

А. Г. Леонов

Логическое проектирование педагогических программных средств

Эффективность проектирования, внедрения и использования педагогических программных средств в значительной мере определяется исследованиями эргономического человеко-машинного интерфейса, наиболее важными из которых здесь выделены методические инварианты как в учебном процессе пропедевтического курса, так и на этапе непрерывного перехода к базовому школьному курсу информатики.

Эти методические инварианты, наблюдаемые и обобщенные, приводят к обоснованию принципов логического проектирования педагогических программных средств, особенно значимых для курса, в котором методические обновления тесно сопрягаются с содержанием, выходящим на учебные языки программирования. В этом обосновании учтены и эргономические аргументы.

Они (принципы логического конструирования – от программных исполнителей к учебно-ориентированным языкам) иллюстрируются на примерах известных учебных программных систем ПиктоМир и КуМир.

Ключевые слова: школьная информатика, интерфейс, инвариантные свойства учебных программ, педагогическое программное средство, непрерывный курс, программирование, КуМир.

A. G. Leonov

The logical design of pedagogical programming systems

The effectiveness of design, implementation and use of the educational software are determined by the research of the ergonomic human-computer interface, the most important of which are methodological invariants in the educational process of the propedeutic course and at the stage of the non-stop transition to the basic course of Informatics in school.

These methodological invariants, observations and generalizations lead to the justification of the principles of logic design of educational software tools. Such principles are especially important for the course in which the methodological updates are closely connected with the content of the teaching language for programming. This justification also took into consideration the ergonomic arguments.

The principles of logic design (from the programme executor to the teaching language) are illustrated by famous educational software systems PiktoMir and KuMir.

Keywords: school Informatics, interface, invariant property of the teaching programme, educational software, a continuous course of Computer Science, programming, KuMir.

Умения и знания, которыми должен обладать современный человек, включают широкую группу понятий и навыков, имеющих близкое отношение к информатике. Такие понятия, как *робот, команды, управление, программирование* и т. п., давно обрели привычный смысл для молодого поколения. Современный человек, слабо разбирающийся в вопросах финансового рынка, сможет реализовать себя во многих других областях, но отсутствие элементарной информационной культуры (как и неспособность управляться с сотовым телефоном) приведут его в лагерь функционально неграмотных людей. Таким образом, тезис академика А. П. Ершова «Программирование – вторая грамотность» [1] сейчас приобретает особую значимость.

Сегодняшние дошкольники обладают определенным набором навыков и умений, позволяющих им ориентироваться в информационном пространстве и использовать современные информационные и компьютерные технологии. Дети четырех–пяти лет имеют опыт «пультового» управления бытовыми приборами: телевизорами, проигрывателями, электронными игрушками. Они с помощью пульта без труда могут найти интересующий их канал или программу на телевизоре, «запустить» с пульта сказку на проигрывателе и т. п. Более старшие дети, или, по крайней мере, многие из них, являются обладателями своеобразных персональных компьютеров – игровых приставок (SonyPlayStation, GameBoy, Wii и пр.). Большинству из них не только знакомы такие понятия, как *компьютер, Интернет, мышка* и пр., но они уже обладают навыками по использованию этих компьютеров (иногда

не только для игр), умеют писать электронные письма, пользоваться текстовым редактором, находить в Интернете интересующую их информацию. Компьютеры, в том или ином виде, плотно вошли в повседневную жизнь современного подростка. Таким образом, сегодняшний уровень владения современными информационными и компьютерными технологиями дает дошкольникам определенный набор навыков (привычек), уже сформированными до начала изучения информатики и использования педагогических программных средств на уроках в школе.

Даже на первых уроках, где используются компьютеры, таким детям не потребуется объяснять, как пользоваться мышкой или клавиатурой. Многим из них не нужно объяснять, как вводить текст в редакторе и как составлять и отправлять электронное письмо. Тем самым, педагог не только сможет более эффективно вести урок по своему предмету, используя компьютерные технологии. Он даже обязан скорректировать программу урока и, использовать те педагогические программные средства, которые надлежащим образом адаптированы к современному школьнику. То есть, педагогические программные средства должны учитывать уровень владения школьником современными информационными технологиями, а также использовать понятийную базу, которой уже владеет ученик.

С другой стороны, чем меньше новых терминов и навыков потребуется школьнику, чтобы освоить учебное программное средство, тем эффективнее будут его занятия на уроках, и тем дальше он сможет продвинуться в поставленной ему учителем задаче, часто напрямую не связанной с информатикой.

Поэтому при проектировании педагогического программного средства требуется продумать и зафиксировать понятийную базу, необходимую для освоения этого программного продукта, и минимизировать эту базу, ориентируясь на набор знаний и навыков современного школьника.

Так в системе для дошкольников и младших школьников ПиктоМир понятие *Исполнитель*¹, не знакомое и трудное в понимании для малышей, заменено на понятие *Робот*. Трудно найти ребенка, которому не знаком этот термин. Исполнитель обладает определенной системой команд, но и каждый ребенок знает, что Робот умеет выполнять определенные действия и (так же, как и Исполнителем) им можно управлять при помощи команд. Дети знают, что Робот не человек, и поэтому не догадывается о цели, которой хочет добиться управляющий им человек. То же можно сказать про Исполнителя, которому не известна конечная цель выполняемого им алгоритма. Таким образом, для младшего школьника, понятие Исполнитель становится полностью эквивалентным знакомому понятию Робот. Замена одного понятия в педагогическом программном средстве на другое, но уже известное учащемуся, позволяет сократить набор новых терминов, необходимых для освоения программного продукта и решения поставленной педагогом задачи. Такой подход сокращает время изучения учащимся нового программного продукта.

Но ученик не застрахован от ошибок при освоении не знакомого ему педагогического программного средства. Ему необходимо время на изучение, освоение, адаптацию к новой программной среде, даже к самой простой. Пока идет этот достаточно мучительный процесс, у обучаемого могут возникнуть стрессовые состояния не только из-за непонимания, но и из-за боязни (неумения) обращаться с новым программным средством. Такую проблему обычно решают при помощи видео-курса и (или) учебных занятий по освоению программного средства. К такому же эффекту приводит набор последовательно усложненных заданий. В дополнение можно предложить включить в программное средство режим типа «тест-драйва», когда учащийся знакомится с ПО еще до постановки задачи².

Важнейшей частью для успешной реализации проекта педагогического программного средства, является интерфейс, проектирование которого можно выделить в отдельную задачу. Дополнительные специальные требования к интерфейсу накладывает и эргономика.

Не только ЭВМ, но и все машины и механизмы, взаимодействующие с человеком физически, инженеры разрабатывают с учетом возможностей человека – его скелета, мускулатуры, органов чувств и т. д. Совокупность сведений в этой области и составляет эргономику. Используя знания об физических особенностях человека, сегодня проектируют не только удобные стулья и столы, но и клавиатуры и мышки, дисплеи программиста, пульт оператора атомной электростанции. Учет человеческих особенностей крайне важен для конечного программного продукта. Например, вряд ли рабочему будет удобно за станком, у которого кнопки управления разнесены друг от друга на расстояние нескольких метров. Более того, работа на таком станке просто опасна для здоровья, т. к. нажать кнопку аварийного отключения за долю секунды не удастся. С точки зрения эргономики, легко включать и выключать сильно «разнесенные» переключатели не сможет ни один человек.

Эргономическое проектирование подразумевает выбор таких параметров устройства управления, при котором оно будет удобно среднестатистическому большинству, например, автомобильное кресло окажется удобным, подходящим для 95 % водителей, тогда как остальным 5 % такие кресла покажутся неудобными, из-за слишком высокого или слишком маленького роста.

Аналогично, при проектировании интерфейса учебной программы для школьников, требуется тщательным образом учитывать эргономические ограничения, которые определяются не только механическими или медико-биологическими особенностями ребенка, но и психологическими и культурологическими особенностями. Также многие ограничения определяются, в первую очередь, возрастом. Так, у младших школьников ладонь имеет меньший размер, чем у их старших товарищей, поэтому исполнение некоторых фортепьянных произведений в музыкальной школе или управление полноразмерной мышкой на уроке в средней школе может потребовать больших физических усилий, что, в свою очередь, приведет к быстрой физической утомляемости на занятиях как за фортепиано, так и за компьютером.

Для младших школьников является серьезной проблемой обычное движение по перетаскиванию некоторого объекта мышкой на экране, которое состоит из нажатия кнопки, удерживания ее, и перемещения мышки по столу, продолжая удерживать эту кнопку до нужного места. Именно удержание кнопки мыши с одновременным перемещением мышки по столу создает большую нагрузку на опорно-двигательный аппарат школьника. С такой проблемой легко справляются взрослые люди или старшие школьники, маленьким детям эта задача не под силу.

В системе ПиктоМир для настольных ЭВМ стандартная операция перетаскивания иконки с командой на поле программы состоит из следующей последовательности действий (рис. 1):

- Кликом мышки активируется нужная иконка (активная иконка начинает подпрыгивать).
- Мышь перемещают в нужное поле программы (кнопку нажатой держать не требуется).
- Кликом мыши указывают выбранное поле (иконка копируется в программу).

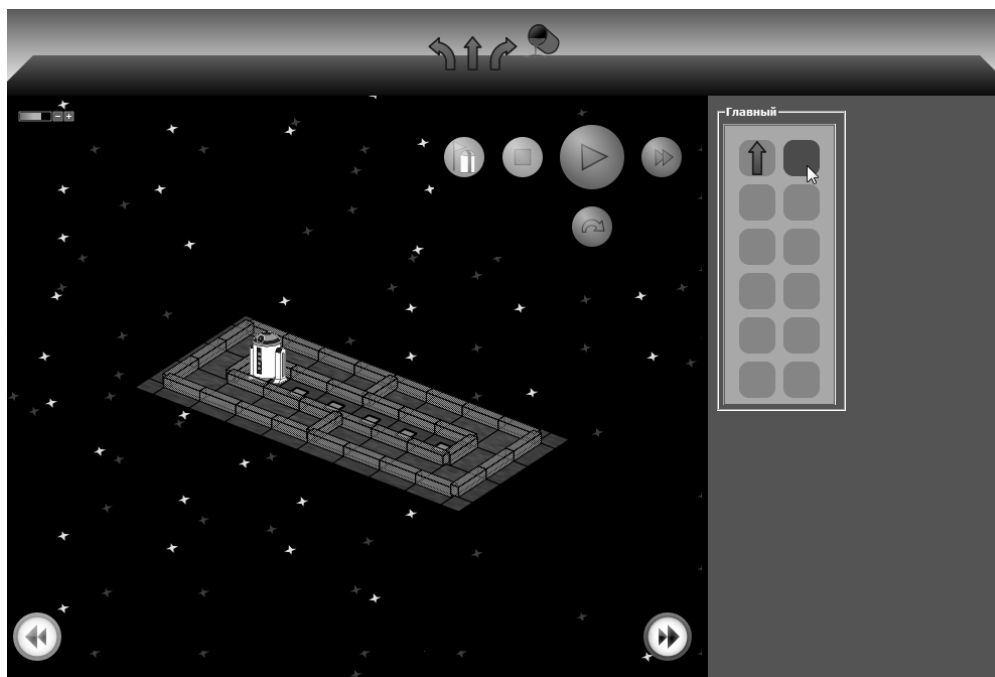


Рис. 1. ПиктоМир. Указание, вместо перетаскивания иконки.

Можно добавить еще целый ряд возрастных эргономических (физических) ограничений на интерфейс педагогических программных средств. Не рассматривая их по отдельности, можно, тем не менее, сделать один неоспоримый вывод – в начальной школе время непосредственного «сидения» за компьютером должно быть небольшим.

Однако не только традиционная эргономика накладывает ограничения на интерфейс педагогических программных средств. Так как большинство механизмов, созданных цивилизацией, взаимодействуют с нами в основном механически, то накоплен большой опыт и ограничения, связанные с меньшими размерами и весом ребенка, как правило, легко сформулировать и постараться учесть. Но как учесть взаи-

модействие с программной средой, которое имеет большее отношение к области интеллектуальной, нежели физической. В своей книге «Интерфейс: новые направления в проектировании компьютерных систем» известный специалист в области проектирования человеко-машинного интерфейса Джеф Раскин дает ответ: «Мы должны овладеть эргономикой сознания, если мы хотим создавать интерфейсы, которые могли бы хорошо работать. Удивительно, но мы часто не замечаем собственные ментальные ограничения, поэтому для определения границ возможностей нашего сознания мы должны прибегнуть к тщательному наблюдению и экспериментированию. Изучение прикладной сферы наших ментальных способностей называется когнитивным проектированием, или когнетикой» [2].

Некоторые когнитивные ограничения вполне естественны: например, как нельзя ожидать от обычного школьника способности возводить в квадрат и куб в уме многозначные числа, так нельзя проектировать педагогически программные средства, полноценное использование которых требует детального изучения томов документации, например, требуя от школьника профессионально освоить программные системы, аналогичные Microsoft Word. Для этого у учащихся просто нет ни сил, ни времени, ни, главное, мотивации. К сожалению, еще достаточно часто попадаются программные системы и разрабатываются компьютерные интерфейсы (и интерфейсы в широком смысле слова), созданные с расчетом на некие когнитивные способности, которыми взрослый человек или ребенок на самом деле не обладает. В таком случае уместно говорить о низком качестве таких человеко-машинных интерфейсов.

Но подобная проблема «некачественных» интерфейсов в педагогических программных продуктах ведет не только к сложностям использования программного продукта, но и к приобретению учениками ненужных и даже вредных навыков и привычек по использованию компьютера. Приведем еще одну цитату Джефа Раскина: «На языке когнитивной психологии любая задача, которую вы научились выполнять без участия сознания, становится автоматичной. Автоматизм позволяет выполнять сразу несколько действий одновременно. Все одновременно выполняемые задачи, за исключением не более, чем одной, являются автоматичными.

Не менее важным, чем понимание того, что в каждый отдельный момент времени нельзя осознавать более одной задачи, является тот факт, что человек не может избежать формирования автоматических реакций. Эта невозможность не зависит от повторения: никаким количеством повторений нельзя научиться не формировать привычки при регулярном использовании того или иного интерфейса. Формирование привычек является неотъемлемой частью нашего ментального аппарата. Его невозможно остановить волевым действием».

Сформированные неправильные привычки по использованию программного интерфейса вступают в противоречие с уже сформированными навыками и впоследствии потребуют переучивания.

Как уже говорилось выше, к началу занятий по информатике ученик, еще не знакомый с тем педагогическим программным средством, которым ему придется использовать, обладает набором не просто навыков и умений, а привычных навыков и умений, которые также необходимо учитывать при разработке педагогических компьютерных систем.

Поставим перед педагогом задачу: в минимальное время научить детей основам алгоритмизации. Или, еще более конкретно, организовать первое занятие так, чтобы каждый ученик сумел написать первую в его жизни содержательную программу. Сложности этого процесса для детей несомненна, так как ребенок от рождения не обладает алгоритмическим мышлением. Попытаемся упростить эту сложную проблему.

Основная нагрузка на ученика начинается, когда требуется составить программу, т. е. в некотором виде (как правило, словесном, если идет речь о языках программирования) записать будущую последовательность действий. Этот язык программирования, вне зависимости от того, учебный он или производственный, должен быть некоторым способом формализован для записи алгоритмов. Но в этом и кроется основная проблема на начальном этапе освоения алгоритмизации. Для школьника даже учебный язык программирования, который использует лексику родного языка, является трудным в освоении на начальном этапе. В любом случае, учащийся должен изучить и попытаться запомнить команды нового языка, понять их назначение, чтобы впоследствии он мог их применить, т. е. составить программу на этом языке. Таким образом, для решения задачи по написанию первого алгоритма, которую поставил перед учеником учитель, нужно сделать следующее:

1. продумать алгоритм;
2. записать его на некотором (уже изученном) алгоритмическом языке, причем запись должна быть достаточно формальной;
3. используя (изученный заранее) интерфейс некоторой педагогической программной системы, загрузить записанный алгоритм в ЭВМ и проверить его работоспособность.

Каждый из перечисленных выше трех пунктов несет самостоятельную сложность для ученика. Труднее всего решить первую проблему, т. к. достаточно сложно проследить путь, как человек придумывает что-то новое. Однако, в решении двух оставшихся проблем можно продвинуться вперед, сняв с них ряд искусственных трудностей: освоение алгоритмического языка и изучение педагогического программного средства.

Ниже излагается подход к конструированию педагогического программного обеспечения на примере цепочки программных средств по информатике и ИКТ для начальной, базовой и старшей школы обучения основам алгоритмизации.

В непрерывном курсе информатики стартовую позицию в цепочке учебного программного обеспечения занимают инструментальные средства пропедевтической части курса – *исполнители* и, в первую очередь, *программные исполнители* [3]. Здесь мы не будем детально рассматривать методические аспекты использования исполнителей на школьных уроках, в многообразии программных исполнителей, входящих в состав программно-методической системы КуМир [4].

1. Пульт

Как известно, школьники умеют управлять домашними приборами с помощью пультов управления. Это умение, сформированное в привычные действия, приобретенное ими с детства. Для непосредственного управления программными исполнителями можно использовать *обобщенный* пульт. Действительно, дома пульты управления имеют более-менее схожий интерфейс. Одни и те же действия обозначаются одинаковыми значками. При этом, даже ребенок, взяв в руки, например, мобильный телефон (вне зависимости от производителя) сумеет набрать номер и сделать звонок. Поэтому различные исполнители в системе КуМир имеют сходные пульты управления. Так называемый, обобщенный пульт.

Этот пульт не только имеет кнопки для управления исполнителем (например, Роботом) и индикаторы для обратной связи, но и макрокоманды, знакомые детям по умным домашним пультам и электронное табло, на котором отображаются выполненные команды. Макрокоманды, по сути своей, представляют собой первые вспомогательные алгоритмы, с которыми сталкивается школьник. При этом пульт позволяет не только непосредственно управлять тем или иным исполнителем, но и «запоминать» последовательные фрагменты из команд для упрощения управления исполнителем.

Во главу угла поставлена задача, чтобы школьники преодолевали не комбинацию сложностей информатики и, например, математики, а «чистые» алгоритмические сложности информатики.

Именно эти цели преследует простейший исполнитель – чрезвычайно простой исполнитель Робот, которым можно управлять с помощью обобщенного пульта.

Робот имеет небольшой набор базовых команд:

- Вверх
- Вниз
- Вправо
- Влево
- Закрасить

С одной стороны, Робот достаточно прост, с другой стороны, жизненный багаж учеников уже содержит достаточные знания о том, что такое Робот, и поэтому для них не будет удивительным, что Робот управляется дистанционно, и у него есть пульт управления. То, что пульт управления имеет много кнопок, также вряд ли удивит ребенка. При этом можно быть уверенным, что если ученик думает над задачей по управлению Роботом, то все сложности у него алгоритмически.



Рис. 2. Пульт Робота как обобщенный пульт.

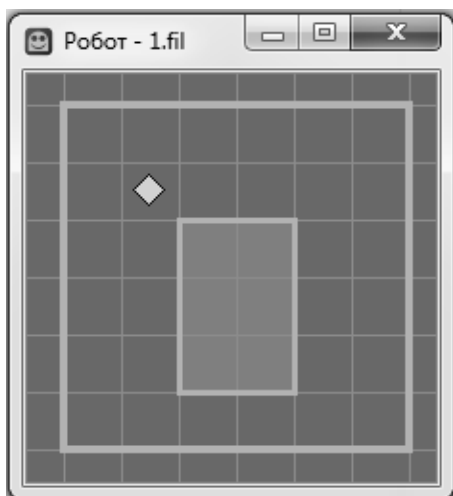


Рис. 3. Робот у прямоугольного препятствия.

Также отсутствует сложность не только в Роботе, но и в постановке задачи: если нарисовано поле Робота с прямоугольным препятствием, Робот стоит в верхнем левом углу, и нужно попросить придумать ученика алгоритм, по которому Робот обошел бы вокруг этого прямоугольника «неизвестных размеров», то такая постановка задачи тоже ясна, хотя на экране нарисован вполне конкретный прямоугольник.

Последовательность команд, набранную на пульте для решения конкретной задачи, можно сохранить для изучения и дальнейшего использования уже в программном управлении в системе КуМир.

2. ПиктоМир

Однако пультовое управление Роботом (как и другими исполнителями) имеет свои алгоритмические границы. Если простейшие линейные алгоритмы еще можно создавать при помощи пульта, то циклы приходится организовывать при помощи управляющей системы «ученик с пультом».

Для создания более сложных управляющих алгоритмов разумно воспользоваться управляющей средой ПиктоМир, использующий визуальное (пиктографическое) программирование³.

В настоящее время пиктограммы являются неотъемлемым признаком современных интерфейсов. По мнению компании AppleComputer, «пиктограммы могут существенным образом увеличить ясность и усилить привлекательность приложения. Кроме того, использование пиктограмм позволяет намного упростить процесс перевода программ на другие языки. Всякий раз, когда требуется добавить объяснение или надпись, попытайтесь вместо текста использовать пиктограмму». В педагогическом программном средстве, призванном для изучения начал программирования, использование пиктограмм вполне уместно. ПиктоМир предназначен для устранения противоречия, когда «в голове» у ребенка есть программа – план будущей деятельности, а вот средств выразить этот план в какой-то текстовой форме еще нет. Система ПиктоМир безтекстового, пиктографического программирования позволяет школьнику «собрать» программу, управляющую программными исполнителями, например, тем же Роботом.

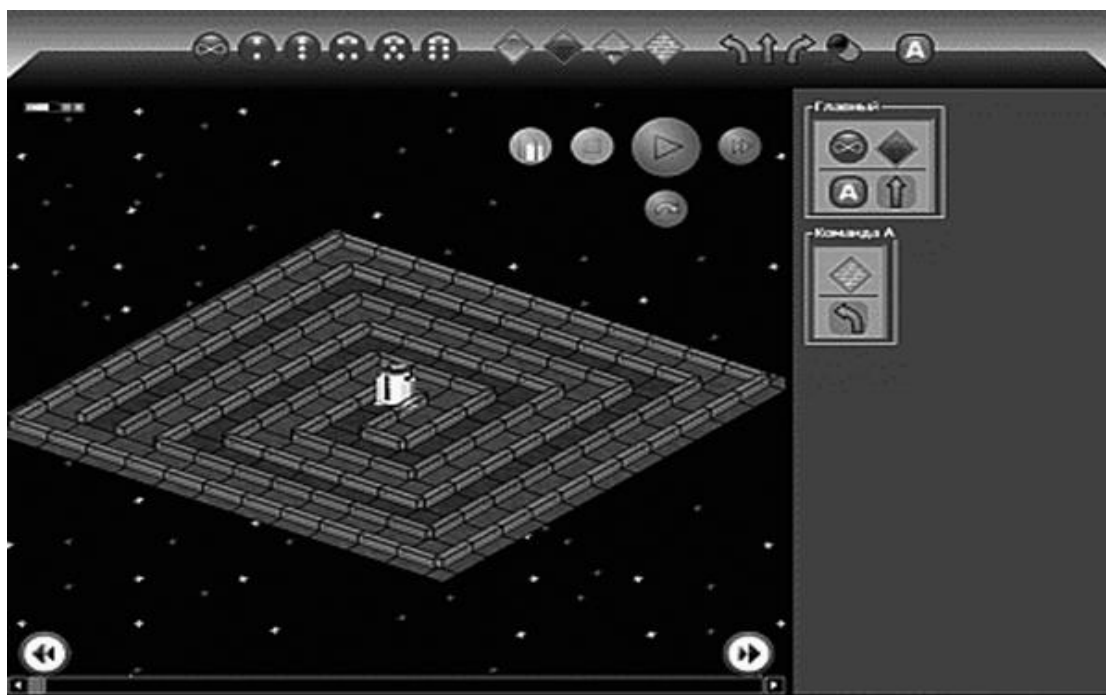


Рис. 4. Педагогическая программная среда ПиктМир.

В Пиктомире, кроме главного алгоритма, с которого всегда начинается выполнение программы, можно использовать несколько вспомогательных алгоритмов, подобно макрокомандам на обобщенном пульте. В ПиктоМире уже есть циклы: можно задать, сколько раз нужно выполнять все команды вспомогательного алгоритма 1, 2, 3, 4, 5 или 6, или до бесконечности, точнее, пока не будет выполнено определенное условие. Таким специальным способом можно задать в ПиктоМире частные случаи общих конструкций цикл «N раз» и «цикл пока» и команду выбора «если-то».

```
использовать Робот_Вертун

алг Главный алгоритм
нач
. нц пока клетка закрашена
. . А
. . вперед
. кц
кон

алг А
нач
. если впереди стена то
. . повернуть налево
. . иначе выход
. все
кон
```

Но при всем удобстве набор решаемых при помощи ПиктоМира задач по управлению исполнителями ограничен. Хотя созданную программу можно экспортировать в систему КуМир.

Так, например, в Пиктомире, отсутствуют величины, которые есть в системе КуМир.

3. КуМир

Переход к системе КуМир (использующий алгоритмический язык) можно упростить, что поможет освоению школьного алгоритмического языка. Все результаты визуального программирования в ПиктоМире (и на Пульте), т. е. составленную школьником программу, можно загрузить в системе КуМир. При этом программа будет выполнять тот же самый алгоритм, только он будет записан на школьном алгоритмическом языке. Знакомство с синтаксисом нового, более сложного текстового алгоритмического языка пойдет проще, если начнется с изучения знакомых алгоритмов, уже освоенных в ПиктоМире.

Рис. 5. Экспортирование из ПиктоМира в КуМир.

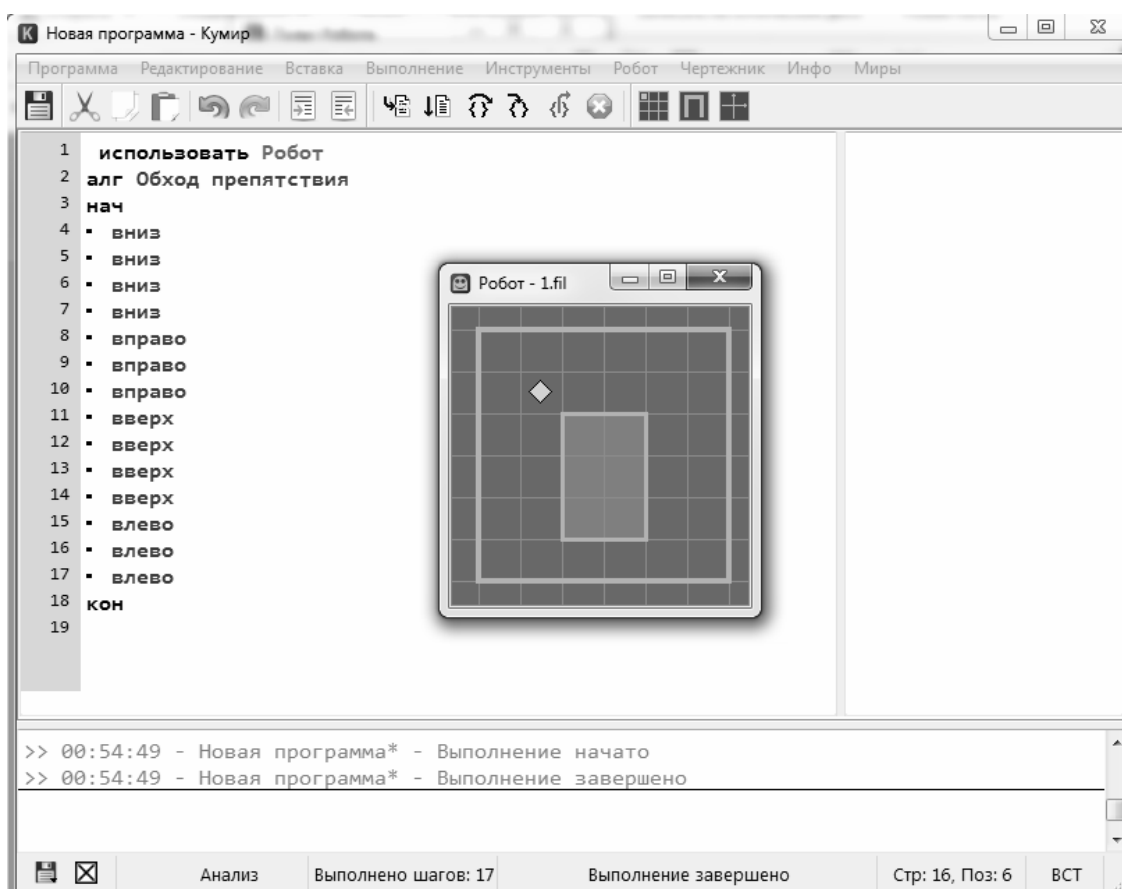


Рис. 6. Программа «Обход препятствия».

Важно, чтобы на начальном этапе обучения задачи, которые решают ученики, можно было отделить от лишних технических деталей, например, от английского алфавита и английских слов и т. п.

Системы КуМир специально разрабатывалась таким образом, чтобы ученик мог решать задачи и концентрироваться на преодолении соответствующей алгоритмической сложности, а не разбираться с особенностями программного обеспечения. Это касается и средств «порождения» конструкций языка, и немедленной диагностики ошибок при вводе алгоритма, и средств исполнения, и показа результатов работы написанного алгоритма. В процессе исполнения на экране изображаются и алгоритм, и результаты его работы – перемещения Робота по полю. Выполнение можно прервать, алгоритм подправить и запустить вновь.

КуМир – учебная система. Она сводит к минимуму «накладные расходы» на освоение, имеет развитую систему диагностики ошибок, средства, позволяющие ученику следить за выполнением программы и т. п. Ученик, никогда ранее не программировавший, может начать писать и выполнять алгоритмически относительно сложные программы через 1–2 часа после первого знакомства с Кумиром. В то же время система КуМир позволяет создавать достаточно большие и сложные программы.

Работа в КуМире очищена от понятия «компиляции» и специальных отладчиков. Можно сказать, что КуМир (на начальном уровне обучения) очищен даже от понятия файловой системы, хотя работа с файлами включена в него как отдельный исполнитель. При создании прототипа системы, в 1986 г. [5] перед разработчиками действительно стояла задача подготовить за кратчайшие сроки первых учителей информатики, большинству из которых впервые предстояло преподавать этот предмет. Поэтому создатели системы сэкономили не только время учеников, но и педагогов.

Библиографический список

1. Ершов, А. П. Программирование – вторая грамотность [Текст] / А. П. Ершов // 3-я Всемирная конференция ИФИП (Международной федерации по обработке информации) и ЮНЕСКО по применению ЭВМ в обучении. – Лозанна (Швейцария), 1981.
2. Раскин, Д. Интерфейс: новые направления в проектировании компьютерных систем [Текст] / Джеф Раскин. – М. : Символ-Плюс, 2005. – 272 с.
3. Звенигородский, Г. А. Первые уроки программирования [Текст] / Г. А. Звенигородский. – М. : Наука, Библиотечка «Кванта», 1985. – 208 с.
4. Леонов, А. Г. Методика преподавания основ алгоритмизации на базе системы «КуМир» [Текст] / А. Г. Леонов, А. Г. Кушниренко. – М. : 1 сентября, 2009.
5. Лебедев, Г. В. Основы информатики и вычислительной техники [Текст] / Г. В. Лебедев, Р. А. Сворень, А. Г. Кушниренко. – М. : Просвещение, 1991. – 224 с.

Bibliograficheskiy spisok

1. Ershov, A. P. Programmirovaniye – vtoraya gramotnost' [Tekst] / A. P. Ershov // 3-ya Vsemirnaya konferenciya IFIP (Mezhdunarodnoj federacii po obrabotke informacii) i JuNESKO po primeneniju JeVM v obuchenii. – Lozanna (Shvejcarija), 1981.
2. Raskin, D. Interfejs: novye napravlenija v proektirovanii komp'juternyh sistem [Tekst] / Dzhef Raskin. – M. : Simvol-Pljus, 2005. – 272 s.
3. Zvenigorodskij, G. A. Pervye uroki programmirovaniya [Tekst] / G. A. Zvenigorodskij. – M. : Nauka, Bibliotekha «Kvanta», 1985. – 208 s.
4. Leonov, A. G. Metodika prepodavaniya osnov algoritimizacii na baze sistemy «KuMir» [Tekst] / A. G. Leonov, A. G. Kushnirenko. – M. : 1 sentjabrja, 2009.
5. Lebedev, G. V. Osnovy informatiki i vychislitel'noj tehniky [Tekst] / G. V. Lebedev, R. A. Svoren', A. G. Kushnirenko. – M. : Prosveshhenie, 1991. – 224 s.

¹Исполнитель – это устройство, приспособление, робот, организация и т. п., способное выполнять определенные действия, строго фиксированный набор команд.

²Так, в системах КуМир и ПиктоМир включен режим кнопочного управления исполнителями. Ещё до получения задачи, ученик может провести по полю Робота, освоиться с его набором команд.

³Пиктограммы (англ. icons) – маленькие картинки, служащие здесь для обозначения кнопок и других объектов на экране.