

ЭВМ в образовании

**Педагогические
программные
средства**

Министерство народного образования Латвийской ССР

Латвийский ордена Трудового Красного Знамени
государственный университет имени Петра Стучки

Вычислительный центр

Лаборатория проблем школьной и вузовской информатики

Э В М В О Б Р А З О В А Н И И
П Е Д А Г О Г И Ч Е С К И Е П Р О Г Р А М М Н Ы Е С Р Е Д С Т В А

Межвузовский сборник научных трудов

Латвийский государственный университет им. П. Стучки
Рига 1989

ЭВМ в образовании. Педагогические программные средства: Межвузовский сборник научных трудов / Под ред. Н.Н.Устинова. Рига: ЛГУ им. П.Стучки, 1989. 190 с.

Предлагаемый сборник содержит 18 статей, отражающих результаты исследований по общим и прикладным вопросам применения компьютера в образовании. Рассматриваются научные, методические и практические аспекты разработки программно-методического обеспечения для обучения информатике, педагогических программных средств (ППС) для предметов естественнонаучного цикла, а также базового и инструментального программного обеспечения.

Сборник научных трудов предназначен для научных работников, занимающихся исследованиями и разработкой ППС, методическими вопросами применения ППС, для преподавателей, аспирантов и студентов, применяющих персональные ЭВМ в качестве инструмента в своей учебной и профессиональной деятельности, а также для учителей средних учебных заведений.

Рис. 25, табл. 4, библиогр. 83 назв.

РЕДКОЛЛЕГИЯ:

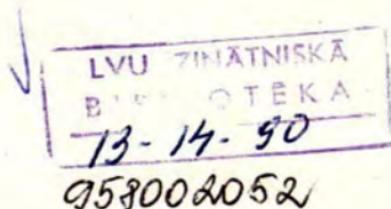
М.В.Витиных, А.С.Корхонена,
К.А.Краст, С.И.Павлов,
Н.Н.Устинов (отв.ред.),
В.Л.Цилевич

Печатается по решению Издательского совета
ЛГУ им.П.Стучки

2405000000-137у

3 ----- Доп. 19
М 812(11)-89

© Латвийский
государственный
университет
им.П.Стучки, 1989



ВВЕДЕНИЕ

Научные исследования по компьютеризации образования и разработка педагогических программных средств (ППС) координируются Академией педагогических наук СССР в рамках комплексной программы "ЭВМ в школе". В Латвийской ССР Министерством народного образования разрабатывается Комплексная программа "Компьютеризация учебного процесса".

Настоящий сборник вносит определенный вклад в разработку проблем компьютеризации образования. Основу сборника составляют исследования, проведенные в 1988-1989 г.г. в лаборатории проблем школьной и вузовской информатики Вычислительного центра при ЛГУ им. П. Стучки, научная тематика которой координируется названными программами исследований. В сборнике представлены статьи сотрудников других подразделений ВЦ при ЛГУ, а также ряда организаций, с которыми установились научные связи: НИИ содержания и методов обучения АПН СССР и НИИ школьного оборудования и технических средств обучения АПН СССР, Института прикладной математики АН СССР (г. Москва), Московского государственного университета, Московского авиационного института, Белорусского политехнического института (г. Минск) и др.

Статьи сборника разделены по следующим четырем разделам:

1. Образование и компьютер. Общие вопросы [1-6].
2. Программно-методическое обеспечение обучения информатике [7-12].
3. Программное обеспечение предметов естественно-научного цикла [13-15].
4. Базовое и инструментальное программное обеспечение [16-18].

Широкий спектр рассматриваемых проблем отражает складывающиеся направления научных исследований, связанных с разработкой содержания учебных курсов с использованием ЭВМ, созданием ППС, в том числе основанных на применении методологии моделирования и экспертных систем.

ОБРАЗОВАНИЕ И КОМПЬЮТЕР
ОБЩИЕ ВОПРОСЫ

Межвузовский сборник научных трудов
ЭВМ В ОБРАЗОВАНИИ. ПЕДАГОГИЧЕСКИЕ ПРОГРАММНЫЕ СРЕДСТВА
Рига: ЛГУ им. П. Стучки. 1989. С. 5-28

УДК 37.01:007+371.3+519.6+681.3.06

П. Л. Брусилковский
ВНИИЦентр ГКВТИ СССР,
г. Москва

Н. Л. Буланова
ВНИИ проблем вычислительной
техники и информатики
ГКВТИ СССР, г. Москва

И. Н. Бурмирова-Зуева
ВНИИ "Эпран" ГКНТ СССР,
г. Москва

Т. Б. Горская-Белова
Средняя школа № 259,
г. Москва

Н. С. Келлин, С. Н. Михайлов, И. В. Репин, П. Д. Ширков
Институт прикладной математики им. М. В. Келдыша
АН СССР, г. Москва

А. Н. Крутов, А. Я. Нодельман, С. Я. Нодельман, А. Н. Сотский
Московский авиационный институт им. С. Орджоникидзе

О. И. Мельникова, Г. Л. Семашко
Объединенный институт
ядерных исследований,
г. Дубна

В. Б. Хозилов
Московский государственный
университет им. М. В. Ломоносова

МЕЖДУНАРОДНАЯ КОМПЬЮТЕРНАЯ ШКОЛА:
ПРОБЛЕМЫ И ПЕРСПЕКТИВЫ

"Деятельность и мышление, мышление и деятельность - вот итог всей мудрости. Оба должны неустанно двигаться в жизни взад и вперед, как выдыхание и вдыхание. Кто делает для себя закон, испытать деятельность мышлением, а мышление - деятельностью, тот не может заблудиться, а если и заблудится, то скоро вернется на верную дорогу". Этому принципу И. В. Гете пытались следовать и мы, авторы проекта Международной компьютерной школы (МКШ), проходившей в г. Дубне в августе 1989 г.

Такого рода школы сегодня становятся популярными, поскольку несут в себе гигантский творческий заряд одержимых педагогическими новациями взрослых и свободных от традиционных педагогических пут детей. Различное содержание от "экзотического" - экологического, археологического и т. д. - до программно-обыденного - физико-математического, питает эти шко-

лы, определяя каждый раз свою особую форму их проведения. Компьютерная школа - достаточно новый жанр, и необходимо пристрасно рассмотреть все ее основания, чтобы отделить экзотику от ценного общечеловеческого материала.

Компьютеризация жизненно необходима нашему образованию. Пожалуй, нет сегодня сравнимых с ней сил, способных так комплексно и радикально затронуть все проблемы обучения, вдохнуть гармонию и жизнь в изрядно утомленную нововведениями и реформами школу. Однако компьютеризация в том виде, как она осуществляется повсеместно сейчас, по крайней мере, не улучшает это положение, а скорее усугубляет его. Собственно, сама проблема сводится к оснащению школ компьютерами и введению курса "Информатика", причем материально-техническое обеспечение находится на переднем плане: "бескомпьютерные" школы стремятся получить современные ПЭВМ, а те, что уже получили, задумчиво решают вопрос: что с ними делать? Ведь курс "Информатики" редуцирован к изучению языков программирования, и желание подготовить "квалифицированного пользователя" все чаще наталкивается на трудности освоения еще одного, теперь уже формализованного "иностранныго" языка (с наперед заданной эффективностью и традициями в этой области). Все менее ясным становится понятие "квалифицированный пользователь" - не компьютерный же "игроман"! А кто? Теоретические воззрения на компьютеризацию пока стыдливо тяготеют к тридцатилетней давности концепциям программированного обучения Б.Скиннера и Н.Краудера, все активнее реализуя принцип "ну хоть что-нибудь практически нужное!". Движущей силой любой инновации является общественная практика, а указанная этапность - это следствие революционных изменений внутри самой практики. Однако в случае с компьютеризацией мы имеем дело с революцией "сверху", и неэффективное общественное производство в подавляющей своей массе нацелено на неэффективное использование компьютеров. Собственно, одним из основных препятствий на пути компьютеризации, по мнению А.А.Самарского, является не недостаток компьютеров как таковых, а отсутствие методологии и технологии их применения.

Понятно, что природа такой "оппозиции" инновациям в конечном счете имеет экономические основания, однако попробуем быть чуткими к современным тенденциям и за всеобщей "обломовщиной" различим истинно человеческое неприятие новой интеллектуальной кабалы, пусть даже и с супер-современным, эргономически притягательным ликом цветного дисплея. Расширяя возможности человека в плане ликвидации рутинных функций, компьютер в известной степени закрепощает мышление исследователя, подчиняет его своеобразной компьютерной логике, имеющимся программно-языковым стереотипам: "чему ты господин, тому и раб". Это состояние довольно остро переживает большинство направлений человеческой культуры. Видимо, повсеместная (в том числе и на Западе) проблема компьютерного освоения действительности и сопутствующая ей невысокая эффективность научно-исследовательского (а не утилитарно-бытового!) использования компьютера, могут быть сняты в том случае, когда человеком эта особая логика осознана, и в процессе решения задачи компьютер выступает как средство, подчиняясь законам постановки и достижения целей. Такое кредо совершенно естественно в нынешней общественной ситуации, когда и теория и практика народного образования обнаруживают полное бессилие в попытках вписать компьютеризацию в среднюю общеобразовательную школу. Совершенствование технической и материальной базы компьютеризации в школе не влечет за собой, к сожалению, параллельных изменений содержания учебных предметов и не приводит к торжеству новых методов обучения.

Поскольку одни теоретики и практики педагогики эти актуальнейшие проблемы не могут самостоятельно решить, необходимо участие самых широких кругов научной общественности, тех, кому не безразличны интересы нашего образования. Это и привело нас - неформальный научно-исследовательско-педагогический коллектив "естествоиспытателей", "технарей" и "гуманитариев" - в Дубну под знамена МКШ с лозунгом "Программное и методическое обеспечение для школы - руками школьников".

1. Концепция МКШ

МКШ задумывалась нами как этап апробации целого ряда предположений концептуального характера.

Во-первых, вслед за Э.В.Ильенковым, Д.В.Элькониним, П.Я.Гальпериним, В.В.Давыдовым предлагалось, что школа должна учить мыслить. МКШ открыла друг другу взрослых и детей для обмена опытом и идеями. Для взрослых представилась возможность в обозримые сроки разработать и проверить соответствующим образом выбранный и методически оформленный фрагмент научного знания, предназначенный для переложения на язык моделирования и последующей трансформации в учебную компьютерную программу. Для школьников - возможность решить цикл задач из предметной области, а также моделировать саму область и, наконец, составить программу, реализующую полученную модель. Обучение мышлению заключается не только в решении различных циклов задач, но и в анализе самих путей и способов решений, в рефлексии средств и методов анализа проблемных ситуаций, без чего для человека невозможно подлинное "квалифицированное пользование" компьютерной технологией. Кроме того, солидный задел предполагалось сделать в преодолении тех аспектов формальной школьной организации, которые, на наш взгляд, активно противостоят целям компьютеризации образования. Таким образом, МКШ стала экспериментальным полигоном для проверки эффективности возможных направлений компьютеризации.

Во-вторых, "компьютерность", рассматриваемая нами в широком контексте освоения методов моделирования действительности, ключевых понятий учебных предметов, дает возможность школьникам осуществлять реальную социально-значимую деятельность. Перед компьютером равны все: и дошколята, и академики. Это обстоятельство является весьма притягательным для детей, а россыпь понятий и интеллектуально-мировоззренческих задач в науках цикла "computer science" становится благодаря активной и организованной учебной деятельности доступной и содержит в себе развивающий эффект. Освоение "Информатики",

начиная с программирования (как оно сейчас веде происходит) без поддержки "онтологического" осмысления школьниками предметной действительности. без освоения пространства научных моделей. технократизирует и угнетает их и без того эмпирическое мышление. Задача иного подхода к содержанию образования состоит в том, чтобы ввести систему понятий одновременно с теорией и практикой моделирования. Тогда реализация модели в компьютерной программе становится для учащегося закономерным и естественным шагом познания мира.

В-третьих, "международность" нашей школы позволяет сопоставить уровень интеллектуальной и компьютерной грамотности школьников разных стран и национальностей, проследить естественное преодоление языковых и этнических барьеров, отработать схему управления творческим разновозрастным коллективом, решающим общие задачи. Применительно к этой сложной педагогической обстановке, на наш взгляд, наиболее отчетливо могут выступить все "плюсы" и "минусы" реализуемого подхода.

В-четвертых, МКШ должна иметь конкретный продукт - фрагменты учебных программ, наиболее точно определяемые по жанру как компьютерный практикум по различным учебным предметам.

Намеченный эксперимент потребовал серьезной научной и методической подготовки. В рамках секции "Компьютер и познание" Международного компьютерного клуба на постоянно действующем семинаре была осуществлена разработка модели МКШ и ее научной программы. В качестве основной единицы работы МКШ выступил "проект", понимаемый как совместная деятельность (учеба, работа, отдых) преподавателя - автора проекта (как правило, это были сотрудники ведущих НИИ и вузов), студентов - помощников, обеспечивавших техническую часть проекта, и школьников - исполнителей проекта по созданию учебных программ для компьютеризации школы (ячейка проекта представляла группу в 4-9 человек).

Проект на стадии подготовки - а она началась в январе 1989 г. - неоднократно докладывался на семинаре, корректировался и в итоге должен был содержать следующие структурные элементы: цель проекта (формулировка и обоснование); опреде-

ление места соответствующего учебного фрагмента в системе учебных предметов в нынешней и предполагаемой системе дисциплин; система учебных задач; характеристика развивавшихся возможностей самого проекта и создаваемой учебной программы; диагностика наличного и конечного (по окончании школы) уровня осведомленности школьников в данном предметном разделе, области моделирования и в возможностях компьютерной техники. "Проектная" схема проведения школы потребовала развития особых организационных форм сотрудничества детей и взрослых, и в результате оказалась довольно эффективной. После утверждения семинаром намеченных проектов (их было 14) их аннотации были посланы для ознакомления участникам МКШ, и на первой пресс-конференции в школе каждый автор проекта излагал свой замысел и "вербовал" себе команду помощников и исполнителей.

2. Принципы формирования научной программы МКШ

В качестве концептуальной основы для проведения МКШ были намечены идеи, разрабатываемые П. Я. Гальпериным, Д. Б. Элькониним, В. В. Давыдовым и др., в русле теории планомерно-поэтапного формирования умственных действий, а также эффективные методические находки, тяготеющие к "педагогике сотрудничества". В первую очередь целью предполагаемых проектов являлось пересмотр традиционного содержания фрагментов школьных учебных предметов, который был проведен по ряду направлений.

1. Каждый учебный предмет (раздел, тема) должен был отражать логику научного поиска, образовываться через раскрытие исходной базовой "клеточки" (единицы) материала и выступать как целостная система понятий. Движение учащегося в учебном предмете предполагало восхождение от абстрактного к конкретному и самостоятельное разворачивание, посылкой для осмысления исторической эволюции (а также и революции) основных понятий данной предметной области (обычно как логики построения учебного предмета, так и цель и смысл изучаемого материала скрыты от ученика, он принимает их необходимость и данность "на веру", без должного личного обоснования, не вы-

водит их из действительных отношений природы и обстоятельств развития человеческого знания).

2. Освоение учебного предмета означает при таком подходе формирование мышления (физического, химического, исторического и т.д.), и новая, особая роль "Информатики" как фундаментальной отрасли науки может существенно изменить расстановку школьных дисциплин. Освоение гносеологического содержания "Информатики" ведет к формированию научного мировоззрения, овладение методами формализации знаменует новый этап в развитии мышления школьника, а постижение основ прикладной компьютерной грамотности означает его готовность в качестве квалифицированного пользователя включиться в компьютерную технологию. Настоящая ситуация в образовании требует решительно преодолеть разрозненность учебных предметов, консолидировать их, и на наш взгляд, системообразующая роль в этом процессе могла бы быть отдана именно информатике. При этом еще раз подчеркнем, что саму информатику не следует трактовать как отрасль программирования, а, напротив, вернуть ей изначально присущее и зафиксированное в названии предназначение науки о предельных формах (*in forma* - содержащейся в форме).

От многих последствий бессистемного построения учебного курса крайне трудно избавиться, если не ставить принципиальной целью среднего образования формирование целостного мировоззрения. Каждый предмет, равно как и каждая наука, имеет свою особую логику разворачивания и стремится к автономности. Столь же сильна и тенденция к синтезу наук. Эти полярные движения должны быть открыты учащемуся; причем не через хаотическую картину взаимосвязей фактов и объяснений в естествознании (как зачастую делается в научно-популярной литературе и телевизионных передачах), а через диалектическое движение предмета в природе и общественном познании, через столкновение подходов и точек зрения, через естественную, революционную, а не искусственно-обывательски-эволюционную модель научного развития. Особенность нынешней ситуации в том, что бурный прирост информатики и арсенала методов моделирования

происходит на фоне обострения автономных и синтетических направлений научных отраслей. По-видимому, не имеет смысла переоценивать влияние информатики на эти процессы, хотя активное привлечение ее аппарата в области оформившегося (сложившегося) научного знания вызвало их существенное обновление, "переоткрытие", переосмысление многих проблем.

3. Каждой научной дисциплине свойственно свое особое сочетание формализованных и неформализованных методов моделирования явлений, доказательных и объяснительных процедур и т.д., однако лишь информатика как субъект предельной формализации легко и естественно восстанавливает межпредметные связи, преодолевает предметные и концептуальные границы, обогащает различные области научного познания, и в этом беспредельном и прогрессирующем процессе утверждает и обогащает сама себя. Такого рода центральное положение в сообществе наук может быть эффективно воспроизведено и при построении системы учебных предметов. Образуются как бы два плана: с одной стороны, раскрытие первоначальной абстракции - "клеточки", единицы, предмета каждой науки, а с другой, - параллельное движение в освоении методов и средств формализации (в первую очередь - математических), без которого фактически невозможно движение самой науки. Многообразие возможностей формализации уже само по себе требует рефлексии и представления моделирования как особой реальности. Культура мышления нашего современника должна основываться на максимально корректном использовании формализованных и неформализованных методов познания. Иными словами, наш ученик должен не закрывать предмет уже известной формулой, а четко представляя ограничения, свойственные любой формуле, любому методу, открывать предмет в многообразии его проявлений и опосредований, в средоточии адекватных и равносторонних методов.

4. Основой методического обеспечения МКШ является использование прогрессивного опыта ведущих учителей-новаторов и опыт зарубежных концепций компьютеризации. Принципиальными для нашего подхода являются два положения.

Во-первых, смена традиционной педагогической парадигмы.

когда учитель есть своего рода "хранитель знаний" и весь процесс обучения сводится к непрерывному рассказу и показу учителем тех или иных фрагментов осваиваемой учеником деятельности. Этот процесс крайне неэффективен, поскольку практически не предполагает того результата, который особенно необходим в условиях современного развития науки и культуры, а именно: чтобы ученик был лучше, умнее, образованнее учителя. Такая цель может быть достигнута при условии, если учитель является организатором учебной деятельности ученика, т.е. когда это необходимо с методической точки зрения, он меняет свое амплуа "лектора-ментора" на "коллегу", "помощника" и т.д. Иными словами, ведущая роль учителя в обучении должно стать деликатное посредничество между учеником и различными системами знания. Итогом здесь может быть активное освоение учениками способов самостоятельной работы.

Во-вторых, при такой организации учебного взаимодействия компьютер выступает как средство познания. Освоение различных приемов и способов моделирования в изучаемых предметных областях может вестись одновременно с раскрытием школьником компьютерных возможностей.

И, наконец, помимо содержательного и методического аспектов работы, важнейшей задачей научной программы было фактическое проведение школы-эксперимента с широким варьированием предметности (различных учебных предметов, учебных задач, фрагментов материала и т.д.), субъектов исполнения (преподавателей, учеников, студентов), форм и методов работы и т.д. В условия каждого проекта входило практическое ведение мониторинга по основным психолого-педагогическим показателям работы каждой учебной группы.

3. Краткая характеристика проектов МКШ

Всю совокупность проектов МКШ можно достаточно условно разбить на три группы: проекты физико-математического цикла, естественно-научные и гуманитарные проекты и проекты, связанные с поиском эффективных форм начального обучения компью-

терной грамотности.

1. Первая группа проектов предполагала достаточно традиционное использование компьютера для моделирования физических процессов и явлений. Цель каждого из проектов - освоение школьниками методов и средств моделирования в рамках физического метода исследования. В основе этого метода со времен Ньютона лежит допущение о возможности изучения окружающей действительности с помощью схематических представлений, допускающих математическую интерпретацию (то, что в последние десятилетия начали называть математическим моделированием).

Для реализации этой цели были выбраны задачи моделирования управляемого движения таких объектов, как килевая парусная яхта, самолет и автомобиль. В процессе решения этих задач учащиеся должны были сделать следующее:

- разработать физическую модель управляемой системы, определить необходимый минимальный набор существенных факторов и взаимосвязь между ними, а также степень их влияния на параметры движения;

- разработать математическую модель управляемой системы;
- найти приемлемый алгоритм решения модели;
- создать необходимое программное обеспечение;
- разработать дружелюбный интерфейс пользователя;
- выполнить ряд исследований по определению зависимости эксплуатационных характеристик управляемых систем от основных конструктивных параметров.

Разработанные прикладные программы могли бы стать средством первоначального обучения, позволяющим получить представление о физических законах движения, управлении движением рассматриваемых объектов.

Проект "парус". Цель - разработка программы моделирования движения килевой яхты.

За время работы школы участникам проекта удалось разработать простую физическую модель движения яхты, отражающую основные особенности этого процесса и соответствующую ей математическую модель.

В процессе этой работы учащиеся получили первые пред-

ставления о методике применения ПЭВМ для решения подобных задач. численных методах решения уравнений движения, способах экстраполяции функций, влиянии шага интегрирования на устойчивость решения и т. д. Наибольшую сложность представляла разработка физической модели движения яхты. Проведенный входной контроль показал также, что весьма трудным является планирование последовательности действий, необходимых для решения физической задачи. Меньше всего проблем возникало при написании программ для ПЭВМ, что объясняется хорошей программистской подготовкой двух из четырех участников проекта.

Физическая модель, реализованная при работе над содержательной частью проекта, достаточно упрощена и отражает лишь основные особенности взаимодействия одномаховой парусной яхты с водой и воздухом. Эта модель учитывает динамику движения только в горизонтальной плоскости и позволяет исследовать поведение яхты при отклонении руля, изменении угла установки паруса и влияние отдельных конструктивных элементов (длины корпуса, площади паруса и руля, а также положение мачты) на ходкость и управляемость яхты.

С учетом индивидуальных склонностей учащихся, были созданы две версии прикладной программы: с использованием системы Turbo Pascal 3.0 для ПЭВМ "Yamaha MSX-1" и системы Turbo Pascal 5.0 для ПЭВМ "ПРАВЕЦ-16". Обе программы снабжены минимальным интерфейсом пользователя, достаточно полно прокомментированы и в существующем виде могут быть использованы при наличии минимального руководства для пользователя. Контрольная версия программы была написана на языке MSX BASIC.

Существующие версии программ представляют пользователю следующие возможности:

- наблюдать за движением яхты в пределах акватории "с высоты птичьего полета";
- наблюдать за положением паруса, руля, направлением ветра с точки зрения члена экипажа;
- изменять положение руля, паруса и площадь паруса путем нажатия соответствующих клавиш.

Разработанные программы могут быть использованы в соста-

ве компьютерного практикума при изучении такого раздела курса физики как "Механика" (движение тела под действием суммы сил при наличии сопротивления среды).

Проект "Самолет". Цель - создание программы, которая бы не только моделировала взлет, полет и посадку самолета, но и помогла пользователю, работающему с ней, получить начальные сведения из области аэродинамики. Исходя из этого, авторы проекта предполагали: создать "игрушку" - тренажер, с которым было бы интересно работать школьнику, а также предусмотреть возможность использовать ее как серьезную обучающую программу. Из первого коллективного обсуждения задачи выяснилось, что школьники раньше практически не работали с компьютером, и это вызвало значительные сложности в дальнейшей работе. В отличие от других проектов в проекте "Самолет" не было явного разделения функций между участниками, однако школьники прошли вместе все стадии решения поставленной задачи - от разработки математической модели до ее реализации на компьютере с помощью конкретного алгоритмического языка.

Результат напряженного труда участников проекта - программа, которая позволяет моделировать взлет и полет самолета. Однако говорить об использовании данной программы в курсе физики средней школы несколько преждевременно. Очевидно, что она нуждается в серьезной доработке. Сами участники проекта считают, что не потратили время даром, а двое из них высказали желание работать над ней в дальнейшем. Кроме того, ребята получили первый опыт работы с ПЭВМ и освоили новый язык программирования.

Проект "Пушка". Основная цель - создание законченного программного продукта, который впоследствии можно было бы применить не только как наглядное пособие по физике, но и как мощный инструмент физического практикума.

В ходе решения задачи участники проекта разделились на три группы:

- группа "физиков", занимающаяся разработкой физической модели изучаемого явления;

- группа "математиков", на которую была возложена задача

преобразования полученных "физиками" уравнений в форму, наиболее удобную для реализации на компьютере, а также составление алгоритма решения поставленной задачи:

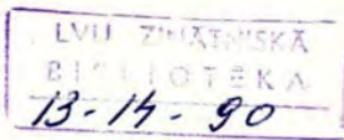
- группа "программистов", которая занималась проблемами ввода/вывода информации в удобной форме.

Все три группы активно работали с компьютером на всех стадиях разработки проекта.

В ходе работы была написана и отлажена программа моделирования движения материальной точки и полета тела в поле действия силы тяжести с учетом сопротивления окружающей среды. Эта программа, по замыслу авторов, должна занять важное место в процессе изучения курса физики в средней школе. Ее можно применять как в шестых-седьмых классах при изучении кинематики, так и в старших классах при изучении динамики и основ аэродинамики.

Проект "Автомобиль". Цель - овладение методикой применения ПЭВМ при моделировании динамической системы "автомобиль - внешняя среда". Конечным продуктом работы должна быть программа моделирования, позволяющая учитывать влияние конструктивных параметров автомобиля и получать зависимости эксплуатационных свойств от параметров внешней среды. Совместно со школьниками были проанализированы возможные способы моделирования. Для реализации на ПЭВМ была выбрана простейшая модель. Автомобиль рассматривался как материальная точка, находящаяся под действием сил трения и сопротивления воздуха. Модель дороги состояла из отрезков прямых и дуг окружностей. Управляющими параметрами были выбраны: расход топлива и угол поворота руля. При этом полагалось, что угол отклонения продольной оси автомобиля пропорционален углу поворота руля, кривизна поворота пропорциональна углу поворота руля, дорога не имеет неровностей. Учитывались ограничения: на силу тяги, которая не должна превосходить максимальную силу трения покоя, действующую на автомобиль, на скорость, при которой на определенном радиусе кривизны нет проскальзывания.

Самостоятельная разработка модели встретила трудности, поэтому в итоге модель была предложена преподавателем. Мате-



математическая модель и алгоритм расчетов по модели разработаны совместно со школьниками. Работа была разбита на этапы:

- ввод исходных данных;
- имитация "внешней среды";
- моделирование объекта (автомобиля);
- управление объектом;
- расчет кинематических параметров;
- выбор системы отсчета;
- графическое отображение информации.

Инструментальным языком был выбран Pascal, как наиболее удобный для реализации блочных конструкций. В процессе решения конкретных задач, основы его были усвоены школьниками достаточно легко. Каждым участником самостоятельно разработано и отлажено по несколько процедур.

Проект "Космический разведчик". Перед группой была поставлена задача по доставке космического зонда (спасателя, разведчика) в заданный район Вселенной и проведению в этом районе аварийных или исследовательских работ. Предполагалось, что в ходе ее решения будет конкретизирована постановка, скорректировано содержание и уточнена совокупность тех более простых задач (проблемных ситуаций), разрешая которые, школьники освоят существенные понятия и соотношения данной предметной области в их взаимосвязи.

Для решения задачи о "космическом разведчике" учащимся необходимо было точно сформулировать задачу: выбрать критерий оптимальности перелета (связанный со структурой пространства и времени), построить способ описания поля взаимодействия; понять особенности реального взаимодействия космических объектов и их внешнее проявление, в том числе на примере реальных систем.

В первые полторы недели основной формой работы над проектом была дискуссия. Все участники проекта высказывали свободно свою точку зрения по обсуждаемому вопросу. При этом роль руководителя и ассистентов в основном сводилась к управлению дискуссией: учащимся по ходу обсуждения задавались вопросы, подчеркивавшие либо неудачность их гипотез, либо их

сильные стороны.

Следует отметить, что ассистенты (а ими были студенты двух ведущих вузов страны), приняли такую роль учителя не сразу. Сначала они пытались познакомить учащихся с правильными результатами сами, оставив им только техническую часть работы. Однако после того, как исполнители самостоятельно сформулировали два основных способа описания поля взаимодействия и предложили алгоритм построения линий равного потенциала на экране монитора, ассистенты перестроились. В результате учащиеся достаточно полно освоили понятия и явления рассматриваемой области.

Отметим, что целью проекта не являлось обучение ребят программированию на языке высокого уровня. Несмотря на это в ходе самостоятельной деятельности - выполнения заданий на ЭВМ - большинство учащихся контрольной группы практически самостоятельно освоили новый для себя язык программирования Pascal и успешно справились с заданиями по программированию: переезду фрагментов программ с языка BASIC на язык программирования Pascal. При составлении этих контрольных заданий учитывались структурные отличия языков.

2. Вторая группа проектов естественно-научного и гуманитарного циклов была ориентирована на моделирование предметности в нетрадиционных для компьютера областях.

Проект "Остров". Цель проекта - осуществить переход от разрозненных и упрощенных понятий географии к компьютерному моделированию их существенных взаимосвязей в условиях единого природного комплекса.

Главная задача проекта заключалась в создании программного продукта по курсу географии за седьмой класс, способного смоделировать и проиллюстрировать структуру и основные особенности природного комплекса на примере любого острова, выбранного произвольно в северной части Атлантического океана. Остров как объект исследования выбран не случайно. Он является единицей, объединяющей все компоненты природы и представляющей их взаимосвязь. Для упрощения остров рассматривался как точка на карте, а его природный комплекс определялся как единый для всей его площади. Содержательная часть

программы была полностью намечена еще до начала работы МКШ. Она включала в себя графическую характеристику всех компонентов природы, а также их основные взаимосвязи и закономерности, определенные как система правил.

Цель-минимум проекта в процессе работы МКШ была достигнута - программная часть модуля сделана полностью. Содержательная часть реализована только наполовину из-за значительного объема графической работы (сорок картинок и диаграмм). Помимо основной системы, в духе здоровой конкуренции, были разработаны два графических редактора (немецкий и словацкий), которые могут быть использованы как самостоятельный программный продукт. Также была разработана экспертная мини-система. Основной программный модуль оформлен таким образом, что общение с ним возможно на любом языке, который выбирает пользователь (в данный момент на немецком, русском или словацком). Работа с данной программой может органически завершать курс школьной физической географии материков и океанов, а также использоваться в течение всего года для формирования ряда понятий по отдельным разделам географии.

Созданная учебная программа может работать в двух режимах: обучение и эксперимент. В зависимости от этого изменяется и предъявляемый ей материал. Порядок работы по программе таков: ученик выбирает географическое положение острова, а затем последовательно в соответствии с географическими закономерностями им описываются компоненты природы, то есть выбираются географические изображения соответствующих компонент. Всего их в программе девять: географическое положение, тектоническое строение, тип земной коры, рельеф, растительность, животный мир, почва, происхождение острова. Каждая компонента имеет несколько графических изображений. Ученик как бы конструирует природный комплекс в данной точке океана. Правильность выбора определяется экспертной системой, в которую включены все основные правила-закономерности. Правильные ответы оцениваются определенным количеством баллов, что дает возможность быстро выявить учащихся, слабо разобравшихся в материале.

В основе задания лежит идея целостности географической оболочки Земли. Поэтому в дальнейшем есть возможность доработать программу с точки зрения экологии, а также расширить область изучения, добавив в программу информацию по всем остальным океанам.

Проект "Генетика". Цель - разработка программы, моделирующей передачу генетической информации от родителей потомкам и наглядно изображающей этот процесс на экране. Кроме того, для введения пользователя в курс дела потребовалось сделать программу, выводящую некоторый пропедевтический текст по генетике, а также толковый словарь терминов с машинной реализацией ссылок - задача чисто программистская.

Каков же итог? Степень разработанности предметной области соответствует 100% замысла. Что заняло в программе 200 строк основного модуля. Выполнена предлагавшаяся программа-минимум: реализована возможность постановки компьютерных экспериментов по скрещиванию и мутагенезу; имеется текст начальной информации и словарь - все в финальной стадии отладки. Предлагаемая программа может быть рекомендована к использованию в общеобразовательной школе для организации практикума при изучении генетики в курсе общей биологии. Разработанную программу рекомендуется сочетать с пакетом программ, созданным в НИВЦ АН СССР (г. Пушкино).

Проект "Словарь" имел первоначально несколько разноплановых целей, которые должны были конкретизироваться в процессе работы коллектива. Одной из таких целей было создание электронного словаря-справочника в помощь начинающему программисту или лицу, не владеющему иностранным языком, принятым в качестве официального для общения в среде МКШ. В результате обсуждения проблемы со школьниками постановка задачи была сведена к следующему:

1. Минимальная цель - построение электронного словаря-справочника в англо-русском варианте;
2. На основе выработанной структуры хранения информации предложить оптимальный алгоритм доступа и обработки данных;
3. Реализовать выработанные алгоритмы в виде программ на

языке Pascal при строгом распределении работ:

4. Предложить альтернативные варианты информационно-обучающей системы по овладению основами английского языка и попытаться их реализовать в виде программ на языке Pascal:

5. В целях повышения уровня информированности освоить работу в различных программных средах. В том числе в среде редактора SNI-WRITER реализовать армянский шрифт:

6. При помощи созданных программ произвести заполнение базы данных словаря, и на основе полученной выборки из словаря провести работы по оценке быстродействия и приступить к сложному семантическому анализу состава слов.

На данном этапе все перечисленные выше пункты, выдвинутые в качестве целевых, реализованы.

Проект "История". Цель - изучение связей гуманитарных (история, лингвистика) с точными и естественными (астрономия, биология, computer science, физика) науками.

Традиционно гуманитарные науки изучались в школе совершенно изолированно от естественных. Поэтому те старше школьники, которым нравится заниматься физикой и другими точными науками, в подавляющем большинстве забыли то, чему их учили в 5-6 классах на уроках истории. Аналогично, ученики, избравшие гуманитарное образование, "не знакомы" с точными науками. А вместе с тем, связи этих наук были гораздо теснее. Историография, например, в XVI-XVIII веках считалась наукой точной (как сказали бы сегодня, это была прикладная астрономия). Что касается лингвистики, то один из ее создателей Г.В. Лейбниц вообще не разделял ее с математикой: в XVII веке, разумеется, еще не было деления языков на искусственные и естественные.

В настоящее время вновь стала возможной математизация некоторых разделов истории:

а) применение современных ЭВМ позволяет за разумное время обрабатывать значительные объемы информации, необходимые при формализации и исследовании задач с гуманитарной предметной областью:

б) методы теории вероятности и математической статисти-

ки развиты к настоящему времени настолько, что с их помощью стало возможным строгое обоснование результатов историографического и лингвистического характера, получаемых в численных экспериментах.

В силу сказанного, правомерной является такая постановка задачи для МКШ: обучение школьников основам количественного анализа информации на примере некоторых задач историографии и лингвистики.

Работа интернационального коллектива над данным проектом имела особенности в организационном плане. Советские школьники, обучавшиеся по программе, разработанной в ИПМ им. М. В. Келдыша АН СССР, имели возможность узнать раньше о простейших принципах математического моделирования явлений, изучавшихся в гуманитарных (история, лингвистика) науках. Иностранцы такой возможности, как оказалось при сравнении учебных планов и программ, принятых в странах-участницах МКШ, не имели. Поэтому лекционный курс в данном проекте был разделен на две части. Иностранцам делегациям были прочитаны лекции по теоретическим основам применения естественных методов для анализа древних документов и литературных текстов.

В итоге был разработан ряд программ, позволяющих смоделировать и представить в удобном для пользователя виде историографические и лингвистические массивы информации.

Проект "Квартет". Перед школьниками ставилась задача создания программы, которая по заданной музыкальной теме давала возможность выстраивать и озвучивать простейшие музыкальные формы.

Программу предполагалось делать на "Yamaha MSX" на языке BASIC. Кроме того, проект преследовал и другие не менее важные цели:

- эстетическое развитие участников проекта, знакомство с музыкальными произведениями различных направлений;
- освоение музыкального материала, знакомство с теорией музыки, разбор композиционных приемов в демонстрируемых произведениях.

В конечном счете задача моделирования была сведена к созданию программы, позволяющей по заданной музыкальной фразе:

- строить противосложение;
- строить развернутую мелодию с заданной гармонией;
- давать ее графическое изображение.

На компьютере "Правец-16" был сделан двухголосный музыкальный редактор, позволяющий озвучивать заданное музыкальное произведение, сохранять его запись в памяти компьютера. Данные программы могут быть использованы в процессе преподавания музыки и в школе. Школьник, не владея никаким музыкальным инструментом, может прослушать звучание сочиненной им самим или взятой от другого автора музыкальной композиции. С другой стороны, данные программы могут сами строить некоторые музыкальные композиции, а учащийся может сам дополнять и изменять эти программы.

Современные персональные компьютеры, обладая звуковым генератором и возможностью управления им с помощью программы, дают уникальную возможность для разработки новой методики обучения музыке. Такая методика позволяет с совершенно новой стороны взглянуть на природу звука, помогает лучше понять и по-иному взглянуть на стилистику различных музыкальных направлений, открывает широкий простор для экспериментов с цвето-графическим изображением мелодий и шумовыми эффектами.

3. Третья группа проектов (обучение компьютерной грамотности) предполагала проведение ряда экспериментальных курсов обучения на материале computer science.

Проект "Обучение-исполнитель". Его содержание составляло начальное знакомство с программированием группы шестиклассников школ г. Дубны. Ранее школьники с программированием знакомы не были. Подход к обучению осуществлялся следующим образом.

В качестве языка программирования предлагался мини-язык, позволяющий управлять отображающимся на экране роботом. Язык содержит все структуры управления реальных языков, а все действия языка вполне наглядны.

Точка зрения авторов проекта: для получения начального

знакомства с программированием достаточно получить некоторый опыт программирования на мини-языке. Основные понятия при этом усваиваются не хуже, а лучше, чем при изучении обычных языков, а обучение идет легче и быстрее.

Цели проекта:

а) для авторов - опробовать в обучении несколько конкретных исполнителей и мини-языков. Необходимо понять, какие черты мини-языков и их реализации способствуют обучению, а какие мешают;

б) для школьников - освоить идеологию применения исполнителей, понять их преимущества, составить и отладить набор задач для исполнителей.

Цели проекта реализованы полностью. Со школьниками проведён полный цикл занятий, и за это небольшое время ими были освоены два исполнителя "жук" и "строитель". К концу работы все дети могли составить достаточно сложные по структуре программы, содержавшие вложенные циклы и ветвления.

Проект "Мульти". Основная цель - пробудить стремление к творчеству при работе с компьютерами.

Конкретной целью, поставленной перед ребятами, была разработка обучающих программ по математике для младших классов. В результате создано около двадцати программ для первых и шестых классов. Они могут использоваться в учебном процессе школ, где ПЭВМ оснащена трансляторами с расширенного языка BASIC.

Полученные ребятами навыки позволят им работать самостоятельно практически с любыми компьютерами. Пробудившийся интерес выразился в том, что ребята, используя литературу, начали изучать новые возможности ПЭВМ.

Проект "Базы данных и систем управления базами данных (БД и СУБД)". Цель - создание баз данных и пакетов программного обеспечения БД для дальнейшего их использования и развития в процессе школьного обучения.

Программное обеспечение БД для работы МКШ было разработано и создано (на 75%) заранее, до начала работы школы. Это обстоятельство сыграло решающую роль в привлечении ребят для

участия в проекте. Существование наглядного примера позволило выполнить поставленные задачи быстро и легко.

БД участников школы и БД проектов школы предоставляет пользователю следующие возможности:

- эстетически оформленный ввод/вывод информации (в БД по участникам МКШ - два формата вывода); уничтожение информации (по номеру записи), любая статистика - пользователь сам выбирает БД, поле БД и необходимый признак.

В БД по библиографии книг МКШ у пользователя имеются следующие возможности: ввод списка всех книг библиотеки в алфавитном порядке; вывод информации о книгах по ключу, введенному с экрана; вывод на экран списка книг, взятых читателями; ввод информации о новых книгах; ввод информации о возвращении книги в библиотеку.

По мнению автора проекта, наиболее эффективным способом является обучение школьников по уже существующим БД, что позволяет избежать проблем их построения и помогает перейти к составлению довольно сложных программ по обработке соответствующей информации, а также совершенствованию созданных. В такой ситуации обучаемые быстрее понимают суть проблемы, овладевают системой управления и могут наиболее полно раскрыть свои возможности.

Проект "Детский сад". Цель - апробация идей, которые могут лечь в основу курса, знакомящего школьников младших классов с алгоритмическими языками, элементами программирования, работой на персональном компьютере.

Обучение осуществлялось на "теоретических" и практических занятиях, условно их можно разбить на две части. Первая - это достаточно свободное обсуждение тем, представленных цепочкой понятия "ЭВМ - "машина" - "исполнитель" - "множество "умений" исполнителя" - "управление исполнителем для достижения некоторой цели" - "необходимость "общения с исполнителем" - "сообщение = язык + способ сообщения". Здесь же дети играли в "роботов", причем решались задачи следующего типа:

1. Дана "цель" и робот с жестко закрепленным набром

действий. Как следует управлять роботом, чтобы достигнуть этой цели?

2. Дается некоторая "цель". Каким набором действий должен владеть робот, достигающий этой цели?

В качестве игровых выступили хорошо знакомые ситуации, поэтому в основном задачи решались легко, алгоритмизация проходила естественно.

Затем детям был предложен компьютерный исполнитель "жук", на языке которого учащиеся составили свои первые (только линейные) программы (этот исполнитель явился переходным звеном ко второй части обучения). Здесь ученики начали осваивать алгоритмические конструкции на примере исполнителя "жук", языков программирования BASIC и Pascal. Обучение проходило в процессе решения последовательности графических задач (в том числе: введение координат сначала полярных, а затем декартовых; знакомство с графическими операторами языков программирования), каждый последующий тип задач являлся развитием предыдущего, приводил к необходимости использования новых конструкций языков (цикл, условие). Для каждой задачи, когда это было возможно, составлялось три программы: на всех вышеуказанных языках. Там, где невозможно использовать "жука", пользовались двумя другими языками.

З а к л ю ч е н и е

За две с половиной недели работы МКШ практически во всех проектах достигнуты ожидаемые результаты. В атмосфере творческого сотрудничества созданы фрагменты методик и программных продуктов. Более детальную психолого-педагогическую информацию о школе еще предстоит обработать и осмыслить, но уже сейчас видно, что чуда не произошло. Обстановка раскованного труда в содружестве со сверстниками и "демократичными" взрослыми как бы аккумулировала все творческие потенции школьников. Компьютерные игры - бич подобного рода мероприятий - уступили свое место работе над проектами. Та же участь грозила спортивной и культурной программе МКШ.

Пока трудно сказать, в какой степени опыт МКШ переносим на аналогичные формы внешкольной работы, а также может быть внедрен в повседневную практику общеобразовательных школ. Однако очевидно, что сама идея МКШ, концепция компьютеризации, лежащая в ее основе, созданные фрагменты компьютерного практикума доказали свою жизнеспособность и требуют дальнейшего развития.

Принцип "задача в задаче", когда освоение компьютерной технологии не является самостоятельной целью, а составляет лишь этап решения более широкой предметной проблемной ситуации, на наш взгляд, может составить генеральное направление в ходе компьютеризации. Опыт МКШ показал, что подход ко всему комплексу средств информатики через деятельность моделирования и освоение предметной области значительно эффективнее, нежели движение в каком-либо одном из этих фрагментов знания. Их единение как бы создает естественную взаимодополняемость моментов учебной работы школьников и придает ей продуктивность, законченность, осмысленность и социальную значимость.

Именно такая компьютеризация с разработанной системой-синтезом учебных предметов, учебными программами, практикумами, новыми формами взаимоотношений учителей и учеников может поставить преподавательский корпус в принципиально новую ситуацию: запечатленный в технике эффективный способ учебной деятельности не позволит отойти от логики предметного содержания или методически произвольно дробить его на части вне раскрытия внутренней связи.

Мышление и деятельность, деятельность и мышление - не только итог, но и исток всей мудрости. Осознания этого мы ждем от учителей, пытаемся донести эту идею до своих учеников, ее понимание хотели бы открыть в своих детях.

Межвузовский сборник научных трудов
ЗВМ В ОБРАЗОВАНИИ. ПЕДАГОГИЧЕСКИЕ ПРОГРАММНЫЕ СРЕДСТВА
Рига: ЛГУ им. П. Стучки. 1989. С. 29-34

УДК 37.01:007+371.3+378.147+519.6+681.3.06

А. Н. Крутов	С. И. Павлов, В. Л. Шилевич
Московский авиационный институт им. С. Орджоникидзе	Вычислительный центр при ЛГУ им. П. Стучки, г. Рига

В. Б. Хозиев	П. Д. Ширков
Московский государственный университет им. М. В. Ломоносова	Институт прикладной математики им. М. В. Келдыша АН СССР, г. Москва

**ИНТЕГРИРОВАННЫЙ КУРС "ЕСТЕСТВОЗНАНИЕ":
ДЕЯТЕЛЬНОСТЬ. МОДЕЛИРОВАНИЕ. КОМПЬЮТЕР**

Особым свойством компьютерной технологии, приносимой в любую предметную область, является способность качественно изменять традиционно присущую этой области расстановку целей и средств. Поэтому в каждом новом направлении компьютерной экспансии необходимо стремиться к анализу роли и функций этой технологии в наиболее общем контексте развития предметной области, что может помочь усмотреть решающие тенденции и в какой-то мере уберечь от ошибок.

Первые электромеханические и электронные вычислительные машины были созданы в 40-е годы XX века для решения задач, связанных с моделированием сложных физических и технических систем при проектировании объектов авиационной и ракетной техники, вооружения и т. д. Бурный прогресс элементной базы и архитектуры вычислительной техники, снижение ее габаритов и стоимости, увеличение вычислительной мощности, расширение возможностей периферийных устройств и развитие системных и прикладных программных средств предопределили прогрессирующее расширение "побочных" областей применения компьютера: управление технологическими процессами, хранение и обработка

данных, система телекоммуникаций, автоматизация делопроизводства, игры и т.п. В результате число пользователей, применяющих компьютер по "изначальному" предназначению, т.е. для моделирования, составляет явное меньшинство среди всех пользователей ЭВМ, и относительное их число продолжает неуклонно сокращаться.

Этот эффект в полной мере проявился к моменту, когда компьютер "пришел" в систему образования. Поэтому вполне естественно, что структура применения ЭВМ в образовании представляет собой проекцию описанной ситуации. Внедрение ЭВМ в учебный процесс заметно увеличивает объем работы как для учителя, так и для ученика, отнюдь не обеспечивая резкого повышения качества образования. Кроме того, отсутствует осознанная потребность в компьютеризации, поскольку достижение нынешних целей обучения (с существующими содержанием, формами и методами обучения) вовсе не требует помощи компьютера. Эти причины, наряду с обычной реакцией любой инерционной системы на любую инновацию, приводят к отторжению компьютера как инородного тела из системы образования.

Можно выделить три этапа компьютеризации образования:

- 1) компьютер как новый объект изучения;
- 2) компьютер как средство будущей профессиональной деятельности; основная задача - подготовка пользователя;
- 3) компьютер как инструмент познания в процессе обучения.

В плане теоретического осмысления мы сегодня находимся на уровне перехода от второго этапа к третьему, а в плане практического применения ЭВМ в образовании - на уровне перехода от первого ко второму этапу. Отметим, что переход от одного этапа к другому означает не полный отказ от деятельности, характеризующей предыдущий этап, а лишь отказ от гипертрофизации роли этой деятельности, определение для нее соответствующего места в учебном процессе. Если на первом и втором этапах мы пытаемся включить компьютер в имеющуюся систему образования, то на третьем этапе возникает качественно новая ситуация, характеризующаяся как объективной необходимостью, так и объективной возможностью изменения целей и содержания образования.

Объективность перехода от технологического общества к информационному (изменение социально-экономической ситуации, массовая инженеризация и интеллектуализация труда) предопределяют новые требования к работнику. Другими словами, необходимость пересмотра целей образования и появление компьютера имеют общие причины. Теперь от работника требуется не столько исполнение готового "алгоритма", сколько конструирование самого "алгоритма" или даже постановка задачи, причем указанное изменение характера труда происходит не только в "традиционной" промышленно-технологической области.

Результатом образования должно стать не овладение частными видами деятельности, а овладение средствами, позволяющими постоянно осваивать новые частные виды деятельности. Не овладение суммой знаний должно служить критерием зрелости выпускника, а овладение методологией приобретения знаний.

Наряду с изменениями в содержании обучения необходимы изменения и в организации учебной деятельности. В такой ситуации становится нужным не учитель - "источник знания", а учитель - "диспетчер", организатор процесса приобретения нового знания. Активное взаимодействие учащегося с изучаемым предметом, поддержанное компьютером, дает возможность выйти на принципиально иное качество индивидуализации обучения.

Рассмотрим новое содержание обучения, соответствующее изложенному подходу, на примере курса естествознания.

Центральным элементом предлагаемого подхода является понятие модели, понимаемое в самом широком смысле. При этом моделирование выступает:

- во-первых, как наиболее активная форма деятельности обучаемого;
- во-вторых, как наиболее адекватная методология приобретения объективно и субъективно новых знаний в естествознании;
- в-третьих, изначальная сфера применения компьютера, возврат к которой позволяет качественно повысить эффективность использования компьютера в учебном процессе.

Не останавливаясь здесь на определении понятия модели, отметим, что учащиеся довольно легко усваивают его смысл.

Действительно, игра, являющаяся эффективнейшим способом изучения окружающего мира, непосредственно основана на модельном подходе к действительности, реализуемом в процессе деятельности играющих. В ходе игры дети моделируют различные взаимоотношения, конфликты, трудовой процесс, пользуясь при этом языком, наиболее адекватным осваиваемой ситуации.

Если во главу угла ставится моделирование как метод приобретения знания, то по-новому выглядит проблема межпредметных связей, поскольку становится ясно, что различия между учебными дисциплинами - это только различия между методами изучения единого предмета - Вселенной во всем ее многообразии. Представление о единстве предмета естествознания необходимо закладывать на самой ранней стадии обучения - по окончании начальной школы и сохранить при последующем разделении естествознания на физику, астрономию, химию и другие, в значительной мере обусловленном избранной системой моделей.

В зависимости от сложности рассматриваемого объекта и требуемого математического аппарата могут применяться различные виды моделей, доступные обучаемому, в том числе, описательные (вербальные), физические, математические и др.

С давних пор модельный подход активно используется в процессе обучения физике, хотя и в неявной форме. При решении физических задач учащиеся должны составить для себя ясную физическую картину явления или процесса, заданного условиями задачи, выбрать параметры, характеризующие это явление или процесс, выявить законы природы, "управляющие" его развитием, и записать их в математической форме, т.е. построить его математическую модель. Далее наступают этапы решения уравнений модели с целью нахождения связей между искомыми и заданными параметрами явления или процесса и анализа полученного результата. Несвершенство действующей системы образования проявляется, в частности, в том, что в подавляющем большинстве случаев учащиеся опускают этап анализа, интерпретации результата, ограничиваясь получением "ответа" - сведений о явлении или процессе. Но ведь полученный в аналитическом виде результат решения задачи является не чем иным, как элементарной моделью рассматриваемого явления или процесса, да-

юшей возможность проследить его поведение при варьировании определяющих параметров.

В целом курс естествознания должен строиться как расширяющаяся система взаимодействующих моделей. В процессе анализа и интерпретации результатов моделирования, сравнения их с результатами натурального эксперимента неизбежно возникновение противоречий. Эти противоречия являются причиной для модификации. "расширение" уже рассмотренных моделей, а также введения новых моделей. Вполне допустимо и даже желательно применение различных моделей для описания одного и того же явления или процесса. Особую дидактическую эффективность могут иметь "ложные" модели.

В определенной мере в курс естествознания интегрируется и математика. В контексте последовательного применения моделирования многие математические понятия, приемы и методы вводятся естественно, с ясной целью. Значительно менее вероятна типичная ныне ситуация, когда учащиеся не могут применить полученные на уроках математики знания для решения конкретных задач на уроках физики.

Принципиально новые возможности для применения метода моделирования в процессе учебной деятельности предоставляет компьютер. В этом отношении решающими факторами являются:

- возможность проведения большого объема вычислений с малым временем реакции;
- возможность обработки большого количества информации с малым временем выборки при активном взаимодействии обучаемого с обширной информационной базой;
- эффективная визуализация процессов с применением динамической графики, цвета, звука;
- возможность в интерактивном режиме управлять ходом моделируемого процесса, варьируя параметры, начальные и граничные условия.

Указанные свойства компьютера позволяют исключить трудоемкие, слишком сложные или по иным причинам недоступные обучаемому этапы процесса моделирования, тем самым создавая возможность активного взаимодействия с моделями различной слож-

ности.

Изложенная методология таит в себе опасность абсолютизации "компьютерной реальности". Надежной защитой от такого восприятия служит формирование у обучаемых четкого представления о модели как отражении реальности.

Известные программные среды, ориентированные на активное взаимодействие обучаемого с моделью реальности (например, LOGO, MicroWorld и другие) недостаточно эффективны, поскольку используются, как правило, вне органической взаимосвязи с содержательной концепцией курса в целом.

Таким образом, актуальной задачей коренной перестройки содержания образования является разработка интегрированного курса естествознания на основе методологии моделирования. Неотъемлемой составной частью курса должны быть соответствующие педагогические программные средства.

Межузовский сборник научных трудов
ЗВМ В ОБРАЗОВАНИИ. ПЕДАГОГИЧЕСКИЕ ПРОГРАММНЫЕ СРЕДСТВА
Рига: ЛГУ им. П. Стучки. 1989. С. 35-40

УДК 37.01:007+53(07.07)+371.3+378.147+519.6+681.3.06

А. Г. Головейко, В. Г. Севастьяненко,
В. В. Сидорик, Л. Н. Смурага
Белорусский политехнический институт, г. Минск

МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ В КУРСЕ ФИЗИКИ ДЛЯ ШКОЛЬНИКОВ И СТУДЕНТОВ

Хотя компьютерная технология все больше охватывает промышленное производство, науку, технику и другие сферы жизни и деятельности общества, ее возможности применительно к образованию еще остаются нераскрытыми.

В настоящее время компьютеризация образования эффективно коснулась лишь тех аспектов учебного процесса, которые связаны с применением компьютерной технологии для решения конкретных, преимущественно количественных задач. При всей своей неоспоримости и большой значимости это направление тем не менее не вторгается в дидактический аспект образования. Большинство учебных дисциплин, их научные основы и содержание изучаются классическими методами, сущность которых сводится к тому, чтобы познать самому то, что уже познано другими. Задача заключается в том, чтобы эта учебная проблема познания решалась на основе компьютерной технологии при значительном снижении трудоемкости обучения и одновременно существенном повышении его эффективности.

Можно утверждать, что эта цель вполне достижима, особенно по тем вопросам обучения, которые поддаются математическому моделированию. В целом это относится к обширному комплексу учебных дисциплин физико-математического и технического профиля. Именно через математическую модель компьютерная технология становится эффективным методом обучения, раскрывающим особенности процесса познания.

Математическая модель физического или технологического

процесса адекватно и на строгом количественном уровне отражает все особенности процесса, прогнозирует его развитие и дает надежную информацию о нем. Математическая модель становится исходным звеном компьютерной технологии для извлечения информации о процессе.

Полученная аналогичным образом информация о процессе используется в автоматическом проектировании (САПР) для решения конкретных конструкторских или технологических задач реального производства. При изучении естественных и технических дисциплин информация о процессе должна быть использована для дидактических целей, то есть для познания и изучения самого процесса.

Таким образом, математическая модель выступает как исходный источник информации, а компьютерная технология - как технология автоматизированного извлечения информации из математической модели с представлением ее в обработанном виде на цветном графическом дисплее в легко воспринимаемой форме для решения дидактических задач. Важно подчеркнуть, что при этом высокая степень сложности математической модели перестает быть тем барьером, из-за которого не включаются в учебные программы многие важные вопросы, которые могут оказаться недоступными учащимся.

Устранение указанного барьера создает принципиально новые условия для соединения учебного процесса с реальной научно-технической практикой по самым актуальным вопросам. Так, например, широкий круг задач механики, физики, химии, биологии, техники, социологии описываются системами дифференциальных уравнений. Алгоритмы решения этих уравнений, оформленные в виде пакетов программ, делают решение прикладных задач доступными для студентов.

Но не менее важны и другие дидактические возможности, связанные с применением компьютерной технологии на основе математического моделирования. Сложная математическая модель - это всегда трудный, а иногда и непреодолимый барьер на пути к познанию изучаемого явления или процесса. Обычно преодоление этого барьера искивается в порядке помощи студенту пу-

тем упрощения реальных задач, обращения к их частным тривиальным вариантам. Но есть и другая возможность, а именно, не прибегать к таким приемам, а возложить преодоление вычислительного барьера на ЭВМ, сохранив математическую модель в ее максимальном приближении к действительности.

Студенту предлагается предварительно осмыслить содержание задачи, разобраться в сущности математической модели, осознать конечную цель решения задачи и выделить те условия, от варьирования которых зависит развитие процесса. На начальном этапе, при недостаточной компьютерной грамотности студентов, им можно предложить готовую программу вычислений и вывода конечных результатов на цветной монитор.

Получая прямой доступ к ЭВМ, студент освобождается от рутинной работы и может сосредоточиться на осмыслении физического содержания рассматриваемого явления, процесса или задачи. Варьируя условия, студент на экране терминала получает всю гамму конечных результатов в наиболее удобной для анализа и легкой для восприятия иллюстративной форме.

Студент фактически "проигрывает" явление в различных аспектах и "видит" на экране все его особенности, зашифрованные в математической модели, часто не очевидные или совсем недоступные при классическом подходе. Студент чувствует себя участником реальной работы, сам управляет процессом познания. В условиях творческого поиска в процессе обучения студент на собственном опыте убеждается в высокой эффективности компьютерной технологии.

Необходимо, однако, отметить, что изложенная постановка учебного процесса далеко не проста по своему программному обеспечению. Учебная программа по любому избранному вопросу требует тщательной методической проработки. От составителей учебных программ требуется не только знание рассматриваемого вопроса и компьютерной технологии, но и понимание психологии и особенностей мышления учащихся. В настоящее время программное обеспечение нового метода обучения пока отстает от его принципиального обоснования и технической реализации.

В Белорусском политехническом институте изложенный прин-

цип обучения реализован на кафедре технической физики в специализированном дисплейном классе математического моделирования физических процессов. Моделируются явления механики, молекулярной физики, термодинамики, квантовой механики и полупроводников. Моделируются также сложные уникальные физические эксперименты, которые не могут быть реализованы в учебном физическом практикуме из-за отсутствия соответствующих экспериментальных установок или необходимого дорогостоящего капитального оборудования. Тот факт, что математическая модель и компьютерная технология без создания уникальных установок обеспечивает достижение тех же конечных учебных целей по изучению существа эксперимента и его результатов, еще раз подчеркивает, насколько исключительны возможности рассматриваемого принципа обучения.

Сотрудниками кафедры технической физики разработан начальный цикл учебных программ, большинство из которых уже внедрены в практику учебного процесса. Ниже дается краткий комментарий к некоторым действующим программам.

Математическая модель движения материальной точки в центральном гравитационном поле Земли основана на уравнениях траектории движения и времени полета. На экран выводится часть поверхности Земли или весь земной шар, траектория движения материальной точки, численные данные об основных параметрах траектории, о дальности и времени полета. Студент по своему усмотрению задает начальную высоту, начальную скорость и угол ее наклона к местному горизонту и получает с экрана дисплея всю информацию о движении в центральном силовом поле по замкнутым и не замкнутым траекториям в зависимости от конкретных условий запуска.

Математическую модель столкновения упругих шаров составляют уравнения, выражающие законы сохранения энергии и импульса. Студент варьирует начальные скорости и их направления, прицельное расстояние, размеры и массы и получает с экрана дисплея всю информацию о конечных результатах столкновения.

Моделирующая программа, реализующая законы электростати-

ческого поля, позволяет студенту, задав положение, знаки и величины зарядов, наблюдать на экране картину силовых линий и эквипотенциальных поверхностей.

Модель дифракции монохроматического излучения составляет уравнения, выражающие распределение интенсивности света в зависимости от угла дифракции, числа, размеров и расположения щелей.

В дисплейном классе кафедры непрерывно занимаются не только студенты, но и много школьников города Минска от первого до десятого класса. Вначале школьники изучают программирование на языке BASIC, главным образом его графические возможности. Для этого разработано специальное методическое обеспечение с большим числом программ.

После освоения начал программирования школьники и студенты под руководством преподавателей сами создают простые программы для моделирования явлений физики, часто с элементами игры. Так школьниками и студентами младших курсов созданы программы, которые моделируют движение тел в однородном поле тяжести, построение изображений в зеркалах и линзах, колебание различных маятников, электрические процессы в простых линейных цепях и другие физические явления.

Распространенным является мнение, что дети работе предпочитают компьютерные игры за терминалом. Опыт работы дисплейного класса кафедры технической физики опровергает этот стереотип. Наблюдается обратное. Освоив основы графического программирования и почувствовав свои реальные возможности решать интеллектуальные задачи, большинство учащихся предпочитают компьютерным играм захватывающие возможности математического моделирования.

Новый вариант учебного процесса реализован на кафедре технической физики для школьников, студентов и аспирантов, для переподготовки инженеров и преподавателей.

В работе изложена лишь часть прикладных программ, реализованных в дисплейном классе кафедры технической физики ВПИ. Некоторые другие детали рассмотрены в [1-3].

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Севастьянов В.Г., Сидорик В.В., Смугага Л.Н. Моделирование физических процессов с использованием интерактивной графики // IX Зональная методическая конференция преподавателей общей физики высших учебных заведений Белоруссии, республик Прибалтики и Калининградской области РСФСР: Тезисы докладов. Минск. 1986.

2. Орлов В.И., Севастьянов В.Г., Смугага Л.Н. Графический дисплейный класс математического моделирования физических процессов // Перспективы развития и проблемы эффективного использования ЭВМ общего назначения и персональных ЭВМ: Тезисы докладов научно-технической конференции. Минск. 1987.

3. Севастьянов В.Г., Головейко А.Г., Сидорик В.В., Мартынихина В.И. Расширение возможностей самостоятельной работы по физике на основе применения ЭВМ // Самостоятельная работа студентов в условиях перестройки учебного процесса. Челябинск. 1988.

УДК 37.01:007

Р. Я. Чевере

Вычислительный центр при ЛГУ П. Стучки

НОВЫЕ ВОЗМОЖНОСТИ АВТОМАТИЗИРОВАННОГО ОБУЧЕНИЯ С ПРИМЕНЕНИЕМ ЭКСПЕРТНЫХ СИСТЕМ

В разработке автоматизированных систем обучения (АСО) можно выделить несколько этапов, которые тесно связаны с развитием вычислительной техники и программирования в целом. Исследования в области искусственного интеллекта и развитие работ по созданию экспертных систем (ЭС), естественно, породило желание применять эти идеи и в области обучения. В настоящей статье даю краткий исторический обзор развития идей интеллектуализации АСО, особенно выделяются новые возможности, которые предоставляют ЭС.

1. Основные этапы разработки АСО

Первые шаги в создании АСО связаны с использованием универсальных языков программирования. Для этих систем характерно предъявление учебного материала студенту в зависимости от его ответов, а также организация некоторой обратной связи. При этом самой тяжелой проблемой оказалось создание и приведение к необходимой форме учебного материала, так как это является исключительно трудоемкой задачей. Это привело к разработке специализированных языков создания курсов машинного обучения [1].

В семидесятые годы наблюдается переход на новый, более сложный уровень в разработке АСО. Появилась возможность в некоторых проблемных областях таких, как, например, арифметика, формировать обучающий материал машинным путем. Кроме того, эти системы были способны отвечать на определенные вопросы студентов. С тех пор развитие АСО, в основном, было

направлено на расширение функциональных возможностей обратной связи и увеличение степени индивидуализации учебного курса. Однако уровень знаний по изучаемой проблеме, заложенный в АСО, остался довольно низким. В АСО не содержались знания, свойственные человеку. Кроме того, эти системы не могли отвечать на вопросы студентов типа "как?", "почему?" и т.д.

Таким образом, одним из основных недостатков традиционных АСО является отсутствие модели пользователя и модели процесса принятия решений. Это особенно чувствуется в тех учебных курсах, которые имеют концептуальный характер, и в которых необходимо использовать знания о процессе принятия решений. Кроме того, традиционные АСО не могут обосновать или пояснить принятые решения и сделанные выводы [2].

Разработка т.н. интеллектуальных обучающих систем направлена на устранение перечисленных недостатков. Идеи искусственного интеллекта открывают путь к разработке модели пользователя и реализации гибкого диалога между человеком и системой. В области искусственного интеллекта в последние годы большое внимание уделяется исследованию и разработке ЭС. Поэтому естественными являются поиски интеллектуализации АСО с помощью ЭС.

Согласно определению Э.Фейгенбаума, ЭС - это "интеллектуальная программа, которая использует знания и процедуры вывода для решения достаточно сложных проблем, требующих использование человеческой экспертизы. Знания, необходимые для выполнения действия, совместно с использованными процедурами вывода могут быть рассмотрены как модель экспертизы, выполняемой лучшими практиками данной области" [2].

Применительно к АСО, можно создать ЭС, которая формирует модель пользователя и действует согласно этой модели.

Как известно, основными составными частями ЭС являются база знаний, механизм вывода и интерфейс с пользователем. Кроме того, каждая ЭС содержит более или менее развитые средства пояснения процесса принятия решения и средства формирования базы знания. Все эти компоненты могут быть использованы в АСО.

2. Место и роль ЭС в автоматизированном обучении

Прежде, чем рассматривать построение АСО с применением ЭС, кратко остановимся на основных способах построения ЭС. Здесь наблюдаются два основных подхода.

Во-первых, главное внимание можно уделять построению новой концепции базы знаний и механизма вывода, наиболее подходящих для данной проблемной области. В этом случае определяющими являются свойства и структуры, найденные и использованные при извлечении, приобретении и формализации знаний. В этом случае реализация на ЭВМ является подчиненной и действует в качестве ограничений только в такой мере, что каждая новая форма представления знаний в принципе должна быть реализуема на ЭВМ. Этот путь является сложным и трудоемким, так как требует разработки всего программного обеспечения ЭС.

Во-вторых, работы по созданию ЭС могут быть ориентированы на использование готовых программных средств - т.н. оболочек ЭС. В этом случае реализация доминирует над всеми остальными этапами создания ЭС, так как сбор, приобретение и формализация знаний обязательно должны привести к форме представления, определяемой данной оболочкой.

В настоящей статье рассматриваются некоторые новые возможности, предоставляемые автоматизированному обучению в случае его интеллектуализации с помощью ЭС, построенных, в основном, на базе готовых оболочек.

Использование готовой оболочки намного сокращает трудоемкость создания ЭС. Естественно, этот подход применим только тогда, когда доступная оболочка ЭС в достаточной мере подходит для данной проблемной области:

Возможны два варианта использования ЭС при автоматизированном обучении. Во-первых, саму ЭС в явном виде можно использовать как средство обучения. Во-вторых, ЭС как составная часть может быть включена в более сложную АСО. На рис. 1 [3] показаны наиболее важные сферы возможного применения ЭС. Такими являются создание учебного курса, планирование учебного процесса, моделирование поведения пользователя, принятие решений и т.д. Рассмотрим эти применения несколько подробнее.

Основные сферы применения ЭС

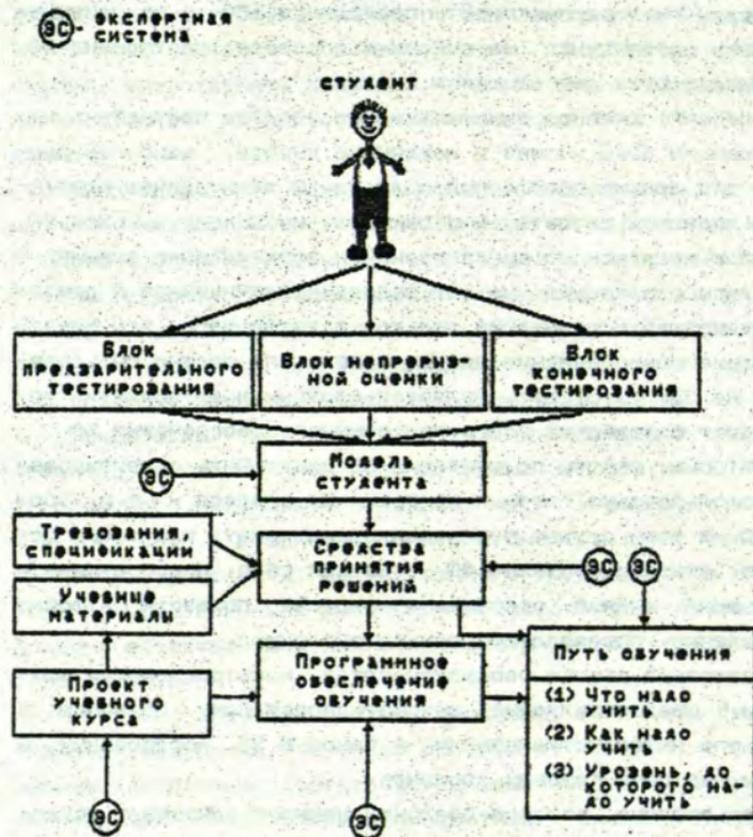


Рис. 1. Использование экспертных систем (ЭС) в автоматизированных системе обучения (АСО).

Создание учебного курса - это один из наиболее сложных, трудоемких и трудно формализуемых этапов создания АСО. Если по аналогии рассматривать процесс создания ЭС, то здесь выделяется этап приобретения знаний. В области ЭС этому этапу уделяется большое внимание: исследуется поведение экспертов, разрабатывается методика извлечения знаний, методика поэтапной формализации знаний и разрабатываются вспомогательные

программные средства приобретения знаний как в составе ЭС, так и в виде самостоятельного инструмента. Очевидно, что здесь имеется много общего с построением учебного курса для АСО. В ЭС могут быть заложены различные уровни знаний по данному курсу: знания по изучаемой проблеме, а также мета-знания, т.е. знания о знаниях.

На различных этапах создания учебного курса можно использовать:

1) методику и средства построения модели экспертных знаний, которые предназначены для систематизации, документирования и поэтапного преобразования к внутрисистемному представлению знаний, получаемых из всевозможных источников: в случае АСО такими источниками являются учебники, справочники, методические указания, и, что является самым сложным, - опыт и субъективные знания опытных преподавателей;

2) средства автоматизации процесса приобретения знаний.

Известны реализации средств автоматизации различной сложности. Одно направление - это разработка удобных, развитых текстовых редакторов. Примером такого является система **ОВИЕ:КНОВЕ** [3] (предметно - ориентированный интеллектуальный редактор: редактор базы знаний), который включает следующие составные части: графический редактор, редактор когнитивных связей и редактор упорядочения панелей. **ОВИЕ:КНОВЕ** является интеллектуальным средством, базирующимся на знаниях, и предназначенным для построения учебного курса. Более сложные системы представляют собой рабочее место инженера по знаниям. Примером таких систем является система **ETS** и ее расширенная версия **AQUINAS**, которые разработаны как средства приобретения знаний для широкого спектра ЭС [4].

Из отечественных разработок к средствам автоматизации приобретения знаний относится, например, система **АКВИЗИТОР**, которая базируется на анализе понятийной структуры.

Сложность планирования учебного процесса заключается в том, что требуется одновременное соблюдение различных, в том числе и противоречивых ограничений. Облегчить эту работу может созданное на базе ЭС рабочее место-консультант, который имеет сведения по целому ряду вопросов: наличие ресурсов, их образовательные достоинства и эффективность, их свойства,

существенные для планирования учебного процесса, местонахождение ресурсов, стоимость их использования, технические требования, организация доступа, возможное объединение ресурсов, завершенность и целостность учебных материалов и т.д.

В индивидуализации учебного процесса важную роль играет модель пользователя. Здесь также большой потенциал имеют ЭС и инженерия знания. Модель пользователя может быть разработана с различных точек зрения. В системном проектировании различают следующие виды моделей: 1) модель пользователя с точки зрения проектировщика системы; 2) пользовательское представление о прикладной задаче; 3) пользовательское представление о системе, применяемой для решения прикладной задачи; 4) модель пользователя, заложенная в систему. В обучающих системах разработчик системы должен учитывать требования четырех классов пользователей. Это проектировщики учебных ресурсов, разработчики ресурсов, преподаватели и студенты. Для моделирования первых трех классов пользователей сделано относительно мало - в существующих работах основное внимание уделяется поведению студента. При этом целью является обеспечение оптимального соответствия обучающего курса физическим и когнитивным потребностям каждого конкретного студента. Для построения модели студента могут быть использованы различные средства инженерии знания. Наиболее простым способом является построение набора правил типа IF-THEN и базы данных, содержащей значения атрибутов. База данных заполняется на основе предварительной информации и дополняется динамически во время проведения диалога пользователя с системой. Основной вклад ЭС в моделирование пользователя ожидается именно от использования декларативных и процедурных средств ЭС, освобождающих пользователей от сложного и трудоемкого программирования.

Кроме рассмотренного выше применения ЭС как составной части АСО, возможно использование ЭС, разработанных специально для обучения, - в основном, систем-советчиков. Так, при существующей за рубежом системе высшего образования, на базе ЭС разрабатывается система-советчик, которая помогает студенту выбрать набор учебных дисциплин, определить глубину их изучения и т.д. в зависимости от осваиваемой специаль-

ности и получаемой степени [3].

3. Примеры построения интеллектуальных АСО

В [5] описываются основные принципы создания системы КВЕТ, которая предназначена для обучения инженеров, занимающихся устройствами с числовым программным управлением. В проекте КВЕТ преследуется цель создания интеллектуальной оболочки для инженерного обучения общего назначения. Аналогично вышерассмотренным работам в составе интеллектуальной обучающей системы выделяются три основные компоненты знания: знания об изучаемой проблеме, знания о студенте и знания о процессе обучения. Однако использование готовых оболочек ЭС в обучении имеет свои особенности. По своему характеру знания, необходимые для решения некоторой проблемы, довольно существенно отличаются от знаний, необходимых для проведения процесса обучения. Представление знаний в ЭС, в свою очередь, существенно отличается от описания учебных курсов в традиционных АСО. Последние в большинстве случаев организованы по принципу электронного пролистывания страниц. В интеллектуальной системе обучения элементы знаний связываются между собой различными способами, которые могут включать несколько уровней метаописаний. Следовательно, система не только хранит знания о предмете, но и умеет эти знания использовать для решения проблемы в данной области.

В КВЕТ представляются общие инженерные и специфические знания, связанные с функционированием устройств с числовым программным управлением.

Система КВЕТ предназначена для работы различных пользователей: одних интересует просмотр вложенных в КВЕТ возможностей, других - получение совета, третьих - обучение. Поэтому проведена классификация возможных пользователей по типу, уровню подготовленности и потребностям, и разработаны два вида моделей для каждого типа пользователей. Первая модель отражает знания пользователя о различных аспектах инженерного обучения. Вторая модель - представление о потребностях пользователя. Модели описывают некий стереотип пользователя. Описание не является строго фиксированным, его можно на-

полнять и изменять как со стороны пользователя, так и системы. Кроме того, реализованы несколько форм общения пользователя с системой. Одни пользователи хотят иметь возможность полностью управлять диалогом, особенно при поиске или просмотре, других устраивает диалог, управляемый системой. Упрощенной формой такого диалога является стандартная форма взаимодействия с ЭС, в которой после ответа пользователя на ряд вопросов выдается решение или некоторые рекомендации. Дополнительные возможности общения заложены в средства пояснения решений. В КВЕТ разработана более сложная и разнообразная форма диалога.

Рассмотренный проект предусматривает создание всего программного обеспечения обучающей системы. Этот путь является трудоемким, но обеспечивает более полное удовлетворение требований пользователя.

Другой подход - это "быстрее" создание интеллектуальной обучающей системы путем интеграции существующей АСО и ЭС. В [6] приводится описание архитектуры, основных проблем и опыта создания такой системы на базе АСО "AUTHOR" и оболочки ЭС "EST". При решении проблемы интеграции важным моментом является определение точек, в которых системы обмениваются данными или управляющей информацией. Предлагается обмен данными организовать на уровне файла, в котором содержатся знания о пройденных каждым студентом курсах, оценке его знаний, прохождении тестов соответствия и т.п. На основе этой информации с помощью ЭС определяется дальнейшая стратегия обучения. Все остальное время каждая компонента может работать самостоятельно.

4. Новые методические возможности в обучении на базе ЭС

Все вышерассмотренные работы направлены на включение ЭС в состав АСО с целью их интеллектуализации. Однако методически процесс обучения организуется аналогично традиционному АСО, т.е. в систему закладываются знания о данной проблемной области и в более или менее интеллектуальной форме организуется прохождение учебного курса. Выдача информации чередуется с ответами на вопросы, выполнением учебных заданий и оцен-

кой уровня знаний.

В [7] предлагается совершенно новый подход к применению ЭС в автоматизированном обучении. Главная идея этого подхода заключается в тезисе: наиболее эффективный способ чему-то научиться - это учить других. В данном случае в качестве обучаемого объекта предлагается использовать ЭС. Другими словами, проектирование базы знаний и ее реализация с применением оболочки ЭС являются многообещающими средствами обучения. Конечно, сомнения о мере их универсальности являются справедливыми. Методика, основанная на практическом опыте работы [7], заключается в следующем. Группа студентов разбивается на подгруппы по два-три человека, и каждой из них дается задание - создать ЭС по теме изучаемой дисциплины, используя готовую оболочку ЭС. Студенты должны выбрать экспертов, определить ресурсы и источники знаний, реализовать базу знаний в рамках оболочки и проверить правильность принятых на ее основе решения.

Выполняя такое задание, студенты должны:

- 1) определить источники информации, структурировать и преобразовать знания в форму правил;
- 2) разработать и применить способ извлечения экспертных знаний;
- 3) во время проверки созданной ЭС выявить и путем коллективного обсуждения устранить найденные неточности.

Как показывает опыт, такой способ обучения является более активным, он позволяет студенту не только освоить данную проблему, но и учит, как более ясно и логично мыслить, точно формулировать свои мысли, делать выводы и умозаключения. В традиционном обучении эти навыки развиваются в значительно меньшей степени. Конечно, при такой форме обучения новые требования предъявляются к преподавателю. Он должен умело, в зависимости от предварительной подготовленности слушателей, провести начальное обучение по использованию оболочки ЭС, определить начальную стратегию разработки проекта.

Конечно, метод имеет и свои недостатки. Во-первых, это трудоемкий способ обучения. Так, по приведенным данным, разработка базы знаний объемом 12-15 правил потребовало от 25 до 60 часов. Во-вторых, это предъявляет жесткие требования к

уровню подготовки преподавателей. Они должны поддерживать правильное направление работы и в любой момент дать советы и оказать помощь. В-третьих, оболочка ЭС должна быть легко осваиваемой, "дружелюбной" и содержащей некоторые средства для создания базы знаний, по крайней мере, на уровне хорошего текстового редактора.

В принципе этот подход применим в любой сфере обучения - как при изучении грамматических правил, так и при решении задач по физике.

З а к л ю ч е н и е

Разработка АСО является одной из тех областей, в которых интеллектуализация программных средств может обеспечить качественно новый уровень функционирования. Учитывая достаточно большой опыт в области исследования и разработки ЭС, целесообразными являются работы по их интеграции в состав АСО. Основными являются следующие направления работ.

1. Использование накопленного опыта, разработанных средств и методики приобретения знаний для разработки учебных курсов.

2. Разработка специализированных ЭС как составных частей АСО. При этом возможна как разработка оригинального программного обеспечения, так и интеграция ЭС, разработанной на базе готовой оболочки, в состав АСО.

3. Самостоятельное использование ЭС как средств обучения. В данном случае возможны три основных варианта:

- каждая ЭС, которая создана как более или менее законченный продукт, содержит средства обучения по использованию и механизм пояснения вывода, например, в форме ответа на вопросы типа "как?" и "почему?". Этот режим можно использовать для обучения по заложенной в базу знаний проблеме;

- базу знаний ЭС можно заполнить с ориентацией непосредственно на обучение. Назначение системы может быть различным. Можно создать систему-эксперт по изучаемой проблеме, и использовать ее для контроля правильности решения обучаемого. Можно создать экспертную систему-советчик, которая предоставляет информацию, дает советы и направляет ход процесса

обучения:

- процесс создания ЭС требует тщательного изучения проблемной области и классификацию знаний. Приходится также построить некую модель процесса мышления. Очевидно, что все это способствует освоению данного предмета и может быть использовано как своеобразная форма обучения. т.е. ЭС используется в качестве объекта, который нужно обучать.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Yazdani M. Intelligent tutoring system survey // Artificial Intelligence Review. 1986. No. 1. Pp. 43-52.
2. Shaw K. The application of expert systems principles to teaching and training // Technology Based Training. 1986. Pp. 123-131.
3. Barker P. Expert systems in engineering education // Engineering Applications of Artificial Intelligence. 1988. Vol. 1. March. Pp. 47-58.
4. Kitto C.M. Knowledge acquisition tools for different problem-solving paradigms: research at Boeing Computer Services // Cognitive Engineering in the Design of Human-Computer Interaction and Expert Systems. 1986. Pp. 515-522.
5. Rivers R., Kriss B. Knowledge-based engineering training // Computer-Aided Engineering Journal. 1987. Vol. 4. No. 1. Pp. 68-70.
6. Crews P. tbt EXPERT: A case study in integrating expert systems technology with computer assisted instruction // Proceedings of Third International Conference on Data Engineering. 1987. Pp. 556-562.
7. Trollips S.R., Lippert R.C. Constructing knowledge bases: a promising instructional tool // Journal of Computer-Based Instruction. 1987. Vol. 14. No. 2. Pp. 44-48.

УДК 37.01:007+681.3.06

А.Р.Спектор, Р.Я.Чевере

Вычислительный центр при ЛГУ им. П. Стучки

НЕКОТОРЫЕ ВОЗМОЖНОСТИ ПРИМЕНЕНИЯ ПЭВМ ПРИ ОБУЧЕНИИ ЛАТЫШСКОМУ ЯЗЫКУ

Появление и широкое распространение персональных ЭВМ позволяет их использовать во многих областях, в том числе и в автоматизированном обучении. Малые габариты, простая операционная система, широкий набор программного обеспечения, которое в основном использует гибкий и развитый диалог, делает эти машины привлекательными для различных пользователей, не связанных непосредственно с вычислительной техникой. Эти преимущества следует использовать и при создании автоматизированных систем обучения. В настоящей работе рассматриваются некоторые возможности применения ПЭВМ при обучении латышскому языку, используя программное обеспечение экспертных систем (ЭС).

1. Назначение и сферы применения

Основная идея разработки заключается в следующем: на базе оболочки ЭС продукционного типа создать систему-справочник по грамматике латышского языка и русско-латышский и латышско-русский словари с ограниченным набором слов. Учитывая технические ресурсы ПЭВМ, быстродействие и возможности программного обеспечения ЭС, было решено ограничиться одной из сфер применения ПЭВМ - делопроизводством.

Закладываемые в систему функции делятся на две части, условно называемые "словарь" и "грамматика".

1.1. "Словарь" содержит русско-латышский и латышско-русский словарь терминов делопроизводства. При разработке макета используется набор слов, отобранных авторами из общего словаря и относящихся к делопроизводству. После разработки согласованного словаря, этот набор будет корректироваться. Функционально словарь делится на четыре уровня. Для всех включенных в словарь терминов дается перевод. Эта часть словаря содержит только специальные термины делопроизводства. Выделен также набор тем, соответствующих основным документам, используемым в делопроизводстве: письмо, приказ и т.д. Для каждой темы даются переводы: 1) термина; 2) группы атрибутивных словосочетаний; 3) группы глагольно-именных словосочетаний; 4) конструкция; 5) примеров. В конструкциях приводятся наиболее употребительные схемы предложений с заглавным словом темы. Примеры представляют предложения, в которых содержатся наиболее употребительные конструкции. Пример содержания многоуровневой части словаря для термина "план" имеет следующий вид:

ПЛАН - PLĀNS

1. намеченный	:		:	ieskicēts
общий	:		:	vispārējs
подробный	:		:	sīki izstrādāts
предложенный	:	план	:	ierosināts
развернутый	:	plāns	:	izvērsts
утвержденный	:		:	apstiprināts
четкий	:		:	precīzs
	:	работы	:	darba
план	:	отдела	:	nodalas ; plāns
	:	строительства	:	celtniecības ;
план	:	развития	:	attīstības ; plāns
	:	исследования	:	pētījumu ;

2. выполнить	:	:	izpildīt
изложить	:	:	izklāstīt
обсудить	:	:	apspriest
одобрить	:	план	atzīt par labu
осуществить	:	плану	realizēt
предложить	:	:	ierosināt
разработать	:	:	izstrādāt
рассмотреть	:	:	izskatīt
составить	:	:	sastādīt

3. а) на основе плана (чего) uz (ka, kāda) plāna
разрабатывается (что) pamata tiek izstrādāts (kas)
б) при составлении плана sastādot plānu, tiek ņemts
учитывается, (что) vērā (kas)
4. План был обсужден и одобрен Zinātniskās padomes sēdē
на заседании ученого совета plānu apsprieda un atzina
par labu

1.2. Подсистема "грамматика" содержит основные сведения по грамматике латышского языка и предназначена для работы в качестве консультанта по общим понятиям языка, конкретным грамматическим правилам или конкретному вопросу правописания. При структурировании грамматического материала за основу взята работа [1] и набор терминов делопроизводства. Последнее означает, что грамматические ситуации, которые в данном подмножестве слов не встречаются, в отобранный материал не включаются. Например, среди терминов делопроизводства практически нет имен существительных мужского рода с окончанием -уа (третье склонение). Следовательно, правила описания форм имен существительных третьего склонения в грамматический материал можно не включать.

Разработка ориентирована, в первую очередь, на пользователей, для которых латышский язык не является родным. Она может быть использована в качестве консультанта, а также в процессе обучения или самообучения для освоения правил и тренировки правописания.

2. Реализация подсистем

на базе оболочки ЭС продукционного типа

Использованная оболочка ЭС содержит продукционные правила как единственную форму представления знаний. Для работы системы необходима ПЭВМ со следующей конфигурацией: ОЗУ - 256 К и один накопитель на гибких магнитных дисках. Используются три алфавита - латинский, латышский и русский.

Каждое правило может содержать одно или несколько утверждений в IF части, а также одно или несколько предложений в THEN части. Кроме того, с каждым правилом связаны примечание и ссылки для дополнительной информации. Для формирования правил необходимо усвоить несколько основных понятий.

УСЛОВИЕ - это утверждение в виде текста на естественном языке или алгебраическое выражение.

Текстовое условие формируется из двух частей - КВАЛИФИКАТОРА (описателя) и одного или нескольких ЗНАЧЕНИЙ. Квалификатор представляет собой начальную часть предложения и обычно заканчивается глаголом. Любое из значений должно формировать правильное продолжение этого предложения. Алгебраические выражения могут содержать переменные, знаки операций и вызовы стандартных функций. При этом следует отметить, что в данной оболочке могут быть только числовые переменные.

Следующим из основных понятий является ВАРИАНТ. Варианты представляют собой все возможные решения проблемы, которые выдаются пользователю как результат. Особенностью варианта является то, что он выводится вместе с вероятностью его наступления.

В THEN части правила может быть предложение, алгебраическое выражение или вариант. В данной реализации результатом является текст на естественном языке. Если предложение в THEN части сформировано из квалификатора, то оно используется при внутренней обработке правил, но не выдается на экран как результат. Вариант, в свою очередь, влечет за собой вывод вероятности появления, что в данном случае неприемлемо. Поэтому для выдачи чистой текстовой информации как результата приходится ее задавать в виде поясняющего текста к переменной. Несмотря на перечисленные ограничения, данная

оболочка позволяет реализовать макеты как подсистемы "грамматика", так и "словарь". Пример грамматического правила в формате, заданном оболочкой, выглядит следующим образом:

Rule number 3

IF:

(1) Вас интересует принадлежность существительного к склонению OR склонение конкретного существительного

AND (2) существительное мужского рода

AND (3) существительное в форме NOMINATIVS (именительный падеж) ЕДИНСТВЕННОГО числа имеет окончание -is

THEN

(1) существительное принадлежит второму склонению.

AND (2) существительное принадлежит ВТОРОМУ склонению

AND (3) например, существительное KANTORIS, SKAITLIS

Для записи такого правила необходимо сформировать следующие квалификаторы:

Qualifier #1

Вас интересует

- 1 основные правила склонения существительных
- 2 принадлежность существительного к склонению
- 3 склонение конкретного существительного
- 4 формы существительного определенного склонения

Qualifier #2

Существительное

- 1 мужского рода
- 2 женского рода

Qualifier #3

Существительное в форме NOMINATIVS (именительный падеж) ЕДИНСТВЕННОГО числа имеет окончание

- 1 - s
- 2 - is
- 3 - us
- 4 - a
- 5 - e
- 6 - s

В ТНЕН части данного правила одно утверждение содержится два раза. Первое предложение сформировано на основе квалификатора и предназначено для использования в других правилах. Второе записано как поясняющий текст к переменной и обеспечивает вывод результата на экран.

Система обрабатывает правила последовательно в порядке возрастания их номеров. Если для обработки очередного утверждения не хватает информации, то выдается запрос в форме соответствующего квалификатора.

Процесс приобретения знания для данной системы включает анализ курса грамматики, выбор и формулировку тех правил, которые целесообразно поместить в базу знаний, определение входных и выходных данных и разработку сценариев взаимодействия пользователя с системой.

Грамматический материал, в основном, состоит из общедоступных знаний, которые можно получить из книг, справочников и других источников. Непосредственное участие эксперта - специалиста по языку и обучению, требуется по отдельным вопросам отбора материала и организации последовательности его изложения.

В каждое правило закладываются два вида знаний.

1. Знания, определяющие потребность в данном материале и регулирующие запрос информации у пользователя (т.н. метазнания). В приведенном примере знания этого вида содержатся в первом утверждении IF части правила.

2. Знания, относящиеся непосредственно к грамматике латышского языка. В приведенном примере такие знания содержатся во всех остальных утверждениях IF части.

Фрагмент структурирования грамматического материала, закладываемого в базу знаний, приводится на рис. 1. Учитывая возможности данной оболочки и желаемое время реализации, каждая тема оформляется в виде отдельной базы знаний. Информация об этом делении, в свою очередь, содержится в отдельной базе знаний. Такая организация позволит упростить правила по каждой отдельной теме, уменьшит объем файлов, обеспечивает более гибкую реорганизацию и расширение каждой от-

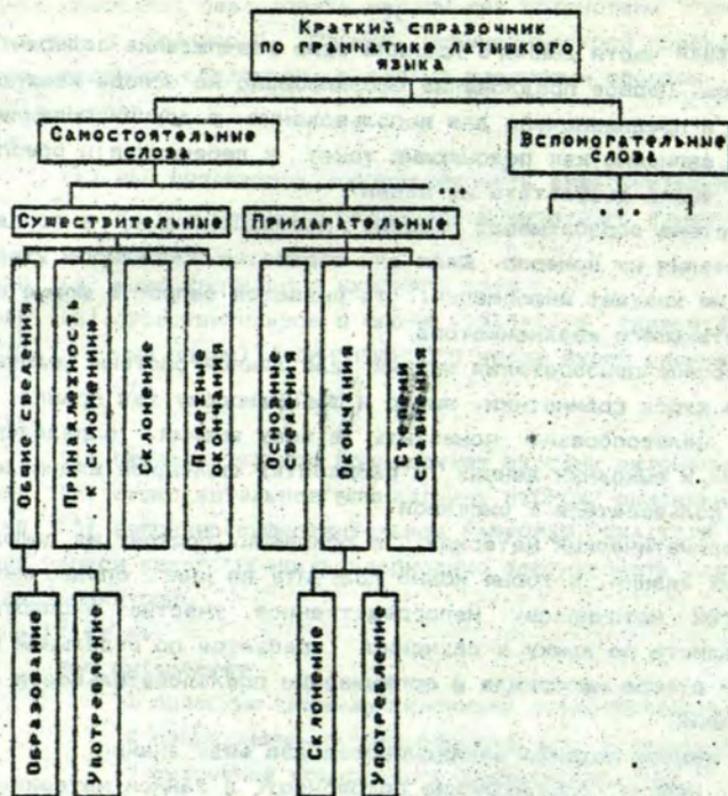


Рис. 1. Фрагмент структуризации темы.

дельной темы и сокращает диалог с пользователем. В случае наращивания отдельной подтемы возможна вложенная реализация баз знаний, т.е. некоторая ветвь может заканчиваться информацией о том, в какой базе следует искать более подробное изложение интересующего вопроса.

В настоящее время на ПЭВМ "Искра-1030" для подсистемы "грамматика" реализован макет темы "имя существительное". Текст правил дается на русском и латышском языках.

Для подсистемы "словарь" реализован макет первого уровня. Здесь используется аналогичная форма правил. В одно правило объединяются все термины, которые начинаются на данную

букву. При большом количестве терминов возможно их разбиение на подгруппы, например, по первым двум буквам. Правило для выдачи группы слов выглядит следующим образом.

Rule number 2

IF:

(1) Выбор слова производится (vārda izvēle notiek)
по алфавиту (alfabeta kārtība)

AND (2) исходный термин дан (sākotnējs terminš dots)
на русском языке (krievu valodā)

AND (3) слово начинается на букву Б

THEN

(1) багаж - bagāža

AND (2) база - bāze

AND (3) баланс - bilance

AND (4) бандероль - bandrole

AND (5) чистовик - tīrāksta

Аналогичным образом строятся правила и для остальных уровней словаря.

3. Обучающие возможности программного обеспечения ЭС

Согласно основным принципам создания ЭС, в каждую оболочку закладываются средства, которые можно использовать в целях обучения. В разработанном макете эти средства обеспечивают работу пользователя с базой "грамматика" в следующем виде. Пользователь отвечает на вопросы системы, описывая интересующую его ситуацию, и получает ответ в виде текста на русском языке. Отдельные термины, правильные окончания, примеры и т. д. выдаются на латышском языке. В общем случае, текст, или отдельную его часть можно подготовить на любом из трех языков (английском, латышском и русском). Ниже приводится пример текста, выдаваемого пользователю:

- 1 Существительные второго склонения в падежах имеют следующие окончания: N. kas? -is; G. ka? -a; D. kam? -im; A. ko? -i; I. ar ko? -i; L. kur? -a; V. -i.

2 У существительных ВТОРОГО склонения в форме GENITIVS ЕДИНСТВЕННОГО числа и во ВСЕХ ПАДЕЖАХ МНОЖЕСТВЕННОГО числа происходит чередование согласных перед окончанием, например, lietvedis-lietveža; pārzinis-pārzina; modelis-modela.

3 Существительные ВТОРОГО склонения в форме LOKATIVS ЕДИНСТВЕННОГО числа оканчиваются на ДОЛГУЮ ГЛАСНУЮ (-ā).

После получения такого текста пользователь может получить описание исходного запроса, сформированного на основе введенной им информации. В данном примере выдается следующее описание:

1 Вас интересуют формы существительного определенного склонения

2 Существительное принадлежит второму склонению.

Кроме того, для каждого пункта в тексте результата можно запросить выдачу правил, которые использовались при формировании соответствующего утверждения. В данном примере после набора номера пункта 1 выдается следующее правило:

Rule number 14

IF:

(1) Вас интересуют формы существительного определенного склонения

AND (2) Существительное принадлежит второму склонению
THEN

(1) Существительные второго склонения имеют следующие окончания: N. kas? -is; G. kas? -a; D. kam? -im; A. ko? -i; I. ar ko? -i; L. kur? -ā; V. -i.

Во время работы можно задать такой режим, при котором после ввода определенной порции информации выдаются все правила, которые эту информацию обрабатывают. Это позволяет последовательно прочитать все грамматические правила на родном языке.

Совместное использование всех этих возможностей обеспечивает достаточно удобное обучение, получение консультаций и выполнение упражнения.

Кроме того, данную оболочку можно использовать как объект обучения для реализации методики, предложенной в [2]. Редактор формирования правил достаточно прост в освоении и

позволяет работать с текстами на родном языке. В этом случае целесообразно по заданному набору тем создать эталонную базу знаний. После того, как студент сам составит ряд правил, он имеет возможность сравнить их с правилами, выдаваемыми эталонной базой знаний.

В ы в о д ы

Оболочка ЭС производственного типа может быть использована для быстрого построения средств обучения латышскому языку и изучению грамматики. Базу знаний рекомендуется строить для ограниченного грамматического материала, базируясь, например, на словаре терминов определенной проблемной области.

Основное преимущество заключается в простоте создания базы знаний, что позволяет быстро создавать и настраивать систему на изучаемый материал различного характера.

Используемая оболочка ЭС имеет ограниченные возможности: в ней не предусмотрена работа с символьными переменными. Это не позволяет ввести анализируемое слово, затрудняет формирование выходного текста. Кроме того, выдача информации на экран происходит в строгом соответствии с порядком ее ввода при создании системы. Это затрудняет логическое изложение материала и упорядочение терминов по алфавиту. Очевидно, что при продолжении работ желательно использовать оболочку ЭС производственного типа, имеющую более гибкие возможности работы с символьными переменными, или разработать собственное программное обеспечение, реализующее рассмотренные функции.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Гужане В., Еча А., Спрогис В. Краткий справочник по грамматике латышского языка. Рига: Звайгзне, 1988. 237 с.
2. Trollips S.R., Lippert R.C. Constructing knowledge Bases: a promising instructional tool // Journal of Computer-Based Instruction. 1987. Vol. 14. No. 2. Pp. 44-48.

Межвузовский сборник научных трудов
ЭВМ В ОБРАЗОВАНИИ. ПЕДАГОГИЧЕСКИЕ ПРОГРАММНЫЕ СРЕДСТВА
Рига: ЛГУ им. П. Стучки. 1989. С. 62-65

УДК 37.01:007+681.3.06

Е. Ю. Диркова

НИИ школьного образования и
технических средств обучения
АПН СССР, г. Москва

М. Л. Фокин

НИИ содержания и методов
обучения АПН СССР,
г. Москва

КРИТЕРИИ ЭКСПЕРТНОЙ ОЦЕНКИ
ПЕДАГОГИЧЕСКИХ ПРОГРАММНЫХ СРЕДСТВ

Все расширяющееся оснащение средних школ электронно-вычислительной техникой и установка на компьютеризацию процесса обучения основам наук породили большой интерес к созданию педагогических программных средств (ППС). Создание ППС типа программированных заданий с выбором правильного ответа по силам даже старшеклассникам. Так ввод в компьютер дидактических материалов [1] не требует от исполнителя высокого профессионализма. Однако потребность в оригинальных обучающих программах необычайно высока, и на успех здесь могут рассчитывать авторские коллективы, состоящие, как минимум, из методиста, разрабатывающего сценарий, программиста и учителя-экспериментатора, осуществляющего испытание ППС в учебном процессе. За последние годы количество ППС стремительно возросло, однако этого нельзя сказать о качестве большинства программ - зачастую они оставляют желать лучшего. Низка эффективность как у программ, написанных педагогами с недостаточными знаниями программирования, так и у программ, созданных профессиональными программистами, не владеющими методикой преподавания учебного предмета и не учитывающими психо-физиологические аспекты взаимодействия школьников с ЭВМ. В итоге можно обнаружить множество ППС, которые не оказывают сколь-нибудь заметного положительного влияния на качество обучения. Среди широко распространенных недостатков ППС - неоправданное программирование тривиальных

сюжетов и ситуаций, успешно поддающихся изучению традиционными методами, перегрузка экрана дисплея текстами, однообразие и монотонность деятельности обучаемого. Вместе с тем, имеется ряд ограничений для компьютеризации обучения школьников, и среди них важнейшие - санитарно-гигиенические нормы пребывания детей и подростков перед экраном дисплея. Нельзя не учитывать недостаточную оснащенность школ компьютерной техникой. Кроме того, в учебном процессе должны быть представлены разнообразные виды деятельности учащихся, а не только работа за ПЭВМ. По этим причинам рекомендуемые для массового внедрения ППС должны проходить тщательный отбор, осуществляемый специализированными аттестационными комиссиями. С другой стороны, сегодня признается целесообразным существование альтернативных средств и методик обучения, в том числе компьютерных. Право окончательного выбора методик и средств остается за учителем. Например, по одной и той же теме может быть предложено несколько методически равноценных ППС. В этих условиях остро стоит вопрос о критериях оценки ППС [1].

На наш взгляд, рассмотрение ППС с точки зрения возможности его использования в учебном процессе целесообразно проводить в два этапа.

1. ЭКСПЕРТНАЯ ОЦЕНКА. Достоинство ППС оценивается группой компетентных и авторитетных специалистов по применению ЭВМ как средств обучения.

2. ЭКСПЕРИМЕНТАЛЬНАЯ ПРОВЕРКА. Под ней мы подразумеваем апробацию данного ППС в базовых дисплейных классах.

Ограничимся здесь обсуждением вопроса о том, какими критериями должна руководствоваться группа экспертов. Ситуация, однако, осложняется тем, что существует множество типов ППС: можно выделить программы управляющие, диагностические, тестовые, изменяющие и контролируемые, имитационные, моделирующие, базы данных и т.д. [2]. Каждый из типов реализует соответствующую специфическую дидактическую функцию ЭВМ. И если от имитационной программы мы вправе требовать, чтобы имитируемое явление наглядно изображалось на экране дисплея, и, следовательно, чтобы в должной степени использовались графические возможности ЭВМ, то для программы-теста это требование вообще может не иметь смысла. Этот пример с очевид-

ностью показывает, что требования к программам разных типов могут существенно отличаться.

Следовательно, при разработке критериев оценки ППС можно действовать двумя способами:

- для каждого типа ППС сформулировать свою оценку критериев;
- сформулировать систему критериев, пригодную для всех видов ППС, отобрав в нее наиболее общие, инвариантные критерии.

Слабое место первого подхода - в том, что зачастую конкретную программу сложно отнести к какому-то определенному типу ППС. В одной программе могут сочетаться элементы моделирования, контроля, сообщения нового знания и т.д. Соответственно, окажется затруднительным определить, какую именно из конкретных систем критериев следует применять для оценки такой программы.

Отрицательная сторона второго подхода в том, что в нем не учитывается специфика конкретного вида ППС, к которому принадлежит данная программа.

Изучение большого количества ППС позволило выявить ряд наиболее общих, часто встречающихся у них недостатков, наличие хотя бы одного из которых делает нецелесообразным использование ППС в учебно-воспитательном процессе. Анализ этих недостатков позволяет сформулировать систему общих критериев оценки ППС, пригодную для ППС всех видов.

КРИТЕРИИ ЭКСПЕРТНОЙ ОЦЕНКИ ППС:

1. Соответствие содержанию учебного курса.
2. Соответствие целям учебного процесса.
3. Педагогическая эффективность применения (на уровне предварительного заключения).
4. Наличие диалога между обучаемым и ЭВМ.
5. Техническая надежность работы программы.
6. Качество графики на экране дисплея.
7. Наличие исчерпывающих инструкций для пользователя.
8. Наличие дифференцированного подхода к обучению.

Каждый из критериев, вообще говоря, имеет несколько аспектов. Например, для третьего из них имеет значение оригинальность ППС и преимущества его воздействия по сравнению с

другими средствами обучения. Однако детальный анализ возможных аспектов потребует от экспертов значительных затрат времени. Поэтому в практике работы аттестационных комиссий по экспертизе ППС целесообразно использовать предварительное анкетирование авторов программ по вышеуказанным критериям. Их краткие характеристики ППС позволят выявить наиболее существенные преимущества программ.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Кабардин О.Ф., Кабардина С.И., Орлов В.А. Задания для контроля знания учащихся по физике в средней школе. М.: Просвещение. 1983.
2. Лаутербах Р., Фрей К. Программное обеспечение процесса обучения // Перспективы. Вопросы образования. 1988. № 3. С. 70-79.

**ПРОГРАММНО-МЕТОДИЧЕСКОЕ
ОБЕСПЕЧЕНИЕ ОБУЧЕНИЯ
ИНФОРМАТИКЕ**

УДК 37.01:007+371.3+681.3.06

И. А. Шкерстена

Вычислительный центр при ЛГУ им. П. Стучки

СИСТЕМА ЛОГО ДЛЯ ПЭВМ "УАМАНА MSX-2"

I. "Черепашья графика"

Наличие компьютеров в школе еще не гарантирует их активное использование на занятиях по информатике, поскольку решающий фактор - наличие соответствующего программного обеспечения. Недостаток программных средств, предназначенных для курса "Основы информатики и вычислительной техники" ("ОИИВТ"), превращает его в курс программирования на каком-либо конкретном языке [1], выбор которого обусловлен типом вычислительной техники, имеющимся или доступным в школе. Однако, выработка умения программировать - отнюдь не основная цель курса "ОИИВТ", а лишь один из практических аспектов курса, который предполагает также приобретение навыков работы с готовым программным обеспечением. Кроме практического необходимым является и теоретический аспект, направленный на формирование алгоритмической культуры учащихся, развитие навыков анализа и формализации предметных задач, ознакомление с такими понятиями, как исполнитель, алгоритм, структура алгоритма, типы величин и т.д. [1].

В первой части курса "ОИИВТ" [2] рассматриваются основные этапы решения задач на ЭВМ, причем уделяется большое внимание обучению алгоритмизации. В новом пробном учебном пособии [3] для объяснения понятия алгоритма используются конкретные исполнители - Чертежник, Робот, Черепашка. Алгоритмы управления этими исполнителями записываются на алгоритмическом языке. Выполнить их школьники могут в основном только на бумаге, "вручную". Если в школе имеется вычислительная техника, возможна практическая реализация алгоритмов с использованием исполнителей, имеющих простую систему ко-

манд для кодировки алгоритмов, например, графическая система ШПАГА для ПЭВМ "Агат" [4], а также система "Джойстик" для микроЭВМ "Искра-226" [5], в состав которой входят два исполнителя - "таракан" и "корректор".

Однако большинство исполнителей не позволяют реализовать "дословный" переход от алгоритма, записанного на алгоритмическом языке, на входной язык исполнителя, поскольку не всегда имеются аналоги соответствующих алгоритмических конструкций. В идеальном варианте входной язык исполнителя, кроме графических возможностей, должен содержать все конструкции алгоритмического языка [3]: команды цикла и ветвления, команды ввода-вывода информации, возможность выполнения арифметических и логических операций, возможность работы с числовыми, литерными и табличными величинами.

Все эти возможности обеспечивают языки программирования высокого уровня. Однако, во-первых, нереально параллельное изучение в неспециализированной общеобразовательной школе алгоритмического языка и одного из языков программирования. Во-вторых, для всех учащихся освоение программирования обязательно.

Учащимся необходимо предоставить средство, требующее минимальных затрат времени на освоение и дающее возможность исполнить и проверить алгоритмы, записанные на алгоритмическом языке. Таким средством, практически полностью отвечающим перечисленным требованиям является система Лого.

Компьютерная система обучения Лого разработана в 1969 г. под руководством Сеймура Пейперта (лаборатория искусственного интеллекта Массачуссетского технологического института (США)). Система Лого практически используется в учебных заведениях США, Англии, Дании, Польши и др. [6-11]. В проектах компьютеризации образования нашей страны также предполагается применение системы Лого [12, 13].

Разработанная автором система реализует обучающую среду Лого, т.н., среду "черепаший графики" для ПЭВМ "Yamaha MSX-2" с системой команд на русском языке и возможностью перевода основного словаря исполнителя - "черепашки" на другой язык диалога.

Основная цель статьи - проиллюстрировать использование обучающей среды "черепаший графики" при преподавании курса

"ОМИВТ" [3].

В статье приводятся процедуры на языке Лого, которые являются реализацией алгоритмов задач из учебника [3] для исполнителя Черепашки.

1. Лого - что это? Обучающая среда Лого

Во-первых, Лого - язык программирования, настолько простой, что он легко осваивается дошкольником, и настолько мощный, что и для опытного программиста в нем найдется много интересного. Во-вторых, Лого - философия обучения, система взглядов на процесс и организацию обучения [14], которые отличаются от традиционных.

Философия языка программирования Лого [15] базируется на психологических исследованиях, связанных с развитием навыков мышления у детей. Ключевая идея: дети - не пассивные обучаемые, а активные "строители" своих интеллектуальных инструментов. Учителя младших классов, разделяя это мнение, высоко оценивают роль игры в развитии детей. В среде Лого, играя с черепашкой, дети решают достаточно абстрактные задачи с помощью своих интуитивных знаний.

Психическое развитие ребенка определяет та социальная среда, в которой он действует. Из этой среды черпаются понятия, схемы, способы деятельности и т.д. без специально организованного обучения. Среда определяет как специфику, так и границы развития ребенка.

Например, усвоение ребенком родного языка происходит естественным путем непосредственно из среды, окружающей его. Чем же характеризуется среда, в которой ребенок учится своему первому языку? Существенно, что среда насыщена "материалами", соответствующими элементам языка, которые дети могут использовать для исследования, экспериментов, а также полны людьми - пользователями языка, ответы которых помогают детям убедиться в правильности или неправильности своих предположения о том, как "работает" язык. Изучаемый язык дает и вознаграждение - желания детей выполняются быстрее и точнее с каждым успешным шагом в освоении языка [16]. Исходной посылкой создателей Лого являлось использование представлений об естественной среде при создании соответствующих сред для

изучения школьных дисциплин: математики, естественных наук, музыки.

Объекты среды Лого - это компьютер ("математически говорящее средство") и черепашка ("кибернетическое животное"), управляемое посредством компьютера. В начале это был полусферический робот, передвигающийся по полу или по большому листу бумаги [15], теперь - небольшой указатель на экране дисплея компьютера, рисующий фигуры в соответствии с программой на языке Лого, составленной учащимся. В процессе обучения ученик "беседует" с черепашкой и компьютером. В отличие от традиционной организации обучения, в среде Лого не компьютер управляет учащимся, а учащийся "обучает" компьютер, отдавая ему "приказы" на языке Лого.

Язык программирования Лого составляет основу среды Лого, обычно термин "язык Лого" понимают как синоним "среды Лого".

Разработчики Лого базировались на трех принципах [15]:

- 1) преэмптентность - язык, на котором дети программируют, должен находиться во взаимосвязи с их личными знаниями;
- 2) мощность - язык предоставляет детям возможность реализации их собственных проектов;
- 3) "культурный резонанс" - язык должен быть применим и иметь смысл в широком социальном контексте, т.е. для решения самых различных прикладных задач.

Черепашка применяется для обучения геометрии. Однако геометрия черепашки отличается и от геометрии Евклида, и от геометрии Декарта. Она является компьютерной, в ее основе лежат понятия компьютерной науки - геометрические фигуры представлены в "черепашьей" геометрии процедурами их построения. Усвоение в этой среде геометрии связано с усвоением понятий программирования. Работая с "черепашьей графикой", дети могут развивать интуицию, лежащую в основе дальнейшего изучения геометрии и тригонометрии.

В среде Лого нет резкого разрыва между "научными" теориями, изучаемыми в школе, и "ненаучными" теориями, создаваемыми самим учеником [15]. В среде Лого переход от "ненаучных" детских теорий к "научным" можно сравнить с отладкой программы. В философии Лого иным является и взгляд на ошибку учащегося. Если в традиционном школьном обучении ошибочное решение объявляется неверным и отбрасывается, то в Лого ошибки рассматриваются как неотъемлемая часть процесса обу-

чения. Ошибки необходимо выделять, выяснять их причины и исправлять. В процессе исправления ошибок учащийся постепенно совершенствует свою программу, получая при этом новые знания. Переход ученика к "научным" теориям рассматривается как постепенное исправление ошибок, совершенствование "ненаучных" теорий.

2. "Черепашья графика"

Среда "черепашьей графики" представляет собой введение в язык. В центре экрана дисплея находится треугольный указатель - черепашка, которая "понимает" несколько простых команд, указывающих на характер движения (вперед или назад). Черепашка всегда передвигается в том направлении, в котором она сориентирована (или, соответственно, в противоположном направлении). Расстояние определяется числом "черепашьих" шагов, указанным в команде (в разных версиях системы Лого "длина" шага черепашки может быть различной). Черепашка поворачивается на месте по и*против часовой стрелки на указанное число градусов в локальной полярной системе координат. Движение черепашки по экрану дисплея относительно, т.е. точкой отсчета всегда является ее местонахождение. Черепашка по указанию пользователя оставляет или не оставляет след своего пути на экране. Она также может повторить некоторую последовательность команд (возможна конструкция цикла). Уже только эти команды позволяют создавать интересные рисунки. Далее будет рассмотрена возможность "обучения" компьютера новым словам или командам.

В разработанной и реализованной автором версии системы Лого словарь черепашки состоит из следующих команд:

- ВПЕРЕД n - черепашка проходит вперед n "черепашьих" шагов (в данной версии один шаг равен одной графической точке экрана);
- НАЗАД n - черепашка идет назад (не поворачиваясь) n шагов;
- НАПРАВО x - черепашка поворачивается на месте по часовой стрелке на x градусов;

- НАЛЕВО x - черепашка поворачивается на месте против часовой стрелки на x градусов;
- РИСОВАТЬ - черепашка при передвижении по экрану дисплея оставляет "след" - линию (по умолчанию, в исходном положении черепашка рисует);
- НЕ_РИСОВАТЬ - отменяется команда РИСОВАТЬ - черепашка не оставляет "следа" на экране;
- ПОВТОРИТЬ r <команды>, находящиеся между командами ПОВТОРИТЬ и КОНЕЦ, называемые телом цикла, будут исполнены r раз.

Величины n , x , r называются входными параметрами команд. Значение входного параметра указывается через пробел за именем команды без скобок. Исходное положение черепашки в центре экрана "носом" вверх (в учебнике [3] это направление названо "север").

Хотя в командах НАПРАВО и НАЛЕВО требуется указывать число градусов, совсем не обязательно, чтобы обучаемый заранее имел знания о понятии угла, величины угла. Экспериментируя с черепашкой, он получает представление об этом и других понятиях геометрии.

Язык Лого, созданный как язык для обучения, характеризуется: 1) интерактивностью; 2) процедурностью; 3) расширяемостью; 4) рекурсивностью.

1) Интерактивность языка означает, что ученик получает немедленно результат введенных в компьютер команд или программ на языке Лого. Это будет либо передвижение черепашки по экрану дисплея, либо сообщение, указывающее на ошибку, если черепашка "не поняла" введенную команду.

Рассмотрим пример [3, § 4.10]. Для создания на экране дисплея геометрической фигуры (в данном примере - квадрата) необходимо "проинструктировать" черепашку, как ей пройти путь в виде квадратного контура, рисуя линию. Алгоритм решения задачи, записанный на алгоритмическом языке (в командах "вперед" расстояние 1 заменено на 20) [3, § 4.10], переводится на язык Лого:

```
ВПЕРЕД 20      ; ВЛС СЛЕД
НАПРАВО 90     ; НАЧ
ВПЕРЕД 20      ;           опустить хвост
НАПРАВО 90     ;           вперед (20), направо (90)
ВПЕРЕД 20      ;           вперед (20), направо (90)
НАПРАВО 90     ;           вперед (20), направо (90)
ВПЕРЕД 20      ;           вперед (20), направо (90)
НАПРАВО 90     ;           поднять хвост
                ; КОН
```

Команды исполнителя Черепашки [3] на алгоритмическом языке "опустить хвост" и "поднять хвост" соответствуют командам Лого РИСОВАТЬ и НЕ_РИСОВАТЬ. Так как черепашка Лого по умолчанию рисует, нет необходимости это указывать специально.

Результат выполнения алгоритма показан на рис. 1.

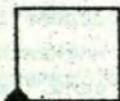


Рис. 1.

Алгоритм записывается короче с использованием команды ПОВТОРИТЬ (соответствующая команда алгоритмического языка - "НЧ r раз"; команде Лого КОНЕЦ соответствует команда алгоритмического языка "КЧ"):

```
ПОВТОРИТЬ 4    ; ВЛС СЛЕД
  ВПЕРЕД 20     ; НАЧ
  НАПРАВО 90   ;           опустить хвост
КОНЕЦ          ;           НЧ 4 раз
                ;           вперед (20), направо (90)
                ;           КЧ
                ;           поднять хвост
                ; КОН
```

2) Язык Лого, подобно языку Pascal и другим современным языкам программирования, является процедурным. По сравнению с программой на языке BASIC (MSX), которая является длинным списком команд, программа на языке Лого состоит из имен маленьких понятных процедур, каждая из которых выполняет свою задачу. Одна процедура может вызывать другую. Эта возможность позволяет учащимся сначала рассмотреть большую задачу в целом, распланировать свои действия и затем разделить большую задачу на маленькие подзадачи, каждая из которых решается элементарно. Таким образом, дети начинают осваивать разделение задач на процедуры, а также комбинирование процедур при создании сверхпроцедур, т.е. процедур, для решения больших, сложных задач. В среде Лого естественно усваиваются принципы процедурного (структурированного) программирования, рациональная техника построения алгоритмов.

3) Посредством включения созданных процедур в состав языка осуществляется его расширение (развитие) в желаемом направлении.

Если определить последовательность команд, рисующих квадрат, как процедуру с именем КВАДРАТ, то с этого момента черепашка будет "знать" сама, как рисовать фигуру, когда вводится слово "КВАДРАТ" - к ее словарю добавляется новое слово. В действительности, конечно, черепашка "не понимает" смысл слова "КВАДРАТ". Если процедуру, рисующую квадрат, назвать "ЗОНТИК", черепашка все равно нарисует квадрат, поскольку она выполняет команды, записанные в процедуре, а смысл имени процедуры имеет значение только для самих детей.

Рассмотрим возможности создания и комбинирования процедур, т.е. пополнение словаря Лого новыми словами. Для этого составим алгоритм, рисующий улицу из шести домиков [3, § 5.12, зад. 2а]. Хотя предполагается составить алгоритм для исполнителя Чертежник [3], не будет принципиального различия, если мы напишем алгоритм для черепашки.

Чтобы нарисовать улицу из шести домиков, черепашке необходимо "поставить" подряд шесть домиков. Сперва составим алгоритм рисования одного домика, для чего воспользуемся процедурой КВАДРАТ. К нарисованному квадрату необходимо лишь "достроить" крышу.

ДОМИК	:	ВЛГ ДОМИК
КВАДРАТ	:	НАЧ
КРЫША	:	КВАДРАТ
	:	КРЫША
	:	КОН

Определим вспомогательную процедуру КРЫША:

КРЫША	:	ВЛГ КРЫША
ВПЕРЕД 20	:	НАЧ
НАПРАВО 45	:	вперед (20)
ВПЕРЕД 14	:	направо (45)
НАПРАВО 90	:	вперед (14)
ВПЕРЕД 14	:	направо (90)
НАПРАВО 45	:	вперед (14)
ВПЕРЕД 20	:	направо (45)
	:	вперед (20)
	:	КОН

Имя новой процедуры черепашка "запоминает", и оно фактически является новой командой. Если сейчас ввести команду ДОМИК, черепашка нарисует в центре экрана домик (см. рис. 2).



Рис. 2.

Процедура, рисующая улицу, использует уже созданные процедуры или, другими словами, новые команды черепашки:

УЛИЦА	:	ВЛГ УЛИЦА
НЕ_РИСОВАТЬ	:	НАЧ
НАЛЕВО 90	:	поднять хвост
ВПЕРЕД 78	:	налево (90)
НАПРАВО 90	:	вперед (78)

РИСОВАТЬ	:	направо (90)
ПОВТОРИТЬ 6	:	опустить хвост
ДОМИК	:	еще 6 раз
ПЕРЕХОД	:	ДОМИК
КОНЕЦ	:	ПЕРЕХОД
	:	кц
	:	кон

Первые шесть команд (после имени процедуры) реализуют перемещение черепашки на "начало" улицы. Процедура ПЕРЕХОД, которая используется процедурой УЛИЦА, предназначена для перемещения черепашки к месту рисования следующего домика:

ПЕРЕХОД	:	вдл ПЕРЕХОД
НЕ_РИСОВАТЬ	:	нач
НАЛЕВО 90	:	поднять хвост
ВПЕРЕД 5	:	налево (90)
НАЛЕВО 90	:	вперед (5)
РИСОВАТЬ	:	налево (90)
	:	опустить хвост
	:	кон

Если теперь введем команду:

УЛИЦА

черепашка нарисует на экране улицу из шести домиков (см. рис. 3).

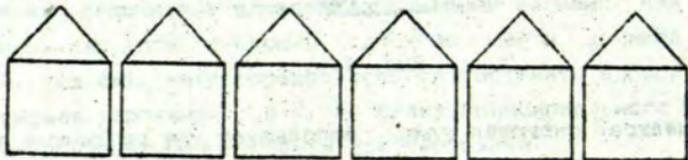


Рис. 3.

С помощью определения новых процедур, учащийся может создать свой собственный язык и использовать его для проверки своих идей. Кроме решения геометрических задач (рисуя черепашки картинки с использованием "черепаший графики") при

дальнейшем изучении языка, возможно освоение и других областей как математики, так и других предметов.

Таким образом, среда Лого позволяет сформировать своеобразную компьютерную культуру путем создания новых слов словаря языка Лого. Важную роль в ее развитии играет учитель.

4) **Рекурсивность** означает, что любая процедура может вызывать сама себя. Используя рекурсию, можно не только создавать на экране дисплея интересные рисунки с периодически повторяющимися элементами, но и удобно решать довольно сложные задачи. Например, процедура ЗВЕЗДА (результат ее выполнения см. на рис. 4) в последней команде содержит рекурсивный вызов:

```
ЗВЕЗДА           ; адг ЗВЕЗДА
ВПЕРЕД 30        ; нач
НАПРАВО 135     ;   вперед (30)
ЗВЕЗДА          ;   направо (135)
                 ;   ЗВЕЗДА
                 ; кон
```

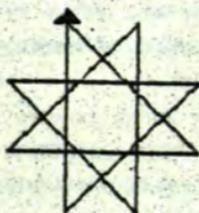


Рис. 4.

Кроме вышеперечисленных четырех характеристик языка Лого, необходимо подчеркнуть простоту и естественность усвоения языка Лого [14-16]: учащимся не составляет труда понять язык по следующим двум причинам [15].

Во-первых, основные команды словаря черепашки являются словами, выражающими соответствующие действия. Команды Лого, ВПЕРЕД 10, НАЗАД 20, НАПРАВО 90, НАЛЕВО 60 позволяют распорядиться черепашкой на экране дисплея компьютера или на полу, и эти же команды учащийся может использовать для описания своего движения по классу.

В большинстве языков программирования, в противоположность Лого, мы должны работать с Декартовыми координатами. Например, оператор языка BASIC (MSX)

```
10 LINE (100,50)-(150,50)
```

переместит "черепашку Бейсика" на 50 шагов (единиц) на экране дисплея, рисуя линию. Соответственно, команда Лого

```
ВПЕРЕД 50,
```

которая передвинет черепашку на 50 шагов вперед в том направлении, как она сориентирована, гораздо ближе представлению ребенка о том, как черепашка "обдумывает" и выполняет команды перемещения.

Таким образом, размышляя над поведением черепашки, учащийся приучается анализировать свои действия, и наоборот, наблюдение над своим собственным поведением является источником идеи для "разговоров" с черепашкой.

Например, учащийся хочет, чтобы черепашка нарисовала круг, но не знает, как составить соответствующую процедуру. Учитель рекомендует ему "играть в черепашку", т.е. самому встать на место черепашки. Ребенок проходит по кругу и обнаруживает, что для того, чтобы описать круг, надо сделать шаг и чуть-чуть повернуться, еще шаг и опять повернуться и т.д. После этого принцип составления процедуры становится для него ясным.

Во-вторых, учащемуся легко понять команды словаря Лого, поскольку Лого позволяет ему пополнять словарь по своему желанию и впоследствии работать со "своим" словарем.

3. Особенности разработанной системы

3.1. Дополнительные команды.

Кроме упомянутых выше команд управления черепашкой в разработанной системе имеются три вспомогательные команды. ЦЕНТР - устанавливает черепашку в центре экрана, в ее исходное положение. СТЕРЕТЬ - не меняет местонахождение черепашки, но стирает все ее следы с экрана. СБРОС - реализует функции обеих этих команд, т.е. очищает экран дисплея и помещает черепашку в ее исходное положение.

Для создания и редактирования процедур служат команды ПОСТРОИТЬ и ИЗМЕНИТЬ. Эти команды как входной параметр тре-

буют имя процедуры (например, КВАДРАТ). Созданные процедуры во время работы с системой хранятся в оперативной памяти ПЭВМ. Узнать, какие процедуры уже созданы, можно введя команду СПИСОК. Если команда СПИСОК вводится с параметром - именем процедуры, на экран выводится последовательность команд данной процедуры.

Уничтожить построенную процедуру, т.е. "вычеркнуть" ее из списка можно командой ВЫЧЕРКНУТЬ, в которой указано имя процедуры. В этом случае процедура стирается из оперативной памяти.

Система предполагает сохранение и вызов процедур с диска. Для этого служат команды СОХРАНИТЬ и ВЫЗВАТЬ, в которых тоже надо указывать имя процедуры.

Имеются еще такие команды, как НАЧАЛО, которая приводит систему в начальное положение, т.е. очищает экран, помещает черепашку в его центр, уничтожает все созданные процедуры, и ВЫХОД, при которой работа с системой заканчивается.

3.2. Перевод словаря Лого.

Система предполагает возможность перевода словаря черепашки. Имеется набор команд управления черепашкой на русском языке и программные средства для его перевода на другой язык. Это и является основной особенностью разработанной системы.

Команды Лого - это слова естественного, первоначально английского, языка, описывающие определенные действия. Поскольку стандарт на язык Лого отсутствует, созданы и создаются версии Лого на польском (для ПЭВМ "Sinclair Spectrum") [9], на русском (для ПЭВМ "Правец", "Yamaha", "Корвет", "Электроника БК-0010") [17], литовском (для ПЭВМ "Электроника БК-0010") языках.

Разработанная система позволяет пользователю создать и работать на своем "диалекте" Лого. Для этого служит пакет программ, с помощью которого в режиме диалога можно дополнить латинский или русский алфавит необходимыми знаками, перевести команды Лого и сообщения об ошибках с русского языка и записать новый словарь Лого на диске. При запуске системы пользователю дается возможность выбрать "диалект" Лого из существующих, т.е. из русского и вновь созданных.

Существующая версия системы позволяет организовать практическую работу учащихся по отдельным вопросам темы "Алгоритмический язык" курса "ОИИВТ": "Черепашка и ее команды", "Вспомогательные алгоритмы", "Команда повторения н раз".

Экспериментальный вариант системы Лого реализован на языке программирования MSX BASIC. Версия системы для тиражирования создается на языке С для ПЭВМ "Yamaha MSX-2".

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Кузнецов А., Долматов В. Методическая система обучения ОИВТ: структура и функции, состояние и перспективы // Информатика и образование. 1989. № 1. С. 3-8.
2. Основы информатики и вычислительной техники: Пробное учебное пособие для средних учебных заведений / Под ред. А.П.Ершова, В.М.Монахова. М: Просвещение. 1985. Ч. I. 96 с.
3. Ершов А.П., Кушниренко А.Г., Лебедев Г.В., Семенов А.Л., Шень А.Х. Основы информатики и вычислительной техники: Пробный учебник для средних учебных заведений / Под ред. А.П.Ершова. М: Просвещение. 1988. 207 с.
4. Юнрман Н. Использование графической системы ШПАГА при введении основных понятий программирования // Информатика и образование. 1988. № 1. С. 52-55.
5. Луванов А. Информатика без ЭВМ // Информатика и образование. 1987. № 3. С. 50-55.
6. Micros in the primary classroom / Edited by Ron Jones. London: Edward Arnold. 1984. 104 p.
7. Obrist J. Microcomputers and children. A guide for parents and teachers. London: Hodder and Stoughton. 1985. 124 p.
8. Brack E.J. Meet the micro. A first guide to computers for primary teachers. London: Edward Arnold. 1984. 61 p.
9. Waligorski S. Logo na Sinclair Spectrum. Warszawa. 1987.
10. Wang C., Tovgaard N. What can we use microcomputers in the schools? // Phase two. 1984. Vol. 4. No. 2. Pp. 68-72.
11. Clark S. A trip to the states // Phase two. 1984. Vol. 4. No. 3. Pp. 110-113.

12. Бешенков С. Экспериментальная программа преподавания начал информатики в младших классах // Информатика и образование. 1987. № 2. С. 44-45.
13. Смирнов Е. К концепции обучения информатике в младших классах // Информатика и образование. 1988. № 2. С. 22-25.
14. Бруснецова Т.Н. О психолого-педагогических принципах компьютерной системы обучения Лого // Вопросы психологии. 1986. № 2. С. 164-170.
15. Douglas H. Clements. Computers in early and primary education. New-Jersey: Prentice Hall Inc. 1985. 322 p.
16. Goodyear P. LOGO: A guide to learning through programming. London: Ellis Horwood Ltd. & Heinemann Computers in Education Ltd. 1984. 206 p.
17. Школа начинающего программиста // Наука и жизнь. 1988. № 9. С. 78-83.

УДК 37.01:007+371.3+681.3.06

И. А. Шкерстена

Вычислительный центр при ЛГУ им. П. Стучки

СИСТЕМА ЛОГО ДЛЯ ПЭВМ "YAMAHA MSX-2"

II. Перспективы развития разработанной системы

Реализованная система Лого [1] дает возможность пользователю работать в обучающей среде "черепашьей графики" с системой команд исполнителя на русском языке и предполагает возможность перевода набора команд на любой другой язык.

Среда "черепашьей графики" предоставляет учащемуся исполнителя-черепашку, "понимающую" несколько простых команд и способную "обучаться" новым. Задачи для черепашки берутся из области планиметрии, но решая их, учащиеся усваивают основы алгоритмизации и структурного программирования.

Однако популярность "черепашьей графики" имеет нежелательный побочный эффект - она отвлекает внимание от таких не менее мощных и интересных возможностей Лого, как обработка числовой и символической информации [2-4].

Дальнейшее развитие описываемой системы Лого включает расширение среды "черепашьей графики" добавлением возможностей использования переменных, арифметики, команд ввода-вывода информации, а также создание среды для работы с текстовой информацией.

Ниже подробнее рассматриваются возможности языка Лого, которые предполагается реализовать.

1. Переменные

Процедуры, подобно командам управления черепашкой, могут требовать один или несколько входных параметров. Входной параметр процедуры указывается при создании процедуры в строке

заголовка за именем процедуры и заключается в скобки. Если параметров несколько, они разделяются пробелом. При вызове процедуры значение входного параметра указывается за именем процедуры (через пробел, без скобок), подобно значению входного параметра, например, команды НАЗАД. Если параметров несколько, они разделяются пробелом.

Входными параметрами процедур являются переменные. Переменная определяется как объект, имеющий постоянное имя, но значение которого может меняться. Значение переменной бывает как числовое, так и литерное.

Один из доступных для детей способов объяснения организации механизма переменных - это представление переменных в виде маленьких "ящиков", находящихся в памяти компьютера. Каждый "ящик" имеет свою оригинальную этикетку, которая остается неизменной, в то время, как содержание "ящика" может меняться. Этикетка на "ящике" эквивалентна имени переменной, а содержание - значению переменной.

Значение переменной задается (в "ящик" помещается новое содержание) с помощью команды присваивания (:=):

ДЛИНА := 30

Встретив такую команду, черепашка проверяет, имеется ли в памяти компьютера "ящик" с этикеткой ДЛИНА. Если "ящик" имеется, она помещает в него значение 30. Если такого "ящика" нет, черепашка создает новый, отмечает его этикеткой ДЛИНА и помещает в него значение 30.

В качестве литерных величин Лого использует списки, состоящие из слов. Слово - это группа символов (букв или цифр), например, 22П4, СТУЛ, 55-45-23. Список - это последовательность слов, чисел или других списков. Для разделения элементов списка служит пробел.

Литерное значение присваивается переменной также как числовое, только кавычки перед словом в правой части команды присваивания указывают черепашке, что это литерная константа.

ИМЯ := "ШУРА"

После выполнения этой команды в "ящике" с этикеткой ИМЯ будет находиться слово "ШУРА".

Для обозначения списка в Лого используются квадратные

скобки. пустой список обозначается пустыми квадратными скобками:

```
ИМЯ := [ШУРА ВАЛЯ МАША ЖЕНЯ]
```

```
СПИСОК := []
```

Переменные используются как входные параметры команд Logo, например:

```
ВПЕРЕД :ДЛИНА
```

```
НАПРАВО :УГОЛ
```

Двоеточие перед именем переменной выполняет определенную функцию. Так, ":ДЛИНА" читается как "содержание "ящика" с этикеткой ДЛИНА".

В [1] была определена процедура КВАДРАТ, реализующая алгоритм рисования квадрата со стороной 20 [5, § 4.10]:

```
КВАДРАТ                : вдг КВАДРАТ
ПОВТОРИТЬ 4            : нач
  ВПЕРЕД 20             : нц 4 раз
  НАПРАВО 90            : вперед (20)
  КОНЕЦ                 : направо (90)
                        : кц
                        : кон
```

Если процедуру КВАДРАТ определить с входным параметром СТОРОНА, указывающим длину стороны, с ее помощью можно будет рисовать квадраты любого размера:

```
КВАДРАТ (СТОРОНА)     : вдг КВАДРАТ (вдг нач СТОРОНА)
ПОВТОРИТЬ 4           : нач
  ВПЕРЕД :СТОРОНА     : нц 4 раз
  НАПРАВО 90          : вперед (СТОРОНА)
  КОНЕЦ                : направо (90)
                        : кц
                        : кон
```

Если ввести команду КВАДРАТ с параметром 60, т.е.

```
КВАДРАТ 60 .
```

черепашка нарисует квадрат со стороной 60 единиц. Бесконечное значение 60 присваивается переменной СТОРОНА, которая потом

используется в команде ВПЕРЕД.

В качестве примера процедуры с несколькими входными параметрами рассмотрим задачу, в которой требуется составить алгоритм, рисующий любой правильный многоугольник [5, § 8.7, зад. 5]:

МНОГОУГОЛ (КОЛСТОРОН УГОЛ ДЛИНА)

ПОВТОРИТЬ : КОЛСТОРОН

НАЛЕВО : УГОЛ

ВПЕРЕД : ДЛИНА

КОНЕЦ

; ВЛГ МНОГОУГОЛ (ВРГ НАТ КОЛСТОРОН, УГОЛ, ДЛИНА)

; НАЧ

; НЦ КОЛСТОРОН РВА

; налево (УГОЛ)

; вперед (ДЛИНА)

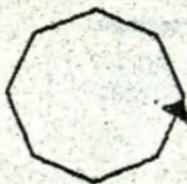
; КЦ

; КОН

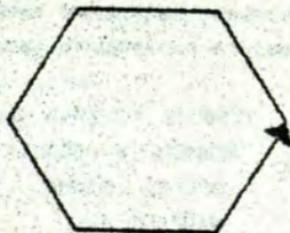
На рис. 1 а, б, в показан результат выполнения команд: МНОГОУГОЛ 4 90 20; МНОГОУГОЛ 8 45 15; МНОГОУГОЛ 6 60 30.



а



б



в

Рис. 1.

При вызове процедуры, в списке входных параметров указываются как числа, так и имена переменных, например:

МНОГОУГОЛ 6 45 : ДЛИНА

2. Арифметика

В правой части команды присваивания, а также в списке входных параметров при вызове процедуры и в других командах Лого (см. дальше) может находиться арифметическое выражение. Для обозначения арифметических операций Лого использует следующие символы:

+ сложение	* умножение	^ возведение в степень
- вычитание	/ деление	

Например,

```

ДЛИНА := :ДЛИНА + :ЧИСЛО
УГОЛ := 360 / :ЧИСЛО_СТОРОН
ПЛОЩАДЬ := :ПИ * :РАДИУС ^ 2

```

В силе обычные приоритеты выполнения арифметических операций:

- 1) возведение в степень (^);
- 2) умножение и деление (* и /);
- 3) сложение и вычитание (+ и -).

Для наглядности и во избежание ошибок при написании арифметических выражения используются скобки. Операции, заключенные в скобки, выполняются в первую очередь. Например,

```
ДЛИНА := 3 * ( :ШИРИНА + 5 )
```

Процедура СТИРАЛЬ, рисующая спираль с указанным количеством звеньев, является решением задачи [2, § 8.7, зад. 4]. При вызове процедуры пользователь указывает количество звеньев, начальную длину звена и число, определяющее прирост каждого следующего звена спирали.

```
СТИРАЛЬ (ЗВЕНЬЯ ДЛИНА ЧИСЛО)
```

```
ПОВТОРИТЬ :ЗВЕНЬЯ
```

```
ВПЕРЕД :ДЛИНА
```

```
НАПРАВО 90
```

```
ДЛИНА := :ДЛИНА + :ЧИСЛО
```

```
КОНЕЦ
```

```
: ВДС СТИРАЛЬ (ВДС НАЧ ЗВЕНЬЯ ДЛИНА ЧИСЛО)
```

```
: НАЧ
```

```
: НАЧ ЗВЕНЬЯ ДВА
```

```
вперед (ДЛИНА)
```

```
: направо (90)
: ДЛИНА := ДЛИНА + ЧИСЛО
: КЦ
: КОН
```

На рис. 2. показан результат выполнения команды
СПИРАЛЬ 15 10 5

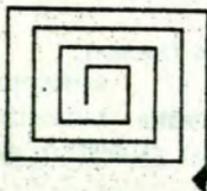


Рис. 2.

Допускается применение стандартных функций: sin, cos, sqrt, др.

3. Команда повторения с условием

Используя команду цикла ПОВТОРИТЬ *n*, необходимо знать заранее число повторов, что не всегда возможно. В Лого [3] имеется еще одна команда цикла ПОВТОРИТЬ ПОКА... В алгоритмическом языке этой команде соответствует цикл "пока". Форма записи этой команды в Лого следующая:

```
ПОВТОРИТЬ           : НИ ПОКА <условие>
ПОКА <условие>      :     <команды>
<команды>           : КЦ
КОНЕЦ               :
```

Условие - это логическое выражение, в котором используются операции сравнения:

= равно	> больше	>= больше или равно
<> не равно	< меньше	<= меньше или равно

Команды тела цикла выполняются до тех пор, пока условие имеет значение "истина".

В процедуре СПИРАЛЬ необходимо указать количество звень-

ев спирали. Не исключена ситуация, когда при определенных (достаточно больших) значениях введенных параметров спираль не поместится на экран, и процедура закончится сообщением "Черепашка достигла границы поля" или т.п. Можно изменить процедуру так, чтобы она проверяла значение переменной ДЛИНА и прекратила выполнение при достижении определенного значения этой переменной. Предельная величина задается как параметр процедуры и записывается в переменную ПРЕДЕЛ.

СПИРАЛЬ (ДЛИНА ЧИСЛО ПРЕДЕЛ)

ПОВТОРИТЬ

ПОКА :ДЛИНА <= :ПРЕДЕЛ

ВПЕРЕД :ДЛИНА

НАПРАВО 90

ДЛИНА := :ДЛИНА + :ЧИСЛО

КОНЕЦ

; ВЛГ СПИРАЛЬ (ВРГ НАТ ДЛИНА, ЧИСЛО, ПРЕДЕЛ)

; НАЧ

; НЧ ПОКА ДЛИНА <= ПРЕДЕЛ

; вперед (ДЛИНА)

; направо (90)

; ДЛИНА := ДЛИНА + ЧИСЛО

; КЦ

; КОН

4. Команда ветвления

Другая конструкция, где используются операция сравнения - это команда ветвления ЕСЛИ...ТО или ЕСЛИ...ТО...ИНАЧЕ:

ЕСЛИ <условие>	:	ЕСЛИ <условие>
ТО <команды>	:	ТО <команды>
ИНАЧЕ <команды>	:	ИНАЧЕ <команды>
ВСЕ	:	ВСЕ

Эта команда необходима, когда выполнение одних или других команд зависит от наличия какого-либо условия. Если значе-

ние логического выражения (условия) - "истина", выполняются команды, написанные за ключевым словом ТО (ветвь "ТО"), если значение логического выражения - "ложь", выполняются команды ветви "ИНАЧЕ". Ветвь "ИНАЧЕ" может отсутствовать.

Далее следуют два примера записи конструкции на языке Лого и на алгоритмическом языке.

1. В результате выполнения команды ЕСЛИ процедура ПРЯМОУГОЛЬНИК будет вызвана так, чтобы первый входной параметр был больше второго:

```
ЕСЛИ :СТОРОНА1 > :СТОРОНА2
  ТО ПРЯМОУГОЛЬНИК :СТОРОНА1 :СТОРОНА2
  ИНАЧЕ ПРЯМОУГОЛЬНИК :СТОРОНА2 :СТОРОНА1
ВСЕ
```

```
; ЕСЛИ СТОРОНА1 > СТОРОНА2
;   ТО ПРЯМОУГОЛЬНИК (СТОРОНА1, СТОРОНА2)
;   ИНАЧЕ ПРЯМОУГОЛЬНИК (СТОРОНА2, СТОРОНА1)
; ВСЕ
```

2. Значение переменной ДИАМЕТР будет увеличено на 10, если оно меньше десяти. Затем вызывается процедура КРУГ с входным параметром ДИАМЕТР:

```
ЕСЛИ :ДИАМЕТР < 10
  ТО ДИАМЕТР := :ДИАМЕТР + 10
ВСЕ
КРУГ :ДИАМЕТР
```

```
; ЕСЛИ ДИАМЕТР < 10
;   ТО ДИАМЕТР := ДИАМЕТР + 10
; ВСЕ
; КРУГ (ДИАМЕТР)
```

5. Ввод и вывод информации

Используя команды ввода и вывода информации Лого учащиеся могут писать интерактивные процедуры.

Команда ВВЕСТИ позволяет засылать информацию с клавиатуры в память компьютера при работе процедур. Команда ВВЕСТИ выдает информацию, например, значение переменной или текст на экран дисплея. Соответствующие команды алгоритмического языка - ВВОД и ВЫВОД.

Например, процедура ВЫЧИСЛИТЬ_ПЛОЩАДЬ вычисляет площадь произвольного прямоугольника (размеры прямоугольника задаются как входные параметры) и выводит результат на экран дисплея:

```
ВЫЧИСЛИТЬ_ПЛОЩАДЬ (ДЛИНА ШИРИНА)
  ПЛОЩАДЬ := :ДЛИНА * :ШИРИНА
  ВВЕСТИ :ПЛОЩАДЬ

; алг ВЫЧИСЛИТЬ_ПЛОЩАДЬ (арг нат ДЛИНА, ШИРИНА)
; нач (нат ПЛОЩАДЬ)
;   ПЛОЩАДЬ := ДЛИНА * ШИРИНА
;   вывод ПЛОЩАДЬ
; кон
```

Если выполнить процедуру с разными значениями ДЛИНЫ и ШИРИНЫ, результат команды "ВЫЧИСЛИТЬ_ПЛОЩАДЬ 10 15" будет 150, а результат команды "ВЫЧИСЛИТЬ_ПЛОЩАДЬ 8 7" будет 56.

С помощью команды ВВЕСТИ на экран дисплея выводятся также списки слов (текст). Команда

ВВЕСТИ [Площадь прямоугольника] напечатает на экране дисплея строку "Площадь прямоугольника". Аргументом команды ВВЕСТИ также может быть арифметическое выражение.

Изменим процедуру ВЫЧИСЛИТЬ_ПЛОЩАДЬ так, чтобы она дала результат в более наглядной форме:

```
ВЫЧИСЛИТЬ_ПЛОЩАДЬ (ДЛИНА ШИРИНА)
  ВВЕСТИ [ Площадь прямоугольника ]
  ВВЕСТИ :ДЛИНА * :ШИРИНА

; алг ВЫЧИСЛИТЬ_ПЛОЩАДЬ (арг нат ДЛИНА, ШИРИНА)
; нач
```

```
 ;   вывод "Площадь прямоугольника"  
 ;   вывод ДЛИНА * ШИРИНА  
 ;   КОН
```

Процедура `ВЫЧИСЛИТЬ_ПЛОЩАДЬ` станет интерактивной процедурой и будет более удобной для использования, если применить команду `ВВЕСТИ`, которая позволит пользователю самому вводить размеры прямоугольника:

```
ВЫЧИСЛИТЬ_ПЛОЩАДЬ  
  ВЫВЕСТИ [ Какая длина? ]  
  ВВЕСТИ ДЛИНА  
  ВЫВЕСТИ [ Какая ширина? ]  
  ВВЕСТИ ШИРИНА  
  ВЫВЕСТИ [ Площадь прямоугольника ]  
  ВЫВЕСТИ : ДЛИНА * : ШИРИНА
```

```
 ;   алг ВЫЧИСЛИТЬ_ПЛОЩАДЬ  
 ;   нач (нач ДЛИНА, ШИРИНА)  
 ;   вывод "Какая длина?"  
 ;   ввод ДЛИНА  
 ;   вывод "Какая ширина?"  
 ;   ввод ШИРИНА  
 ;   вывод "Площадь прямоугольнике"  
 ;   вывод ДЛИНА * ШИРИНА  
 ;   КОН
```

6. "Процедуры-функции"

В учебнике [5. § 12.10] введено понятие алгоритма-функции, задающего новую функцию, применяемую наравне со стандартными функциями `sqrt`, `sin`, `cos` и др. На Лого учащийся также может создать пакет "процедур-функции", которые потом используются как готовые "блоки" при построении других процедур. Для определения результата "процедуры-функции" используются команда `РЕЗУЛЬТАТ`. В алгоритмическом языке для этой цели используется переменная `знач`. Приведем примеры таких "процедур-функций":

```
ФУНКЦИЯ1 (ЧИСЛО)      :  АЛГ НАТ ФУНКЦИЯ1 (АРГ ЦЕЛ
РЕЗУЛЬТАТ :ЧИСЛО * :ЧИСЛО :
                        :
                        :  НАЧ
                        :
                        :    ВНАЧ := ЧИСЛО * ЧИСЛО
                        :
                        :  КОН
```

```
ФУНКЦИЯ2 (ЧИСЛО)      :  АЛГ НАТ ФУНКЦИЯ2 (АРГ ЦЕЛ
РЕЗУЛЬТАТ (:ЧИСЛО ^ 2) / 2 :
                        :
                        :  НАЧ
                        :
                        :    ВНАЧ := (ЧИСЛО ^ 2) / 2
                        :
                        :  КОН
```

Следующая процедура КВАДРАТ вызывает "процедуру-функцию", вычисляющую квадрат введенного пользователем числа. Рядом записан соответствующий алгоритм-функция:

```
КВАДРАТ                :  АЛГ КВАДРАТ
ВВЕСТИ [Введите число] :  НАЧ НАТ ЧИСЛО
ВВЕСТИ ЧИСЛО           :
                        :    ВЫВОД "Введите число"
ВВЕСТИ [Квадрат равен:] :  ВВОД ЧИСЛО
ВВЕСТИ ФУНКЦИЯ1 :ЧИСЛО :
                        :    ВЫВОД "Квадрат равен:"
                        :
                        :    ВЫВОД ФУНКЦИЯ1 (ЧИСЛО)
                        :
                        :  КОН
```

7. Обработка списков

Режим обработки списков позволяет учащимся оперировать целыми списками или отдельными элементами списков - словами. Как было сказано выше, в переменную можно записать список. Определим три переменные:

```
СПИСОК1 := [ ИВАРС МАРА ПЕТЕРИС ЗАНЕ ]
СПИСОК2 := [ ПОЕТ ПИШЕТ ЧИТАЕТ ЧИНИТ ]
СПИСОК3 := [ ПИСЬМО СТУЛ ПЕСНЯ КНИГУ ]
```

7.1. Выбор элемента из списка.

Для иллюстрации работы со списками, запишем процедуру.

которая выводит на экран случайные предложения. Схема работы такой процедуры следующая: случайным образом берем по одному слову из каждого списка (СПИСОК1...3) и выводим на экран.

Чтобы выбрать элемент из списка надо использовать т.н. у к а з а т е л ь, позволяющий указать номер интересующего нас элемента списка. Указатель обозначается символом "#". Так, результатом выполнения команды

```
ВЫВЕСТИ [ ИВАРС МАРА ПЕТЕРИС ЗАНЕ ] # 2
```

будет выведенное на экран дисплея слово "МАРА" - второй элемент списка. Значение указателя можно записать в переменную, которую назовем, например, .УКАЗАТЕЛЬ:

```
УКАЗАТЕЛЬ := 3
```

Если сейчас ввести:

```
ВЫВЕСТИ :СПИСОК2 # :УКАЗАТЕЛЬ
```

на экран будет выведено слово "ЧИТАЕТ".

Команда ВЫБОР позволяет использовать генератор случайных чисел. Например,

```
ВЫБОР (5)
```

дает случайное целое число в пределах от 1 до 5. Используем ее для написания процедуры, которая выводит на экран случайные предложения:

```
ПИСАТЬ_ПРЕДЛОЖЕНИЕ
```

```
ВЫВЕСТИ :СПИСОК1 # ВЫБОР (4)
```

```
ВЫВЕСТИ :СПИСОК2 # ВЫБОР (4)
```

```
ВЫВЕСТИ :СПИСОК3 # ВЫБОР (4)
```

Процедура ПИСАТЬ_ПРЕДЛОЖЕНИЕ берет случайно выбранный элемент из первого списка и выводит на экран, потом то же проделывает со вторым и с третьим списками. Так можно получить предложение типа "ПЕТЕРИС ЧИНИТ ПЕСНЮ". Процедура ПИСАТЬ_ПРЕДЛОЖЕНИЕ с точки зрения пользователя имеет два недостатка: первый - каждое слово будет напечатано в своей строке, поскольку команда ВЫВЕСТИ начинает вывод на экран с новой строки, второй - в процедуре задана длина списков - 4, что накладывает ограничения на применение процедуры, т.е. невозможно использовать списки другой длины.

В Лого существуют еще и другие средства для обработки

списков, позволяющие выбрать из списка определенные элементы. Это команды ПЕРВЫЙ, БЕЗ_ПЕРВОГО, ПОСЛЕДНИЙ и БЕЗ_ПОСЛЕДНЕГО. Например, результат выполнения команды

ПЕРВЫЙ : СПИСОК1

будет "ИВАРС" - первый элемент списка, результат команды

БЕЗ_ПЕРВОГО : СПИСОК1

будет "МАРА ПЕТЕРИС ЗАНЕ" - тот же список, но без первого элемента. Эти команды можно комбинировать, например:

ВЫВЕСТИ ПЕРВЫЙ БЕЗ_ПЕРВОГО : СПИСОК3

Тогда будет выведено второе слово из списка СПИСОК3: "СТУЛ".

7.2. Формирование новых списков.

Для соединения двух списков служит команда СОЕДИНИТЬ. Например, команда

СОЕДИНИТЬ [ИСПОЛНИТЕЛЬ] [ЧЕРЕПАШКА]

в результате даст список [ИСПОЛНИТЕЛЬ ЧЕРЕПАШКА]. Если нужно соединить больше, чем два списка, надо несколько раз использовать команду СОЕДИНИТЬ:

СОЕДИНИТЬ [ГРАФИЧЕСКИЙ] СОЕДИНИТЬ [ИСПОЛНИТЕЛЬ]
[ЧЕРЕПАШКА]

Другой способ добавления элемента к списку - используя команды В_НАЧАЛО и В_КОНЕЦ. Команда

В_НАЧАЛО "СТРЕКОЗА [И МУРАВЕЙ]

присоединит слово "СТРЕКОЗА" в начало списка и в результате даст список [СТРЕКОЗА И МУРАВЕЙ], соответственно

В_КОНЕЦ [БРАТЕЦ КРОЛИК И] "ДРУГИЕ

даст список [БРАТЕЦ КРОЛИК И ДРУГИЕ].

Узнать число элементов в списке можно с помощью команды СОСЧИТАТЬ. Например, результатом выполнения команды

СОСЧИТАТЬ [ЭТОТ СПИСОК СОДЕРЖИТ ПЯТЬ ЭЛЕМЕНТОВ]

будет число 5.

7.3. Процедуры обработки списков.

"Процедуры-функции" также возможно использовать для выполнения операций над списками, например, процедуру:

ВЗЯТЬ_СЛОВО_ИЗ (СПИСОК)

РЕЗУЛЬТАТ : СПИСОК # ВЫБОР (СОСЧИТАТЬ : СПИСОК)

Результатом выполнения "процедуры-функции" ВЗЯТЬ_СЛОВО_ИЗ

будет случайно выбранный элемент из списка, который указывается при вызове процедуры. Отметим, что процедура обрабатывает списки произвольной длины. Если ввести команду

ВЫВЕСТИ ВЗЯТЬ_СЛОВО_ИЗ :СПИСОК1 ,

будет выведено случайно выбранное слово из списка СПИСОК1.

Теперь изменим процедуру ПИСАТЬ_ПРЕДЛОЖЕНИЕ, используя только что введенные команды:

ПИСАТЬ_ПРЕДЛОЖЕНИЕ

ПРЕДЛОЖЕНИЕ := СОЕДИНИТЬ ВЗЯТЬ_СЛОВО_ИЗ :СПИСОК1

СОЕДИНИТЬ ВЗЯТЬ_СЛОВО_ИЗ :СПИСОК2

ВЗЯТЬ_СЛОВО_ИЗ :СПИСОК3

ВЫВЕСТИ :ПРЕДЛОЖЕНИЕ

Сперва в переменной ПРЕДЛОЖЕНИЕ формируется случайное предложение, а потом значение этой переменной выводится на экран дисплея. Таким образом, предложение будет напечатано в одной строке. Процедура также не накладывает ограничение на длину используемых списков.

7.4. Подписки.

Лого, также, как и другие языки программирования, которые имеют возможности обработки списков (например, LISP [6]), удобно использовать для работы с иерархическими организованной информацией. Таким образом, организованная информация позволяет легко справляться с большим объемом данных, несмотря на ограниченную память микроЭВМ. В частности, может моделироваться структура человеческого мышления и языка.

Возможность иерархической организации информации в Лого следует из факта, что элементами списков могут быть списки. Например, определим переменную СЕМЬЯ:

СЕМЬЯ := [[ЛОДА СЕРГЕЙ] [САША ОЛЕГ МАША]]

Список СЕМЬЯ имеет два элемента, каждый из которых в свою очередь является списком. [ЛОДА СЕРГЕЙ] - подсписок списка СЕМЬЯ, первый элемент этого подсписка - ЛОДА, а первый элемент списка СЕМЬЯ - это [ЛОДА СЕРГЕЙ]. Иерархическую структуру СЕМЬЯ можно представить в виде (рис. 3):



Рис. 3.

Для этого списка результатом выполнения команды

`ПЕРВЫЙ ПЕРВЫЙ :СЕМЬЯ`

будет "ЛЮДА", а результатом выполнения команды

`ПЕРВЫЙ БЕЗ_ПЕРВОГО :СЕМЬЯ`

будет "САША". Новая "член семьи" к списку СЕМЬЯ добавляется следующей командой:

`В_КОНЕЦ БЕЗ_ПЕРВОГО :СЕМЬЯ "ОРА`

Учащиеся могут создать списки информации, которые для них представляют интерес, например, списки прочитанных книг, своих друзей, оценок по учебным дисциплинам, или список школьных спортивных команд (вместе с полученными очками). Используя простые рекурсивные процедуры, дети могут оперировать с информацией: добавить к списку новую информацию, выбросить информацию из списка, переместить информацию на другое место в списке, проделать арифметические операции.

В качестве примера рассмотрим список ЖУРНАЛ, в котором содержатся оценки по учебным предметам одного ученика и сведения о нем, и запишем несколько процедур для обработки находящейся в журнале информации. Определим структуру списка ЖУРНАЛ в виде (рис. 4):

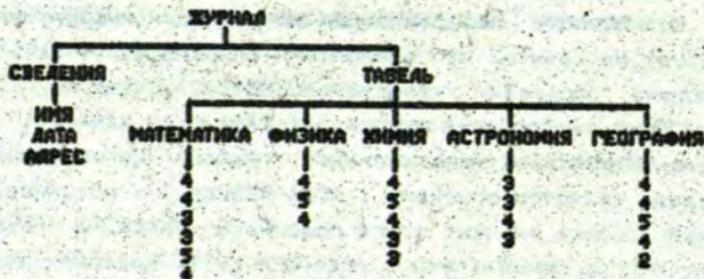


Рис. 4.

Список ЖУРНАЛ содержит два элемента, которые являются списками. Первый список, состоящий из трех элементов, содержит сведения об ученике: фамилия, имя и отчество, дата рождения и адрес. Элементами второго подсписка являются списки из двух элементов: первая - название учебного предмета, а вторая - список оценок ученика по этому предмету. Список оценок в начале каждой четверти пустой.

Список ЖУРНАЛ возможно определить двумя способами. Во-первых, непосредственно:

```
ЖУРНАЛ := [ [КРУМИНЬШЛИВАРС_ЯНОВИЧ 17.09.73 РИГА, 226005,  
            УЛ.ЗАЛЕС.10-8, 527103]  
           [ [МАТЕМАТИКА [ ] ] [ФИЗИКА [ ] ] [ХИМИЯ [ ] ]  
             [АСТРОНОМИЯ [ ] ] [ГЕОГРАФИЯ [ ] ] ] ]
```

Во-вторых, список можно строить, начиная с подписков. Используем переменные ИМЯ, ДАТА и АДРЕС:

```
ИМЯ := "КРУМИНЬШЛИВАРС_ЯНОВИЧ"
```

```
ДАТА := "17.09.73"
```

```
АДРЕС := "РИГА, 226005, УЛ.ЗАЛЕС.10-8, 527103"
```

Затем формируется переменная СВЕДЕНИЯ:

```
СВЕДЕНИЯ := [ИМЯ : ДАТА : АДРЕС]
```

Отсюда видно, что внутри списков используются имена переменных.

Установить набор предметов, содержащийся в переменной ДНЕВНИК, можно используя рекурсивную процедуру НАЧАЛО_ГОДА, сперва определив переменную ПРЕДМЕТЫ, содержащую список всех учебных предметов, и переменную ОЦЕНКИ, которая пока является пустым списком:

```
ПРЕДМЕТЫ := [ МАТЕМАТИКА ФИЗИКА ХИМИЯ АСТРОНОМИЯ  
              ГЕОГРАФИЯ ]
```

```
ОЦЕНКИ := [ ]
```

```
ДНЕВНИК := [ ]
```

Определим процедуру:

```
НАЧАЛО_ГОДА (ПРЕДМЕТЫ)
```

```
ЕСЛИ СОСЧИТАТЬ :ПРЕДМЕТЫ > 0
```

```
ТО СОЕДИНИТЬ :ДНЕВНИК
```

СОЕДИНИТЬ ПЕРВЫЙ : ПРЕДМЕТЫ : ОЦЕНКИ
НАЧАЛО_ГОДА БЕЗ_ПЕРВОГО : ПРЕДМЕТЫ

ВСЕ

Процедура вызывается строкой:

НАЧАЛО_ГОДА : ПРЕДМЕТЫ

Процедура берет первый элемент описки ПРЕДМЕТЫ, присоединяет к нему список ОЦЕНКИ, и добавляет к списку ДНЕВНИК. После этого процедура вызывает сама себя, но сейчас рассматривает список ПРЕДМЕТЫ уже без первого элемента. Это продолжается до тех пор, пока список ПРЕДМЕТЫ не исчерпан. Алгоритм такого типа характерен для процедур обработки списков.

Когда переменные СВЕДЕНИЯ и ДНЕВНИК определены, можно определить список ЖУРНАЛ:

ЖУРНАЛ := (: СВЕДЕНИЯ : ДНЕВНИК)

Полученные учеником во время четверти оценки необходимо занести в ЖУРНАЛ. Для этого используется процедура ЗАНЕСТИ_ОЦЕНКУ:

ЗАНЕСТИ_ОЦЕНКУ (ОЦЕНКА)

ДНЕВНИК := БЕЗ_ПЕРВОГО : ЖУРНАЛ

НАПЕЧАТАТЬ : ДНЕВНИК

КАКОЙ_ПРЕДМЕТ

ОФОРМИТЬ_ЗАПИСЬ : ДНЕВНИК . *

ЖУРНАЛ := СОЕДИНИТЬ ПЕРВЫЙ : ЖУРНАЛ : ДНЕВНИК

Если ученик получил оценку "4" по физике, нужно ввести команду:

ЗАНЕСТИ_ОЦЕНКУ 4

Сперва на экран будет выведен список названия предметов (процедура НАПЕЧАТАТЬ), и необходимо будет ввести название предмета, по какому получена оценка (процедура КАКОЙ_ПРЕДМЕТ). Процедура ОФОРМИТЬ_ЗАПИСЬ найдет соответствующий элемент в списке ДНЕВНИК и добавит к нему новую оценку. Наконец, обновляется список ЖУРНАЛ.

Вот, определения процедур, вызываемых процедурой ЗАНЕСТИ_ОЦЕНКУ:

```
НАПЕЧАТАТЬ (СПИСОК)
ЕСЛИ СОСЧИТАТЬ :СПИСОК > 0
    ТО ВЫВЕСТИ ПЕРВЫЙ :СПИСОК
        НАПЕЧАТАТЬ БЕЗ_ПЕРВОГО :СПИСОК
ВСЕ
```

(Вместо условия "СОСЧИТАТЬ :СПИСОК > 0" может также быть условие " :СПИСОК <> []".)

```
КАКОЙ_ПРЕДМЕТ
ВВЕСТИ [Введите название предмета]
ВВЕСТИ ПРЕДМЕТ

ОФОРМИТЬ_ЗАПИСЬ (ДНЕВНИК)
ЕСЛИ ПЕРВЫЙ ПЕРВЫЙ :ДНЕВНИК = :ПРЕДМЕТ
    ТО В_КОНЕЦ БЕЗ_ПЕРВОГО ПЕРВЫЙ :ДНЕВНИК :ОЦЕНКА
        ИНАЧЕ ОФОРМИТЬ_ЗАПИСЬ БЕЗ_ПЕРВОГО :ДНЕВНИК
ВСЕ
```

В конце четверти с помощью процедуры ОЦЕНКА_ЧЕТВЕРТИ получим список средних оценок ученика по всем предметам:

```
ОЦЕНКА_ЧЕТВЕРТИ (ДНЕВНИК)
ЕСЛИ СОСЧИТАТЬ :ДНЕВНИК > 0
    ТО ПРЕДМЕТ := ПЕРВЫЙ ПЕРВЫЙ :ДНЕВНИК
        ОЦЕНКИ := БЕЗ_ПЕРВОГО ПЕРВЫЙ :ДНЕВНИК
        ДЛИНА := СОСЧИТАТЬ :ОЦЕНКИ
        СУММА := 0
        СТРОКА := СОЕДИНИТЬ :ПРЕДМЕТ ВЫЧИСЛИТЬ :ОЦЕНКИ
        ВЫВЕСТИ :СТРОКА
        ОЦЕНКА_ЧЕТВЕРТИ БЕЗ_ПЕРВОГО :ДНЕВНИК
ВСЕ
```

Чтобы получить на экране дисплея список оценок ученика в четверти по всем предметам, нужно ввести строки:

```
ДНЕВНИК := БЕЗ_ПЕРВОГО :ЖУРНАЛ
ОЦЕНКА_ЧЕТВЕРТИ :ДНЕВНИК
```

Процедура ОЦЕНКА_ЧЕТВЕРТИ рассматривает каждый элемент списка ДНЕВНИК. В переменную ПРЕДМЕТ записывается название предмета, а в переменную ОЦЕНКИ - список оценок по этому предмету. Используются вспомогательные переменные: СУММА - для вычисления средней оценки, ДЛИНА - для запоминания количества оценок по данному предмету, и СТРОКА - для формирования информации, выводимой на экран дисплея. Процедура ОЦЕНКА_ЧЕТВЕРТИ вызывает "процедуру-функцию" ВЫЧИСЛИТЬ, которая прорабатывает работу вычисления средней оценки ученика по данному предмету. Ее определение следующее:

```
ВЫЧИСЛИТЬ (СПИСОК)
ЕСЛИ СОСЧИТАТЬ :СПИСОК > 0
    ТО СУММА := :СУММА + ПЕРВЫЙ :СПИСОК
    ВЫЧИСЛИТЬ БЕЗ_ПЕРВОГО :СПИСОК
ВСЁ
РЕЗУЛЬТАТ :СУММА / :ДЛИНА
```

Допустим, что поменялся адрес ученика. В таком случае необходимо рассматривать подпись СВЕДЕНИЯ:

```
СВЕДЕНИЯ := ПЕРВЫЙ :ЖУРНАЛ
```

Новый адрес запишем в переменную АДРЕС:

```
АДРЕС := "РИГА, УЛ. КУДРАС, 10-36, 610887"
```

Как известно, адрес ученика содержится в третьем - последнем элементе подписки СВЕДЕНИЯ. Этот элемент нужно изменить:

```
СВЕДЕНИЯ := В_КОНЕЦ БЕЗ_ПОСЛЕДНЕГО :СВЕДЕНИЯ :АДРЕС
```

Наконец, измененный подпись СВЕДЕНИЯ необходимо опять включить в список ЖУРНАЛ:

```
ЖУРНАЛ := СОЕДИНИТЬ :СВЕДЕНИЯ БЕЗ_ПЕРВОГО :ЖУРНАЛ
```

Аналогично создается список с оценками всех учеников класса, но такой список будет уже больше по объему и его структура будет сложнее.

З а к л ю ч е н и е

Система Лого, описанная в [1], позволяет организовать практическую работу учащихся по отдельным вопросам темы "Алгоритмический язык" курса "СИИВТ" [5]: "Черепашка и ее ко-

манды", "Вспомогательные алгоритмы", "Команда повторения n раз".

Описанные в этой статье возможности языка Лого находятся в стадии проектирования и реализации и войдут в расширенную (в сравнении с [1]) версию системы, которая предназначена для изучения следующих тем курса "ОИИВТ" [5]: "Арифметические выражения и правила их записи", "Команда повторения пока", "Условия, команды ветвления и контроля", "Величина в алгоритмическом языке, команда присваивания", "Результаты алгоритмов и алгоритмы-функции", "Символьные и литерные величины", "Команды ввода-вывода в алгоритмическом языке".

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Шкерстена И. А. Система Лого для ПЭВМ "УАМАНА MSX-2". I. "Черепашья графика" // ЭВМ в образовании. Педагогические программные средства. 1989. С. 67-81.
2. Douglas H. Clements. Computers in early and primary education. New-Jersey: Prentice Hall Inc. 1985. 322 p.
3. Morrell B. Turtle Graphics on the BBC microcomputer and Acorn Electron. Cambridge: Acornsoft Ltd. 1984. 70 p.
4. Govier H., Neave M. Logo Challenge. London: Addison-Wesley Publishers Ltd. 1983. 40 p.
5. Ершов А. П., Кушниренко А. Г., Лебедев Г. В., Семенов А. Л., Шень А. Х. Основы информатики и вычислительной техники: Пробный учебник для средних учебных заведений / Под ред. А. П. Ершова. М: Просвещение. 1988. 207 с.
6. Norman A., Cattell G. LISP on the BBC Microcomputer. Cambridge: Acornsoft Ltd. 1983. 197 p.

Межвузовский сборник научных трудов
ЗВМ В ОБРАЗОВАНИИ. ПЕДАГОГИЧЕСКИЕ ПРОГРАММНЫЕ СРЕДСТВА
Рига: ЛГУ им. П. Стучки. 1989. С. 102-112

УДК 37.01:007+371.3+378.147+681.3.068

О. М. Елкина, И. Я. Озиш.

С. И. Павлов, Е. В. Перфильева, В. Л. Цилевич

Вычислительный центр при ЛГУ им. П. Стучки

УЧЕБНЫЙ ТРЕХАЛФАВИТНЫЙ ТЕКСТОВЫЙ РЕДАКТОР ДЛЯ КУВТ-86

Возможности распространения персонального компьютера обусловлены наличием широкого набора программных средств общего назначения, к которым в первую очередь относятся системы подготовки текстов. Для персонального компьютера IBM PC, являющегося по существу мировым стандартом, популярны текстовые процессоры WordStar, Easy Writer, CHI Writer и др. Системы обработки текстов являются составной частью интегрированных систем обработки информации Lotus 1-2-3, Symphony, Framework и др. [1].

Редакторы текстов для компьютеров, поставляемых для системы образования, по сравнению с указанными, выше имеют ограниченный набор команд. Примером могут служить редакторы текстов VIEW для ПЭВМ "Acorn" [2,3], TOR и "Микромир" для ПЭВМ "Yamaha" [4-6]. Надо отметить, однако, что при проектировании названных редакторов для комплектов учебной вычислительной техники (КУВТ) авторы тяготели скорее к "идеологии" построения редактора для пользователя, нежели редактора для обучения.

Разработка программного обеспечения, в том числе учебного, для микроЭВМ "Электроника БК-0010" (далее "БК"), поставляемой в систему образования СССР в составе локальной сети КУВТ-86 с управляющей микроЭВМ "ДВК-2М", наталкивается на определенные трудности вследствие крайне ограниченных возможностей микроЭВМ "БК", спроектированной как бытовой компьютер. Поэтому попытки разработки для "БК" редактора текста по типу пользовательских даже при высоком качестве програм-

мирования, как правило, не очень удачны, поскольку развитость одних функций, например, определение алфавитов *) или обмен с диском управляющей ЭВМ [7] оборачивается чрезмерными ограничениями других функций - в основном организации "дружественной" среды, особенно необходимой для работы в режиме диалога учащегося с соответствующим программным средством учебного назначения.

Принимаясь за разработку "очередного" текстового редактора для "БК", авторы видели свою задачу в создании программного средства для использования его прежде всего в условиях насыщенного курса "Основы информатики и вычислительной техники" ("ОИИВТ"), с целью ознакомления потенциальных пользователей компьютера - учащихся старших классов средней школы - с основными возможностями пользовательских текстовых процессоров на примере специально созданной "действующей модели" - учебного текстового редактора (УТР).

1. Принципы разработки УТР

При проектировании УТР разработчики исходили из следующих основных принципов:

1. Необходимость создания "дружественной среды" диалога с учащимся, наличие "меню", комментариев, сообщений об ошибках, контекстно зависимых подсказок.
2. Возможность создания и редактирования текстов, содержащих символы одновременно трех алфавитов - латинского, латинского и русского.
3. Необходимость реализации минимального набора функций - перемещения курсора, редактирования символа, строки, блока (в том числе форматирования), разметки текста для печати.
4. Необходимость сохранения текстовых файлов на диске управляющей ЭВМ.

*) Кузьмина Л.М. Текстовый редактор с переменным алфавитом для КУВТ-87 // Программа 48-я научной конференции ЛГУ им. П.Стучки (г.Рига, 10-17 февраля 1989 года). Секция прикладной информатики. Рига: ЛГУ им. П.Стучки, 1989. С. 40.

5. Возможность распечатки текстовых файлов на печатающем устройстве ("D-100") управляющей ЭВМ с максимальным использованием стандартных шрифтов и режимов печати.

6. Возможность подготовки образцов текстов, наиболее часто встречающихся в учебном процессе, с использованием разработанного УТР.

7. В составе комплекса программ УТР имеется набор простых программ на языке программирования BASIC, изучение которого входит в программу курса "ОИИВТ", для анализа на уроках информатики некоторых алгоритмов редактирования текстов.

Далее остановимся на некоторых возможностях УТР.

2. Режимы УТР. Планировка экрана

Диалог с учащимися ведется на латышском или русском языке по выбору пользователя УТР. Выбор языка можно сделать при появлении заставки УТР. Все сообщения, содержащиеся в кадре заставки (рис. 1), приведены на обоих языках.

После выбора языка диалога обучаемому доступны два режима: режим обмена с диском и режим редактирования (соответственно рис. 2 и 3, 4; все сообщения для примера даны на латышском языке).

На экране информация располагается в трех областях:

- строки 19-23 - область комментариев. В режиме обмена с диском в области располагается "меню" доступных команд. В режиме редактирования - полный список доступных команд редактирования и соответствующие им комбинации клавиш. Последний список разделен на пять "страниц", которые появляются циклически в указанной области. Клавиши, соответствующие командам, отмечены инверсией символа (на рис. 1-4 подчеркнуты);

- строки 16-17 - область сообщений об ошибках, контекстно зависимых подсказок и директив. "Мигающие" сообщения являются реакцией на действия пользователя;

- строки 0-14 - информационная область. В режиме обмена с диском в области размещается справочная информация о текстовых файлах. В режиме редактирования - область содержит текущую "страницу" редактируемого текста.

Строки 15 и 18 являются разделительными. В командной

(c) Rīga: LVU-SC, SAIPL Versija 1989-1.1
(c) Рига: ВЦ при ЛГУ, ЛШВИ Версия 1989-1.1

M A C I V U T E K S T U
R E D A K T O R S

У Ч Е В Н Ы Й Т Е К С Т О В Ы Й
Р Е Д А К Т О Р

I z v e l e j i e t i e s d i a l o g a v a l o d u !
В ы б е р и т е я з ы к д и a л o г a !

L - latviešu valoda СТОП - izeja no redaktora
P - русский язык выход из редактора

Рис. 1. Планировка экрана для режима выбора языка диалога.

(c) Rīga: LVU-SC, SAIPL Versija 1989-1.1

A R M A I N A S R E Z I M S A R D I S K U

Rediģejamais teksts ir nolasis no faila _____

Vel var ievadīt 7500 simbolus

Komanda: Lasīšana
Faila nosaukums: _____

I e v a d i e t f a i l a n o s a u k u m u !

СТОП - izeja no redak ora U - jaunais teksts
P - faila lasīšana LO - faila lasīšana
RED - rediģēšanas režims P - faila papildināšana
G - faila saglabāšana
K - diska katalogs

Рис. 2. Планировка экрана для режима обмена с диском.

```
-----  
! 1,5 * * * 7365 * ЗАГЛ * ЛАТ *  
!  
! 59  
!  
!  
! 10  
!  
!  
! 60  
!  
! -----  
! I e v a d i e t t e k s t u v a i k o m a n d u !  
!  
! -----  
! РЕД - а р м а и п а s р е ж и м s а р d и s k u АР2 ? - k о м е н т а р у т у р п и н а ј u m s  
! K u r s o r a п а р в и e t о ш а n a  
! v -> <- - p a r v i e n u p o z и c и ј u  
! АР2 = / АР2 > - p a r v i e n u л а s п u c и (15 r и n d a s) u z a u g s u / л e ј u  
! АР2 < / АР2 ≥ - u z и e п р и e k ш e ј а / н а k о ш а в а р d a с а k u m u  
! -----  
! ○
```

Рис. 3. Планировка экрана для режима редактирования.

```
-----  
! S i m b o l a r e d и g e ш а n a  
! КТ - s i m b o l a и e s п р а u ш а n a s / a и z v и e t о ш а n a s р e ж и m a п а р s л e g ш а n a  
! !--> - и e s п р а u c t т u k ш u m u !<-- - и z d ж e c t c и m b o l y  
! -----  
! R и n d a s r e d и g e ш а n a  
! АР2 0 - и e s п р а u c t т u k ш o р и n d y АР2 7 - с a d a л и t r и n d y  
! АР2 0 - и z m e c t r и n d y АР2 8 - с a v и e n o c t r и n d y  
! -----  
! B л o k a r e d и g e ш а n a  
! АР2 2 - u c т a д и t m a r k e r и АР2 4 - п а р v и e t o c t б л o k y  
! (и e z и м e t б л o k y) АР2 5 - n o k п e t б л o k y  
! АР2 3 - f o r m a t и z e t б л o k y АР2 6 - и z d ж e c t б л o k y  
! -----  
! I z d p r u k a s р e ж и m и  
! АР2 + - u c т a д и t и n t e р в а л y c т a р п r и n d а m u n п а р n y м e р e t r и n d a s  
! АР2 1 - u c т a д и t л a s п a c п u c e s н y м e р a р а d и t a ј u  
! -----
```

Рис. 4. Комментарии при циклическом нажатии комбинации клавиш АР2 ?

строке выводятся краткие сообщения о разработчике, версии, а также некоторые команды монитора.

3. Режим обмена с диском

В режиме доступны следующие команды (список команд на латинском языке приведен на рис. 2):

- **Новый.** Освобождается область памяти, отведенная для редактируемого текста.

- **Чтение.** Автоматически выполняется команда **Новый**. Файл с указанным именем читается с диска. Прочитанный файл размещается в предварительно освобожденной области, отводимой для редактируемого текста.

- **Добавление.** Прочитанный файл (или если не хватит места, то его часть) размещается в области памяти, еще не занятой ранее введенным текстом.

- **Запись.** Отредактированный текст из области памяти записывается в файл на диске под указанным именем.

- **Каталог.** В информационную область экрана выдается список хранящихся на диске файлов, ранее подготовленных с помощью текстового редактора.

4. Режим редактирования

Информационная область (строки 0-14) режима редактирования представляет собой "окно", относительно которого с помощью курсора перемещается текст. "Окно" отсекает от текста "текущую" страницу (15 трок) - см. рис. 3.

Каждая строка содержит 64 позиции (нумерация от 0 до 63). Для размещения редактируемого текста отводятся 60 позиций (с 3 по 62). Строки могут быть пронумерованы - первые две позиции (0 и 1). Возможна нумерация страниц (отмечается первая строка каждой страницы).

В командной строке содержится разметка строки, некоторые команды монитора, а также число свободных байт в области памяти, отведенной для редактируемого текста. В первых четырех позициях (от 0 до 3) командной строки указан выбранный интервал для последующей печати текста.

В режиме редактирования обучаемому доступны следующие группы команд (список команд на латышском языке приведен на рис. 3.4):

- перемещение курсора:
- редактирование символа:
- редактирование строки:
- редактирование блока:
- режимы печати.

Минимальный набор выполняемых команд текстового редактора, необходимый для ознакомления на занятиях по курсу "ОИИВТ", по мнению авторов, следующий:

1. Перемещение курсора: на одну позицию вверх, вниз, влево или вправо; на одну "страницу" (15 строк) вверх и вниз; с первого символа слова на первый символ соседнего слова.

2. Редактирование символа: вставка пробела; стирание символа; переключение режима замена/вставка (при вводе текста - замена ранее введенного символа на новый или сохранение ранее введенного символа посредством его сдвига).

3. Редактирование строки: вставка пустой строки; стирание строки; объединение двух строк; разделение одной строки на две.

4. Редактирование блока: разметка блока - обозначение части текста двумя курсорами; стирание блока; перенос блока в указанное курсором место; дублирование блока на указанное курсором место.

Ключевым моментом является наличие в УТР команды форматизации блока с выравниванием редактируемого текста по правому краю. Команда может применяться после других команд редактирования для придания тексту эстетически оформленного вида.

5. Режимы печати: установка интервала для печати (числа строк в странице); нумерация страниц.

5. Распределение памяти и структура программы.

Кодовая таблица и алфавиты редактора.

Память "БК" (без экранного ОЗУ) - 16 К. доступная для программы, разделена на следующие зоны:

- область редактируемого текста;
- область программы УТР.

Программа УТР имеет оверлейную структуру и содержит:

- корневой сегмент (в том числе программу вывода букв латышского алфавита; программу обработки ошибок; общие переменные и ядро УТР);
- транзитная область (в том числе программы обмена с диском или программы редактирования текста; текстовые константы для диалога на латышском или русском языках).

Независимо от УТР как в режиме МОНИТОРА, так и в режиме BASICа может применяться программа LPLPLP вывода на экран букв латышского алфавита.

Клавиатура "БК", состоящая из 74 клавиш, позволяет получить полный набор 8-ми разрядных кодов.

Изображение символа на экране дисплея "БК" формируется драйвером ТВ-приемника, находящимся в ПЗУ по системному запросу программы пользователя. Изменение изображения символа возможно с помощью специальной программы, постоянно находящейся в оперативной памяти, перехватывающей системные запросы и формирующей, если это необходимо, другое изображение символа. Для работы с латышским алфавитом разработана программа LPLPLP, позволяющая перехватывать системный запрос драйверу ТВ-приемника и после вывода стандартного изображения символа дополнять его диакритическим знаком.

Для кодов символов латышского алфавита с диакритическими знаками использована часть кодовой таблицы, отведенная под графические символы (табл. 1). Для обеспечения удобства работы с клавиатурой коды символов латышского алфавита формируются при одновременном нажатии клавиши <AP2> и соответствующей клавиши без диакритического знака (например, <S>). Для этого программа LPLPLP перехватывает обработку кодов, формируемых по регистру <AP2>. В регистре <PUC> при одновременном нажатии <AP2> и алфавитно-цифровой клавиши код обрабатывается стандартно, и на экран выводится графический символ. В регистре <ЛАТ> соответствующие символы выводятся с диакритическим знаком.

Таблица 1. Кодовая таблица для трехалфавитного редактора текстов.

0	1	2	3	4	5	6	7	8	9	а	в	с	д	е	ф
0	СБР ТАБ	ПР	0	Q	P	'	p		ШАГ	к	↑	ю	л	Ю	П
1		1	А	Q	a	q	ПОВТ	КРАС	А	←	а	я	А	Я	
2	К	"	2	В	R	b	r	ИИД СУ	ЗЕЛ	з	З	б	р	Б	Р
3	КТ	ВС	#	3	С	S	с	з	СИН	С	↑	ц	с	Ц	С
4	ГТ	X	4	D	T	d	t	БЛОК РЕД	ЧЕРН	с	з	д	т	Д	Т
5	П	%	5	E	U	e	u		ГРАФ	Е	0	е	у	Е	У
6	К	&	6	F	V	f	v		ЗАП	е	д	е	х	Ф	Ж
7	ЗБ	⇒	7	G	W	g	w		СТИР	б		г	в	Г	В
8	←	←	<	8	H	X	h	x	РЕД	б	◆	х	ь	Х	Ь
9	⇒	→	9	I	V	i	v	ТАБ	⇐	1	↓	и	ы	И	Ы
а	↪	↑	к	J	Z	j	z		КУР- СОР	1	ž	й	э	Й	Э
в	↓	+	.	K	C	k	c		32, 64	к	ž	к	ш	К	Ш
с	СБР	↪	.	L	\	l	l	РП	ИИВ СИИ	к	Л	л	э	Л	Э
д	УСТ ТАБ	↪	-	=	M	J	m	J	ИИВ ЭКР	↓	1	м	щ	М	Щ
е	РУС	↪	.	>	N	-	n	-	УСТ ИИД	N	→	н	ч	Н	Ч
ф	ЛАН	↪	/	?	0	-	o	ЗБ	ПОДЧ	р	⌘	о	ь	О	Ь

6. Вывод текстового файла на печать с управляющей ЭВМ

Вывод на печать содержимого текстового файла, сохраненного на диске, производится с использованием печатающего устройства (в описываемом варианте - "D-100"), подключенного к управляющей ЭВМ "ДВК-2М". Специальная программа, работающая в "ДВК-2М", проводит построчный анализ текстового файла, содержащего символы трех алфавитов. Строка печатается в два приема:

- выводятся стандартные символы латинского и русского алфавитов:

- символы с кодами латышского алфавита дополняются необходимыми диакритическими знаками после программного переключения печатающего устройства в графический режим.

Такой подход позволяет программно управлять режимами печатающего устройства, в том числе:

- изменять "качество" печати, например, задавать выделительный шрифт и/или шрифт в двойной пропечаткой;
- изменять интервал между строками, например, один, два или полтора интервала;
- изменять шрифт, например, уплотненный или укрупненный.

7. Использование УТР для подготовки текстов

С использованием разработанного УТР подготовлены некоторые образцы текстов из приведенного ниже списка, который, естественно, не является исчерпывающим:

- календарь;
- расписание уроков, дневник, табель успеваемости;
- план урока, текстовый опорный конспект;
- список приборов и технических средств;
- список литературы к уроку, теме, курсу;
- латышско-русско-английский словарь;
- диктандо, план, сочинение;
- таблица умножения, таблица дифференцирования элементарных функций;
- список адресов и телефонов родителей учеников;
- объявление, график дежурств, заметка в стенную газету;
- заявление, деловое письмо, приказ.

8. Алгоритмы редактирования на языке BASIC

Для анализа на уроках информатики составлены алгоритмы, блок-схемы и программы на языке BASIC, реализующие следующие команды редактирования: перемещение курсора на одну позицию, на одну страницу, в начало предыдущего или следующего слова; переключение режима вставка/замена символа, вставка пробела и удаление символа; работа со строками.

По составленным алгоритмам подготовлены действующие на

"БК" программы-упражнения (за исключением обработки текстовых строк - из-за дефектов в применяемой в настоящий момент версии языка BASIC). Программы-упражнения могут применяться для развития у учащихся навыков работы с клавиатурой и с перечисленными командами полной версии УТР.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Брябрин В.М. Программное обеспечение персональных ЭВМ. М.: Наука. 1988. 272 с.
2. VIEW Guide. A guide to the VIEW Word Processor for the BBC Microcomputer. Cambridge: Acornsoft. 1984. 51 p.
3. Into VIEW. An introduction to word processing on the BBC Microcomputer. Cambridge: Acornsoft. 1984. 70 p.
4. Бараз Л.С. Многофайловый текстовый редактор // Тематическая выставка "Компьютеризация образования". Аннотации программных средств. М.: ГКНО СССР, ВДНХ СССР. 1988. С. 32.
5. Кушниренко А.Г., Лебедев Г.В., Дымченко А.Г. и др. Пакет программных средств ("Микромир", "Е-, Альфа-, ЭВМ-практикум") для серийных школьных ЭВМ (КУВТ "Yamaha", "Корвет", "УК-НЦ", ЭВМ "ДВК-2М", ПЭВМ "БК-0010") // Тематическая выставка "Компьютеризация образования". Аннотации программных средств. М.: ГКНО СССР, ВДНХ СССР. 1988. С. 93
6. Сапир М.В., Романов С.Г., Комаров К.В. Учебный экраный редактор текстов // Тематическая выставка "Компьютеризация образования". Аннотации программных средств. М.: ГКНО СССР, ВДНХ СССР. 1988. С. 27-31.
7. Гриценко А. Система подготовки текстов для КУВТ-86 // Информатика и образование. 1989. № 3. С. 55-59.

УДК 37.01:007+371.3+681.3.06

Л. П. Климанс. И. В. Опмане

Вычислительный центр при ЛГУ им. П. Стучки

ПРОГРАММА "ЭЛЕКТРОННАЯ ТАБЛИЦА" В УЧЕБНОМ ПРОЦЕССЕ

Компьютерная грамотность подразумевает наличие у пользователя определенного объема знаний о роли и месте компьютера в той или иной сфере человеческой деятельности и навыков применения компьютера и программного обеспечения. При этом для различных категорий пользователей объем знаний может существенно различаться. Однако имеются такие прикладные программы, которые широко применяются практически всеми пользователями независимо от сферы их деятельности. К ним относятся в первую очередь текстовые редакторы, базы данных, электронные таблицы [1].

В состав курса информатики и вычислительной техники для средней школы [2] включена тема "Электронная таблица" (ЭТ). Изучение конкретной прикладной программы в процессе решения задач включает:

- понимание цели, назначения, возможностей программы;
- освоение возможностей программы;
- изучение способов обработки и анализа информации;
- изучение возможных применений программы.

Пользователю "ЭТ" необходимо освоить также управление программой, структуру информации на экране, виды сообщений и возможную реакцию на них.

Цель программы "ЭТ" - создание таблицы, поля которой могут содержать числа, названия, формулы, и обеспечение возможностей выполнения действий с элементами таблицы. Организация экрана должна способствовать получению информации о

данных в элементах таблицы, в также информации, связанной с управлением программой, организацией диалога и др.

В работе приводится описание реализации программы "ЭТ" для микроЭВМ "Электроника БК-0010" (далее "БК"), входящей в состав КУВТ-86. Для загрузки блоков программы, файлов таблицы и распечатки таблицы используются возможности локальной сети КУВТ-86.

1. Логическая структура таблицы

Программа дает возможность создавать в памяти микроЭВМ "БК" таблицу, состоящую не более, чем из 25 строк и 12 столбцов. Элементами таблицы могут быть действительные числа, содержащие не более 7 значащих цифр.

Каждая строка имеет поле для названия строки. Каждый столбец имеет два поля для его названия. Название может содержать не более 8 символов.

Кроме того, в каждой строке и в каждом столбце предусмотрено поле для хранения номера. Номер состоит из цифр (начиная с "1", с шагом "1") и буквы К (для столбца) или R (для строки).

Числовые значения таблицы вводятся с клавиатуры или вычисляются по формуле. Каждый столбец и каждая строка имеет поле, куда помещается формула, и ее длина не должна превышать 30 символов. Номера строк и столбцов используются при записи формул для указания соответствующих аргументов. Например, если во второй строке записана формула $2 * R1$, то каждый элемент второй строки будет вдвое больше соответствующего элемента первой.

Логическая структура таблицы показана на рис. 1. Числовые значения, формулы, названия соответствуют конкретной решаемой задаче. Номера строк и столбцов $R1, R2, \dots, R25$; $K1, K2, \dots, K12$ зачисляются при формировании таблицы в начале работы программы, а поля определяются пользователем во время работы.

f	f	f	f	f	f	f	f	f	f	f	f	f
K1	K2	K3	K4	K5	K6	K7	K8	K9	K10	K11	K12	
n	n	n	n	n	n	n	n	n	n	n	n	n
n	n	n	n	n	n	n	n	n	n	n	n	n
f	K1	n	a	a	a	a	a	a	a	a	a	a
f	R2	n	a	a	a	a	a	a	a	a	a	a
f	R3	n	a	a	a	a	a	a	a	a	a	a
f	R24	n	a	a	a	a	a	a	a	a	a	a
f	R25	n	a	a	a	a	a	a	a	a	a	a

Рис. 1. Логическая структура таблицы:
 а - поле для числовых значений;
 n - поле для названий;
 f - поле для формул.

2. Запись формул

Формулой является любое арифметическое выражение, записанное по правилам языка BASIC. Аргументами выражения могут быть константы и номера строк и столбцов.

В записи формулы для строки допустимы только номера строк R1, R2, ..., R25, а также обозначения RS и RX, где RS означает сумму всех предыдущих (до строки формулы) строк, RX означает умножение всех предыдущих строк.

В записи формулы для столбца допустимы только номера столбцов: K1, K2, ..., K12, а также KS и KX (аналогично RS и RX).

В записи формулы допускаются знаки арифметических операций: "+"; "-"; "*"; "/"; "—" (возведение в степень); "\" (целая часть от деления); "MOD" (остаток от деления). Дополнительно введена операция "%" (процент). Могут использоваться математические функции: SIN; COS; LOG; TAN; SQR; ATN; EXP; ABS; FIX; INT; SGN. Как для арифметических выражения, так и для математических функций тип и область определения аргументов соответствует правилам языка BASIC для "ЕК".

3. Действия с элементами таблицы

Программа позволяет выполнять следующие действия:

- определение значений элементов таблицы (первоначальный ввод с клавиатуры числовых данных, названий, формул);
- замену значений элементов таблицы (повторный ввод с клавиатуры числовых данных, названий, формул);
- вычисление значений элементов таблицы для тех строк и столбцов, для которых заданы формулы;
- автоматическое вычисление значений элементов таблицы при каждой их замене;
- запись таблицы на дискету или кассету;
- чтение таблицы с дискеты или кассеты в память "БК";
- распечатку части таблицы, отображенной на экране;
- распечатку копии всей таблицы без формул;
- распечатку всех формул для данной таблицы;
- распечатку любой строки или столбца;
- вывод на экран значений строк или столбцов в виде столбиковой диаграммы;
- вывод на экран значений строк или столбцов в виде линейного графика.

4. Структура экрана

Для отображения результатов работы программы, а также для организации диалога с пользователем, используется экран, разделенный по типу информации на три зоны.

Верхняя, основная часть экрана (15 строк) используется для вывода элементов таблицы. На экране не может быть отображена вся таблица из 25 строк и 12 столбцов, поэтому поле экрана используется в качестве "окна", которое накладывает-ся на нужную ее часть. "Окно" позволяет отобразить 12 строк и 6 столбцов, а также соответствующие им названия и номера. Используя управляющий курсор можно выбрать нужное поле таблицы для ввода или изменения значения. Используя этот же управляющий курсор, можно изменить положение "окна" на таблице.

Следующая зона экрана (с 15 по 18 строку) служит для вывода сообщений программы.

Во время диалога с пользователем эта зона используется

также при вводе информации. Введенная с клавиатуры информация (числа, формулы, названия) отображается в 16 строке экрана. Только после нажатия <ВВОД> эта информация помещается в соответствующее поле таблицы.

Формулы отображаются на экране в зоне сообщения в тот момент, когда управляющий курсор находится в поле номера строки или столбца. "Меню" содержит список команд. Некоторые команды могут иметь параметры. При помощи клавиш ">" и "<-" управляющий курсор устанавливается на название команды или параметр. Нажатием <ВВОД> устанавливается значение параметра, либо выполняется команда.

5. Структура программы

Программа состоит из четырех блоков. Первый управляющий блок выводит на экран список трех основных режимов, каждому из которых соответствует один блок программы. При выборе пользователем нужного режима, загружается и запускается соответствующий блок программы. Каждый из трех блоков заканчивает работу загрузкой и запуском управляющего блока. Названия блоков соответствуют названиям режимов в строках "меню".

При загрузке и запуске любого блока программы таблица в памяти "БК" не сохраняется и ее нужно считать заново.

Логическая структура программы приведена на рис. 2.

6. Функциональные возможности основных блоков программы

Для краткого описания возможностей блоков программы представлен список команд в "меню". Если команда имеет параметры, то для каждого из них представлена совокупность возможных значений.

Команды и параметры блока программы для создания таблицы показаны на рис. 3:

ТАБЛИЦА - установка управляющего курсора для определения и изменения элементов таблицы;

СЧЕТ - вычисление значения элементов таблицы по формулам (если такие заданы);

1-я параметр задает порядок выполнения формул

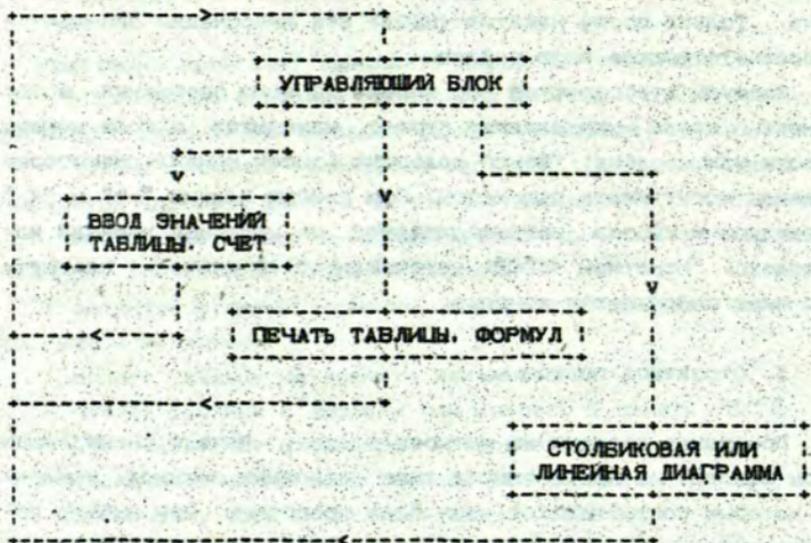


Рис. 2. Структура программы.

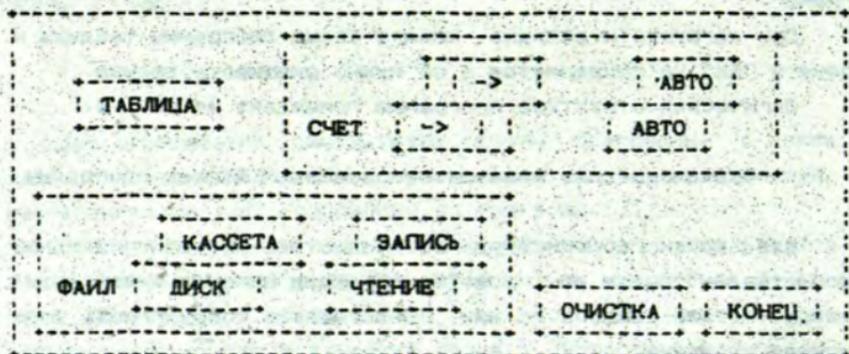


Рис. 3. Команды и прометры блоков создания таблицы.

2-я параметр позволяет установить режим автоматического вычисления формул при каждом вводе или замене элемента таблицы.

ФАЙЛ - запись или чтение таблицы;

ОЧИСТКА - очистка всех полей таблицы;

КОНЕЦ - загрузка и запуск управляющего блока программы.

На рис. 4 представлены команды и параметры блока вывода информации на печать:

ТАБЛИЦА - установка управляющего курсора и просмотр таблицы;

ФАЙЛ - чтение файла таблицы;

ПЕЧАТЬ - распечатка информации:

1-я параметр задает вид печатаемой информации (копия экрана, таблица, формулы, строка, столбец);

2-я параметр указывает тип подключения принтера;

3-я параметр указывает номер, если 1-я параметр задает печать строки или столбца;

4-я параметр задает режим использования буфера;

КОНЕЦ - загрузка и запуск управляющего блока программы.

Рис. 5 содержит список команд и параметров для выдачи диаграмм или графиков.

ТАБЛИЦА - установка управляющего курсора и просмотр таблицы для выбора данных;

ФАЙЛ - чтение таблицы;

ГРАФИК - построение на экране диаграммы или графика:

1-я параметр задает вид;

2-я параметр указывает выбор переменной (строка или столбец);

В нижней части за элементами СТРОКИ и СТОЛБЦЫ указываются конкретные номера строк или столбцов для переменных и конкретные позиции переменных для построения графика. Как видно из рис. 5, программа допускает не более пяти переменных и для каждой переменной не более пяти значений.

КОНЕЦ - загрузка и запуск управляющего блока программы.

7. Использование электронных таблиц обработки информации

Первоначально "ЭТ" использовались в коммерческом деле [3]. "ЭТ", в основном, предназначены для работы с числовой информацией и применяются менеджерами, экономистами, бухгал-

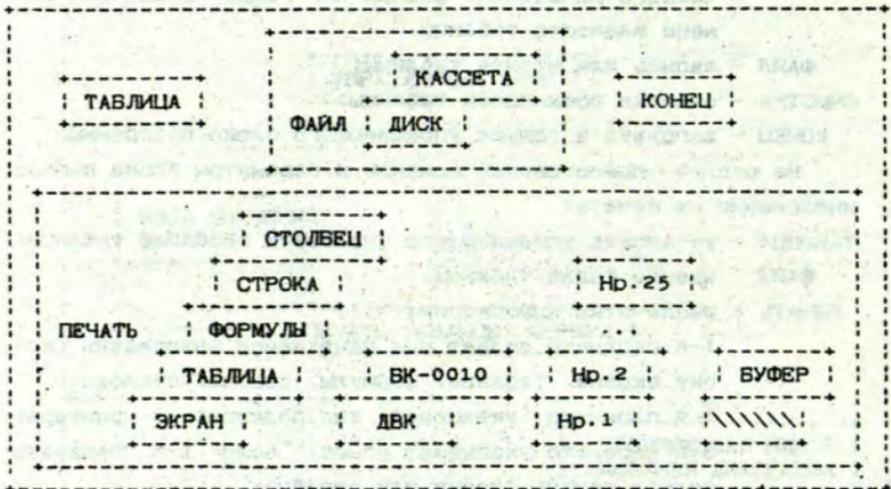


Рис. 4. Команды и параметры блока вывода на печать.

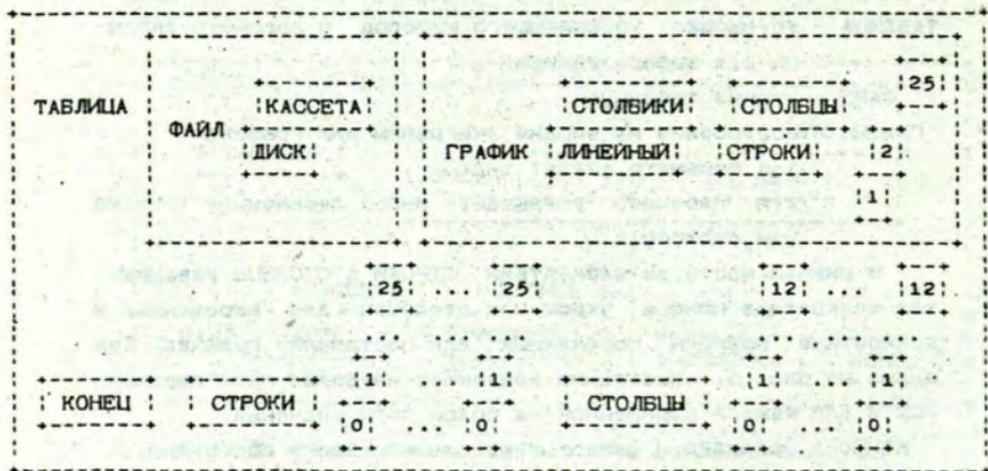


Рис. 5. Команды и параметры блока графического изображения данных.

терами, работниками сберкасс. При анализе экономической информации часто возникают ситуации, когда целесообразно рассмотреть несколько вариантов: "что будет, если...". С помощью "ЭТ" легко моделировать такие ситуации, для каждого набора данных вычислять характерные показатели и выбирать наиболее подходящее решение проблемы.

С помощью "ЭТ" может быть освоена обработка экономической информации. Для этого полезно рассмотреть задачи планирования финансовых операций, а также распределения средств семейного бюджета [4].

В табл. 1 приведен пример планирования семейного бюджета. В первых столбцах располагается месячная заработная плата членов семьи и общая сумма. В следующих столбцах - предполагаемые расходы и общая сумма расходов. Последний столбец содержит накопление. Задача состоит в выборе оптимального варианта расходов для накопления определенной суммы в течение года.

Может быть поставлена задача выбора списка газет и журналов из набора желаемых изданий при отведенной сумме денег для этих нужд.

Другой класс экономических задач связан с операциями в сберегательной кассе. Табл. 2 иллюстрирует состояние вкладов за разное время при различных процентах прироста.

"ЭТ" нашли применение при обработке экспериментальных данных. На начальном этапе вычислительная техника в основном использовалась для сбора и хранения информации и довольно ограниченно для обработки и анализа данных [5]. Для решения каждой задачи приходилось осваивать разные программы, разработанные на различных принципах. Для экономии времени школьников и студентов во время лабораторных исследований достаточно иметь простую программу для обработки и анализа данных. Таким инструментом являются "ЭТ". В случае обработки экспериментальных данных "ЭТ" могут быть использованы для промежуточных расчетов и расчетов ошибок в курсе физики [5].

В математике "ЭТ" удобно использовать для иллюстрации методов решения задач. С помощью программы решаются уравнения следующими приближенными методами: методом деления отрезка пополам, методом хорд или касательных. Имеется также возможность проследить по шагам выполнение алгоритма вычис-

Таблица 1. Таблица и формулы для анализа бюджета семьи.

	K1	K2	K3	K4	K5	K6	K7	K8	K9
	ЗАРПЛА: ТА КУВА:	ЗАРПЛА: ТА ХЕМИ:	СТИПЕНДИ: СИНА	ДОХОД: СЕМЬИ	НА: ПИТАНИЕ	ДОПЛАТ: ПОКУПКИ	ОБЩЕ: РАСХОД:	НАКОП: РАСХОД:	ЛЕННЕ
R1: ЯНВАРЬ	350:	150:	60:	560:	200:	0:	110:	310:	250:
R2: ФЕВРАЛЬ	350:	150:	60:	560:	200:	200:	110:	510:	50:
R3: МАРТ	410:	150:	60:	620:	200:	0:	110:	310:	310:
R4: АПРЕЛЬ	350:	150:	60:	560:	200:	250:	110:	560:	0:
R5: МАЙ	350:	150:	60:	560:	200:	0:	110:	310:	250:
R6: ИЮНЬ	430:	150:	60:	640:	200:	100:	110:	410:	230:
R7: ИЮЛЬ	350:	150:	260:	760:	200:	100:	110:	410:	350:
R8: АВГУСТ	350:	150:	260:	760:	200:	0:	110:	310:	450:
R9: СЕНТЯБРЬ	400:	150:	60:	610:	200:	700:	110:	1010:	-400:
R10: ОКТЯБРЬ	350:	150:	60:	560:	200:	0:	110:	310:	250:
R11: НОЯБРЬ	350:	150:	60:	560:	200:	200:	110:	510:	50:
R12: ДЕКАБРЬ	500:	300:	60:	860:	200:	0:	110:	310:	550:
R13: ГОДОВОЕ	4540:	1950:	1120:	7610:	2400:	1550:	1320:	5270:	2340:
R14: МЕСЯЧНОЕ: 37E+03:	167.5:	79E+02:	63E+03:	200:	12E+03:	110:	43E+03:	195:	

$$K4 = K1 + K2 + K3$$

$$K8 = K5 + K6 + K7$$

$$K9 = K4 - K8$$

$$R13 = R8$$

$$R14 = R13 / 12$$

Таблица 2. Таблица. формула для вычисления накопления и распечатка 7. столбца.

	K1	K2	K3	K4	K5	K6	K7
	ВШАА	ПРОЦЕНТ:		ЧИСЛО		КОНЕЧН:	
		%		ЛЕТ		С ЧИСЛА	
R1	500:		2:		1:		510:
R2	500:		2:		5:		1.55E+03:
R3	1000:		3:		1:		1030:
R4	1000:		3:		5:		1.11E+04:
R5	10000:		10:		5:		16105.1:
R6	10000:		10:		10:		1.25E+05:

$$K7 = K1 * (1 + K3 / 100)^{K5}$$

1. СТРОКА 7. СТОЛБЕЦ: 510
2. СТРОКА 7. СТОЛБЕЦ: 552.0404
3. СТРОКА 7. СТОЛБЕЦ: 1030
4. СТРОКА 7. СТОЛБЕЦ: 1159.274
5. СТРОКА 7. СТОЛБЕЦ: 16105.1
6. СТРОКА 7. СТОЛБЕЦ: 23937.42

ления и повторить итерации необходимое количество раз [6]. С помощью "ЭТ" возможно вычисление значений функций, числовых рядов, разложение в ряд Тейлора.

В задачах алгебры при помощи "ЭТ" (иногда "ЭТ" называют табличными процессорами) можно найти сумму матриц, можно преобразовать детерминант методом сложения строк и столбцов, решить систему линейных уравнений методом сложения уравнений.

"ЭТ" можно применять для анализа статистической информации, например, для определения средней оценки ученика или класса по предмету.

Для использования "ЭТ" пользователь должен решить в некотором смысле проблему "программирования". Для каждой конкретной задачи пользователь решает проблему размещения данных (по строкам или по столбцам) и обработки данных (выбор формул, размещение вычисленных данных в строках и столбцах, порядок вычисления по формулам). Как и при программировании, расположение данных тесно связано с формулами, с порядком их выполнения. При проектировании таблиц перечисленные вопросы решаются взаимосвязанно.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Bork A. *Personal Computers for Education*. New York: Harper & Row, Publishers. 1985. 179 p.
2. Геян А.Г., Житомирский В.Г., Липицкий Е.В., Сапир М.В., Шолохович В.Ф. *Основы информатики и вычислительной техники*. Свердловск: Уральский госуниверситет. 1989. 272 с.
3. Дойл У. *Табличный процессор СУПЕРКАЛК для персонального компьютера*. М.: Финансы и статистика. 1987. 320 с.
4. Brown J.M. *Spreadsheets in the Classroom // The Computing Teacher*. 1987. No. 2. Pp. 9-12.
5. Feinberg R., Knitell M. *Microcomputer spreadsheet programs in the physics laboratory // American Journal of Physics*. 1985. Vol. 53. No. 7. Pp. 631-634.
6. Wilson J.W. *VisiCalc in the Elementary School // The Computing Teacher*. 1985. No. 6. Pp. 22-30.

Межвузовский сборник научных трудов
ЭВМ В ОБРАЗОВАНИИ. ПЕДАГОГИЧЕСКИЕ ПРОГРАММНЫЕ СРЕДСТВА
Рига: ЛГУ им. П. Стучки. 1989. С. 124-128

УДК 37.01:007+371.3+378.147+681.3.06

М. В. Витиньш, Л. В. Озолиня, А. Ф. Раудис
Вычислительный центр при ЛГУ им. П. Стучки

УЧЕБНАЯ СИСТЕМА УПРАВЛЕНИЯ БАЗОЙ ДАННЫХ ДЛЯ КУВТ-86

С 1988/1989 учебного года в общеобразовательных школах Латвии вводится новая экспериментальная программа курса "Основы информатики и вычислительной техники" [1]. По сравнению с действующей программой [2]:

во-первых, она ориентирована на школы, имеющие кабинеты вычислительной техники;

во-вторых, вместо отдельного практикума (до сих пор основной формы практических занятий на ЭВМ) в новой программе предусмотрено систематическое использование компьютера на каждом уроке информатики;

в-третьих, переставлены акценты в изучении курса - главное внимание уделяется не основам алгоритмизации и программирования, а современной технологии решения задач на ЭВМ с широким использованием прикладных программных средств;

в-четвертых, из-за сокращения объема учебного материала по ряду тем и интенсификации учебной деятельности в условиях постоянного использования на уроках компьютеров уменьшено время, отведенное на изучение курса, со 102 до 68 часов, что в свою очередь, дает возможность пройти курс за один учебный год (10 класс) в режиме два часа в неделю.

Программа курса, как и программа любого другого общеобразовательного предмета, включает две группы задач:

- формирование общей научной картины мира;
- предоставление специфического набора конкретных знаний, умения и навыков.

Для предложенного курса информатики первая группа задач предполагает ознакомление учащихся с информационными процессами в природе и обществе, а другая составляющая образует компьютерную грамотность школьников.

В программу кроме введения (2 часа) включены следующие темы: первоначальное знакомство с компьютером, компьютер в современном обществе (6 часов); структура и принципы работы ЭВМ (4 часа); алгоритмы и их построение (19 часов); знакомство с программированием (14 часов). Курс завершает блок тем об информационной технологии решения задач (23 часа). Более детализированная развертка последнего блока следующая: текстовый редактор (4 часа); графический редактор (3 часа); базы данных (4 часа); электронная таблица (4 часа); электронная почта (1 час); пакеты прикладных программ, реализующих простейшие численные методы (3 часа); компьютер в учебном процессе (4 часа).

На тему "Базы данных" программой отведены 4 часа. Разбору подлежат: основные понятия баз данных; модели представления данных; язык запросов; базы данных в делопроизводстве, с использованием микроЭВМ.

Во время практических занятия на этих же уроках следует заниматься созданием, заполнением и корректировкой (добавлением, изменением и удалением записей) базы данных, поиском (маркировкой) и просмотром записей, а также статистическим анализом информации.

В результате учащиеся должны знать основные понятия баз данных, модели представления данных, назначение и возможности языка запросов, применение баз данных в делопроизводстве с использованием микроЭВМ и уметь вести практическую работу с базой данных.

Авторами статьи разработана система управления базой данных, предназначенная для поддержки темы "Базы данных" вышеприведенной программы в среде КУВТ-86. В основу разработки легли идеи и решения, изложенные в [3-5]. В базе используется реляционная модель представления данных. Диалог пользователя и системы организован по принципу "меню" и мо-

жет вестись как на русском, так и на латышском языках. Программные модули системы и базы данных в виде файлов размещаются на гибком магнитном диске. Для загрузки программных модулей и обмена данными используется программа локальной сети NET.

Система поддерживает базу данных с максимальным количеством записей 999 (количество записей коррелирует с длиной записи). Допускаются два типа полей:

- символьные, т.е. любые символы (включая буквы латышского алфавита), кроме "*"; над символьными полями нельзя выполнять никакие арифметические операции и проводить сравнения; максимальная длина - 255;

- числовые - содержат только числа; максимальное количество позиций - 14; позиции могут содержать знак числа, шифры и десятичную точку.

В системе реализованы следующие функции:

- создание базы данных;
- заполнение базы и корректировка записей;
- просмотр записей;
- сохранение базы данных;
- поиск (маркировка) записей;
- статистический анализ;
- удаление всех записей;
- удаление базы данных.

В процессе создания базы данных задаются имя базы и структура записей. Имя базы используется одновременно и как имя файла, поэтому может содержать только 6 заглавных букв латинского алфавита (первая буква - не "D") и цифры. Структура записи определяется как последовательность, так и описаниями полей. Поля задаются именами (12 символов) и атрибутами (текст, число, длина, расположение десятичной точки). Изменение структуры заполненной базы не допускается.

Заполнение базы осуществляется в классическом понимании. Под корректировкой записей подразумевается их добавление (вставка), изменение и удаление.

Просмотр записей можно вести как по горизонтали (фик-

сированная запись). так и по вертикали (фиксированное поле).

Сохранение базы данных - запись базы данных из памяти "БК" в файл с именем базы данных на дискете. Запись выполняется только после дополнительного подтверждения желания пользователя. Если на дискете уже имеется файл с данным именем, то пользователю дается возможность изменить имя базы и, следовательно, файла.

Загрузка базы данных - противоположное сохранению базы данных действие: данные из файла на дискете пересылаются в память "БК".

Поиск (маркировка) записей является одной из наиболее важных функций системы управления базой данных. При поиске записей задаются условия, налагаемые на значения полей. Все записи, значения полей которых удовлетворяют заданным условиям, маркируются и образуют подбазу. С подбазой, как и со всей базой, можно проводить просмотр записей, статистический анализ и удаление всех записей. Средства данной системы позволяют строить как простые логические выражения, содержащие одно утверждение, так и сложные выражения с использованием логических связей "и" и "или". Применение скобок не предусмотрено. Для числовых полей имеют место шесть сравнения: =; <; >; >; >; <; <. Значение для сравнения может быть только числовой константой, что существенно обедняет возможность системы. Для текстовых полей допускаются два вида сравнения: на совпадение и на несовпадение. В записи текстовых значений для сравнения имеют особое назначение два символа - "?" и "*". Символ "?" означает, что на соответствующем месте в значении поля может быть любой символ, а символ "*" - любое сочетание символов (в том числе, пустое).

Средства статистического анализа позволяют получить наименьшее и наибольшее значение, сумму и арифметическое среднее значения числовых полей.

Удаление всех записей приводит к удалению данных в базе данных в памяти "БК". Описание базы данных сохраняется. В файле базы данных на дискете изменения нет.

В результате удаления базы данных из памяти "БК" уда-

ляются как данные, так и описание базы данных. Как и в предыдущем случае файл базы данных на дискете не меняется. Удаление файла с дискеты возможно при помощи известных стандартных средств операционной системы.

Следует указать несколько направлений развития системы. В первую очередь это обеспечение вывода данных базы на печать, как при помощи встроенных средств генератора отчетов, так и путем создания текстовых файлов в формате текстового редактора. Другое направление представляют средства сервиса, например, слияние баз данных. Внимания также заслуживает обеспечение возможности ведения обработки данных базы на уровне языка программирования BASIC.

При удачной реализации расширений есть основания считать, что предложенная система управления базой данных учебного назначения может стать системой с признаками профессиональной.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Informātikas un skaitslošanas tehnikas pamati. Ekspērimētālā programma 10. klasei. Rīga: Zvaigzne, 1989. 15 lpp.
2. Программа курса "Основы информатики и вычислительной техники" X-XI классам (102 ч) // Микропроцессорные средства и системы. 1986. № 2. С. 86-89.
3. Крамм Р. Системы управления базами данных dBASE II и dBASE III для персональных компьютеров. М.: Финансы и статистика. 1988. 283 с.
4. Лори П. Базы данных для микроЭВМ. М.: Машиностроение. 1988. 136 с.
5. Database for the BBC Microcomputer Model B. Cambridge: Acornsoft Ltd. 1984. 44 p.

Межвузовский сборник научных трудов
ЗВМ В ОБРАЗОВАНИИ. ПЕДАГОГИЧЕСКИЕ ПРОГРАММНЫЕ СРЕДСТВА
Рига: ЛГУ им. П. Стучки, 1989. С. 129-135

УДК 37.01:007+371.3

Л. А. Андерсоне
Республиканский институт
усовершенствования учителей
Министерства народного
образования Латвийской ССР

А. У. Гринфелдс, У. К. Гринфелдс,
А. Р. Кангро, Ю. Я. Кузьмин,
ЛГУ им. П. Стучки

ПРЕДМЕТ "ОСНОВЫ ИНФОРМАТИКИ И ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ"
В СРЕДНИХ УЧЕБНЫХ ЗАВЕДЕНИЯХ ГУМАНИТАРНОГО ПРОФИЛЯ

В настоящее время все большую популярность приобретает идея гуманитаризации учебного процесса. Одним из способов реализации этой идеи является создание в средних школах классов гуманитарного профиля. В 1988/89 учебном году в Латвийской ССР создано 34 класса гуманитарного профиля. Количество подобных классов продолжает расти. Разрабатываются и проверяются на практике новые экспериментальные учебные программы и планы по всем предметам. Для новых учебных планов характерно увеличение объема часов по гуманитарным предметам за счет естественно-научных предметов и математики.

Как показывает мировая практика, информатизация становится все более важным фактором развития гуманитарной сферы. Это проявляется и в образовании: ЗВМ активно используются при изучении гуманитарных предметов. В СССР по ряду причин ЗВМ применяется лишь при изучении предмета "Основы информатики и вычислительной техники" ("ОИИВТ"). В дальнейшем предполагается внедрение ЗВМ и в другие предметы. На наш взгляд, уже на данном этапе целесообразно переработать курс "ОИИВТ" с учетом специфики гуманитарного профиля.

В новом экспериментальном учебном плане для предмета "ОИИВТ" выделено 34 часа. При этом предусмотрено изучать "ОИИВТ" в 11 классе (2 часа в неделю в течение второй половины учебного года). Все занятия проводятся в кабинетах вычисли-

тельной техники в школах с классами гуманитарного профиля.

Поэтому актуальными являются вопросы разработки соответствующей учебной программы, методических материалов, программного обеспечения ЭВМ.

I. Основы информатики и гуманитарные средние школы

В изучении основ информатики в школах гуманитарного профиля встречаются как бы два полюса - современная технология информационной обработки и гуманитарный образ мышления человека, традиционно противящийся узкому техницизму. Поэтому учитель гуманитарной школы находится в особых условиях и должен непрерывно искать опорные точки информатики в гуманитарных школьных предметах, учитывая особенности восприятия школьников и выбирая соответствующие ему приемы мотивации и пояснения. Одной из таких точек опоры может быть работа с текстами и графической информацией. Важно продемонстрировать возможности информатики не только в школьных предметах, но и в других сферах гуманитарной деятельности человека.

В настоящее время во всем мире наблюдается тенденция увеличения людей занятых в информационной сфере, за счет уменьшения работников материального производства. Быстро развивается промышленность информационных средств. Расширяется перечень работ, для выполнения которых необходимы ЭВМ и методы информатики. Появляются новые средства, меняющие привычные формы деятельности человека, в том числе и в искусстве. Успехи информатики влияют и на мыслительную деятельность через новые, неизвестные ранее элементы. Однако несмотря на то, что мы являемся очевидцами, а иногда и соучастниками названных изменений, чаще всего нам видны лишь близкие утилитарные их цели. Причиной тому - инертность мышления и отсутствие полного обзора и понимания происходящих повсюду информационных процессов. К тому же наша страна пока что отстает от мирового уровня автоматизации и компьютеризации информационных процессов. Тем более важно, чтобы учитель информатики показал роль и значение этих процессов, а также их особенности и тенденции развития. Ученики должны приобрести такое информационное образование, чтобы по окончании школы

они смогли в достаточно широкой зоне соприкасаться с информационной техникой. Другими словами, обучение информатике должно быть поставлено так, чтобы не превратиться в освоение узкой технической дисциплины. Наоборот, оно должно ввести школьника в инфосферу современного общества и хоть немного приоткрыть завесу над тайнами информационных процессов в природе и обществе.

В школьные кабинеты информатики начинает поступать разнообразное программное обеспечение - обучающие, контролируемые и другие программы как по информатике, так и по другим предметам. Учитель информатики должен найти время, чтобы ознакомить других учителей и помочь им внедрить эти программы. Иными словами, кабинеты информатики следует превратить в своеобразные методические центры, координирующие внутри школ компьютеризацию учебного процесса.

II. Экспериментальная учебная программа предмета "ОИИВТ" для гуманитарных средних школ

Информатика - это общеобразовательный предмет, цель которого показать школьникам важнейшие информационные процессы в современном обществе, ознакомить их с возможностями ЭВМ при работе с информацией.

Цели и задачи обучения здесь выражены словами "общеобразовательный предмет". Это значит, что от школьника не требуется достижение уровня профессиональных знаний в области ЭВМ и программирования. Достаточно лишь понимание процессов обработки информации с применением пакетов прикладных программ, текстовых и графических редакторов и других готовых программ. Главная задача учителя сориентировать ученика в мире сложных информационных процессов.

Настоящая учебная программа является экспериментальной и рассчитана на освоение курса "ОИИВТ" в классах гуманитарного профиля средних школ (34 часа за полугодие - 2 часа в неделю). Программа ориентирована на работу в кабинетах, оснащенных комплектами вычислительной техники (например, КУВТ-86). Распределение материала по темам ориентировочное, и учитель вправе менять его в зависимости от имеющегося программного

обеспечения и комплектации вычислительной техники. Но любые изменения не должны снижать уровня знаний ученика. Прежде всего по этой причине нельзя пропускать вопросы, помеченные в программе символом (*). Эти вопросы образуют основу программы. Учитель вправе выбрать реализации общеобразовательных и воспитательных приемов при изложении предмета информатики. Важнейшие положения следует излагать без излишней формализации, но корректно и точно.

1. ВВЕДЕНИЕ (2 часа).

Что такое информатика. Информация, ее получение, преобразование, пересылка, хранение, поиск и выдача. Информационные процессы, роль ЭВМ в их ускорении и упрощении.

Знания: информатика (*); информация (*); информационные процессы (*); автоматизация и механизация процессов.

Умения: включение, запуск, выключение ЭВМ; ведение простого диалога с ЭВМ (*).

2. НАЧАЛЬНЫЕ ПРЕДСТАВЛЕНИЯ ОБ ЭВМ И ЕЕ РОЛИ В ОБЩЕСТВЕ (8 часов).

Важнейшие блоки ЭВМ - процессор, память, клавиатура, экран, магнитофон, диски, принтер - их назначение. Прохождение информации через блоки ЭВМ во время решения задачи. Информация как программа действия ЭВМ, данные. Представление о программном управлении блоками ЭВМ. Применение ЭВМ в школе, науке, технике, экономике, медицине, быту и досуге.

Знания: состав ЭВМ (*); взаимодействие блоков (*); техника безопасности (*); виды информации, доступные ЭВМ (*); программный принцип работы ЭВМ (*); история развития ЭВМ; примеры использования ЭВМ.

Умения: работа с клавиатурой, ввод и работа с простыми программами (*); составление собственных набросов из готовых программ.

3. ЯЗЫКИ, АЛГОРИТМЫ И ПРОГРАММЫ (5 часов).

Понятие алгоритма, как описания сложного действия. Отличие алгоритма от других описаний. Примеры алгоритмов из школьной практики. Переход от описания действия на родном

языке к специальным языкам (блок-схемы и языки команд исполнителя). Алфавит, словарь, синтаксис и семантика машинных языков в сравнении с языками человека. Программа как запись алгоритма на языке ЭВМ. Понятие транслятора как механического переводчика с одного языка на другой.

Знания: понятия алгоритма (*), команды (*) и исполнителя (*); отличие механического исполнителя от человека (*); язык блок-схем (*).

Умения: запись простых алгоритмов на языке блок-схем (*); запись простых программ на BASIC.

4. ПРИМЕНЕНИЕ ЭВМ (19 часов).

Текстовые системы (6 часов): Понятие о текстовых редакторах, анализе текстов, их хранении и форматном выводе. Структуры типичных текстов - бланк, заявление, деловое письмо.

Графические системы (2 часа): Понятие о графическом редакторе. Деловая графика и компьютерные изображения.

Базы данных (4 часа): Организация больших объемов информации. Структура данных. Понятие о записи, дескрипторе и языке поиска информации. Справочные системы. Экспертные системы.

Электронные таблицы (1 час): Табличные структуры, запись данных в электронных таблицах. Типичные задачи, решаемые электронными таблицами.

Пакеты прикладных программ (6 часов): Понятие о пакете программ. Пакет обучающих программ. Пакет программ анализа текстовой информации.

Знания: понятие о текстовом и графическом редакторе (*), базе данных (*), электронной таблице (*), пакете программ (*); основные директивы текстовых и графических редакторов, языка поиска информации в базе данных; примеры применения пакетов программ.

Умения: составление делового письма с помощью текстового редактора; поиск информации в учебной базе данных; расчет с помощью электронной таблицы затраты на покупку заданных предметов; объяснение принципа работы простой обучающей программы.

III. Перспективы внедрения предмета "ОИИВТ" в учебный процесс

Обобщая изложенный материал, основанный на опыте работы в области школьной информатики, литературных источниках [1-12], выводах практических семинаров учителей информатики Латвийской ССР, можно сделать следующие выводы:

1. Учитывая неоднородность выпускаемой вычислительной техники и характер ее поступления в школы, следует руководствоваться региональным принципом внедрения информатики в школы, учитывающим обеспеченность школ ЭВМ, программами и кадрами.

2. Опыт внедрения информатики в школы свидетельствует о необходимости дифференцировать как программы предмета "ОИИВТ", так и преподавание по этим программам. Дифференциация программ должна зависеть от характера школы (общая, гуманитарная, с физико-математическим уклоном и т.п.).

3. Особое внимание следует уделять программному обеспечению школьных ЭВМ, получивших массовое распространение. Необходимо признать, что современное состояние как ЭВМ, так и программ к ним не позволяет не только осуществлять подготовку учителей на высоком уровне, но и обеспечивать высокую эффективность использования школьных ЭВМ.

4. Наряду с программным обеспечением школьных ЭВМ необходимо уделять большое внимание разработке и распространению учебно-методических пособий, соответствующих типу школ, включая практику применения программ и пособий без учета особенностей школ.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Andžāns A., Grīnfelda U., Ikaunieks E. Informātika: Macību palīgvidzeks 10.klasei / U.Grīnfelda red. R.: Zvaigzne, 1985. 88 lpp.
2. Andžāns A., Grīnfelda U., Ikaunieks E. Informātika: Macību palīgvidzeks 11.klasei / U.Grīnfelda red. R.: Zvaigzne, 1986. 86 lpp.

3. Jeršova A., Monahova V., Bešenkovs S. u.c. Informatikas un skaitlošanas tehnikas pamati: 2 daļās / A.Jeršova un V.Monahova red. R.: Zvaigzne. 1987. 1 d. 90 lpp.; 2 d. 134 lpp.
4. Informatikas un skaitlošanas tehnikas pamati. Eksperimentāla mācību programma vispārīgizglītojošo vidusskolu 10. klasei. R.: Zvaigzne. 1989. 16 lpp.
5. Freivalds R. Kur lieto un kur nelieto skaitlotājus. R.: LPSR Zinību biedrība. 1988. 28 lpp.
6. Витиньш М.В., Опмане И.В., Павлов С.И., Фрейвалд Р.В. Изучение курса "Основы информатики и вычислительной техники". Р.: ЛГУ им.П.Стучки. 1988. 97 с.
7. Martuzāns B. Ka Pēcis Beisikāns Maiju Saprātību programmēt mācīja. R.: Zinātne. 1986. 220 lpp.
8. Romanovskis T. Elektroniskais skaitlotājs skolā. R.: Zvaigzne. 1986. 255 lpp.
9. Romanovskis T. Elektroniskie kabaņas skaitlotāji stāstos un spēlēs. R.: Zinātne. 1988. 255 lpp.
10. Кузьмин Ю.Я. Обучающая система "Рига-микро". Р.: Звайгана. 1989. 78 с.
11. Кузнецов А., Долматов В. Методическая система обучения ОИВТ: структура и функции, состояние и перспективы // Информатика и образование. 1989. № 1. С. 3-8.
12. Каямин В. Угринович Н. О преподавании курса ОИВТ по машинному варианту // Информатика и образование. 1989. № 2. С. 17-22.

**ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ
ПРЕДМЕТОВ
ЕСТЕСТВЕННО-НАУЧНОГО ЦИКЛА**

УДК 37.01:007-378.147-510.6-681.3.06

А. Я. Лепин, Л. А. Лепин, Т. Я. Трейкале
Вычислительный центр при ЛГУ им. П. Стучки

КОМПЛЕКС ОБУЧАЮЩИХ ПРОГРАММ "ЛОГИКА"

Внедряемые в учебный процесс компьютерные обучающие программы должны соответствовать определенным дидактическим требованиям [1]. Для программ-тренажеров, нацеленных на формирование типовых умений, характерна организация деятельности учащихся по готовому алгоритму, заранее заданному составителем программы. При выборе методики контроля знаний предпочтение следует отдавать заданиям, не требующим описательного ответа [2], что позволяет проводить тестирование быстрее и объективнее.

Комплекс обучающих программ "Логика" является первым в цикле программ, предназначенных для оказания помощи студентам-первокурсникам математических специальностей в усвоении курса математического анализа. Он может использоваться и старшеклассниками, факультативно изучающими высшую математику. Существуют два варианта программы, ведущих диалог с пользователем на русском и латышском языках.

Комплекс "Логика" включает в себя собственно обучающую часть, которую можно использовать как для первичного ознакомления с материалом, так и для тренировочных упражнений, и контролирующую программу ("Экзамен"). Комплекс состоит из программы загрузки, программы-меню ("Оглавление") и 8 учебных программ (7 обучающих и 1 контролирующая). Общая структура комплекса отражена на рис. 1.

Программы комплекса написаны на языке BASIC и предназначены для работы на ПЭВМ "Yamaha MSX-2". В процессе работы используются также двоичные подпрограммы, служащие как для эстетического оформления программ (создания различных шрифтов,

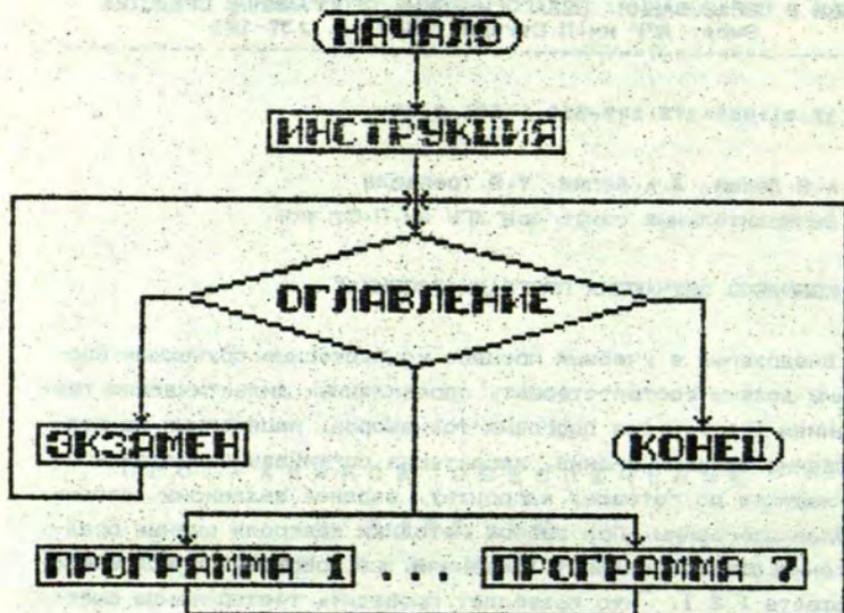


Рис. 1.

спрайтов и т.п.), так и для осуществления управления.

При выборе метода управления диалогом предпочтение было отдано способу, позволяющему максимально упростить эту процедуру, приспособив ее к уровню навыков работы учащегося с клавиатурой ПЭВМ. Управление программой происходит путем выбора пользователем очередного шага: ответа, действия, вызова подпрограммы, — из нескольких перечисленных возможностей, каждой из которых на экране соответствует, т.н. "область выбора". Попадание курсора в одну из областей выбора, о чем свидетельствует изменение его цвета и/или внешнего вида, и последующее нажатие клавиши ПРОБЕЛ указывают, по какой ветви будет передано управление. Если переход из одного состояния программы в другое определен однозначно (например, при чтении теоретического материала, занимающего несколько экранов), то он производится нажатием клавиши ПРОБЕЛ. Движение курсора осуществляется с помощью управляющих клавиш. Не исключается

и использование для этой цели манипулятора "мышь" (его левая кнопка действует аналогично клавише ПРОБЕЛ). "мышь" позволяет более точно управлять движением курсора, что, как показала практика, особенно удобно для пользователей, не имеющих навыка работы на персональных ЭВМ. Таким образом, в работе задействовано всего 5 клавиш, что значительно уменьшает вероятность случайного несправильного нажатия клавиши и, следовательно, вероятность неправильного ответа. Это повышает надежность тестирования и оценки знания.

С порядком управления программой учащийся знакомится посредством инструкции, которая выдается на экран в начале работы комплекса "Логика" (рис. 1). Причем структура инструкции такова, что в процессе чередования кадров с текстовой информацией учащийся не только получает теоретические указания, но и ставится перед необходимостью их немедленной практической реализации. В дальнейшей работе краткая информация-"подсказка" о требуемых действиях пользователя выводится в нижней части экрана.

После ознакомления пользователя с инструкцией на экран выводится основное оглавление. Оно дает информацию об имеющихся учебных программах и возможности завершения работы. Все обучающие программы комплекса работают независимо друг от друга и могут вызываться в любой последовательности.

Каждая обучающая программа состоит из теоретической части и одного или нескольких тестов. Выбор одной из этих частей, как и возврат в основное оглавление, осуществляется по желанию пользователя вне зависимости от предыдущих действий. Например, выбор ветви "Теория" не обязательно должен предшествовать работе с тестами или быть однократным.

При отборе теоретического материала по теме "Логика" авторы опирались на используемую в учебном процессе программу курса математического анализа. При этом в качестве объекта изучения выбирались понятия, факты, правила, широко используемые в различных областях математики. Учебный материал разделялся на логические порции, удобные для одноразового восприятия и подкрепленные соответствующими примерами. В результате комплекс "Логика" включил в себя следующие подпрограммы: "Логические символы", "Приоритет логических операций", "Таб-

лишь истинности", "Тождественные высказывания", "Построение таблиц истинности", "Отрицание" и "Построение отрицания".

Помимо изучения (повторения) теоретического материала в каждой подпрограмме учащиеся имеют возможность проверить и закрепить свои знания на практике, для чего служат соответствующие тесты. В них предлагается ответить на поставленные вопросы (например, назвать логический символ, отметить эквивалентные высказывания), совершить определенные действия (например, построить отрицание логического высказывания). Конкретные задания в тестах формируются с использованием генератора случайных чисел. При этом, как правило, соблюдаются определенные ограничения. В тех случаях, когда эта процедура потребовала бы нерационально больших расходов машинной памяти и времени, задание выбирается из заранее созданного набора случайным образом.

Конструкция всех тестов предполагает выборочный тип ответов (одно- или многоступенчатых). При опросе - с возможностью выбора из двух альтернативных ответов - тотчас по получении ответа оценивается его правильность и, в случае необходимости, приводится правильный ответ. В более сложных тестах алгоритмом предусматривается вывод сообщения-подсказок, корректирующих действия учащихся и помогающих им найти правильный ответ.

Для примера рассмотрим подробнее некоторые из обучающих программ.

Первая из них называется "Логические символы" и состоит из теоретического раздела и одного теста. Как видно из ее названия, программа знакомит пользователя с логической символикой, употребляемой при записи математических высказываний. В тесте ему нужно определить значение следующих друг за другом в произвольном порядке логических символов (один из возможных видов экрана приведен на рис. 2). Для ответа на каждый вопрос дается две попытки. По окончании теста учащемуся сообщается результат - общее количество правильных ответов на 7 заданных вопросов.

Что означает
СИМВОЛ



→ для ЛЮБОГО
существует
НЕ
И
ИЛИ
влечет
ТОЖДЕСТВЕННО

Выбрав с помощью клавиш ↓ ↑ нужный
ответ, нажмите клавишу
ПРОБЕЛ

Рис. 2.

Следующей рассмотрим программу "Таблицы истинности", состоящую из теории и трех тестов, расположенных в порядке возрастания их сложности. В теоретическом разделе дается понятия основных логических операций (отрицание, конъюнкция, дизъюнкция, импликация и эквивалентность) и приводятся таблицы истинности для каждой из них. В первом тесте пользователю предлагается заполнить таблицы истинности для всех пяти операций, следующих одна за другой в произвольном порядке. Во втором тесте случайным образом не только "перемешиваются" логические операции, но и определяются значения операндов. Учащемуся нужно определить истинность логического высказыва-

ния, т.е. воспроизвести таблицы истинности в выборочном порядке. Тест состоит из 6 вопросов и предусматривает помимо текущей оценки правильности ответов сообщение конечного результата. Третий тест, включающий 5 вопросов, похож на предыдущий с той лишь разницей, что абстрактные исходные операнды в нем заменены на конкретные. В их роли выступают математические высказывания, строящиеся с помощью датчика случайных чисел и отражающие отношения сравнения между числами (например, $5 < 2$, $3 > 8$, $6 = 6$ и т.д.): Чтобы выполнить задание - определить истинность высказывания, учащемуся требуется провести предварительную работу, определив истинность операндов. Как и в предыдущем тесте, здесь имеется текущая и окончательная оценка правильности ответов.

Для дальнейшего обучения студентов важным является умение формулировать математические высказывания, противоположные по смыслу исходным. Этому вопросу посвящена программа "Построение отрицания". В ее теоретическом разделе даны общие принципы построения отрицания и правила применения отрицания к отдельным логическим операциям. В тесте требуется построить отрицание логического высказывания, выбранного из набора заданий случайным образом. Задание выполняется учащимся поэтапно, путем выбора очередного, возможного для выполнения действия и замены логического высказывания на эквивалентное. В процессе работы осуществляется постоянный контроль и, в случае необходимости, координация действий учащегося.

Не рассматривая подробнее работу остальных обучающих программ, поясним содержание каждой из них. В программе "Приоритет логических операций" учащийся узнает о порядке выполнения операций в сложном логическом выражении и может закрепить полученные знания на серии однотипных примеров возрастающей сложности. Программа "Тождественные высказывания" описывает один из способов доказательства эквивалентности двух логических выражений - путем построения таблиц истинности для каждого из них. Аналогичным методом устанавливается истинность логического выражения в программе "Построение таблиц истинности". Правила применения отрицания к отдельным логическим операциям приведены в программе "Отрицание".

Подпрограмма "Экзамен" охватывает все темы, рассмотренные в обучающих программах. В настоящем варианте она содержит 20 вопросов: в каждом из которых может быть реализован один из 12 запрограммированных тестов. С целью ускорения тестирования и экономии машинной памяти задания в них сформулированы таким образом, что, предполагая только выборочный (альтернативный) тип ответа, не требуют многоступенчатых действий. Ответы оцениваются по принципу "правильно-неправильно": результат (количество верных ответов на 20 вопросов) сообщается учащемуся по окончании работы. Как показывают расчеты [3], вероятность получения правильного ответа случайным образом не превышает $1/3$. На практике студенты, удовлетворительно изучившие данную тему, дают не менее $2/3$ правильных ответов. Таким образом, может быть разработана довольно точная шкала оценок, позволяющая проверить знания учащихся по теме "Логика".

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Сергеева Т., Чернявская А. Дидактические требования к компьютерным обучающим программам // Информатика и образование. 1988. № 1. С. 48-51.
2. Анастаси А. Психологическое тестирование. М.: Провещение. 1982. Т. 2. 295 с.
3. Габович Ю.Р., Нехода А.А. Система автоматизированного формирования контрольных заданий // Программированное обучение. 1987. Вып. 24. С. 92-95.

УДК 37:007+371.3+378.147+543/547+681.3.06

А. А. Прикулис, С. Я. Такерис

Вычислительный центр при ЛГУ им. П. Стучки

ОБУЧЕНИЕ ПРИКЛАДНОЙ ХИМИИ С ИСПОЛЬЗОВАНИЕМ ЭВМ

Понятие "информационные технологии" означает - технологии на базе ЭВМ, которые позволяют обеспечить различные категории пользователей услугами, связанными с получением и обработкой информации. Базы и банки данных, электронная почта и др. - примеры применения таких технологий. Операции поиска, просмотра информации и обработки данных производятся интерактивно, т.е. в форме диалога между пользователем и ЭВМ [1].

Вышесказанное относится также к обучению химии, особенно ее прикладных вопросов, где приходится иметь дело с большим объемом фактического материала. Например, можно отметить, что ежегодно в нашей стране выпускаются более тысячи различных наименований бытовых химических товаров на общую сумму более 3 млрд. руб. Их классификация, свойства и область применения требуют обширных знания химии как от специалистов, так и от широких слоев населения. Как справиться с этим огромным объемом информации? Очевидно это возможно только с помощью ЭВМ.

Используемые программы можно подразделять на диагностические, тренировочные, обучающие, управляющие, имитационные, моделирующие, измеряющие, справочные и др. Этот перечень не является исчерпывающим и единственно возможным, но для данной области представляется достаточно удобным. Конкретная программа, конечно, может быть отнесена одновременно к двум или нескольким возможным типам. Чисто диагностические про-

граммы в химии не имеют большого самостоятельного значения. Они могут использоваться как промежуточный этап, чтобы определить предварительный уровень знания необходимый для работы с обучающей программой или для допуска к зачету или экзамену. Возможно также применение диагностических программ для определения навыков работы с ЭВМ, особенно в случаях, когда требуется большая скорость работы с клавиатурой. Сравнительно малый интерес представляют также чисто обучающие программы. Они обычно используются для предварительного ознакомления учащихся с данной темой. Например, преподаватель такую программу может использовать во время лекций, демонстрируя на экранах телевизоров (дисплеев) таблицы, графики, определения и т.п. В некоторых программах такое обучение выделено в виде одного из режимов работы. Например, программа "Названия химических элементов" ("Simboli") в обучающем режиме, по запросу пользователя, может выдавать на экран названия элементов на русском, английском, немецком, латинском языках. Вместо французского языка по выбору могут быть использованы латышский, эстонский, литовский, эсперанто, чешский. Одновременно выдается также символ химического элемента, его атомная масса и тип внешней электронной оболочки. В этом режиме не предусмотрена оценка знания, и работа может продолжаться неограниченно долго. Некоторое подобие обучающего режима имеется также в программе "Волнометрическое титрование" [2]. Здесь пользователь может сам выбрать валентность и концентрацию кислоты и концентрацию щелочи. Программа будет рисовать кривую титрования и изображать процесс добавления кислоты к щелочи, при этом в точке эквивалентности меняется окраска индикатора. Чаще используются программы одновременно диагностического и обучающего типа, т.н. обучающие тесты. К таким программам относятся "Химические элементы" [3], "Неорганический анализ" [4], "Органический анализ" [5], "Расшифровка инфракрасных спектров" ("Infra"). Все эти программы содержат определенный объем данных о свойствах или характеристиках веществ. Из массива данных программой выбирается конкретное вещество, и по запросу пользователя выда-

ется информация (полностью или по частям), по которой пользователь должен определить данное вещество. Обучение заключается в том, что в сознании пользователя постепенно накапливаются знания о химических веществах, и с каждым последующим элементом тест решается легче. Однако в отличие от чисто обучающих программ здесь все время контролируется правильность ответов пользователя, а в конце работы подводятся соответствующие итоги, которые выдаются в виде оценки или обобщения. Подобные режимы содержатся также в программах "Simboli" и [2]. Обучающие тесты предполагается создать также по технике безопасности, по анализу катионов или анионов, распознаванию удобрений, пластмасс, вязких веществ, красителей и других химических веществ, применяемых в быту.

Знание некоторых практических вопросов требуется довести до автоматизма. Это касается, например, умения писать и "читать" химические символы, формулы, уравнения реакции. Для этих целей создаются различные тренировочные программы. Так, программа "Уравнения химических реакций" [6] дает возможность довести до автоматизма расстановку коэффициентов уравнений. По желанию обучаемого на экран выдаются уравнения одной (заданной) или различной степени сложности (задается верхний предел сложности). Обучаемому требуется расставить коэффициенты, а программа проверяет их правильность, имитируя действия учителя, т.е. проверяет число атомов каждого элемента в левой и правой частях уравнения и подсказывает, число атомов какого элемента не совпадает. При этом определяется и выводится на экран затраченное время. Опыт показывает, что при помощи этой программы можно достичь скорости около 10 уравнений в минуту. Можно также вводить свои (до 25) уравнения и тренироваться на них (в программе собрано 101 наиболее часто встречающееся в средней школе уравнения). К тренировочным отчасти можно отнести также программы [2] и "Infra".

Моделирующие программы обычно необходимы для того, чтобы лучше разобраться в сути химических процессов. Главным образом здесь имеется в виду построение математической (разумеет-

ся, упрощенной) модели процесса. чтобы можно было, задавая различные исходные параметры, прогнозировать химическое равновесие, выход желаемого продукта, степень превращения исходного продукта, оптимальную продолжительность процесса и т.п. Сюда же относится и получение данных для конструирования необходимой химической установки. Одновременно, используя графические возможности персональных компьютеров, можно моделировать изменения окраски, движение частиц и другие внешние проявления процесса (за исключением запаха и температуры).

Имитационные программы имитируют действия обучаемого с реальными или гипотетическими приборами, установками, моделями. Так, уже отмеченная программа [2] имитирует волюметрическое титрование щелочи кислотой. Обучаемый "открывает" или "закрывает" кран бюретки, и из нее "капает" или "льется" кислота. При этом имитируется звук, а при достижении точки нейтрализации имитируется смена окраски. В отличие от истинной установки в программе имеются некоторые дополнительные возможности: встроенный "калькулятор", повторение титрования с теми же растворами, изображение кривой титрования, доступной только на сложных установках. Программа "Редактор молекул" [7] имитирует построение моделей молекул (стержневых или типа Стюарта-Бриггеба). Эти "модели" можно поворачивать или передвигать, а также измерять межатомные расстояния, валентные и торсионные углы. В отличие от истинных моделей, эти молекулы действительно можно редактировать, т.е. добавлять или удалять атомы, изменять их координаты, сохранять созданные наборы в виде таблиц атомных координат на диске или распечатывать их на бумаге. Нет никаких ограничений на конформации или межатомные расстояния. Модели реальных молекул можно создавать для их дальнейшего использования (квантохимических расчетов, демонстрация на лекционных занятиях и т.п.).

К справочным частично можно отнести уже упомянутую программу "Simboli", в которой имеются справочные данные о химических элементах. Справочные программы можно создавать по

более простому и универсальному принципу: все данные размещать в особьих файлах, которые содержат строчки ("записи") с однотипной информацией. а сама программа только ведет поиск по одному или нескольким признакам (которые можно объединить в виде конъюнкций или дизъюнкций). В этом случае одна и та же программа будет пригодна как для химической, так и любой другой справочной информации, не исключая адресов, телефонов и кухонных рецептов.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Андри Ф. Заочное обучение и коммуникация с помощью ЭВМ // Перспективы. 1989. № 1 (65).
2. Такерис С.Я., Прикулис А.А. Обучающая программа для освоения волнометрического титрования: "TITR" ("Yamaha"; MSX BASIC; русский), "TITRS" ("Yamaha"; MSX BASIC; латышский), "TITROFOK" ("Электроника БК-0010"; ФОКАЛ; русский), "TITRSF" ("Электроника БК-0010"; ФОКАЛ; латышский). Рига: ГосФАП ЛатвССР. 1987. Рег. № 2075895 - 00334, 00335, 00332, 00333.
3. Такерис С.Я., Прикулис А.А., Зекунде А.А., Эглайс М.О. Обучающая программа для освоения свойств химических элементов: "ELEMENTS" ("Yamaha"; MSX BASIC; русский), "ELEMENTI" ("Yamaha"; MSX BASIC; латышский). Рига: ГосФАП ЛатвССР. 1986. Рег. № 2075895 - 00275, 00318.
4. Такерис С.Я., Прикулис А.А., Зекунде А.А. Обучающая программа для освоения основ неорганического качественного анализа: "ANALIZE" ("Yamaha"; MSX BASIC; русский), "NEORVANAL" ("Yamaha"; MSX BASIC; латышский). Рига: ГосФАП ЛатвССР. 1986. Рег. № 2075895 - 00277, 00327.
5. Такерис С.Я., Прикулис А.А. Обучающая программа для освоения основ органического анализа: "ORGANAL" ("Yamaha"; MSX BASIC; русский), "ORVANAL" ("Yamaha"; MSX BASIC; латышский), "ORGAFOK" ("Электроника БК-0010"; ФОКАЛ; русский). Рига: ГосФАП ЛатвССР. 1986. Рег. № 2075895 - 00276, 00328, 00302.

6. Такерис С.Я., Прикулис А.А., Алкснис У.Я. Обучающая программа для освоения химических уравнений: "REAKSIJA" ("Yamaha"; MSX BASIC; русский), "KOEFFICIENTS" ("Yamaha"; MSX BASIC; латышский). Рига: ГосФАП ЛатвССР. 1986. Рег. № 2075895 - 00301, 00315.

7. Такерис С.Я., Прикулис А.А., Вирсис И.Е. Обучающая программа для конструирования молекул и вычисления их параметров: "MOLRED" ("Yamaha"; MSX BASIC; русский). Рига: ГосФАП ЛатвССР. 1987. Рег. № 2075895 - 00425.

УДК 371.302.4+914.7+681.3.06

М. Я. Янкевица

Вычислительный центр при ЛГУ им. П. Стучки

ГЕОГРАФИЯ НА ЭКРАНЕ ПЕРСОНАЛЬНОГО КОМПЬЮТЕРА

Персональные компьютеры удобно использовать при изучении географии в школах и высших учебных заведениях. Графический экран, звук, возможность использовать движущиеся объекты позволяют сделать процесс обучения привлекательным и интересным. По географии студенты и школьники должны запоминать большой фактический материал, учителю в свою очередь приходится тратить много времени для проведения скучной процедуры - проверки знания. Используя персональные компьютеры, этот процесс можно организовать в виде игры. Географическая игра включает элементы обучения, проверку и оценку знания, позволяя обучаемому проводить все действия в индивидуальном режиме.

Современные персональные компьютеры (например, компьютеры типа IBM) снабжены графопостроителями и цветной матричной печатью, что позволяет использовать их для изготовления и размножения различных карт и диаграмм. Обширное системное программное обеспечение (текстовые и графические редакторы, системы управления базами данных, электронные таблицы и т. д.) позволяет пользователям - непрофессионалам использовать ПЭВМ для решения своих проблем даже без написания собственных программ [1]. Существующие системы управления базами данных дают возможность организовать огромные информационные банки, с помощью которых в любой момент можно получить необходимые статистические данные [2]. Географы в своей повседневной работе используют статистическую информацию, которая быстро

меняется. Хранение такой информации в памяти компьютера дает возможность регулярно корректировать ее, что в свою очередь обеспечивает для пользователей доступ к самой новейшей информации, делает ее активной.

В настоящее время видны по крайней мере следующие направления, в которых географы могут применять вычислительную технику [3]:

1) в процессе обучения в школах и высших учебных заведениях;

2) в картографии;

3) для хранения больших баз данных;

4) для моделирования сложных географических проблем.

"Границы компьютерной географии чрезвычайно зыбкие, и ее трудно отделить от работ по формализации географических знаний, развитию теоретической географии, математико-географического моделирования, с одной стороны, и от работ в области информатики, программирования, машинной графики и геометрии и т.п., с другой. На стыке этих направлений и рождается в настоящее время то новое течение в нашей науке, которое называется автоматизацией географических исследований" [3].

Формирование готовности будущих географов к использованию вычислительной техники необходимо начинать уже в школе. Этому способствуют обучающие программы по конкретным темам географии.

1. Характеристика разработанных программ

Рассматривая существующую программу изучения географии Латвии [4] видно, что имеется много тем, для освоения которых можно использовать вычислительную технику: "Реки", "Озера", "Климат", "Транспорт" и т.д. Для апробации выбраны следующие темы: "Административно-территориальное деление" и "Численность и национальный состав населения".

На ПЭВМ "Yamaha" разработаны три программы, которые показывают возможность использования персонального компьютера при обучении географии Латвии. Программа "ГЕО1" показывает на

карте Латвии два населенных пункта и просит обучаемого определить расстояние в километрах между этими пунктами. В верхнем левом углу экрана показан масштаб - отрезок, который соответствует 100 км. Обучаемый вводит число, которое указывает расстояние между населенными пунктами, программа сообщает правильное расстояние и оценивает относительную ошибку ответа. В конце сеанса программа оценивает знания обучаемого (выставляет среднюю оценку).

Программа "ГЕО2" работает в трех режимах :

1) показывает на карте Латвии населенные пункты (автоматический режим);

2) показывает на карте те населенные пункты, которые выбирает обучаемый (режим выбора);

3) географическая игра - программа показывает название населенного пункта, обучаемый должен найти это место на карте. Программа определяет, насколько точно обучаемый находит населенный пункт и показывает его на карте, и в конце сеанса выставляет обучаемому оценку.

Программа "ГЕО3" для каждого населенного пункта показывает общую численность населения, в виде круговой диаграммы - национальный состав населения, а также герб населенного пункта, если такой имеется.

Все программы "знают" 100 населенных пунктов Латвии (их координаты фиксированы в тексте программы). Визуальное оформление программ выполнено манипулятором "мышь" (программа "PAINTER").

Продолжается работа по разработке сценариев и программ по другим темам географии Латвии.

2. Управление работой программ

На уроках и лекциях географии, как правило, присутствуют карта и указка. Обучаемый перемещает указку по карте и ищет необходимые объекты: реки, озера, города и т.д. При использовании в процессе обучения ПЭВМ, карту заменяет экран дисплея, а указку - клавиатура компьютера. Каждому выполняемому

действию соответствует своя клавиша. Для простоты использования программ в правом нижнем углу экрана находится поле подсказок, где перечислены используемые клавиши. Для первоначального знакомства с программами имеется автоматический режим демонстрации возможностей.

Клавиши для управления программами выбраны таким образом, чтобы обозначение клавиши ассоциировалось с выполняемыми действиями: для передвижения указателя по "меню" - клавиши "Стрелка вверх" и "Стрелка вниз", для выбора одного объекта из нескольких - "SELECT", для продолжения работы - "Ввод", для исправления ошибок - "BS", для возврата на "меню" или для выхода из программы - клавиша "ESC" (рис. 1). В некоторых случаях имеется возможность ускорить или замедлить работу программы - для этого используются клавиши "Стрелка вверх" (ускорить) и "Стрелка вниз" (замедлить).

3. Применение программ

Разработанные программы демонстрировались на курсах повышения квалификации учителей информатики в Республиканском институте усовершенствования учителей Министерства народного образования Латвийской ССР. В настоящее время программы используются при изучении курса "Основы геоинформатики" студентами географического факультета Латвийского государственного университета. Имеются упрощенные варианты программ, предназначенные для пересылки по локальной сети КУВТ-2 "Yamaha". Диалог в программах предусмотрен на латышском, русском и английском языках. Они могут быть применены в школах и техникумах для закрепления и углубления знания по курсу географии Латвии, а также в работе школы юных географов.

Для более широкого использования программы могут быть адаптированы для персональных компьютеров, используемых в школах и техникумах. Программы написаны на языке BASIC, поэтому их адаптация на школьных ПЭВМ не потребует много трудозатрат. Следует иметь в виду, что из-за плохой конфигурации школьных ПЭВМ пострадает визуальное оформление, а для обучающихся

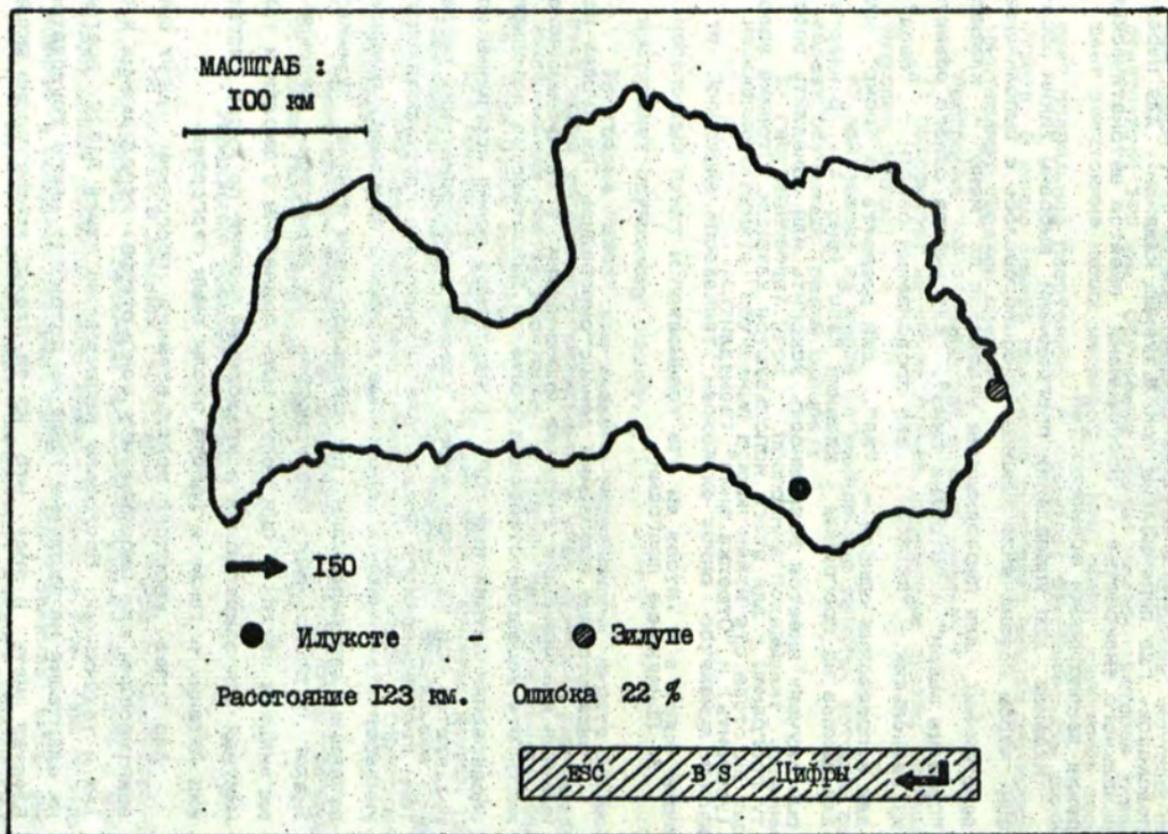


Рис.1. Оформление экрана для программы СВ01 .

программ это существенный фактор.

Применение разработанных программ не отменяет традиционные формы обучения и роль учителя в классе, а является лишь дополнительным средством, обеспечивающим возможность обучаемому работать в индивидуальном темпе, усвоить и закрепить основные умения и навыки и самому оценить уровень освоения темы. Использование вычислительной техники на уроках географии позволяет также сделать процесс обучения более разнообразным и интересным.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Пачиков С.А. Мир персональных компьютеров // В мире персональных компьютеров. 1988. № 1. С. 20-26.
2. Крамм Р. Системы управления базами данных dBASE2 и dBASE3 для персональных компьютеров. М.: Финансы и статистика. 1988. 380 с.
3. Потрянский И.А. Компьютерный арсенал географии. М.: Мысль. 1989. 172 с.
4. Latvijas PSR ģeogrāfijas programma. Rīga: Latvijas PSR Izglītības ministrija. 1987. 4 lpp.

**БАЗОВОЕ И ИНСТРУМЕНТАЛЬНОЕ
ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ**

Межвузовский сборник научных трудов
ЭВМ В ОБРАЗОВАНИИ. ПЕДАГОГИЧЕСКИЕ ПРОГРАММНЫЕ СРЕДСТВА
Рига: ЛГУ им. П. Стучки. 1989. С. 157-170

УДК 681.3.068

В. Г. Лаукс, А. Ф. Раудис

Вычислительный центр при ЛГУ им. П. Стучки

ГРАФИЧЕСКОЕ РАСШИРЕНИЕ ЯЗЫКА ПРОГРАММИРОВАНИЯ
TURBO PASCAL ДЛЯ КУВТ "YAMAHA MSX-2"

Одновременно с внедрением компьютеров в систему образования увеличилась потребность в разработке обучающих программ, с которыми пользователь мог бы работать в интерактивном режиме. Для достижения большего визуального эффекта и улучшения обмена информацией необходимо использование графических возможностей компьютера. Однако не все языки программирования высокого уровня содержат набор графических процедур. Эта проблема существует также для ПЭВМ "Yamaha", графические операторы для которой реализованы, по-видимому, только на рамках языка MSX BASIC, хотя имеются также версии языков Turbo Pascal и C.

Изучение некоторых опытных образцов программной поддержки графического расширения языка Turbo Pascal для ПЭВМ "Yamaha", разработанных в других организациях, привело к выводу о необходимости разработки собственной версии, обеспечивающей необходимую скорость вывода графики и эффективное использование памяти ПЭВМ при создании интерактивных обучающих программ.

Созданный на языке Assembler набор графических процедур для языка Turbo Pascal (далее будем называть этот набор Графическим расширением - ГР) имеет разные дополнительные возможности, например, создавать графические окна, копировать части экрана, создавать наборы выводимых на экран символов.

1. Описание графических процедур

В данной версии ГР в основном сохранены особенности графических операторов языка MSX BASIC [1], часть операторов имеют отличия - даны в более удобной для пользователя форме, например, операторы для определения набора символов.

Существует возможность установки режимов работы, а также проверки наличия ГР:

DetectGraph - возвращает значение TRUE, если ГР инициализировано, или значение FALSE, если ГР отсутствует.

Следует отметить строгие требования к типу параметров в языке Turbo Pascal [2].

1.1. Прямой доступ к видеопамати

Имеются следующие стандартные процедуры доступа к видеопамати:

Vdp(Register, Data) - записывает данные Data в регистр видеопроцессора Register;
Vdpat(Register) - считывает данные из регистра Register состояние видеопроцессора;
Vpoke(Address, Data) - записывает данные Data в видеопамать по адресу Address;
Vpeek(Address) - считывает данные из видеопамати по адресу Address.

1.2. Установка режима экрана

Сохранены стандартные средства включения режима экрана, а также возможность восстановления предыдущего режима.

Screen(ScreenMode) - устанавливает режим экрана;
RestoreScreen - устанавливает предыдущий режим экрана;
Cls - очищает экран.

Возможно отключение вывода информации на дисплей:

Display(Onoff) - активизирует/деактивизирует дисплей.

1.3. Использование цвета.

В более удобной форме предлагается определение новых цветов. Доступны режимы логического переключения цветов в том числе - AND, OR, EOR.

SetColor(FCol, BCol, BRCol)

- устанавливает цвет изображения FCol.

цвет фона BCol, цвет рамки BRCol;

SetRGBPalette(Col, Red, Green, Blue)

- устанавливает цвет Col, в соответствии с заданными значениями интенсивности красного Red, зеленого Green и синего Blue цветов;

DefaultPalette

- устанавливает стандартную палитру цветов;

SetWriteMode(Logic)

- устанавливает режим логики для графических операций: при установке точки на экране считывается предыдущий цвет данной точки, и над ним проводится соответствующая логическая операция.

1.4. Организация графического экрана.

Возможно введение графических окон, относительных координат (отсчитываемых в графическом окне), отсечение частей фигур, выходящих за его пределы. Особенно необходимо отметить процедуру VCopy, позволяющую копировать часть графического экрана.

SetWindow(MinX, MinY, MaxX, MaxY)

- устанавливает графическое окно в соответствии с заданными координатами;

ResetWindow

- устанавливает графическое окно в размере всего экрана;

ClsWindow

- очищает графическое окно;

BorderWindow(Col)

- рисует границы графического окна заданного цвета Col;

Relative(OnOff)

- OnOff=0 - устанавливает начало координат в левый верхний угол экрана;
- OnOff<>0 - устанавливает начало координат в левый верхний угол заданного графического окна;

Clipping(OnOff)

- OnOff=0 - дает возможность вывода графики на весь экран;
- OnOff<>0 - дает возможность вывода графики только в пределах заданного графического окна;

Preset(X, Y)

- устанавливает графический курсор в

- точку с координатами (X,Y):
- PresetRel(dx,dY)** - устанавливает графический курсор в точку с координатами (X+dX,Y+dY), где (X,Y) - предыдущие координаты графического курсора:
- VCopy(X1,Y1,X2,Y2,X3,Y3)** - копирует площадь, ограниченную прямоугольником с диагональю (X1,Y1)-(X2,Y2) на заданное место. Верхним левым углом которого является точка с координатами (X3,Y3).

1.5. Вывод графики

В большей части процедур сохранен синтаксис языка MSX BASIC. В некоторых процедурах уменьшено число параметров (см. Circle) за счет введения новых процедур (см. CircleFill).

- Pset(X,Y,Col)** - выводит на экран точку с координатами (X,Y) и цветом Col:
- PsetRel(dx,dY)** - выводит на экран точку с координатами (X+dX,Y+dY) и цветом Col, где (X,Y) предыдущие координаты графического курсора:
- Point(X,Y)** - возвращает номер цвета точки с координатами (X,Y):
- PointRel(dx,dY)** - возвращает номер цвета точки с координатами (X+dX,Y+dY), где (X,Y) предыдущие координаты графического курсора:
- GetXY(X,Y)** - присваивает координаты графического курсора переменным X и Y:
- Line(X1,Y1,X2,Y2,Col)** - проводит линию из точки (X1,Y1) в точку (X2,Y2) цвета Col:
- LineTo(X2,Y2)** - проводит линию от графического курсора в точку (X2,Y2) ранее заданным цветом изображения:
- LineRel(dx,dY)** - проводит линию от графического курсора в точку (X+dX,Y+dY) ранее заданным цветом:
- Box(X1,Y1,X2,Y2,Col)** - рисует контур прямоугольника с диагональю (X1,Y1)-(X2,Y2) цветом Col:

- BoxFill(X1,Y1,X2,Y2,Col)** - рисует закрашенный прямоугольник с диагональю (X1,Y1)-(X2,Y2) цветом Col.
- Circle(X,Y,R,Col)** - чертит окружность с центром в точке (X,Y), радиусом R и цветом Col.
- CircleTo(R)** - чертит окружность с центром, совпадающим с графическим курсором, радиусом R и цветом Col.
- CircleFill(X,Y,R,Col)** - чертит круг с центром в точке (X,Y), радиусом R и цветом Col.
- Paint(X,Y,Col,BCol)** - закрашивает область, ограниченную цветом BCol, в цвет Col, начиная с точки с координатами (X,Y).

1.6. Спрайты.

В синтаксисе процедур работы со спрайтами сохранены все особенности языка MSX BASIC:

- Sprite(OnOff)** - разрешает/запрещает спрайты;
- SpriteSize(Size1,Size2)** - устанавливает режим спрайтов;
- DefSpritePat(PatNr,Pat)** - формирует шаблон спрайта с номером PatNr; Pat является строкой, которая содержит информацию о шаблоне спрайта.
- DefSpriteCol(Snr,Col)** - формирует цвет спрайта с номером Snr; Col является строкой, которая содержит информацию о цвете спрайта;
- PutSprite(Snr,X,Y,PatNr)** - ставит спрайт с номером Snr и шаблоном PatNr в точку с координатами (X,Y);
- SpriteColl** - возвращает значение TRUE, если есть наложение спрайтов;
- ClsSprite** - очищает все спрайты;
- GetSpriteXY(Snr,X,Y)** - возвращает координаты места расположения спрайта с номером Snr переменным X и Y.

1.7. Вывод текстовой информации.

Процедуры вывода текстовой информации на графический экран разрабатывались с учетом всех возможных вариантов текс-

та и элементов оформления. Предусмотрен также удобный способ формирования шаблонов символов вне зависимости от режима экрана.

- SelectFont(FontNr)** - выбирает один из двух шрифтов букв:
FontNr=0 - стан. артный;
FontNr=1 - шрифт пользователя;
- DefChar(CharNr, Pat)** - формирует шаблон символа с номером CharNr из шрифта пользователя; Pat является строкой, которая содержит информацию о шаблоне символа;
- SetCharSize(HSize, VSize)**
- символы будут выводиться увеличенными в HorSize раз по горизонтали и в VertSize раз по вертикали;
- SetCharInverse(OnOff)**- OnOff=0 - символы будут выводиться в обычном цвете; OnOff<>0 - символы будут выводиться в инверсном цвете;
- SetGap(HorGap, VertGap)**-устанавливает промежуток HorGap (в точках) между выводимыми символами. Если вывод производится по вертикали, указывается VertGap.
- Locate(locX, locY)** - устанавливает графический курсор в точку (X, Y), где
 $X = \text{HorSize} * 8 * \text{locX} + \text{HorGap} * \text{locX}$
 $Y = \text{VertSize} * 8 * \text{locY} + \text{VertGap} * \text{locY}$
- Print(Message)** - печатает строку Message начиная от графического курсора цветом изображения. После выполнения графический курсор устанавливается на нижний правый угол напечатанной строки;
- SetPrintDir(Dir)** - при Dir=0 печать производится по горизонтали, Dir<>0 по вертикали.

2. Распределение памяти

Предложенный подход распределения памяти имеет преимущество в том, что применение процедур ГР практически не влияет на длину программы пользователя, поскольку в рабочей об-

ласти оперативной памяти располагаются только векторы адресов процедур ГР.

Графические процедуры выполняются следующим образом (см. рис. 1):

- 1) графическая процедура вызывается из программы пользователя: управление передается векторам ГР, которые размещены в области системных переменных;
- 2) управление передается подпрограммам в машинных кодах, которые находятся в расширенном ОЗУ;
- 3) из расширенного ОЗУ управление передается в основное;
- 4) осуществляется возврат в программу пользователя.

3. Рабочая конфигурация ГР

Для работы с процедурами ГР языка Turbo Pascal пользователю необходимы следующие два файла:

GRAPEX .COM

GRAPPAS .LIB

Файл GRAPEX.COM содержит графические процедуры и программу установки ГР.

Файл GRAPPAS.LIB содержит названия и описания параметров процедур ГР и адреса вызовов подпрограмм ГР (процедуры являются внешними с помощью ключевого слова external). Пример строки из файла GRAPPAS.LIB:

```
procedure LineRel(dx,dy:integer); external $F9FC;
```

Перед работой пользователь должен запустить файл GRAPEX.COM, который устанавливает ГР в расширенное ОЗУ. После этого можно запускать компилятор TURBO.COM или файлы с расширением .COM, созданные компилятором.

При работе с компилятором сразу после оглавления программы необходимо ставить директиву (\$I GRAPPAS.LIB).

Например: program MYGRAPHICS;

```
($I GRAPPAS.LIB)
```

```
...
```

```
begin
```

```
... (Здесь можно использовать процедуры ГР)
```

```
end.
```

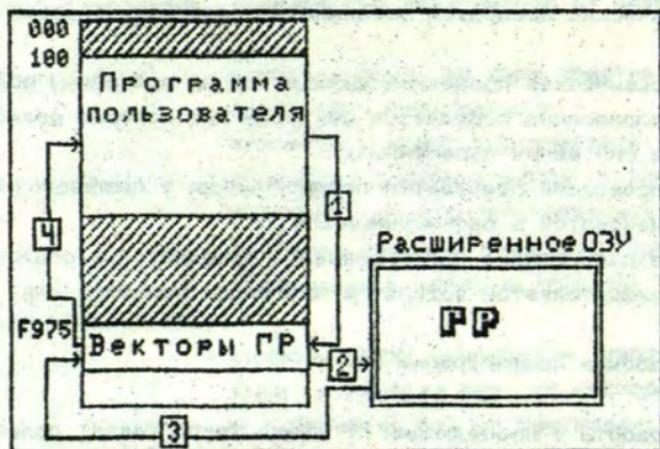


Рис. 1. Распределение памяти и схема выполнения графических процедур.

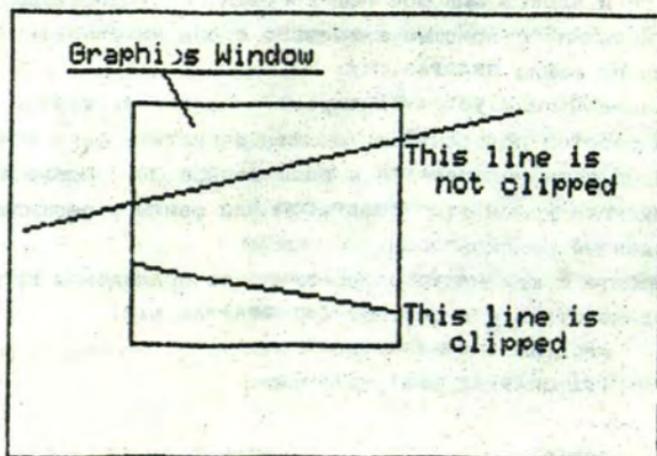


Рис. 2. Графическое окно.

4. Сравнение ГР с возможностями языка MSX BASIC

При разработке данной версии ГР ставилась цель максимально использовать список и синтаксис графических операторов языка MSX BASIC, но между ГР и операторами MSX BASIC имеются различия, которые следует отметить.

В версии ГР для языка Turbo Pascal не реализованы следующие возможности языка MSX BASIC:

- рисование эллипсов и секторов;
- возможность переключения страниц памяти (SET PAGE);
- чтение и запись на диск графической информации.

В версии ГР для языка Turbo Pascal реализованы дополнительные возможности:

- определение графического окна (см. рис. 2);
- есть возможность изменения режима рисования (SET WRITE-MODE);
- расширены возможности вывода символов на графический экран (см. рис. 3).

Графические окна дают возможность работать в системе относительных координат, а также позволяют отсекать части фигур, выходящие за пределы окна. Далее приводится пример определения графического окна (результат выполнения программы показан на рис. 2).

```
program WindowExample;
{$I GRAPPAS.LIB}
begin
  Screen(7);           {Устанавливает графический}
  Locate(5,1);        { режим}
  Print('Graphics Window'); {Выводит текст на экран}
  Line(40,17,170,17,8);
  SetWindow(100,30,300,150); {Устанавливает графическое}
                               { окно}
  BorderWindow(5);    {Рисует границу графического}
                               { окна}
  Clipping(Off);      {Отсечение линий отключено}
  Line(90,17,110,40,8);
  Line(5,90,400,30,6);
  Clipping(On);       {Отсечение линий включено}
```

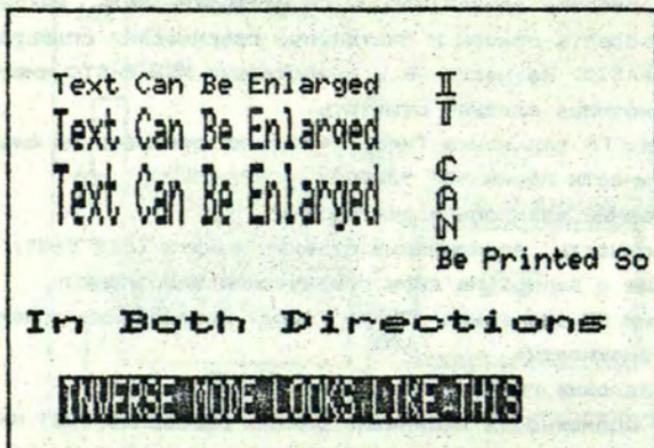


Рис. 3. Примеры вывода текста на графический экран.

Таблица 1. Временны'е характеристики:

Название теста	Кол-во	MSX BASIC		Pascal		Соотноше- ние рав- ств/BASIC
		сек	един/сек	сек	един/сек	
PSET	105544	841	129.06	61	1799.41	13.9
LINE	8192	82	99.90	19	431.15	4.3
CIRCLE	100	27	3.70	11	9.09	2.4

```
Line(5,90,400,150,6);
Preset(305,45);
SetColor(15,1,1);
Print('This line is');
Preset(305,55);
Print(' not clipped');
Preset(305,135);
Print('This line is');
Preset(305,145);
Print(' clipped');
Delay(5000);
Screen(0);
end.
```

Предлагаемые возможности вывода текстовой информации позволяют планировать титульные и информационные кадры обучающих программ, комбинируя буквы, различные по высоте и ширине, буквы в инверсном режиме, а также выводить тексты с изменением направления расположения букв. Далее приводится программа, выводящая на экран в графическом режиме образцы текстов (результат работы программы показан на рис. 3).

```
program TextExample;
{$I GRAPPAS.LIB}
begin
  Screen(7);           {Устанавливает графический режим}
  Locate(10,5);       {Устанавливает курсор}
  Print('Text Can Be Enlarged');
                                     {Выводит текст на экран}
  SetCharSize(1,2);   {Размер символа}
  Locate(10,3);
  Print('Text Can Be Enlarged');
  SetCharSize(1,3);
  Locate(10,3);
  Print('Text Can Be Enlarged');
  SetCharSize(2,1);
  Preset(10,150);
  Print('In Both Directions');
  SetCharSize(1,2);
  SetCharInverse(1);  {Инверсный режим}
```

```
Preset(30,180);      {Устанавливает курсор}
Print('INVERSE MODE LOOKS LIKE THIS');
SelectFont(1);      {Выбирает шрифт пользователя}
SetPrintDir(1);     {Направление печати}
SetCharSize(1,1);
Locate(40,5);
SetCharInverse(0);
Print('IT CAN');
SelectFont(0);
SetPrintDir(0);
Print('Be Printed So');
Delay(1000);
Screen(0);
end.
```

5. Сравнение быстродействия

Все процедуры ПР языка Turbo Pascal выполняются значительно быстрее аналогичных процедур языка MSX BASIC. Результаты проведенных сопоставлений быстродействия показаны в таблице 1. Далее приводятся тексты трех программ соответственно на языках Turbo Pascal и MSX BASIC, использованные для сравнения быстродействия некоторых графических операторов.

ПРОГРАММА "PSET"

```
program pset; {PSET.PAS Ставит 105588 точек}
($I GRAPPAS.LIB)
var i,j : integer;
begin
  Relative(off);ResetWindow;Clipping(off);
  Screen(7);
  for i:=0 to 511 do
    for j:=0 to 211 do Pset(1,j,3);
  Screen(0);
end.

5 'PSET.BAS
10 SCREEN7
20 DEFINT I,J
30 FOR I=0 TO 511:FOR J=0 TO 211:PSET(I,J),3:NEXT J,I
```

40 SCREEN0

Программа "LINE"

```
program line; {LINE.PAS Рисует 8192 линии}
{$I GRAPPAS.LIB}
var i,j : integer;
begin
  Relative(off);ResetWindow;Clipping(off);
  Screen(7);
  for i:=0 to 15 do
    for j:=0 to 511 do Line(j,0,j,211,3);
  Screen(0);
end.

5 'LINE.BAS
10 SCREEN7
20 DEFINTI,J
30 FORI=0TO15;FORJ=0TO511:LINE(J,0)-(J,211),3:NEXTJ,I
40 SCREEN0
```

Программа "CIRCLE"

```
program line; {CIRCLE.PAS Рисует 300 окружностей}
{$I GRAPPAS.LIB}
var i : integer;
begin
  Relative(off);ResetWindow;Clipping(off);
  Screen(5);
  for i:=100 to 200 do Circle(i,106,50,3);
  Screen(0);
end.

5 'CIRCLE.BAS
10 SCREEN5
20 DEFINTI
30 FORI=100TO200:CIRCLE(I,106),50,3:NEXTI
40 SCREEN0
```

6. Совместимость с системами локальной сети

Разработанное ГР может быть использовано совместно с некоторыми системами локальной сети для "Yamaha" КУВТ-2, организуемыми в микроЭВМ учащегося электронным диск. Для этого

служат специально созданные программы GRAPEX и GRAPNET (расширение .COM). Программа GRAPNET резервирует область электронного диска для работы ГР, объем ОЗУ которого уменьшается до 96 К. .

7. Перспективы развития ГР

В настоящее время проводятся следующие разработки:

- ГР для языка программирования высокого уровня - С (версия AZTEC);
- изменение синтаксиса процедур ГР для языков Turbo Pascal и С с целью максимального приближения к синтаксису языков Turbo Pascal (версия 5.0) и Turbo С (версия 2.0). при этом названия процедур ГР дублируются названиями, по синтаксису близкими к языку MSX BASIC (для обеспечения удобств пользователям языка MSX BASIC).

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Справочное руководство по языку программирования MSX Бейсик для комплектов учебной вычислительной техники на базе персональных компьютеров "Ямаха MSX-2". 473 с.
2. Turbo Pascal. Reference Guide. Version 5.0. Borland International. 1988. 493 p.

Межвузовский сборник научных трудов
ЭВМ В ОБРАЗОВАНИИ. ПЕДАГОГИЧЕСКИЕ ПРОГРАММНЫЕ СРЕДСТВА
Рига: ЛГУ им.П.Стучки. 1989. С. 171-174

УДК 681.3.06

А. А. Смилгайс

Вычислительный центр при ЛГУ им.П.Стучки

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ДЛЯ РАБОТЫ
НА ПЭВМ "ИСКРА 1030.11" С ЛАТЫШКИМ АЛФАВИТОМ

Применение персональных ЭВМ широким кругом различных пользователей, в том числе при обучении на латышском языке приводит к необходимости в Латвийской ССР поддерживать трехалфавитную англо-русско-латышскую среду в обработке информации. Для современных ПЭВМ такая возможность предусматривается просто заменой кодовых таблиц. К сожалению, ПЭВМ "Искра 1030.11", также имеющая некоторое распространение, такого средства не имеет. Для установления трехалфавитной среды на ПЭВМ "Искра 1030.11" необходимо либо "перепаять" постоянную память для печати, либо перевести работу ПЭВМ в графический режим и внести необходимые изменения в программы операционной системы. В данной работе рассматривается второй способ.

1. Проблемы при установлении трехалфавитной среды

На ПЭВМ "Искра 1030.11" реализована операционная система АДОС версии 2.0 [1], совместимая с широко распространенной зарубежной операционной системой MS DOS версия 2.0 [2], разработанная для ПЭВМ. В АДОС на ПЭВМ "Искра 1030.11" для установления трехалфавитной среды необходимо провести следующие операции:

1) преобразовать функции базовой системы ввода-вывода (БСВВ) для работы с клавиатурой, чтобы обеспечить достаточно удобный ввод с клавиатуры букв латышского алфавита с диакритическими знаками:

2) изменить дополнительную таблицу изображений знаков в программе GRAFTABL, чтобы в графическом режиме работы на экране отобразить соответствующие буквы латышского алфавита;

3) написать пользовательские программы для печати в трех-алфавитной среде.

Рассмотрим более подробно одну из реализаций перечисленных функций, выполненную в ВЦ при ЛГУ им. П.Стучки.

Перед тем, как приступить к изменениям в программах необходимо разработать кодовую таблицу для представления информации на ПЭВМ и для поддержки трехалфавитной среды. На ПЭВМ "Искра 1030.11" для расширения основной таблицы ASCII используется основной вариант знакогенератора с русским шрифтом. В этой таблице русские буквы размещены в столбцах с кодами 80, 90, D0 и E0, а символы псевдографики в столбцах 80, 90 и A0. Так как здесь имеются отличия от принятой кодировки на IBM PC, то для успешного применения пакетов IBM PC следует внимательно выбрать символы кодовой таблицы, которые заменяются на дополнительные буквы латышского алфавита.

2. Преобразование клавиатурных функций

Для ввода букв латышского алфавита с диакритическими знаками предлагается одновременно нажимать правую клавишу верхнего регистра и соответствующую клавишу латинского алфавита на клавиатуре. Например, для ввода буквы "n" следует нажать правую клавишу верхнего регистра и клавишу "N", а для ввода буквы "A" обе клавиши верхнего регистра и клавишу "A". Этот способ удобен для программирования, так как состояние клавиш верхнего регистра содержится в оперативной памяти в специальном байте состояния клавиатуры. С другой стороны, соответствие букв латышского алфавита с диакритическими знаками клавишам на клавиатуре и относительно редкое появление больших дополнительных латышских букв обеспечивает, по нашему мнению, достаточное удобство также для пользователя.

Преобразование функций РСВВ для работы с клавиатурой реализовано путем замены системных функций клавиатуры, вызываемых по прерыванию 16h, на собственные. Для инициализации собственных функций клавиатуры резидентно вводится в опера-

тивную память, а вектор прерывания 16H настраивается на эти функции. В основе программы клавиатурных функций лежат системные программы, дополненные проверками нажатия правой клавиши верхнего регистра, ситуации CAPS LOCK и наличия сканкода от клавиши для дополнительной буквы латышского алфавита. В случае необходимости код знака заменяется на соответствующий код буквы латышского алфавита.

3. Поддержка трехалфавитной среды для экрана и печати

Для вывода на экран изображений букв латышского алфавита с диакритическими знаками запускается программа, выполняющая АДОС команду GRAFTABL, в которой изменена таблица изображений знаков для графического режима экрана. Необходимо следить за соответствием между значениями кодов дополнительных латышских букв и местонахождением их изображений в таблице программы.

Для печати файла, содержащего буквы трех алфавитов, разработана пользовательская программа на языке BASIC. Эта программа буквы латышского алфавита с диакритическими знаками "рисует" точечным способом, что значительно уменьшает скорость распечатки текстов, в которых такие буквы встречаются часто.

4. Область применения

Разработанные программы по поддержке трехалфавитной среды применимы без дополнительных доработок в программных средствах, не имеющих специальных требований (например, собственных драйверов) к изображению знаков на экране и к вводу с клавиатуры. Такими средствами являются, например, операционная система АДОС [1], строковый редактор EDLIN, текстовый редактор R-1 [3] и макроассемблер МАСМ [4]. К сожалению, большинство программных пакетов требует дополнительной доработки. В настоящее время к работе под управлением АДОС на ПЭВМ "Искра 1030.11" подготовлены программное обеспечение для языка BASIC [5] и оболочка экспертной системы производственного типа.

Кроме того, для языка BASIC разработана подпрограмма печати символьной переменной, содержащей буквы латвийского алфавита с диакритическими знаками.

Примером работы предлагаемого программного обеспечения является настоящая статья.

В ы в о д ы

Предложенное программное средство дает возможность использовать на ПЭВМ "Искра 1030.11" текстовый редактор R-1 в трехалфавитной среде, а также писать литералы и комментарии, создавать файлы на языках программирования МАСМ и BASIC, что является существенным при обучении программированию в школах с латвийским языком обучения.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Программное обеспечение ПП ЭВМ "Искра 1030.11". АДЭС версия 2.00. Руководство программиста. ДШС.00070-01 33 01. 157 с.
2. Брябрин В.М. Программное обеспечение персональных ЭВМ. М.: Наука. 1988. 272 с.
3. Пакет прикладных программ ПП ЭВМ "Искра 1030.11". Редактор текстов R-1. Руководство пользователя. ДШС.00071-01 90 01. 31 с.
4. Программное обеспечение ПП ЭВМ "Искра 1030.11". Макроассемблер МАСМ. Версия 2.00. Руководство программиста. ДШС.00088-01 33 01. 332 с.
5. Программное обеспечение ПП ЭВМ "Искра 1030.11". БЕИ-СИК версия 2.00. Руководство программиста. ДШС.00071-01 33 01. 262 с.

Межвузовский сборник научных трудов
ЭВМ В ОБРАЗОВАНИИ. ПЕДАГОГИЧЕСКИЕ ПРОГРАММНЫЕ СРЕДСТВА
Рига: ЛГУ им. П. Стучки. 1989. С. 175-187

УДК 37.07+519.872.8+681.3.06

Г. Л. Ионин, Я. Я. Канеп, М. Я. Янкевица
Вычислительный центр при ЛГУ им. П. Стучки

ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ СИСТЕМ
С ИСПОЛЬЗОВАНИЕМ ЯЗЫКА SIM/BASIC

В природе, технике и народном хозяйстве функционируют различные системы, в которых события, порядок их поступления и время выполнения действия зависят от случая. В качестве примеров таких систем можно назвать работу скорой помощи, парикмахерской, справочной службы, а также работу таксомоторного парка по обслуживанию пассажиров. К таким системам относятся и многие игры, например, "Цирк" и "Рич-Рач", где ход игры зависит от результата бросания кости.

Функционирование систем может характеризоваться различными величинами, например, среднее время ожидания при обслуживании (пассажиров, покупателей, больных), средняя длина очереди, среднее время игры. Вычисление таких характеристик может представлять сложную математическую проблему.

В простейших случаях решение может быть получено методами теории вероятностей или теории массового обслуживания [1]. В ряде случаев такие характеристики могут быть получены сбором данных в реальных условиях и обработкой измерений методами математической статистики [2], как это и выполняется в лабораторных работах по физике.

Общим способом численного исследования функционирования систем является их имитационное моделирование на вычислительных машинах (метод Монте-Карло) [3]. Реализация метода заключается в построении модели, охватывающей все или основные свойства реальной, проектируемой или гипотетической системы, и составлении программы имитационного моделирования. При запуске программы на ЭВМ имитируется работа модели, со-

бираются и выдаются данные, характеризующие качество ее функционирования. Полученные результаты позволяют анализировать качество функционирования исследуемых систем, что широко используется на стадии проектирования систем в народном хозяйстве.

Для создания имитационных моделей на ЭВМ используются языки имитационного моделирования [4]. Учитывая, что в настоящее время в ряде школ СССР имеются ПЭВМ "Yamaha", мы рассмотрим разработанный в Вычислительном центре при Латвийском госуниверситете язык моделирования SIM/BASIC, программная реализация которого выполнена для вышеуказанной ПЭВМ. Язык SIM/BASIC прост для освоения, так как содержит в основе язык BASIC и дополнен операторами, выполняющими функции имитационного моделирования.

Применение SIM/BASIC в школьной практике позволит приобрести умение формулировать и исследовать задачи, составлять алгоритмы наряду с навыками построения моделей и программирования, используя язык BASIC. Это будет содействовать повышению качества обучения, повысит интерес и активность учащихся при освоении информатики и других учебных предметов.

1. Описание моделей

Описание модели состоит из двух частей: структурной и алгоритмической. В структурной части определяются объекты, характеризующие состояние модели, которые при составлении программы моделирования определяются переменными. Алгоритмическая часть описывает появление и поведение в модели динамических объектов, называемых процессами. Графическое изображение последовательности действия произвольного процесса от момента поступления до ухода из модели называется схемой действий. В произвольный момент в модели может находиться произвольное число процессов, но выполнять действия, может только один (активный) процесс. Каждый процесс может иметь присвоенные ему переменные, число которых задается. Эти переменные будем называть в дальнейшем паспортом процесса.

Основные действия, которые могут совершать процессы, приведены на рис. 1.

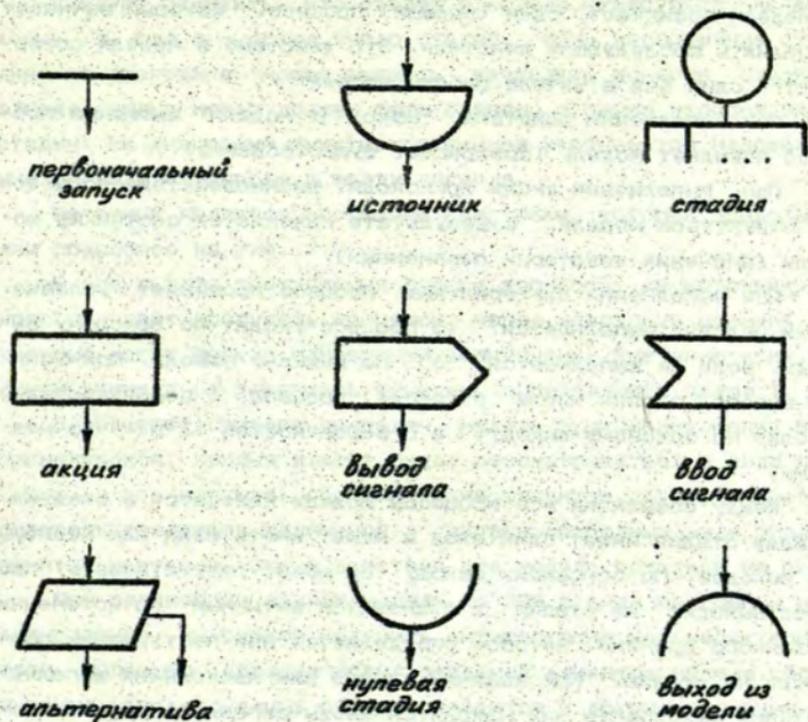


Рис. 1. Основные действия.

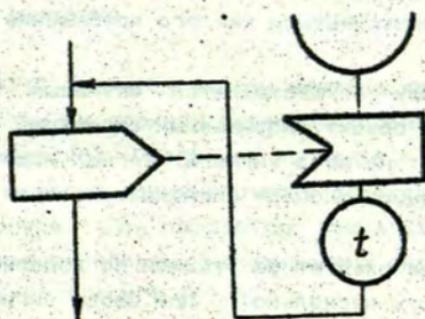


Рис. 2. Источник.

В результате выполнения действия "Первоначальный запуск" в модели возникает один (первый) процесс, который начинает выполнять последующие действия. Это действие в модели совершается один раз в начале работы модели.

При выполнении действия "Выход из модели" активный процесс покидает модель (прекращает существование).

При выполнении акции происходит взаимодействие процесса со структурой модели. в результате изменяется состояние модели (значения некоторых переменных).

При выполнении альтернативы процесс проверяет условие. Если условие выполняется, то процесс уходит по боковому выводу, если не выполняется, то - по нижнему выводу. При записи вместо условия числа p ($0 < p < 1$) процесс с вероятностью p уходит по боковому выводу, а с вероятностью $(1-p)$ - по нижнему.

Между событиями все процессы модели находятся в стадиях. Каждая стадия имеет один вход и может иметь один или несколько выходов. По основному выводу (он может отсутствовать) процесс выходит из стадии и становится активным по истечении заданного времени, которое определяется при поступлении процесса на стадию. При наличии одного или нескольких дополнительных выходов (в них находятся вводы сигналов) процесс может выйти из стадии по сигналу, посылаемому другим процессом (активным), если этот сигнал воспринят вводом и относится к процессу, находящемуся в стадии. Нахождение процесса вне модели может рассматриваться как его пребывание в стадии с нулевым номером.

Вывод сигнала осуществляется активным процессом, при этом образуется объект, называемый сигналом. Сигнал в дальнейшем поступает на ввод сигнала. Сигнал может содержать некоторую информацию (паспорт сигнала), которая воспринимается вводом.

Ввод сигнала следует за стадией по дополнительному выводу (их может быть несколько). При вводе сигнала происходит выход (активизация) процесса из стадии, с которой связано действие ввода сигнала. Причем выходит первый процесс из очереди поступления. Если выход процесса невозможен (в стадии нет процессов), то действие проработает без выхода про-

цесса из стадии, а сигнал в любом случае после действия ввода теряется.

Следует отметить, что стадии в модели нумеруются числами $1, 2, \dots, q$, где q - общее число стадий. Одно изображение стадии на диаграмме может вмещать несколько номеров стадий, однако каждый номер должен быть связан с одним изображением стадии. На диаграмме пунктирной линией указывается направление сигнала от вывода к вводу сигнала.

Источник является составным действием, который представлен подробнее на рис. 2.

В результате поступления одного процесса на источник из него начинают поступать процессы через некоторые времена t , которые могут быть и случайными величинами. Значения этих величин зависят от характера функции, определяющей время t .

В качестве примера приведем модель функционирования парикмахерской. Каждый клиент будет рассматриваться в виде процесса, а все мастера, обслуживающие клиентов, будут характеризовать следующей величиной - числом свободных или числом занятых мастеров. Схема действий для модели показана на рис. 3.

При реализации данной модели на ЭВМ в виде программы моделирования, мы сможем получить интересующие нас характеристики. Например, среднее время ожидания клиентом начала обслуживания, среднюю длину очереди клиентов и другие величины.

2. Язык моделирования SIM/BASIC

2.1. Общие сведения

Язык SIM/BASIC является языком имитационного моделирования простых систем и получен путем расширения языка BASIC с соблюдением синтаксиса. Программа на языке SIM/BASIC состоит из операторов, которые записываются в строках. Каждая строка начинается с номера. При разработке языка SIM/BASIC в качестве первоначального образца использована система имитационного моделирования SPALM [5].

Программа на языке SIM/BASIC (конкретная программа моделирования) описывает не фактическую последовательность действий процессов, а только алгоритм, по которому действует каж-

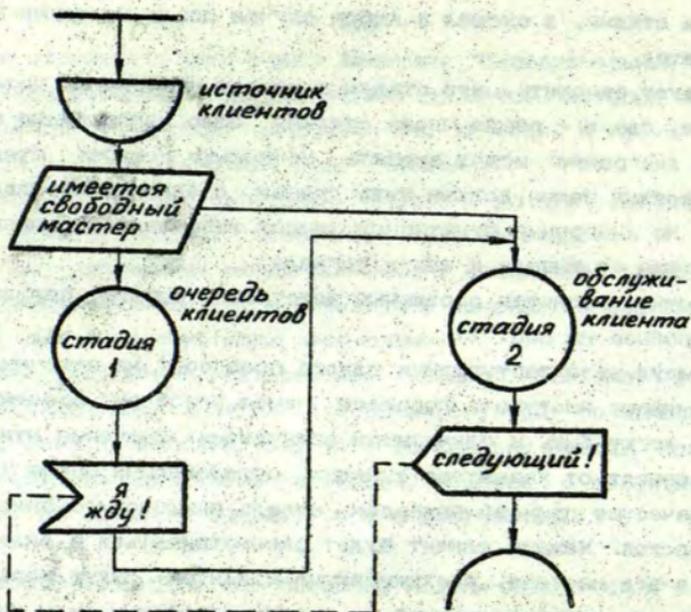


Рис. 3. Схема действия для модели парикмахерской.

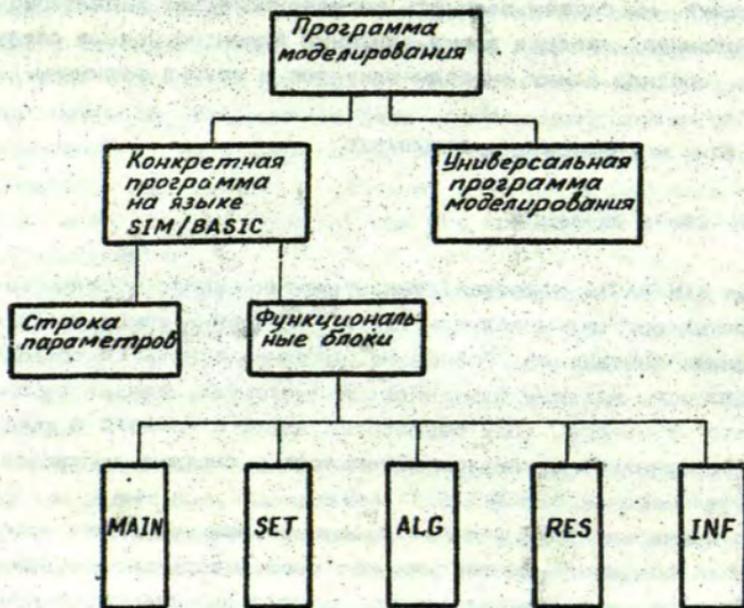


Рис. 4. Структура программы моделирования.

дый отдельный процесс. В системе массового обслуживания одновременно действуют несколько процессов, то же самое должно происходить и в программе, которая имитирует работу этой системы. Поэтому конкретную программу моделирования необходимо дополнить подпрограммами, которые реализуют стандартные части алгоритма моделирования - управление работой нескольких процессов, запуск и останов программы моделирования, переход процессов от пассивных действий к активным, сбор статистической информации. Совокупность этих подпрограмм, написанных на языке BASIC, в дальнейшем будем называть универсальной программой моделирования.

Программа, написанная на языке SIM/BASIC, первоначально обрабатывается специальной программой - предпроцессором языка SIM/BASIC. Предпроцессор добавляет к конкретной программе универсальную программу моделирования, а также заменяет операторы, описывающие специфические для имитационного моделирования действия, обращением к подпрограммам из универсальной программы.

Во время работы универсальная программа взаимодействует с разными частями конкретной программы моделирования. Для облегчения взаимодействия конкретную программу разделяют на блоки. Каждый блок выполняет определенные функции и имеет свое имя, по которому универсальная программа в соответствующие моменты обращается к этому блоку. Структура программы моделирования показана на рис. 4.

Строка параметров предназначена для передачи предпроцессору параметров (максимально допустимое число стадий в модели и процессов, одновременно действующих в модели, наличие или отсутствие паспортов процессов). Если предпроцессору необходимо передавать параметры, конкретная программа должна содержать строку параметров как первую, т.е. с наименьшим номером. В противном случае эта строка отсутствует. Значения принимаются по умолчанию.

Конкретная программа может содержать блоки MAIN, SET, ALG, INF, RES. Каждый блок имеет название, которое задается в первой строке блока. Блок ALG в программе должен присутствовать обязательно, остальные могут отсутствовать. Блоки в программе могут располагаться в произвольном порядке.

Блок MAIN предназначен для управления моделированием. В нем задается число сеансов моделирования, число серий в сеансе, а также условие для окончания серии моделирования. Если программа не содержит блок MAIN, то производится один сеанс моделирования без деления на серии.

Блок SET работает один раз перед каждым сеансом. Он предназначен для присвоения начальных значений переменным, отражающим состояние моделируемой системы.

Блок ALG отражает схему действий моделируемой системы. Он должен обязательно присутствовать в конкретной программе моделирования.

Блок RES предназначен для обработки и вывода результатов моделирования. Обращение к нему происходит после каждой серии.

Блок INF предназначен для сбора информации о состоянии системы. Обращение к нему происходит перед каждым поступлением процесса из источника.

2.2. Операторы моделирования

Операторы языка SIM/BASIC, описывающие характерные для имитационного моделирования действия, также как операторы языка BASIC пишутся в строках с номерами. Одна строка содержит ровно один оператор моделирования. Операторы моделирования начинаются символом "#". Общий вид операторов имитационного моделирования:

<имя оператора> (<список параметров>)

Здесь и далее текст, заключенный в угловые скобки, служит для обозначения некоторого понятия языка SIM/BASIC. При записи оператора в программе этот текст следует заменить конкретными элементами языка. Список параметров заключается в круглые скобки. Он может содержать один или более параметров. Параметры друг от друга отделяются запятыми. Параметры могут быть константами, переменными или выражениями. В некоторых случаях список параметров может быть пустым.

2.2.1. Оператор SIM.

Оператор предназначен для управления сеансом моделирования и применим только в блоке MAIN. Каждое появление этого

оператора в конкретной программе вызывает один сеанс моделирования. Сеанс представляет эксперимент с моделью, описанной в блоке ALG. Сеанс делится на некоторое число серий. Длительность серии определяет значение специального счетчика. Перед началом каждой новой серии фиксируется значение этого счетчика для конца серии. Серия прекращается в тот момент, когда значение счетчика универсальной программы моделирования достигнет значения, зафиксированного для конца серии. Счетчик длины серии может считать число процессов, поступивших на систему, число событий в системе, текущее время модели и т. д. Увеличение счетчика автоматически не происходит, это необходимо запрограммировать в конкретной программе моделирования. Общий вид оператора SIM:

```
# SIM(<параметр1>,<параметр2>)
```

<Параметр1> - число, на которое увеличивается счетчик длины серии за одну серию.

<Параметр2> - число серий в сеансе.

2.2.2. Операторы GEN, GENA, GENE.

Операторы GEN, GENA, GENE предназначены для описания действия "Источник".

Оператор GEN описывает источник простейшего потока. Общий вид оператора:

```
#GEN (<параметр1>)
```

<Параметр1> - интенсивность потока, т.е. среднее число процессов, поступивших за единицу времени.

Оператор GENE описывает также источник простейшего потока. <Параметр1> в этом случае задает не интенсивность потока, а среднее время между поступлениями новых процессов из источника.

Оператор GENA позволяет описать произвольный поток процессов. <Параметр1> задает промежуток времени между смежными процессами.

2.2.3. Операторы ST, STA, STE.

Операторы ST, STA, STE предназначены для описания действия "Стадия". Общий вид оператора ST:

```
#ST(<параметр1>,<параметр2>,<параметр3>)
```

<Параметр1> является обязательным и задает номер стадии. Стадии в модели нумеруются числами 1,2,...,q, где q - общее

число стадий в модели.

<Параметр2> задает продолжительность нахождения процесса в стадии. После истечения этого времени процесс покидает стадию по основному выходу. До истечения этого времени он может покинуть состояние по дополнительному выходу. Оператор ST предусматривает, что процесс находится в стадии случайное время, распределенное по экспоненциальному закону. Среднее значение экспоненциального закона распределения задается в виде $1/\langle\text{параметр2}\rangle$. Значение по умолчанию: $\langle\text{параметр2}\rangle = 1$.

<Параметр3> задается, если имеется стадия с ограничением. Тогда <параметр3> задает путь процесса в случае переполнения объема, заданного для стадии. <Параметр3> указывает либо номер строки в блоке ALG, куда направляется процесс, либо символ "#", - процесс должен покинуть модель.

Оператор STE отличается от оператора ST лишь смысловым содержанием второго параметра. <Параметр2> задает среднее значение времени нахождения в стадии при экспоненциальном законе распределения.

В случае оператора STA <параметр2> задает продолжительность нахождения процесса в стадии.

2.2.4. Оператор QST.

Оператор QST описывает стадию без основного выхода и без ограничения, т.н. "Стадию-ожидание".

Процесс может покинуть стадию только по дополнительному выходу, т.е. по сигналу, присылаемому другим процессом. Общий вид оператора QST:

#QST(<параметр1>)

<Параметр1> является обязательным и задает номер стадии.

2.2.5. Оператор IN.

Оператор IN описывает ввод сигнала. Общий вид оператора:

#IN(<параметр1>, <параметр2>)

<Параметр1> задает имя сигнала. Имя сигнала представляет собой набор символов, отличных от апострофов и кавычек. Длина имени не должна превышать 6 символов.

<Параметр2> задает номер стадии, из которой сигнал выпускает процесс. Номер стадии может принимать значения $0, 1, 2, \dots, q$, где q - общее число стадий в модели. Если номер стадии равен нулю, в модели появляется новый процесс. Новому процессу при-

сваивается паспорт процесса, выводившего сигнал с именем, указанным в операторе IN.

2.2.6. Операторы OUT, EOUT.

Оператор OUT описывает вывод сигнала. Общий вид оператора: #OUT (<параметр1>)

<Параметр1> задает имя выводимого сигнала. Выводимый сигнал после окончания активных действий процесса, выводившего его, посылается к вводу сигнала с таким же именем.

Оператор EOUT описывает вывод экспресс-сигнала. Общий вид оператора:

#EOUT(<параметр1>)

<Параметр1> задает имя выводимого сигнала. В отличие от оператора OUT выводимый сигнал посылается к вводу с таким же именем не после завершения активных действий процесса, выводившего его, а сразу. Приостановленный процесс возобновляет свои действия лишь после того, как активные действия завершил процесс, вышедший из стадии по экспресс-сигналу.

2.2.7. Оператор CR.

Оператор CR предназначен для генерации одного или нескольких процессов. Общий вид оператора:

#CR(<параметр1>, <параметр2>)

<Параметр1> указывает число создаваемых процессов. Значение по умолчанию: <параметр1> = 1.

<Параметр2> указывает номер строки в блоке ALG, на которую направляются все вновь созданные процессы.

2.2.8. Операторы ES, EM, оператор-точка.

Операторы ES, EM и оператор-точка предназначены для прекращения работы серии, сеанса или программы моделирования. Эти операторы не имеют параметров. Общий вид операторов:

#ES #EM #.

Оператор ES прекращает выполнение текущей серии. Программа обрабатывает результаты прерванной серии и продолжает выполнять следующие серии, если такие имеются.

Оператор EM прекращает выполнение текущего сеанса моделирования. Программа обрабатывает результаты прерванного сеанса и начинает выполнять новый сеанс моделирования, если такой предусмотрен в программе.

При встрече оператора-точки программа прекращает работу

окончательно.

2.2.9. Пустой оператор.

Оператор не имеет имени и параметров. Он предназначен для обозначения действия, в результате которого процесс покидает модель. Общий вид оператора: #

2.2.10. Оператор DR.

Оператор DR предназначен для вычисления среднего квадратического отклонения и доверительного интервала случайной величины. Общий вид оператора:

#DR (<параметр1>)

<Параметр1> задает среднее значение случайной величины, для которой необходимо вычислить среднее квадратическое отклонение и доверительный интервал. Вычисляется интервал, соответствующий доверительной вероятности 0.95 при условии, что случайная величина подчиняется закону распределения Стюдента.

З а к л ю ч е н и е

Программное обеспечение языка SIM/BASIC включает редактор ввода, предпроцессор языка SIM/BASIC, универсальную программу моделирования и программу графического изображения действий процессов. Разработанные программные средства предусмотрены для эксплуатации на ПЭВМ "Yamaha" (КУВТ-2) и могут быть использованы как в режиме одиночной машины, так и в сетевом режиме. В 1988/89 учебном году язык SIM/BASIC использовался на кружковых занятиях со школьниками Рижской средней школы № 1. Графическое изображение действий моделирования позволило сделать процесс освоения имитационного моделирования понятным и доступным.

Разработана также демонстрационно-обучающая программа, которая знакомит школьников с основами имитационного моделирования и языка SIM/BASIC, а также на примере модели парикмахерской наглядно показывает все этапы составления и использования имитационной модели.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Гнеденко Б.В., Хинчин А.Я. Элементарное введение в теорию вероятностей. М.: Физматгиз. 1961. 144 с.
2. Касандрова О.Н., Лебедева В.В. Обработка результатов наблюдений. М.: Наука. 1970. 104 с.
3. Соболев И.М. Метод Монте-Карло. М.: Наука. 1978. 64 с.
4. Киндлер Е. Языки моделирования. М.: Энергостроиздат. 1985. 287 с.
5. Ионин Г.Л., Седол Я.Я. Статистическое моделирование систем телетрафика. М.: Радио и связь. 1982. 182 с.

ЗАКЛЮЧЕНИЕ

В сборнике представлены результаты исследований в области применения ЭВМ в образовании. Рассматриваются некоторые научные, методические и практические аспекты. Статьи посвящены вопросам разработки педагогических программных средств (ППС) для общеобразовательных предметов: основы информатики и вычислительной техники (в том числе для наиболее распространенного в школах Латвии КУВТ-86), математика, химия, география. Представлены результаты исследований по перспективным направлениям разработки содержания образования и ППС на основе математического моделирования, с использованием экспертных систем. Описаны разработки программных средств, поддерживающих обучение и подготовку информации на латышском языке, а также расширяющих возможности некоторых версий языков программирования для КУВТ "Yamaha MSX-2".

Результаты исследований могут быть использованы специалистами по разработке программного и методического обеспечения: преподавателями и студентами при изучении основных и специальных курсов; при разработке курсовых и дипломных работ; работниками институтом усовершенствования учителей и центрами учителями при повышении квалификации по информатике и вычислительной технике, а также непосредственно в учебном процессе.

СО Д Е Р Ж А Н И Е

ВВЕДЕНИЕ 3

Образование и компьютер. Общие вопросы

1. Брусиловский П.Л., Буланова Н.Л., Бурмистрова-Зуева И.Н., Горская-Белова Т.Б., Келлин Н.С., Крутов А.Н., Мельникова О.И., Михайлов С.Н., Нодельман А.Я., Нодельман С.Я., Репин И.В., Семашко Г.Л., Сотский А.Н., Хозиев В.В., Ширков П.Д. Международная компьютерная школа: проблемы и перспективы 5

2. Крутов А.Н., Павлов С.И., Цилевич В.Л., Хозиев В.Б., Ширков П.Д. Интегрированный курс "Естествознание": деятельность, моделирование, компьютер 29

3. Головейко А.Г., Севастьяненко В.Г., Сидорик В.В., Смугага Л.Н. Математическое моделирование в курсе физики для школьников и студентов 35

4. Чевере Р.Я. Новые возможности автоматизированного обучения с применением экспертных систем 41

5. Спектор А.Р., Чевере Р.Я. Некоторые возможности применения ПЭВМ при обучении латышскому языку 52

6. Диркова Е.Ю., Фокин М.Л. Критерии экспертной оценки педагогических программных средств 62

Программно-методическое обеспечение обучения информатике

7. Шкерстена И.А. Система Лого для ПЭВМ "Yamaha MSX-2". I. "Черепашья графика" 67

8. Шкерстена И.А. Система Лого для ПЭВМ "Yamaha MSX-2". II. Перспективы развития разработанной системы 82

9. Елкина О.М., Озюш И.Я., Павлов С.И., Перфильева В.В., Цилевич В.Л. Учебный трехалфавитный текстовый редактор для КУВТ-86 102

10. Климанс Л.П., Опмане И.В. Программа "Электронная таблица" в учебном процессе..... 113

11. Витиньш М.В., Озолиня Л.В., Раудис А.Ф. Учебная система управления базой данных для КУВТ-86 124

12. Андерсоне Л.А., Гринфелдс А.У., Гринфелдс У.К., Кангро А.Р., Кузьмин Ю.Я. Предмет "Основы информатики и вычислительной техники" в средних учебных заведениях гуманитарного профиля 129

Программное обеспечение предметов
естественно-научного цикла

13. Лепин А.Я., Лепин Л.А., Трейкале Т.Я. Комплекс обучающих программ "Логика"	137
14. Прикулис А.А., Такерис С.Я. Обучение прикладной химии с использованием ЭВМ	144
15. Янкевица М.Я. География на экране персонального компьютера	150

Базовое и инструментальное программное
обеспечение

16. Лавке В.Г., Раудис А.Ф. Графическое расширение языка программирования TurboPascal для КВБТ "Yamaha MSX-2"	157
17. Смилгайс А.А. Программное обеспечение для работы на ПЭВМ "Искра-1030.11" с латышским алфавитом	171
18. Ионин Г.Л., Канеп Я.Я., Янкевица М.Я. Имитационное моделирование систем с использованием языка SIM/BASIC	175
ЗАКЛЮЧЕНИЕ	188

**ЭВМ В ОБРАЗОВАНИИ
ПЕДАГОГИЧЕСКИЕ ПРОГРАММНЫЕ СРЕДСТВА**

Межвузовский сборник научных трудов

Рецензенты: Л. П. Новицкий, канд. техн. наук доцент
Рижского политехнического института.

Э. А. Икауниекс, канд. физ.-мат. наук доцент
Латвийского государственного
университета им. П. Стучки

Редакторы: Н. Устинов, Р. Павлова
Корректор: К. Краст
Технический редактор: А. Корхонена

Подписано к печати 29 II. 89 ят 18071 Ф/б 60x84/16
Бумага # 1. 13,5 физ. печ. л. 12,6 усл. печ. л. 9,3 уч.-изд. л.
Тираж 500 экз. Зак. # 1411 Цена 1 р. 90 коп.

Латвийский государственный университет им. П. Стучки
226098, Ригв. б. Райниса, 19
Отпечатано в типографии, 226050, Рига, ул. Вейденбаума, 5
Латвийский государственный университет им. П. Стучки

УДК 37.01:007+371.3+519.6+681.3.06

Брусилловский П. Л., Буланова Н. Л., Бурмистрова-Зуева И. Н., Горская-Белова Т. Б., Келлин Н. С., Крутов А. Н., Мельникова О. И., Михайлов С. Н., Нодельман А. Я., Нодельман С. Я., Репин И. В., Семашко Г. Л., Сотский А. Н., Хозиев В. В., Ширков П. Д. МЕЖДУНАРОДНАЯ КОМПЬЮТЕРНАЯ ШКОЛА: ПРОБЛЕМЫ И ПЕРСПЕКТИВЫ // ЭВМ в образовании. Педагогические программные средства. Рига: ЛГУ им. П. Стучки. 1989. С. 5-28.

Описан опыт проведения Международной компьютерной школы (МКШ), основанной на творческой работе школьников 12-16 лет совместно со взрослыми при решении задач моделирования в различных предметных областях. Обсуждаются формы и методы компьютеризации школьного образования.

UDK 37.01:007+371.3+519.6+681.3.06

Brusilovskis P.L., Bulanova N.L., Burmistrova-Zujeva I.N., Gorskaja-Belova T.B., Kellins N.S., Krutovs A.N., Melnikova O.I., Mihailovs S.N., Nodelmans A.J., Nodelmans S.J., Repins I.V., Semashko G.L., Sotskis A.N., Hozijevs V.B., Širkovs P.D. STARPTAUTISKĀ KOMPJŪTERU SKOLA: PROBLĒMAS UN PERSPEKTĪVAS // ЭВМ в образовании. Педагогические программные средства (ESM izglītībā. Pedagoģiskie programmu līdzekļi). Rīga: P.Stučkas LVU. 1989. lpp. 5-28.

Aprakstīta Starptautiskās kompjūteru skolas vadīšanas pieredze, pamatojoties uz skolēnu vecumā no 12 līdz 16 gadiem radošo darbu kopā ar pieaugušajiem, risinot modelēšanas uzdevumus dažādos mācību priekšmetos. Aplūkotas izglītības kompjūterizācijas formas un metodes.

UDC 37.01:007+371.3+519.6+681.3.06

Brusilovskij P.L., Bulanova N.L., Burmistrova-Zueva I.N., Gorskaya-Belova T.B., Kellyn N.S., Krutov A.N., Melnikova O.I., Mihailov S.N., Nodelman A.Y., Nodelman S.Y., Repin I.V., Semashko G.L., Sotsky A.N., Hoziev V.B., Shirkov P.D. INTERNATIONAL COMPUTER SCHOOL: PROBLEMS AND PERSPECTIVES // ЭВМ в образовании. Педагогические программные средства (Computer in Education. Educational Software). Riga: Latvian State University. 1989. Pp. 5-28.

The experience of International Computer School (ICS) is expanded. The ICS was based on creative activities groups of pupils at age 12-16, which solved some problems in different spheres, by using methodology of simulation. The main theme of article is how to put computer into practice of education.

UDK 37.01:007+371.3+378.147+519.6+681.3.06

Крутов А.Н., Павлов С.И., Цилевич В.Л., Хозиев В.В., Широков П.Д. ИНТЕГРИРОВАННЫЙ КУРС "ЕСТЕСТВОЗНАНИЕ": ДЕЯТЕЛЬНОСТЬ, МОДЕЛИРОВАНИЕ, КОМПЬЮТЕР // ЭВМ в образовании. Педагогические программные средства. Рига: ЛГУ им.П.Стучки. 1989. С. 29-34.

Предложен подход к построению интегрированного курса естествознания на основе методологии моделирования. Эффективная учебная деятельность организуется при активном использовании развитых педагогических программных средств для персонального компьютера.

UDK 37.01:007+371.3+378.147+519.6+681.3.06

Krutovs A.N., Pavlovs S.I., Cilavičs V.L., Hozijevs V.V., Širkovs P.D. INTEGRĒTAIS DABAS ZINĀTŅU KURSS: DARBĪBA, MODEĻĒŠANA, SKAITLOTĀJS // ЭВМ в образовании. Педагогические программные средства (ESM izglītībā. Pedagoģiskie programmu līdzekļi). Rīga: P.Stučkas LVU. 1989. Lpp. 29-34.

Tiek piedāvāta pieeja integrētā dabas zinātņu kursa izstrādei, pamatojoties uz modeļēšanas metodoloģiju. Efektīva mācību darbs tiek organizēts, aktīvi izmantojot pedagoģiskos programmlīdzekļus kompjūteram.

UDC 37.01:007+371.3+378.147+519.6+681.3.06

Krutov A.N., Pavlov S.I., Tsilevich V.L., Hoziev V.V., Shirkov P.D. INTEGRATED NATURAL SCIENCES COURSE: ACTIVITY, SIMULATION, COMPUTER // ЭВМ в образовании. Педагогические программные средства (Computer in Education. Educational Software). Riga: Latvian State University. 1989. Pp. 29-34.

The approach to integrated natural sciences course development based on simulation methodology is proposed. The advanced educational software is used to organize effectively active studies.

УДК 37.01:007+53(07.07)+371.3+378.147+519.6+681.3.06

Головейко А.Г., Севастьяненко В.Г., Сидорик В.В., Смур-
га Л.Н. МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ В КУРСЕ ФИЗИКИ ДЛЯ
ШКОЛЬНИКОВ И СТУДЕНТОВ // ЭВМ в образовании. Педагогические
программные средства. Рига: ЛГУ им. П.Стучки. 1989. С. 35-40.

Рассматривается компьютерная технология обучения студен-
тов и школьников на основе математического моделирования.
Библиогр. 3 назв.

УДК 37.01:007+53(07.07)+371.3+378.147+519.6+681.3.06

Goloveiko A.G., Sevast'yanenko V.G., Sidoriks V.V., Smura-
ga L.N. MATEMĀTISKĀ MODEĻŠANA FIZIKAS KURSĀ SKOLENIEM UN
STUDENTIEM // ЭВМ в образовании. Педагогические программные
средства (ESM izglītība. Pedagoģiskie programmu līdzekļi).
Rīga: P.Stučka's LVU. 1989. Lpp. 35-40.

Tiek aprakstīta studentu un skolēnu kompjūteraprācības
tehnoloģija uz matemātiskās modelēšanas bāzes. Bibliogr. 3.

УДК 37.01:007+53(07.07)+371.3+378.147+519.6+681.3.06

Goloveiko A.G., Sevastyanenko V.G., Sidorik V.V., Smura-
go L.N. COMPUTER TECHNOLOGY IN TEACHING STUDENTS AND PUPILS
BASED ON MATHEMATICAL MODELLING // ЭВМ в образовании. Педа-
гогические программные средства (Computer in Education. Edu-
cational Software). Rīga: Latvian State University. 1989.
Pp. 35-40.

Computer technology in teaching students and pupils based
on mathematical modelling is considered. Ref. 3.

УДК 37.01:007

Чевере Р.Я. НОВЫЕ ВОЗМОЖНОСТИ АВТОМАТИЗИРОВАННОГО ОБУЧЕНИЯ С ПРИМЕНЕНИЕМ ЭКСПЕРТНЫХ СИСТЕМ // ЭВМ в образовании. Педагогические программные средства. Рига: ЛГУ им.П.Стучки. 1989. С. 41-51.

Дается краткий исторический обзор развития идеи интеллектуализации автоматизированных систем обучения (АСО). Рассматриваются варианты применения экспертных систем (ЭС) в АСО. Выделяются новые возможности, которые предоставляет применение ЭС в обучении. Ил. 1, библиогр. 7 назв.

UDK 37.01:007

Cevere R.J. EKSPERTSISTĒMU PIELIETOŠANA - JAUNAS IESPĒJAS AUTOMATIZĒTĀJĀ ARMĀCIBĀ // ЭВМ в образовании. Педагогические программные средства (ESM izglītībā. Pedagoģiskie programmu līdzekļi). Rīga: P.Stučkas LVU. 1989. Lpp. 41-51.

Raksts satur īsu apskatu par automatizēto apmācības sistēmu intelektuālizācijas ideju attīstības vēsturi. Aplūkoti iespējamie ekspertu sistēmu pielietojuma varianti, kā automatizētās apmācības procesā. Sevišķa uzmanība veltīta tām jaunajām iespējām, kuras dod ekspertu sistēmu pielietojuma. Il. 1, bibliogr. 7.

UDC 37.01:007

Chevere R.J. USING EXPERT SYSTEMS - THE NEW POSSIBILITIES OF COMPUTER ASSISTED INSTRUCTION // ЭВМ в образовании. Педагогические программные средства (Computer in Education. Educational Software). Riga: Latvian State University. 1989. Pp. 41-51.

Paper continues a short survey of computer assisted instruction development till intelligent tutoring systems. Use of an expert systems within a computer assisted instruction environment is discussed. The emphasis is on the new possibilities given from expert systems. Ill. 1, ref. 7.

UDK 37.01:007+681.3.06

Спектор А.Р., Чевере Р.Я. НЕКОТОРЫЕ ВОЗМОЖНОСТИ ПРИМЕНЕНИЯ ПЭВМ ПРИ ОБУЧЕНИИ ЛАТЫШСКОМУ ЯЗЫКУ // ЭВМ в образовании. Педагогические программные средства. Рига: ЛГУ им. П.Стучки. 1989. С. 52-61.

Рассматриваются основные принципы создания справочника по грамматике латышского языка и проблемно-ориентированного словаря на ПЭВМ с использованием программного обеспечения экспертных систем (ЭС). Проанализированы варианты использования ЭС в обучении латышскому языку. Ил. 1, библиогр. 2 назв.

UDK 37.01:007+681.3.06

Spektors A.R., Čevere R.J. DAŽAS PERSONĀLO ESM IZMANTOŠANAS IESPĒJAS LATVIEŠU VALODAS APMAČĪBAI // ЭВМ в образовании. Педагогические программные средства (ESM izglītība. Pedagoģiskie programmu līdzekļi). Rīga: P.Stučkas LVU. 1989. Lpp. 52-61.

Aprakoti galvenie principi, kā izstrādāt latviešu valodas gramatikas izziņu sistēmu un specializētu terminoloģisko vārdnīcu uz personālajam ESM, izmantojot ekspertsistēmu (ES) programmu nodrošinājumu. Izanalizēti ES pielietošanas varianti latviešu valodas apmācībā. Il. 1, bibliogr. 2.

UDC 37.01:007+681.3.06

Spektors A.R., Čevere R.J. SOME APPLICATIONS OF PERSONAL COMPUTERS IN TEACHING LATVIAN // ЭВМ в образовании. Педагогические программные средства (Computer in Education. Educational Software). Riga: Latvian State University, 1989. Pp. 52-61.

Paper deals with the main principles of development of Latvian grammar 'handbook' and special terminological vocabulary by using expert system (ES) shells on a personal computer. The variants of ES application for teaching Latvian is discussed. Ill. 1, ref. 2.

УДК 37.01:007+681.3.06

Диркова Е.Ю., Фокин М.Л. КРИТЕРИИ ЭКСПЕРТНОЙ ОЦЕНКИ ПЕДАГОГИЧЕСКИХ ПРОГРАММНЫХ СРЕДСТВ // ЭВМ в образовании. Педагогические программные средства. Рига: ЛГУ им. П.Стучки. 1989. С. 62-65.

Рассматривается проблема экспертной оценки разрабатываемых педагогических программных средств (ППС). ППС низкого качества нецелесообразно допускать к использованию в учебно-воспитательном процессе. Для эффективной экспертной оценки по отсеву некачественных ППС предложена система критериев. Библиогр. 2 назв.

UDK 37.01:007+681.3.06

Dirkova J.J., Fokins M.L. PEDAGOGISKO PROGRAMMLIDZEKĻU EKSPERTU NOVĒRTEJUMA KRITĒRIJI // ЭВМ в образовании. Педагогические программные средства (ESM izglītībā. Pedagogiskie programmu līdzekļi). Rīga: P.Stučkas LVU. 1989. Lpp. 62-65.

Aplūkotas izstrādājamo pedagogisko programmlīdzekļu (PPL) ekspertu novērtējuma problēmas. Nav lietderīgi pielaut mācību un audzināšanas procesā zemas kvalitātes PPL izmantošanu. Tiek piedāvāta kritēriju sistēma efektīvam ekspertu novērtējumam nekvalitatīvu PPL atsīšanāsai. Bibliogr. 2.

UDC 37.01:007+681.3.06

Dirkova E.Y., Fokin M.L. CRITERIA OF THE EXPERT EVALUATION OF THE PEDAGOGICAL SOFTWARE // ЭВМ в образовании. Педагогические программные средства (Computer in Education. Educational Software). Riga: Latvian State University. 1989. Pp. 62-65.

The article deals with expert evaluation of pedagogical software. To reject low quality software using in the schools the system of criteria for the expert evaluation is proposed. Ref. 2.

УДК 37.01:007+371.3+681.3.06

Шкерстена И. А. СИСТЕМА ЛОГО ДЛЯ ПЭВМ "YAMAHA MSX-2". I. "ЧЕРЕПАШЬЯ ГРАФИКА" // ЗВМ в образовании. Педагогические программы средства. Рига: ЛГУ им. П. Сту. кн. 1989. С. 67-81.

В статье рассматривается разработанная автором система, реализующая обучающую среду Лого. т.н., "черепашьей графики" с системой команд на русском языке и возможностью перевода основного словаря исполнителя-черепашки на другой язык диалога. Иллюстрируется возможность применения среды "черепашьей графики" при преподавании курса "Основы информатики и вычислительной техники". Приведенные процедуры на языке Лого являются реализацией алгоритмов задач из объединенного учебника под ред. А.П.Ершова. Ил. 4. библиогр. 17 назв.

УДК 37.01:007+371.3+681.3.06

Skerstena I. A. SISTĒMA "LOGO" PERSONĀLAJAM KOMPJŪTERAM "YAMAHA MSX-2". I. "BRUNURUPUŠA GRAFIKA" // ЗВМ в образовании. Педагогические программы средства (ESM izglītībā. Pedagoģiskie programmu līdzekļi). Riga: P. Stučka's LVU. 1989. Lpp. 67-81.

Rakstā aplūkots autora izstrādāta sistēma, kas realizē LOGO arhācības vidi, t.s., "brunurupuša grafikas" vidi. Izpildītāja-brunurupuša komandas ir krievu valodas vārdi. Sistēma atļauj šīs komandas pārtulkot citā dialoga valodā. Ilustrēta "brunurupuša grafikas" izmantošanas iespēja kursa "Informātikas un skaitļošanas tehnikas pamati" pasniegšanā. Tekstā iekļautas LOGO procedūras ir mācību grāmatas (red. A. Jeršovs) uzdevumu algoritmu realizācija. Il. 4. bibliogr. 17.

УДК 37.01:007+371.3+681.3.06

Shkerstena I. A. LOGO FOR "YAMAHA MSX-2". I. "TURTLE GRAPHICS" // ЗВМ в образовании. Педагогические программы средства (Computer in Education. Educational Software). Riga: Latvian State University. 1989. Pp. 67-81.

The article deals with the LOGO-system written by the author. The system realizes the Turtle Graphics environment. The turtle instructions are words in Russian, but the system allows to perform translation of the turtle vocabulary to the any other language. The paper illustrates possibility of using Turtle Graphics in secondary school course "Fundamentals of Informatics and Computing Technique". All LOGO procedures in the text are solutions of exercises from this course. Ill. 4, ref. 17.

УДК 37.01:007+371.3+681.3.06

Шкерстена И.А. СИСТЕМА ЛОГО ДЛЯ ПЭВМ "YAMAHA MSX-2". II. ПЕРСПЕКТИВЫ РАЗВИТИЯ РАЗРАБОТАННОЙ СИСТЕМЫ // ЗВМ в образовании. Педагогические программные средства. Рига: ЛГУ им. П.Стучки. 1989. С. 82-101.

Рассмотрены направления развития ранее разработанной системы Лого, которые включают, во-первых, расширение среды "чепелашей графики" добавлением таких возможностей языка Лого, как переменные, арифметика, команды ввода-вывода информации, во-вторых, создание среды для обработки текстовой информации - списков. Возможности языка иллюстрируются примерами из объединенного учебника по курсу "Основы информатики и вычислительной техники" (ред. А.П.Ершов). Ил. 4. библиогр. 6 назв.

УДК 37.01:007+371.3+681.3.06

Škerstena I.A. SISTEMA "LOGO" PERSONĀLAJAM KOMPJUTERAM "YAMAHA MSX-2". II. IZSTRĀDĀTĀS SISTĒMAS ATTĪSTĪBAS PERSPEKTĪVAS // ZBM в образовании. Педагогические программные средства (ESM izglītība. Pedagoģiskie programmu līdzekļi). Rīga: P.Stučkas LVU. 1989. Lpp. 82-101.

Rakstā aplūkotas autora izstrādātās sistēmas attīstības perspektīvas. Tās ietver, pirmkārt, "brūnurupeša grafikas" vides papildināšanu ar tādām valodas LOGO iespējām, kā mainīgo izmantošana, aritmētisko operāciju izpildīšana, informācijas ievade un izvade. Otrkārt, vides izveidošanu tekstuālās informācijas jeb sarakstu apstrādei. Valodas iespējas ilustrētas ar piemēriem, kas ņemti no mācību grāmatas kursam "Informātikas un skaitļošanas tehnikas pamati" (red. A.Jeršovs). Il. 4, bibliogr. 6.

UDC 37.01:007+371.3+681.3.06

Shkerstena I.A. LOGO FOR "YAMAHA MSX-2". II. DEVELOPMENT PERSPECTIVES OF THE SYSTEM // ZBM в образовании. Педагогические программные средства (Computer in Education. Educational Software). Riga: Latvian State University. 1989. Pp. 82-101.

The paper deals with features of language LOGO which are planning to realise in the existing LOGO-system written by the author. This features are variables, arithmetics, input and output of information and list processing. All features are illustrated by examples from secondary school course "Fundamentals of Informatics and Computing Technique". Ill. 4, ref. 6.

УДК 37.01:007+371.3+378.147+681.3.068

Елкина О.М., Озис И.Я., Павлов С.И., Перфильева Е.В., Шилевич В.Л. УЧЕБНЫЙ ТРЕХАЛФАВИТНЫЙ ТЕКСТОВЫЙ РЕДАКТОР ДЛЯ КУВТ 86 // ЭВМ в образовании. Педагогические программные средства. Рига: ЛГУ им. П.Стучки. 1989. С. 102-112.

Для микроЭВМ "Электроника БК-0010" разработан учебный трехалфавитный текстовый редактор, имеющий режимы редактирования и обмена с диском микроЭВМ "ДБК-2М". С ориентацией на учебный процесс спланирован вывод на экран, сформирована система команд, подобраны комментарии, сообщения и директивы. Текст выводится на печать "D-100" с микроЭВМ "ДБК-2М". Подготовлены образцы текстов. Основные алгоритмы редактирования реализованы на языке BASIC для анализа на уроках информатики. Ил. 4, табл. 1, библиогр. 7 назв.

УДК 37.01:007+371.3+378.147+681.3.068

Jolkins O.M., Ozisa I.J., Pavlovs S.I., Perfiljeva J.V., Šilevičs B.L. MĀCĪBU TRISVALODU TEKSTU REDAKTORS MĀCĪBU SKAITĻOŠANAS TEHNĪKAS KOMPLEKTAM KUVT-86 // ЭВМ в образовании. Педагогические программные средства (ESM izglītībā, Pedagoģiskie programmu līdzekļi). Rīga: P.Stučkas LVU. 1989. Lpp. 102-112.

Trisvalodu mācību tekstu redaktors ir izstrādāts priekš mikroESM "Электроника БК-0010" un paredzēts darbam divos režimos: rediģēšanai un apmaiņai ar mikroESM "ДБК-2М" disku. Paredzot izmantošanu mācību procesā sagatavota teksta izvade uz ekrāna, izveidota komandu sistēma, sagatavoti komentāri, ziņojumi un direktīvas. Tekstu var izdrukāt uz "D-100" no mikroESM "ДБК-2М". Ir sagatavoti tekstu paraugi. Rediģēšanas pamatalgoritmi informatikas stundām realizēti valodā BASIC. Il. 4, tab. 1, bibliogr. 7.

UDC 37.01:007+371.3+378.147+681.3.068

Yolkins O.M., Ozish I.Y., Pavlov S.I., Perfiljeva E.V., Šilevich B.L. THREE-LANGUAGE WORD PROCESSOR FOR EDUCATIONAL PURPOSES FOR SCHOOL COMPUTER COMPLEX KUVT-86 // ЭВМ в образовании. Педагогические программные средства (Computer in Education, Educational Software). Riga: Latvian State University. 1989. Pp. 102-112.

Three alphabets word processor developed for educational purposes for PC "Электроника БК-0010" has two regimes: word processing and textfiles exchange with disk. Textfiles are printed with "D-100" of PC "ДБК-2М". Examples of textfiles are prepared. For analysis purposes some word processing algorithms are realized on BASIC. Ill. 4, tabl. 1, ref. 7.

UDK 37.01:007+371.3+681.3.06

Климанс Л.П., Ормане И.В. ПРОГРАММА "ЭЛЕКТРОННАЯ ТАБЛИЦА" В УЧЕБНОМ ПРОЦЕССЕ // ЭВМ в образовании. Педагогические программные средства. Рига: ЛГУ им.П.Стучки. 1989. С. 113-123.

Дается описание реализации на микроЭВМ "Электроника БК-0010" программы простой электронной таблицы. ее возможности. Рассмотрены некоторые примеры, позволяющие школьнику или студенту приобрести навыки проектирования таблиц. Ил. 5, табл. 2, библиогр. 6 назв.

UDK 37.01:007+371.3+681.3.06

Klimans L.P., Ormane I.V. PROGRAMMAS "ELEKTRONISKĀS TABULAS" IZMANTOŠANA MĀCĪBU PROCESĀ // ЭВМ в образовании. Педагогические программные средства (ESM izglītība. Pedagoģiskie programmu līdzekļi). Rīga: P.Stučkas LVU. 1989. Lpp. 113-123.

Aprakstīta vienkāršas elektroniskās tabulas realizācija uz mikroESM "Электроника БК-0010" un tās iespējas, kā arī daži pielietojumi, kuri ļauj skolniekam vai studentam gūt pieredzi tabulas projektēšanā. Il. 5, tab. 2, bibliogr. 6.

UDC 37.01:007+371.3+681.3.06

Klimans L.P., Ormane I.V. SPREADSHEET PROGRAM IN THE CLASSROOM // ЭВМ в образовании. Педагогические программные средства (Computer in Education, Educational Software). Riga: Latvian State University. 1989. Pp. 113-123.

Simple spreadsheet program for PC "Электроника БК-0010" is described. Some applications, which gives pupils and students experience in making templates and writing formulas are discussed. Ill. 5, tabl. 2, ref. 6.

UDK 37.01:007+371.3+378.147+681.3.06

Витиньш М.В., Озолиня Л.В., Раудис А.Ф. УЧЕБНАЯ СИСТЕМА УПРАВЛЕНИЯ БАЗОЙ ДАННЫХ ДЛЯ КУВТ-86 // ЭВМ в образовании. Педагогические программные средства. Рига: ЛГУ им. П.Стучки. 1989. С. 124-128.

В статье отражены основные положения экспериментальной программы курса "Основы информатики и вычислительной техники" для общеобразовательных школ Латвии. Обсуждаются теоретический материал и практические работы по теме "Базы данных". Описана система управления базой данных, специально разработанная для учебного процесса. Библиогр. 5 назв.

UDK 37.01:007+371.3+378.147+681.3.06

Vitina M.V., Ozoliga L.V., Raudis A.F. MĀCĪBU DATU BĀZES VADĪBAS SISTEMA MĀCĪBU SKAITĻOSANAS TEHNIKAS KOMPLEKTAM KUVT-86 // ЭВМ в образовании. Педагогические программные средства (ESM izglitība, Pedagoģiskie programmu līdzekļi). Rīga: P.Stučkas LVU, 1989. Lpp. 124-128.

Rakstā atspoguļotas kursa "Informatikas un skaitļošanas tehnikas pamati" eksperimentālas programmas Latvijas vispār-izglitojošām skolām pamatatzīnas. Tiek apspriesta tēma "Datubāzes" teorētiskā viela un praktiskie darbi. Apskatīta datubāzes vadības sistēma, kas speciāli izstrādāta mācību procesam. Bibliogr. 5.

UDC 37.01:007+371.3+378.147+681.3.06

Vitins M.V., Ozolina L.V., Raudis A.F. DATA BASE MANAGER SYSTEM FOR EDUCATIONAL PURPOSES FOR SCHOOL COMPUTER COMPLEX KUVT-86 // ЭВМ в образовании. Педагогические программные средства (Computer in Education, Educational Software). Riga: Latvian State University. 1989. Pp. 124-128.

In the article main ideas of experimental schedule for course "Fundamentals of Informatics and Computing Technology" are described. Theoretical material and practical works for theme "Data Base" are considered. Review of Data Base Manager System developed especially for learning process is given. Ref. 5.

УДК 37.01:007+371.3

Андерсоне Л.А., Гринфелде А.У., Гринфелде У.К., Кангро А.Р., Кузьмин Ю.Я. ПРЕДМЕТ "ОСНОВЫ ИНФОРМАТИКИ И ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ В СРЕДНИХ УЧЕБНЫХ ЗАВЕДЕНИЯХ ГУМАНИТАРНОГО ПРОФИЛЯ // ЭВМ в образовании. Педагогические программные средства. Рига: ЛГУ им. П.Стучки. 1989. С. 129-135.

В статье рассмотрена разработанная для классов гуманитарного профиля общеобразовательных средних школ экспериментальная учебная программа "Основы информатики и вычислительной техники" (34 часа) и обсуждаются вопросы обучения информатике в гуманитарной школе. Библиогр. 12 назв.

UDK 37.01:007+371.3

Andersone L.A., Grinfelds A.U., Grinfelds U.K., Kangro A.R., Kuzmins J.J. PRIEKSMETS "INFORMĀTIKAS UN SKAITĻOSANAS TEHNĪKAS PAMATI" HUMANITĀRĀ PROFILĀ VIDĒJĀS MĀCĪBU IESTĀDĒS // ЭВМ в образовании. Педагогические программные средства (ESM izglītībā. Pedagoģiskie programmu līdzekļi). Rīga: P.Stučkas LVU. 1989. Lpp. 129-135.

Rakstā aplūkots eksperimentālā mācību programma priekšmeta "Informātika un skaitļosanas tehnikas pamati" (34 stundas), kura izstrādāta vispārīgizglītojošo vidusskolu humanitārā profila klasēm, kā arī apspriesti informātikas mācīšanas jautājumi humanitārajā skolā. Bibliogr. 12.

UDC 37.01:007+371.3

Andersone L.A., Grinfelds A.U., Grinfelds U.K., Kangro A.R., Kuzmins J.J. THE SUBJECT "FUNDAMENTALS OF INFORMATICS AND COMPUTER TECHNOLOGY" IN THE GENERAL SECONDARY SCHOOLS OF HUMANITARIAN PROFILE // ЭВМ в образовании. Педагогические программные средства (Computer in Education. Educational Software). Riga: Latvian State University. 1989. Pp. 129-135.

In the present item is discussed the experimental study programme in the subject of "Fundamentals of Informatics and computing technology" (34 hours) which is worked out for the humanitarian profile classes of general education secondary schools and are discussed methodical questions of teaching informatics in a humanitarian school. Ref. 12.

UDK 37.01:007+378.147+510.6+681.3.06

Лепин А.Я., Лэпин Л.А., Трейкале Т.Я. КОМПЛЕКС ОБУЧАЮЩИХ ПРОГРАММ "ЛОГИКА" // ЭВМ в образовании. Педагогические программы и средства. Рига: ЛГУ им. П.Стучки. 1989. С. 137-143.

В статье описан разработанный авторами комплекс обучающих программ "Логика", предназначенный для использования студентами-первокурсниками при изучении курса математического анализа. Рассмотрены общие принципы построения обучающих программ и приведены примеры работы некоторых из них. Программы комплекса написаны на языке BASIC и предназначены для работы на ПЭВМ "Yamaha MSX-2". Ил. 2, библиогр. 3 назв.

UDK 37.01:007+378.147+510.6+681.3.06

Lepins A.J., Lepins L.A., Treikale T.J. APMACIBAS PROGRAMMU KOMPLEKSS "LOGIKA" // ЭВМ в образовании. Педагогические программы и средства (ESM izglitiba. Pedagogiskie programmu līdzekļi). Rīga: P.Stučkas LVU. 1989. Lpp. 137-143.

Darba aprakstīts programmu komplekss "Logika", kuru var izmantot pirmā kursa studenti, apgūstot matemātisko analīzi. Aplūkoti vispārīgie aprācību programmu izveides principi. Doti dažu programmu darbības piemēri. Programmu komplekss uzrakstīts valodā BASIC un paredzēts darbam ar personālo ESM "Yamaha MSX-2". Il. 2, bibliogr. 3.

UDC 37.01:007+378.147+510.6+681.3.06

Lepin A.J., Lepin L.A., Treikale T.J. COMPLEX OF TRAINING PROGRAMS "LOGIC" // ЭВМ в образовании. Педагогические программы и средства (Computer in Education. Educational Software). Riga: Latvian State University. 1989. Pp. 137-143.

The article deals with the complex of training programs named "Logic", which is designed for first year students when studying the course of mathematical analysis. Common principles of developing of such programs are given and some examples are considered. Programs are written on MSX-BASIC language and are intended for PC "Yamaha MSX-2". Ill. 2, ref. 3.

УДК 37:007+371.3+378.147+543/547-681.3.06

Прикулис А.А., Такерис С.Я. ОБУЧЕНИЕ ПРИКЛАДНОЙ ХИМИИ С ИСПОЛЬЗОВАНИЕМ ЭВМ // ЭВМ в образовании. Педагогические программные средства. Рига: ЛГУ им.П.Стучки. 1989. С. 144-149.

Приводятся данные о программах для обучения свойствам и названиям химических элементов, неорганических и органических веществ, уравнениям химических реакций, строению молекул, волюметрическому титрованию, инфракрасным спектрам. Обсуждаются принципы классификации педагогических программных средств. Библиогр. 7 назв.

УДК 37:007+371.3+378.147+543/547-681.3.06

Prikuļis A.A., Takeris S.J. ESM IZMANTOSANA LIETISKĀS KĪMIJAS ARMĀSĪBĀ // ЭВМ в образовании. Педагогические программные средства (ESM izglītībā. Pedagoģiskie programmu līdzekļi). Rīga: P.Stučkas LVU. 1989. Lpp. 144-149.

Aplūkotas programmas ķīmisko elementu, neorganisko un organisko vielu īpašību un nosaukumu, ķīmisko reakciju vienādojumu, molekulu uzbūves, infrasarkano spektru apgūšanai. Tiek aplūkoti pedagoģisko programmlīdzekļu klasifikācijas principi. Bibliogr. 7.

UDC 37:007+371.3+378.147+543/547-681.3.06

Prikuļis A.A., Takeris S.J. APPLIED CHEMISTRY TEACHING USING PERSONAL COMPUTER // ЭВМ в образовании. Педагогические программные средства (Computer in Education. Educational Software). Riga: Latvian State University. 1989. Pp. 144-149.

Computer programs for teaching properties and names of chemical elements, inorganic and organic substances, equations of chemical reactions, molecular structure and infrared spectra are dealt with. Principles of classification of didactic software are discussed. Ref. 7.

УДК 371.302.4+914.7+681.3.06

Янкевица М.Я. ГЕОГРАФИЯ НА ЭКРАНЕ ПЕРСОНАЛЬНОГО КОМПЬЮТЕРА // ЭВМ в образовании. Педагогические программные средства. Рига: ЛГУ им.П.Стучки. 1989. С. 150-155.

Рассматриваются обучающие программы по темам "Административно-территориальное деление Латвии" и "Численность и национальный состав населения Латвии". Обсуждаются также некоторые общие проблемы применения персональных компьютеров при изучении географии Латвии. Ил. 1, библиогр. 4 назв.

UDK 371.302.4+914.7+681.3.06

Jankevica M.J. GEOGRĀFIJA UZ PERSONĀLĀ KOMPJŪTERA EKRĀNA // ЭВМ в образовании. Педагогические программные средства (ESM izglītībā. Pedagoģiskie programmu līdzekļi). Rīga: P.Stučkas LVU. 1989. Lpp. 150-155.

Aprakstītas apmācības programmas sekojošu Latvijas ģeogrāfijas tēmu apgūšanai: "Administratīvi-teritoriālais iedalījums" un "Iedzīvotāju skaits un nacionālais sastāvs". Apskatītas arī vispārīgas problēmas, kas saistītas ar personālo skaitļotāju pielietošanu ģeogrāfijas apmācībā. Il. 1, bibliogr. 4.

UDC 371.302.4+914.7+681.3.06

Jankevica M.J. GEOGRAPHY ON THE PERSONAL COMPUTER SCREEN // ЭВМ в образовании. Педагогические программные средства (Computer in Education. Educational Software). Riga: Latvian State University. 1989. Pp. 150-155.

In this publication are described computer assisted learning programs for teaching following themes of Latvia's geography: "Administrative-territorial division" and "Size of the population and nacional structure". There are also described general problems of using personal computers in geography studies. Ill. 1, ref. 4.

УДК 681.3.068

Лаукс В.Г., Раудис А.Ф. ГРАФИЧЕСКОЕ РАСШИРЕНИЕ ЯЗЫКА ПРОГРАММИРОВАНИЯ TURBOPASCAL ДЛЯ КУВТ "YAMAHA MSX-2" // ЭВМ в образовании. Педагогические программные средства. Рига: ЛГУ им. П.Стучки. 1989. С. 157-170.

Дается описание графического расширения языка TurboPascal, предусмотренного для разработки обучающих моделирующих программ. Приведено сравнение разработки с графическими возможностями языка MSX BASIC. Указано на развитие версий графического расширения. Ил. 3, табл. 1, библиогр. 2.

UDK 681.3.068

Lauks V.G., Raudis A.F. PROGRAMMESANAS VALODAS TURBOPASCAL GRAFISKAIS PAPLĀSINĀJUMS PESH "YAMAHA MSX-2" // ЭВМ в образовании. Педагогические программные средства (ESM izglitiba. Pedagoģiskie programmu līdzekļi). Riga: P.Stučkas LVU. 1989. Lpp. 157-170.

Aprakstīts valodas TurboPascal grafiskais paplašinājums, kurš paredzēts modelējošo aprācības programmu izstrādei. Dots salīdzinājums ar valodas MSX BASIC grafiskajām iespējām. Atrīmētas plānojamās grafiskā paplašinājuma versijas. Il. 3, tab. 1, bibliogr. 2.

UDC 681.3.068

Lauks V.G., Raudis A.F. GRAPHICAL EXTENTION OF PROGRAMMING LANGUAGE TURBOPASCAL ON MICROCOMPUTER "YAMAHA MSX-2" // ЭВМ в образовании. Педагогические программные средства (Computer in Education. Educational Software). Riga: Latvian State University. 1989. Pp. 157-170.

Description of Graphical Extention of language TurboPascal is given. Product is oriented on educational simulation software development. Grahical Extention is compared with graphical facilities of MSX BASIC. Ideas for future versions of Graphical Extension are reviewed. Ill. 3, tabl. 1, ref. 2.

УДК 681.3.06

Смилгајс А.А. ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ДЛЯ РАБОТЫ НА ПЭВМ "ИСКРА 1030.11" С ЛАТЫШСКИМ АЛФАВИТОМ // ЭВМ в образовании. Педагогические программные средства. Рига: ЛГУ им. П.Стучки. 1989. С. 171-174.

Предлагается программное средство для поддержки трех-алфавитной среды на ПЭВМ "Искра 1030.11" под управлением операционной системы АДОС. Обеспечивается возможность подключения латышского алфавита для языков программирования МАСМ, BASIC, для текстового редактора R-1 и др. Средство реализовано путем изменения или замены некоторых функций базовой системы ввода-вывода. Библиогр. 5 назв.

УДК 681.3.06

Smilgājs A.A. PROGRAMMU LIDZEKLI LATVIEŠU VALODAS PIESĒGŠANAI UZ PĒSM "ISKRA 1030.11" // ЭВМ в образовании. Педагогические программные средства (ESM izglītība. Pedagoģiskie programmu līdzekļi). Rīga: P.Stučkas LVU. 1989. Lpp. 171-174.

Tiek piedāvāti programmu līdzekļi trīsvalodu vides uzturēšanai uz PĒSM "Iskra 1030.11" operāciju sistēmā АДОС. Nodrošināta latviešu alfabēta pieslēgšana programmēšanas valodām МАСМ, BASIC, teksta redaktoram R-1 u.c. Līdzekļi izstrādāti, izdarot izmaiņas un papildinājumus dažās ievada un izvada pamatsistēmas funkcijās. Bibliogr. 5.

UDC 681.3.06

Smilgājs A.A. THE SOFTWARE TO CONNECT THE LATVIAN LANGUAGE TO PC "ISKRA 1030.11" // ЭВМ в образовании. Педагогические программные средства (Computer in Education. Educational Software). Riga: Latvian State University. 1989. Pp. 171-174.

There are offered the software for supporting three-alphabetical environment on PC "Iskra 1030.11" under the control of operational system АДОС. There are provided the connection of Latvian alphabet to programming languages МАСМ, BASIC, text editor R-1 e.t.c. The software is elaborated by making changes and additions to some BIOS functions. Ref. 5.

УДК 37.07+519.872.8+681.3.06

Ионин Г.Л., Канеп Я.Я., Янкевица М.Я. ИМИТАЦИОННОЕ МОДЕЛИРОВАНИЕ СИСТЕМ С ИСПОЛЬЗОВАНИЕМ ЯЗЫКА SIM/BASIC // ЭВМ в образовании. Педагогические программные средства. Рига: ЛГУ им. П.Стучки. 1989. С. 175-187.

Рассмотрен язык моделирования SIM/BASIC, предназначенный для моделирования на ПЭВМ "Yamaha" простых систем массового обслуживания. SIM/BASIC является расширением языка MSX BASIC посредством введения операторов, позволяющих описать действия, характерные для имитационного моделирования. Использование SIM/BASIC дает возможность решать нетрадиционные задачи, связанные с учетом случайных явлений. Ил. 4, библиогр. 5 назв.

УДК 37.07+519.872.8+681.3.06

Jonins G.L., Kaneps J.J., Jankevica M.J. SISTĒMU STATISTISKĀ MODEĻĒŠANA, IZMANTOJOT VALODU SIM/BASIC // ЭВМ в образовании. Педагогические программные средства (ESM izglītība. Pedagoģiskie programmu līdzekļi). Riga: P.Stučkas LVU. 1989. Lpp. 175-187.

Aplūkota statistiskās modeļēšanas valoda SIM/BASIC, kas paredzēta vienkāršu masu apkalpošanas sistēmu modeļēšanai. Valoda SIM/BASIC izveidota, papildinot valodu MSX BASIC ar speciāliem modeļēšanas operatoriem. SIM/BASIC dod iespēju risināt ar gadījuma lielumiem saistītus netradicionālus uzdevumus. Il. 4, bibliogr. 5.

УДК 37.07+519.872.8+681.3.06

Jonins G.L., Kaneps J.J., Jankevica M.J. THE SIMULATION SYSTEM WITH LANGUAGE SIM/BASIC // ЭВМ в образовании. Педагогические программные средства (Computer in Education. Educational Software). Riga: Latvian State University. 1989. Pp. 175-187.

The simulation language SIM/BASIC is considered for simple public utility system simulation. SIM/BASIC language is developed by supplementing MSX BASIC language with special simulation operators. SIM/BASIC gives an opportunity to solve untraditional problems concerning random quantities. Ill. 4, ref. 5.

СЕМИНАР "ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ОБРАЗОВАНИИ"

Лаборатория проблем школьной и вузовской информатики Вычислительного центра при ЛГУ им. П. Стучки организует ежемесячный семинар "Информационные технологии в образовании".

Основные направления работы семинара:

- роль и место информационных технологий в образовании;
- новейшие идеи в содержании и методике преподавания информатики и других предметов;
- концепции авторских учебных пособий по основам информатики и вычислительной техники, а также другим предметам естественно-научного цикла;
- новые педагогические программные средства для комплектов учебной вычислительной техники и персональных ЭВМ;
- передовой педагогический опыт.

На семинар приглашаются учителя, методисты, педагоги, психологи, организаторы народного образования, создатели программного обеспечения и другие специалисты по информатизации образования.

В качестве референтов выступают ведущие специалисты республики и Союза. Наиболее интересные выступления будут публиковаться в виде статей в межвузовском сборнике научных трудов "ЭВМ в образовании". Иногородние референты обеспечиваются гостиницей.

Семинар проходит во вторник перед последним четвергом каждого месяца. Начало семинара в 16.00.

Семинар проводится на базе лаборатории проблем школьной и вузовской информатики ВЦ при ЛГУ им. П. Стучки по адресу: г. Рига, бульвар Падомью, 5.

Более подробную информацию о семинаре можно получить у секретаря семинара Калейса Арниса Яновича, телефон: 22-37-27 (г. Рига, код - 013-2).

SEMINĀRS "INFORMATIVĀS TEHNOLOĢIJAS IZGLĪTĪBĀ"

LVU Skaitļošanas centra Skolu un augstskolu informātikas problēmu laboratorija organizē ikmēneša semināru "Informatīvās tehnoloģijas izglītībā".

Semināra darbības galvenie virzieni:

- informatīvo tehnoloģiju loma un vieta izglītībā;
- jaunākās idejas par informātikas un citu mācību priekšmetu saturu un pasniegšanas metodiku;
- informātikas un skaitļošanas tehnikas pamatu, kā arī citu dabas zinātņu cikla priekšmetu mācību grāmatu koncepcijas;
- jauni pedagoģiskie programmu līdzekļi skaitļošanas tehnikas mācību komplektiem un personālajiem skaitļotājiem;
- pirmrindas pedagoģiskā pieredze.

Uz semināru tiek aicināti skolotāji, metodisti, pedagogi, psihologi, tautas izglītības organizatori, programnodrošinājuma izstrādātāji un citi izglītības informatizācijas speciālisti.

Seminārā referēs vadošie republikas un Savienības speciālisti. Interesantākie referāti tiks publicēti starppaugstskolu zinātnisko darbu krājumā "ESM izglītībā". Referenti no citām pilsetām tiek nodrošināti ar viesnīcu.

Seminārs notiks otrdienās pirms mēneša pēdējās ceturtdienas. Semināra sākums plkst. 16.00.

Seminārs notiks LVU SC Skolu un augstskolu informātikas problēmu laboratorijas telpās: Rīgā, Padomju bulvāri 5.

Sīkāku informāciju var saņemt no semināra sekretāra Arņa Kalēja, telefons: 22-37-27.

АВТОРАМ СБОРНИКА "ЭВМ В ОБРАЗОВАНИИ"

1. Межвузовский сборник научных трудов "ЭВМ в образовании" (ответственный редактор - Н.Н. Устинов) готовится в лаборатории проблем школьной и вузовской информатики Вычислительного центра при ЛГУ им. П. Стучки.

2. В сборнике публикуются статьи, как правило, после доклада на семинаре лаборатории по следующим направлениям:

- Образование и компьютер. Новые информационные технологии в образовании.

- Содержание образования. Моделирование. Элементы искусственного интеллекта.

- Программное и методическое обеспечение обучения основам информатики и вычислительной техники.

- Программное и методическое обеспечение для предметов естественно-научного цикла.

- Базовое и инструментальное программное обеспечение.

3. В редколлегию необходимо представить по 2 экз.:

- текста статьи, подготовленной на русском, латышском или английском языках, отпечатанной через 1,5 интервала, объемом до 10 стр. вместе с иллюстрациями и библиографическим списком (подписи авторов на обороте титульного листа);

- реферата на русском, латышском (для авторов из Латвии) и английском языках;

- акта экспертизы.

4. Образцы оформления статьи, реферата, библиографического списка, иллюстрация см. в настоящем сборнике.

5. Желательна передача текстового и графического файла, подготовленного на персональной ЭВМ. Текст выравнивается по правому краю, в строке - 60 символов, на странице - 38 строк (включая служебные и информационные).

6. Адрес: 226250, г. Рига, ГСП, бульв. Райниса, 29

ВЦ при ЛГУ им. П. Стучки.

лаборатория проблем школьной

и вузовской информатики

7. Телефоны членов редколлегии (г. Рига, код - 013-2):

М. В. Витиньш - 22-37-27

С. И. Павлов - 22-38-32

В. Л. Шилевич - 22-38-32

83488

LU bibliotēka



958002052

544

2 р. 20 к.

A-90
1

