



ИНФОРМАТИКА И ИКТ В НАЧАЛЬНОЙ ШКОЛЕ

А. Г. Леонов,

Московский государственный университет имени М. В. Ломоносова, механико-математический факультет;

Московский педагогический государственный университет;

Федеральный научный центр Научно-исследовательский институт системных исследований Российской академии наук, г. Москва,

Ю. А. Первин, Я. Н. Зайдельман,

Федеральный научный центр Научно-исследовательский институт системных исследований Российской академии наук, г. Москва

ПРОГРАММНЫЕ ИСПОЛНИТЕЛИ В ЦИФРОВЫХ ОБРАЗОВАТЕЛЬНЫХ СРЕДАХ «ПИКТОМИР», «РОБОТЛАНДИЯ» И «КУМИР»*

Аннотация

В статье анализируются необходимость и возможность изучения основ программирования в дошкольном и младшем школьном возрасте. Рассматривается известная технология изучения программирования на базе программных исполнителей, при этом предлагается существенно снизить возраст начала обучения. Возможность этого снижения обосновывается изменениями среды, в которой живут современные дети, их ранним знакомством с разнообразными программно управляемыми устройствами, накопленным опытом управления такими устройствами. В статье рассматриваются конкретные учебные системы, реализующие этот подход: «ПиктоМир», «Роботландия», «КуМир». Показан постепенный переход от бестекстового программирования с помощью пиктограмм («ПиктоМир») через текстовое управление с помощью минимального набора конструкций («Роботландия») к полноценному языку программирования («КуМир»). При этом работа в среде «КуМир» тоже начинается с управления исполнителями, тем самым обеспечиваются преемственность подходов и постепенное усвоение все более сложных методов программирования. В статье обобщается опыт авторов в разработке и педагогической апробации применения перечисленных систем, делается вывод о возможности, методической подготовленности и образовательной обоснованности раннего изучения основ программирования с использованием программно управляемых исполнителей.

Ключевые слова: информатика, Робот, «ПиктоМир», «КуМир», «Роботландия», алгоритм, программа, язык программирования, школьный алгоритмический язык.

DOI: 10.32517/2221-1993-2019-18-9-54-61

* Работа выполнена по теме 0065-2019-0010 «Разработка, реализация и внедрение семейства интегрированных многоязыковых сред программирования с автоматизированной проверкой заданий для учащихся образовательных организаций, ДОО, младшей, основной и старшей школы и студентов педагогических университетов» госзадания в отделе учебной информатики ФГУ ФНЦ НИИСИ РАН.

Контактная информация

Леонов Александр Георгиевич, канд. физ.-мат. наук, доцент;

ведущий научный сотрудник кафедры вычислительной математики, лаборатории вычислительных методов, механико-математический факультет, Московский государственный университет имени М. В. Ломоносова; *адрес:* 119991, г. Москва, Ленинские горы, МГУ, д. 1;

профессор кафедры математики и информатики в начальной школе, Институт детства, Московский педагогический государственный университет; *адрес:* 119991, г. Москва, ул. Малая Пироговская, д. 1, стр. 1;

зав. секторами, профессор, Федеральный научный центр Научно-исследовательский институт системных исследований Российской академии наук, г. Москва; *адрес:* 119991, г. Москва, Нахимовский пр-т, д. 36, корп. 1;

e-mail: dr.l@infomir.ru

Первин Юрий Абрамович, доктор пед. наук, профессор, ведущий научный сотрудник, Федеральный научный центр Научно-исследовательский институт системных исследований Российской академии наук, г. Москва; *адрес:* 119991, г. Москва, Нахимовский пр-т, д. 36, корп. 1; *e-mail:* ypervin@inbox.ru

Зайдельман Яков Наумович, научный сотрудник, Федеральный научный центр Научно-исследовательский институт системных исследований Российской академии наук, г. Москва; *адрес:* 119991, г. Москва, Нахимовский пр-т, д. 36, корп. 1; *e-mail:* yz@pereslavl.ru

A. G. Leonov,

Lomonosov Moscow State University, Faculty of Mechanics and Mathematics;

Moscow Pedagogical State University;

Federal Research Center Research Institute for System Studies of the RAS, Moscow,

Yu. A. Pervin, Ya. N. Zaidelman,

Federal Research Center Research Institute for System Studies of the RAS, Moscow

SOFTWARE EXECUTORS IN THE DIGITAL PEDAGOGICAL ENVIRONMENTS: PIKTOMIR, ROBOTLANDIA AND KUMIR

Abstract

The article analyzes the necessity and possibility of studying the basics of programming in preschool and primary school. A well-known technology for studying programming based on software executors is considered, while it is proposed to significantly reduce the age at which teaching begins. The possibility of this decrease is justified by changes in the environment in which modern children live, their early acquaintance with a variety of software-controlled devices, accumulated experience in managing such devices. The article discusses specific training systems that implement this approach: PictoMir, Robotlandia, KuMir. A gradual transition from non-text programming using pictograms (PictoMir) through text management using a minimal set of constructions (Robotlandia) to a full-fledged programming language (KuMir) is shown. At the same time, work in the Kumir environment also begins with the management of performers, thereby ensuring continuity of approaches and the gradual assimilation of increasingly complex programming methods. The article summarizes the authors' experience in the development and pedagogical testing of the application of the above systems, the conclusion is made about the possibility, methodological preparedness and educational soundness of the early study of the basics of programming using software-controlled executors.

Keywords: informatics, Robot, PictoMir, KuMir, Robotlandia, algorithm, program, programming language, school algorithmic language.

1. Введение

О необходимости для современного человека обладать определенными компетенциями в области программирования не раз говорил основатель отечественной школьной информатики академик А. П. Ершов. В своей работе «Программирование — вторая грамотность» он прямо указывает на всепроникающую сущность программирования: «Повседневная жизнь человека, особенно городская, — это деятельность по программам. Каждый человек, придерживающийся режима, с гордостью почувствует себя программистом, если вспомнит свои заполненные до предела утренние процедуры, начиная от звонка будильника и кончая началом работы. Поразмышляйте над процедурой уборки в квартире, и вы увидите, что разработка этой программы сделает честь любому профессиональному программисту» [4].

В связи с потребностью общества в новых информационных и алгоритмических компетенциях современного человека многие ученые, педагоги и исследователи как в России [20], так и за рубежом [21] еще в конце прошлого века начали работать над методикой обучения школьников информатике и информационно-коммуникационным технологиям.

Так, еще в СССР в далеком 1985 году был введен курс информатики в IX—X классах школы — предмет «Основы информатики и вычислительной техники» [16]. В курсе использовался школьный алгоритмический язык [3]. При этом предполагалось безмашинное изучение важнейших алгоритмов на этом текстовом учебном языке. Разработанный в Новосибирском государственном университете и Сибирском отделении Академии наук СССР язык Рапира, более сложный, чем школьный алгоритмический язык, было предложено изучать в курсе с использованием компьютера в составе учебной системы программирования «Школьница» [6]. Одновременно школьный алгоритмический язык получил «машинное» внедрение, когда в МГУ имени М. В. Ломоносова была разработана среда программирования «Е-практикум», названная так в честь академика А. П. Ершова [2]. В среде программирования «Е-практикум» можно было решать практически все задачи из курса информатики, но уже с использованием компьютера. В дальнейшем эта среда программирования претерпела коренные изменения — во вновь созданную систему были включены программно управляемые исполнители [12]. Новая система программирования была названа «КуМир» и рассматривалась как средство для составления алгоритмов в различных «мирах» исполнителей. Школьный алгоритмический язык получил достаточно широкое распространение [5, 9, 10] и используется в школах и в настоящее время [1, 8, 15, 19].

В течение ряда лет были предприняты усилия по снижению возраста обучаемых в курсах информатики и ИКТ [7] и более тесной интеграции курса с близкими предметами, такими, как, например, математика [18]. К основным сложностям снижения возраста пропедевтики алгоритмизации можно отнести неподготовленность детей к использованию компьютера (необходимо было отдельно изучать человеко-машинный интерфейс), эргономические ограничения (размер мышки и клавиатуры), сложности текстовых языковых систем программирования, большой объем новой для ребенка

понятийной базы. Однако в последнее время произошли существенные изменения в подходах к изучению основ информатики в раннем возрасте [11] в связи с общей подготовленностью детей в современном цифровом мире.

Сейчас знакомство детей с компьютером и информационно-коммуникационными технологиями происходит в достаточно раннем возрасте. Как правило, в семье еще до посещения дошкольных образовательных учреждений ребенок имеет возможность поиграть со смартфоном или планшетом родителей. Некоторые основные навыки использования планшетных компьютеров и смартфонов появляются у детей не только в процессе игры на планшете, но и при просмотре мультфильмов, прослушивании сказок и т. п. [22]. При этом ребенок познает элементы интерфейса работы с планшетами и смартфонами: выбор иконки, перемещение иконок, изменение масштаба изображения — эти навыки знакомы современным детям еще до их первого появления в образовательном учреждении [23].

Во всем мире важным «действующим лицом» современности становится робот. Роботы собирают станки, управляют автомобилями и поездами, летают в космос. Понятие робота как некоторого автономного электронно-механического изделия знакомо детям из окружающей их действительности. Это радиоуправляемые игрушки, квадрокоптеры, банкоматы, лифты и многие другие устройства. Встроенные в смартфоны электронные помощники Алиса и Siri могут разговаривать с людьми часами. Робота в том или ином виде можно встретить практически повсеместно. Удобно, что понятие робота не нужно отдельно объяснять ученикам. Для них робот понятен, в нем нет ничего сложного. Робот выполняет команды, роботом можно управлять. Для некоторых роботов доступен пульт дистанционного управления, нажимая кнопки которого можно дистанционно управлять умным механизмом. Фактически и компьютер, и планшет, и смартфон — тоже роботы, функционал которых доступен передачей команд нажатиями на кнопки, области экрана или голосом.

Уже в раннем возрасте дети приобретают опыт кнопочного управления роботом. Если дошкольнику попадает в руки игрушка с дистанционным пультовым управлением, то, проводя самостоятельные исследования (нажимая на различные кнопки пульта) или при обучении взрослыми, уже пяти-шестилетний ребенок в состоянии выполнить простейшие задания по управлению роботом-игрушкой. Пультовое программирование понятно маленькому человеку, при этом в голове у него уже фактически сформирована программа для робота, однако средств ее выразить у ребенка пока нет.

Еще одна важная особенность пультового программирования — отсутствие текста. Ребенку, который еще или не умеет, или не очень любит читать и писать, осваивать текстовый язык программирования довольно сложно. Другое дело, когда робот непосредственно управляется с пульта, на котором нанесены значки-пиктограммы, обозначающие команды. Более того, пульт с пиктограммами понятен детям вне зависимости от того, носителем какого языка является ребенок. Можно сказать, что пульт и пультовое управление интернациональны и не требуют предварительного изучения иностранного языка [13].

2. Программирование с помощью пиктограмм в «ПиктоМире»

Не так давно учеными МГУ имени М. В. Ломоносова и Федерального научного центра Научно-исследовательский институт системных исследований Российской академии наук (ФГУ ФНЦ НИИСИ РАН) была придумана и разработана программная среда «ПиктоМир», в которой дошкольники, не обладающие навыками работы с текстами, получили возможность создавать алгоритмы. В «ПиктоМире» вместо традиционных текстовых команд используется набор пиктограмм, обозначающих действия или вопросы. Программа, управляющая виртуальными роботами, фактически собирается ребенком из пиктограмм на экране планшета.

Самый простой робот-исполнитель системы «ПиктоМир» — Вертун. У него всего четыре выполняемые команды, обозначенные соответствующими пиктограммами (рис. 1).

«Настоящность» робота объясняется детям заданиями для Вертуна, который ходит по клетчатому космосдому и ремонтирует (закрашивает) плиты-клетки, разрушенные при взлете космических кораблей. В «ПиктоМире» ученик видит поле Вертуна и самого робота в 2.5D-изображении (изометрия), команды-пиктограммы расположены в правой части экрана («на полке»). Ребенок составляет программу, выбирая нужные команды и помещая их в заготовку-шаблон. В примере задания [11] Вертун «смотрит» на стену. Задача Вертуна — закрасить шесть «испорченных» клеток. Решение из 12 команд приведено на рисунке 2.

Ученик также может выполнять задания в пультовом режиме. В этом случае команды не записываются в программу, а немедленно исполняются. Ученик видит результат каждой команды и выбирает следующую команду, точно зная, в какой ситуации она будет выполняться. При этом все выполненные команды автоматиче-



Рис. 1. Команды робота-исполнителя Вертуна

чески складываются в копилку — прозрачный «стакан», в котором фактически формируется программа. Когда задача решена (Вертун отремонтировал все, что нужно в задании) в пультовом режиме, команды из копилки можно перенести в основной шаблон программы.

Описанный режим становится важным переходным этапом между *непосредственным* и *программным* управлением. Непосредственное управление с пульта просто и понятно ребенку, но оно само по себе не формирует представлений о программировании как о предварительном планировании, не учит заранее думать о следующих действиях. «Чистое» программное управление, когда при составлении программы все следующие действия приходится держать в уме, оказывается слишком сложным. Копилка позволяет сделать этот переход плавным: управление ведется в непосредственном режиме, но, когда задача решена, команды из копилки можно перенести в программу и увидеть, как робот решает задачу, выполняя все команды одну за другой

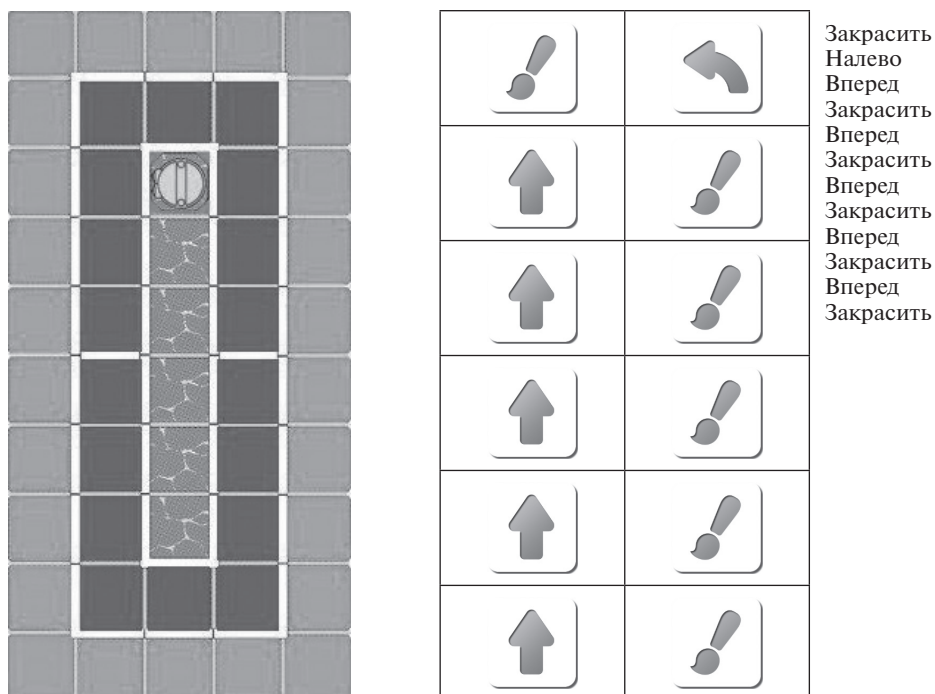


Рис. 2. Программа «ремонта» шести клеток

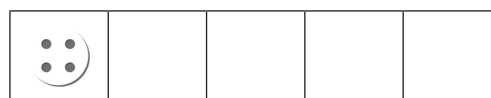
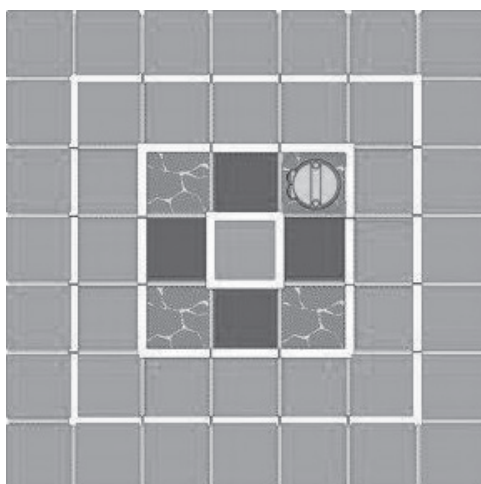


Рис. 3. Использование повторителя в цикле

без вмешательства человека. Так формируется первоначальное представление о программном управлении.

В дальнейшем работу с копилкой удобно использовать для разработки и отладки отдельных фрагментов сложных задач, а также для объяснения новых элементов, например, циклов с повторителем. Перед ребенком ставится задача, для решения которой вроде бы нужно просто последовательно выполнить некоторые команды. Полученная при этом программа решает поставленную задачу, но она содержит слишком много команд и не помещается в заданный шаблон, т. е. ее не удастся использовать в программном режиме. Тем самым у ученика формируются потребность и навыки использования циклов с повторением одних и тех же фрагментов программы вместо длинных линейных алгоритмов.

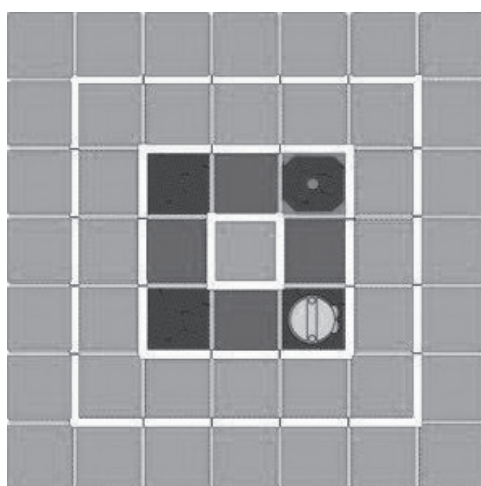
В примере на рисунке 3 роботу нужно закрасить четыре клетки, стоящие по углам квадрата 3×3 .

Чтобы решить задачу линейно в пультовом режиме, потребуется 13 команд (рис. 4).

Но в шаблоне программы предусмотрено всего четыре пустых поля для команд-пиктограмм. При этом в шаблоне есть повторитель, равный четырем, значит, четыре команды можно повторить четыре раза и общее количество команд окажется больше 13. Чтобы решить задачу таким способом, нужно выделить в решении повторяющиеся фрагменты. Можно подсказать ученику, что нужно найти в копилке повторяющиеся куски программы длиной не более четырех команд. Повторяющийся кусок легко обнаруживается (рис. 5).

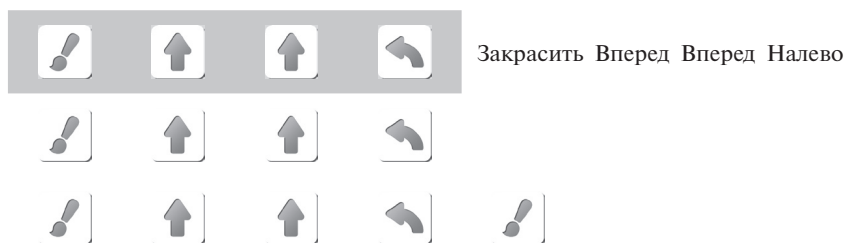
И программа готова! Более того, нет ничего страшного в том, что Вертун выполнит лишние три команды в конце: краткость и простота программы оказываются важнее экономии действий робота (рис. 6).

Робот Вертун не единственный исполнитель в «ПиктоМире». «ПиктоМир» включает набор программно-управляемых исполнителей, работа с которыми позволяет постепенно вводить такие важнейшие понятия



Закрасить
Вперед
Вперед
Налево
Закрасить
Вперед
Вперед
Налево
Закрасить
Вперед
Вперед
Налево
Закрасить

Рис. 4. Копилка



Закрасить Вперед Вперед Налево

Рис. 5. Поиск повторяющегося фрагмента

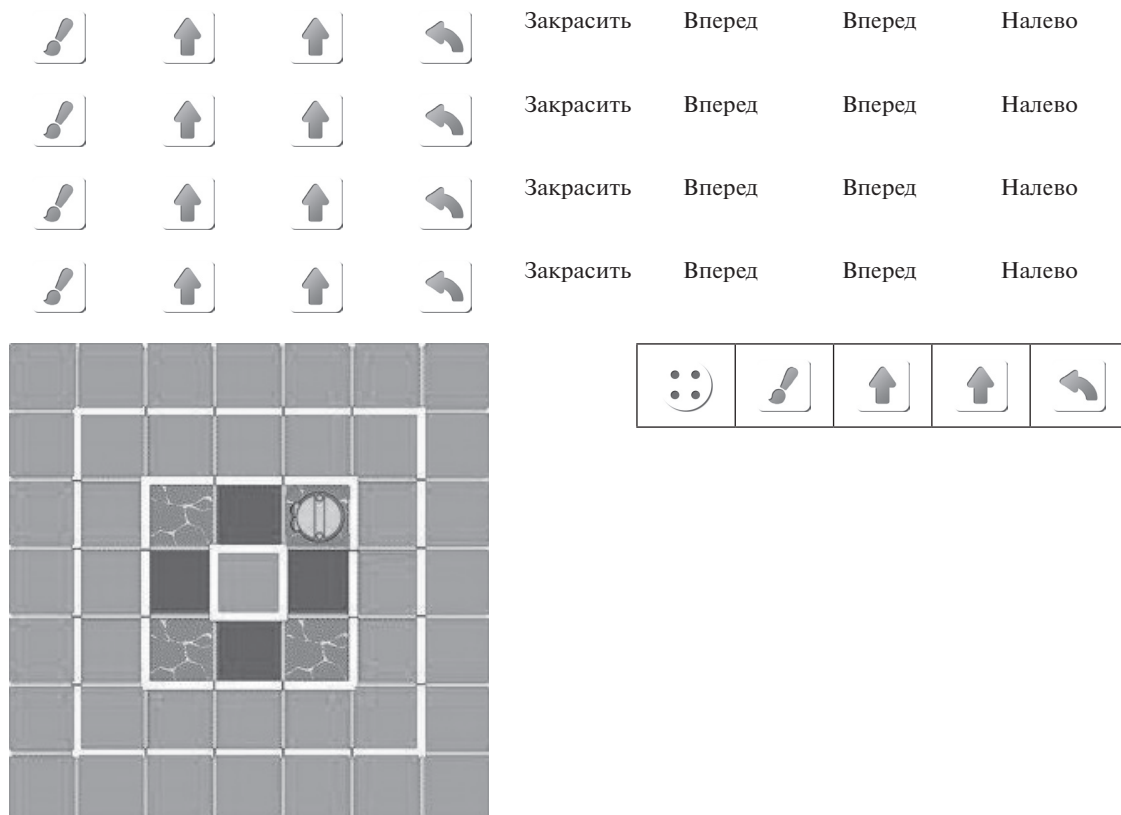


Рис. 6. Программа с повторителем

программирования, как циклы с повторителем, вспомогательные алгоритмы, условные операторы, циклы с условиями и др. Когда бы школьник ни начинал изучать основы алгоритмики, ему придется освоить эти понятия, научиться применять их и рассуждать о программе и алгоритмах, в том числе в повседневной жизни [14].

3. Исполнитель Кукарача в «Роботландии»

Другой известный исполнитель школьной информатики — Кукарача («Роботландия»), «живущий» на клетчатом поле. Кукарача — такой же простой исполнитель, как и робот Вертун. И система команд (предписаний) Кукарачи простая:

ВВЕРХ
ВНИЗ
ВЛЕВО
ВПРАВО

Любое невыполнимое действие Кукарачи приводит к отказу («Не могу!»). На поле исполнителя встречаются кубики с написанными на них символами. Эти кубики можно толкать перед собой и даже выбрасывать за пределы поля. Бывают кубики со знаком вопроса — толкая такой кубик, Кукарача переворачивает его и видит написанный символ. Это позволяет ставить и решать задачи с использованием условий (конструкций «пока» и «если»).

Типичная задача для Кукарачи — преобразование одного слова в другое. Например, ПОБЕДА «превратить» в БЕДА (# обозначает местонахождение Кукарачи):

П	О	Б	Е	Д	А
#					

Задача сводится к составлению алгоритма удаления лишних букв:

- выбросить П;
- подойти к О;
- выбросить О.

Программа очень проста, легко составляется детьми, освоившими кнопочное управление Вертуна и пиктограммное программирование в среде «ПиктоМир»:

ВВЕРХ
ВНИЗ
ВПРАВО
ВВЕРХ

На этом простом примере хорошо видно, что даже для такой короткой программы можно сначала составить обобщенный план действий, а уже потом реализовать его с помощью конкретных команд. Многие задачи Кукарачи имеют похожую структуру и позволяют составить понятный наглядный план, а уже потом переходить от него к составлению работающей программы. Так формируется представление о необходимости планирования, умение разделить задачу на части и реализовать эти части как независимые фрагменты программы.

Важное отличие Кукарачи от исполнителей «ПиктоМира» — программирование становится *текстовым*. Теперь ученик должен сам набирать команды и управляющие конструкции. Но простота самого исполнителя и решаемых задач помогает преодолеть эту сложность. Если у ребенка уже есть опыт бестекстового программирования и он понимает суть программно управления, переход к тексту совершается сравнительно легко.

Исполнитель Кукарача позволяет учителю выбирать для учеников с разным уровнем успеваемости различные модификации курса. Например, Кукарача позволяет реализовать рекурсию, и с сильными учениками можно рассмотреть соответствующие задачи даже в начальной школе [17].

4. Исполнители в «КуМире»

Освоив пропедевтические курсы, школьники с легкостью переходят к изучению более сложной системы «КуМир». В отличие от «ПиктоМира» и «Роботландии», в которых реализованы только базовые элементы программирования, а круг решаемых задач существенно ограничен возможностями исполнителей, в «КуМире» есть полноценный язык программирования, содержа-

щий все основные элементы классической императивной парадигмы и позволяющий решать вычислительные задачи, типичные для большинства курсов основ программирования.

Но даже в «КуМире» начинать обучение проще не с вычислительных алгоритмов, а с управления исполнителями.

Использование простых и наглядных исполнителей Робот и Чертежник во многом обеспечило успех обучения основам программирования в системе «КуМир» [12].

Робот в «КуМире», как и многие другие учебные исполнители, «живет» на клетчатом поле. Он умеет выполнять всего 17 команд. Простейшие команды, аналогично Вертуну и Кукараче, перемещают Робота по полю на соседнюю клетку. Это команды «вверх», «вниз», «вправо», «влево». По аналогии с Вертуном есть команда «закрасить». В некоторых случаях команды перемещения могут быть невыполнимы: например, если выше Робота находится стена, команду «вверх» выполнить невозможно. Простейшая задача для Робота состоит в переходе из некоторой клетки А в другую клетку Б, обходя препятствия и закрашивая заданные клетки [8] (рис. 7).

У Робота в «КуМире» есть пультовое управление (рис. 8), аналогичное «ПиктоМиру». Команды, набранные с пульта, автоматически записываются в протокол (аналог копилки в «ПиктоМире»). Ученик может скопировать протокол прямо в тело программы. Такой подход позволяет быстрее освоиться с исполнителем и языком программирования, облегчает переход от пиктографического программирования к текстовому. Кроме того, Робот в «КуМире» очень похож на Вертуна и Кукарачу. Многие содержательные алгоритмы управления Роботом уже были успешно освоены учениками в пропедевтических курсах и не должны вызвать осложнений.

Но Робот позволяет ставить и решать не только простые задачи. Вот пример задачи сравнительно сложной: «Робот находится в произвольной точке прямоугольного поля без внутренних стен. Одна клетка на поле закрашена. Необходимо найти закрашенную клетку и остановиться в ней».

Для решения этой задачи необходимо построить сравнительно сложный алгоритм, включающий несколько циклов «пока» и условий «если», при этом решение можно упростить, используя вспомогательные алгоритмы (рис. 9).

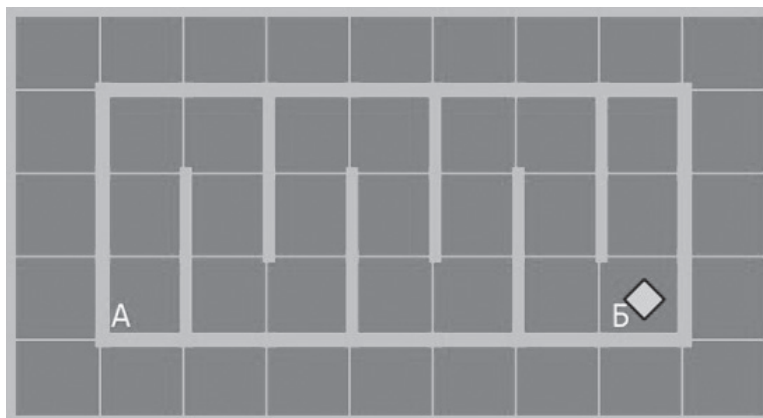


Рис. 7. Робот в «КуМире»



Рис. 8. Пульт в «КуМире»

В современном курсе «Алгоритмики», который читается в МПГУ для будущих педагогов начальной школы и преподавателей информатики, примерно треть занятий тратится на составление управляющих алгоритмов для исполнителя Робот в «КуМире», при этом охватываются практически все методы программирования, включая рекурсию.

Для получения компетенций на основе межпредметных связей (информатика + математика) и первоначальных навыков программирования с использованием величин (переменных) в системе «КуМир» есть исполнитель Чертежник (рис. 10). Он имеет перо и может пере-

мещать его по координатной плоскости, оставляя след в виде линии. Среди команд Чертежника есть основная команда рисования «сместиться на вектор (a, b) » (где a, b — произвольные числа), перемещающая Чертежника из точки с координатами (x, y) в точку с координатами $(x + a, y + b)$. При положительности a и (или) b , значение соответствующей координаты увеличивается, при отрицательных значениях — уменьшается. Например, команда «сместиться на вектор $(-10, 2)$ » переместит Чертежника из точки $(14, 4)$ в точку $(4, 6)$.

5. Заключение

В рамках пропедевтических и базовых курсов информатики и ИКТ [13] педагогу и ученику доступно также множество других программных исполнителей, использование которых не только упрощает процесс освоения дисциплины, но и формирует компетенции учащихся в других учебных предметах, таких как физика, математика и др., реализуя тем самым межпредметный потенциал информатики.

Системы управления исполнителями позволяют начать содержательное изучение программирования в раннем возрасте. Рассмотренные в статье системы «ПиктоМир», «Роботландия» и «КуМир» обладают определенной преемственностью и позволяют, переходя от одной системы к другой, осваивать все более сложные приемы и навыки программирования. При этом освоение более сложных систем не требует обязательного предварительного знакомства с более простыми — в зависимости от возрастной «точки старта» можно начинать с любой системы. Авторы статьи рекомендуют начинать с «ПиктоМира» при обучении дошкольников и учеников начальной школы, «Роботландия» может стать удачной стартовой системой для учащихся IV—V классов, «КуМир» со встроенными в него исполнителями — для учеников V—VII классов.

<pre> 2 алг поиск закрашенной клетки 3 дано Робот на прямоугольном поле без внутренних стен 4 надо одна клетка поля закрашена 5 надо клетка закрашена 6 нач 7 . в левый верхний угол 8 . нц пока клетка чистая 9 . . поиск в строке 10 . . если клетка чистая и снизу свободно 11 . . . то вниз 12 . . все 13 . кц 14 кон 15 16 алг поиск в строке 17 дано слева стена 18 надо клетка закрашена или слева стена 19 нач 20 . нц пока клетка чистая и справа свободно 21 . . вправо 22 . кц 23 . если клетка чистая 24 . . то к левой стене 25 . все 26 кон 27 28 алг в левый верхний угол 29 нач 30 . к левой стене 31 . к верхней стене 32 кон </pre>	
--	--

Рис. 9. Решение задачи — исполнитель Робот в «КуМире»

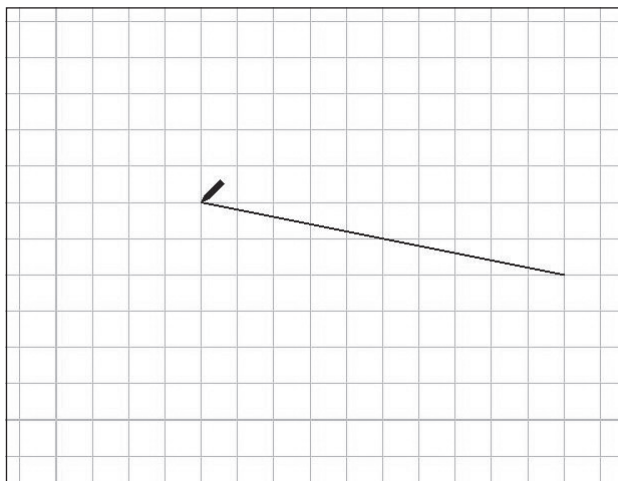


Рис. 10. Исполнитель Чертежник

Список использованных источников

1. Анеликова Л. А., Гусева О. Б. Программирование на алгоритмическом языке КуМир. М.: Солон-Пресс, 2011. 48 с.
2. Варсанюфьев Д. В., Кушниренко А. Г., Лебедев Г. В. Е-практикум — программное обеспечение школьного курса информатики и вычислительной техники // Микропроцессорные средства и системы. 1985. Т. 3. С. 27–32.
3. Ершов А. П. Алгоритмический язык // Квант. 1980. № 1. С. 42–45.
4. Ершов А. П. Программирование — вторая грамотность // Архив академика А. П. Ершова. <http://ershov.iis.nsk.su>
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. Звенигородский Г. А. Основные операторы учебно-производственного языка Рапира // Квант. 1985. № 10. С. 52–55.
7. Коган Е. Я., Первин Ю. А. Курс «Информационная культура» — региональный компонент школьного образования // Информатика и образование. 1995. № 1.
8. Кушниренко А. Г. Информатика. 7–9 классы. Рабочая программа: учебно-методическое пособие. М.: Дрофа, 2017. 88 с.
9. Кушниренко А. Г., Лебедев Г. В., Зайдельман Я. Н. Информатика. 7–9 классы: учебник для общеобразовательных учебных заведений. М.: Дрофа, 2000. 336 с.
10. Кушниренко А. Г., Лебедев Г. В., Сворень Р. А. Основы информатики и вычислительной техники: учебник для средних учебных заведений. М.: Просвещение, 1990–1996. 224 с.
11. Кушниренко А. Г., Леонов А. Г., Ройтберг М. А. Знакомим дошкольников и младших школьников с азами алгоритмики с помощью систем ПиктоМир и Кумир // Труды НИИСи РАН. 2017. Т. 5. № 1. С. 134–137.
12. Леонов А. Г. Обучение информатике: программные исполнители в системе КуМир // Труды большого московского семинара по методике раннего обучения информатике. 2008. Т. 1. С. 17–30.
13. Леонов А. Г., Первин Ю. А. Переход от непосредственного управления исполнителями к составлению программ в пропедевтическом курсе информатики // Ярославский педагогический вестник. 2013. Т. 3. № 3. С. 17–30.
14. Леонов А. Г., Первин Ю. А. Элементы программирования в непрерывном курсе школьной информатики // Ярославский педагогический вестник. 2013. Т. 3. № 1. С. 45–50.
15. Мирончик Е. А., Босова Л. Л., Куклина И. Д. Информатика. 5–6 классы: Изучаем алгоритмику. Мой КуМир. М.: БИНОМ. Лаборатория знаний, 2018.
16. Основы информатики и вычислительной техники: пробное учебное пособие для средних учебных заведений. В 2-х ч. / А. П. Ершов, В. М. Монахов, С. А. Бешенков, Я. Э. Голыц, А. А. Кузнецов, Э. И. Кузнецов, М. П. Лапчик, Д. О. Смекалин; под ред. А. П. Ершова, В. М. Монахова. Ч. 2. М.: Просвещение, 1986. 96 с.
17. Первин Ю. А., Горвиц Ю. М., Зарецкая З. А., Зарецкий Д. В. Модуль первого класса: Пособие для учителя. (Информационная культура). Самара: СИПКПРО, 1994. 206 с.
18. Первин Ю. А., Первин Т. Ю. Множества, элементы, признаки (книга для школьников 2-го класса). Самара: СИПКПРО, 1994. 113 с.
19. Поляков К. Ю., Еремин Е. А. Информатика. 7–9 классы: методическое пособие. М.: БИНОМ. Лаборатория знаний, 2016. 80 с.
20. Brusilovsky P., Calabrese E., Hvorecky J., Kouchnerko A., Miller P. Mini-languages: A way to learn programming principles // Education and Information Technologies. 1997. Vol. 2. No. 1. P. 65–83.
21. Clements D. H. The future of educational computing research: The case of computer programming // Information Technology in Childhood Education Annual. 1999. P. 147–179.
22. Lightbot. <http://lightbot.com>
23. Rogozhkina I. B., Kushnirenko A. G. PictoMir: Teaching programming concepts to preschoolers with a new tutorial environment // World Conference of Educational Technology and Researches. 2011. P. 216–220.