

И. С. Орещенков

Интернет-портал «Банки РБ» (banks.of.by), г. Молодечно, Беларусь

## РЕАЛИЗАЦИЯ УЧЕБНО-ИГРОВОЙ СРЕДЫ «БУКАШКА» СРЕДСТВАМИ КЛИЕНТСКИХ ВЕБ-ТЕХНОЛОГИЙ

### Аннотация

Тема «Алгоритмизация и программирование» — одна из самых сложных в школьном курсе информатики. Наглядно представить учащимся основные понятия этой темы помогают программы, реализующие модели учебных миров, в которых действуют виртуальные исполнители.

Наибольшей наглядности позволяют достичь среды графических исполнителей. Предельной простотой отличается среда «Кенгурёнок Ру», разработанная в начале 1990-х годов в рамках проекта «Пилотные школы» для выполнения на компьютерах IBM PS/2 под управлением операционной системы DOS.

Использование среды ГРИС «Кенгурёнок Ру» на современных компьютерах сопряжено с рядом трудностей. В статье описан вариант среды ГРИС, которая программно совместима с ГРИС «Кенгурёнок Ру», но реализована с помощью технологий HTML, CSS и JavaScript для обеспечения работы в веб-браузерах на широком спектре программно-аппаратных платформ.

**Ключевые слова:** информатика, веб-технологии, графический исполнитель.

**DOI:** 10.32517/2221-1993-2021-20-6-32-38

### 1. Введение

В быту современного человека получили широкое распространение разнообразные компьютеризированные электронно-механические устройства, автоматизирующие повседневные рутинные процессы (уборка, стирка, приготовление пищи), расширяющие возможности коммуникации, получения информации (мобильная текстовая, голосовая и видеосвязь, социальные сети и новостные ресурсы сети Интернет) и способствующие организации досуга (компьютерные игры, виртуальная реальность). Это привело к необходимости введения в программу начальных классов общеобразовательной школы пропедевтического учебного курса информатики и информационно-коммуникационных технологий для формирования у детей правильных представлений о принципах функционирования этих устройств и о предоставляемых ими возможностях.

Одной из фундаментальных основ, на которых базируется учебный курс информатики, является алго-

ритмизация. Федеральным государственным образовательным стандартом начального общего образования в рамках предметов «Математика» и «Информатика» декларируется цель овладения основами логического и алгоритмического мышления, формирования навыков действия в соответствии с алгоритмом и построения простейших алгоритмов [11].

В то же время тема «Алгоритмизация и программирование» является одной из самых сложных при изучении курса информатики, что в значительной степени обусловлено трудностью выбора средства демонстрации основных понятий этой темы. Профессиональные языки программирования и среды разработки требуют больших затрат времени и усилий для освоения специфических парадигм и приемов их использования, тогда как цель обучения в школе не подготовка высококвалифицированных программистов, а изучение способов, алгоритмов и методов программирования при решении задач, т. е. привитие учащимся навыков алгоритмического мышления [10].

### Контактная информация

Орещенков Игорь Сергеевич, инженер-программист, разработчик интернет-ресурса «Банки РБ», г. Молодечно, Беларусь; e-mail: iharsw@tut.by

I. S. Areshchankau,

"Banks of the Belarus" web site (banks.of.by), Molodechno, Belarus

### DEVELOPMENT OF THE LEARNING GAME ENVIRONMENT "BUKASHKA" ("LADYBUG") USING CLIENT WEB TECHNOLOGIES

#### Abstract

The theme "Algorithmization and programming" is one of the most difficult in the school informatics course. The computer programs that implement models of microworlds in which virtual actors operate help to explain the basic concepts of this theme for schoolchildren.

The best visualization can be achieved using learning environments with drawing actors. The learning environment "Kangaroo Ru" provides the most simple model. It was developed in the early 1990s as a part of an educational project for IBM PS/2 computers running the DOS operating system.

The usage of "Kangaroo Ru" is fraught with difficulties on modern computers. This article describes a case of the learning environment that is "Kangaroo Ru" compatible, but is implemented using web technologies HTML, CSS and JavaScript. It is runnable in web browsers of a wide range of hardware platforms.

**Keywords:** informatics, web technologies, virtual drawing actor.

Активный поиск эффективных методических средств обучения основам алгоритмизации происходил в период перехода от безмашинного преподавания к использованию появляющихся в школах учебных компьютерных классов. Наряду с непосредственным обучением языком программирования на примерах решения с их помощью математических задач осуществлялись попытки внедрения в учебный процесс программных продуктов, наглядно и в упрощенной форме демонстрирующих изучаемые концепции.

В странах Запада получила распространение среда программирования Logo, появившаяся в результате начатых в 1966 году многолетних исследований, проводимых ведущими научными центрами. В основу Logo легла идея «микромиров» — учебных моделей, в рамках которых можно программно управлять виртуальными сущностями [13].

Первый микромир Logo представлял собой средства описания логического вывода, позволяющие осуществлять интересные эксперименты на основе языка, близкого к естественному (английскому). Второй микромир реализовывал «черепашью графику» — средства управления пишущим узлом, с помощью которого можно было строить различные изображения. Третий микромир реализовывал модель ньютоновской механики в отношении «черепашки», которая управлялась в терминах физических воздействий. Четвертый микромир позволял использовать анимацию. Помимо этого расширенные средства Logo могли использоваться для программирования роботизированных игрушек LEGO, которые подключались к компьютеру.

В СССР в 1985–1986 годах на механико-математическом факультете МГУ при участии АН СССР был разработан комплекс программных средств поддержки школьного курса «Основы информатики и вычислительной техники» [5], в который вошли: «Альфа-практикум» — система для обучения начальным навыкам работы на ЭВМ, «Е-практикум» — система программирования на школьном алгоритмическом языке, «ЭВМ-практикум» — имитатор микропроцессора, позволяющий вводить, изменять и выполнять программы в машинных кодах.

Теоретические наработки в области использования учебных исполнителей [2] были воплощены в интегрированной программной среде «Школьница» [3]. Это направление получило развитие в программной системе «КуМир» (Комплект Учебных Миров), которая наряду со встроенным алгоритмическим языком позволяла использовать виртуальные программно управляемые сущности (Робот, Чертежник, Вездеход, Двуног, Строитель) для демонстрации различных аспектов алгоритмизации [6].

В начале 1990-х годов в сотрудничестве с корпорацией IBM реализовывался проект «Пилотные школы» [4], в рамках которого в советские учреждения образования поставлялись учебные классы на базе персональных компьютеров IBM PS/2. Программное обеспечение для этих классов было разработано в 1991 году АСП «Компьютерный учебно-демонстрационный издательский центр (КУДИЦ)», г. Москва, в виде «Программно-методических комплексов по курсу информатики» и включало наряду с гипертекстовыми средами и графическими редакторами учебную среду «Кенгурёнок Ру» [9], на которой хотелось бы остановиться подробнее.

## 2. Учебная среда «Кенгурёнок Ру»

Учебная среда «Кенгурёнок Ру» относится к категории графических исполнителей (ГРИС) — моделей, результатом деятельности которых является некоторый рисунок. Она представляет собой исполняемый файл ROO.EXE для операционной системы DOS, который можно выполнить в эмуляторе DosBox. При запуске появляется сообщение о программе и ее разработчике:

```
Roo Teaching Environment. Version 1.4
(C) Forware Systems, 1991. P.O.Box 221
Sverdlovsk 620063 USSR
```

После нажатия клавиши Enter на экране отображается основной интерфейс программы — строка главного меню, область редактора текста программы и рабочая область исполнителя (рис. 1). Исполнитель — забавный кенгуренок Крошка Ру (персонаж сказки Алана Милна о медвежонке Винни-Пухе), — выполняя непосредственные команды или действуя по программе, перемещается в прямоу-

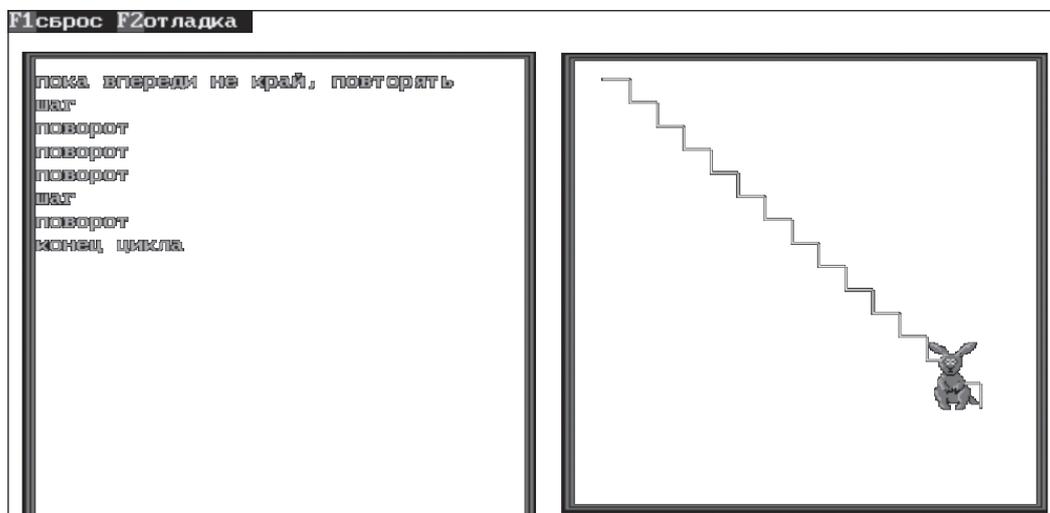


Рис. 1. Интерфейс учебной среды «Кенгурёнок Ру»:

строка главного меню (вверху), область редактора текста программы (слева) и рабочая область исполнителя (справа)

гольной рабочей области и рисует на ней изображение из отрезков вертикальных и горизонтальных линий.

Отличительной чертой среды «Кенгурёнок Ру» является ее исключительная простота. Разработчикам удалось очистить контекст исполнителя, оставив в нем лишь те понятия, о которых дети получают представление в ходе естественного развития в самом раннем возрасте.

Состояние исполнителя в любой момент времени определяется его положением на клетчатом рабочем поле (размером 15 клеток по горизонтали и 19 клеток по вертикали) и текущим направлением — одним из четырех возможных: вправо, вверх, влево, вниз. При этом явно не используется никакая система координат, исполнитель может узнать о своем состоянии лишь то, находится непосредственно перед ним край рабочего поля или нет. Исходное положение исполнителя — в левом верхнем углу рабочего поля, направление — вправо.

Исполнитель может перемещаться по сторонам клеток рабочего поля, закрашивая или не закрашивая их. Для этого он умеет выполнять три императивные команды:

- *прыжок* — перемещение в текущем направлении без рисования;
- *шаг* — перемещение в текущем направлении с рисованием;
- *поворот* — изменение текущего направления на четверть окружности против часовой стрелки.

Помимо этого в систему команд исполнителя (СКИ) входят:

- команды ветвления: *если впереди (не) край, то ... конец ветвления;*
- команды цикла: *пока впереди (не) край ... конец цикла;*
- команды определения подпрограмм: *процедура ИмяПроцедуры ... конец процедуры;*
- команда вызова подпрограмм: *сделай ИмяПроцедуры.*

В языке программирования исполнителя «Кенгурёнок Ру» напрочь отсутствуют математические объекты — нет даже чисел и переменных, не говоря уже о градусной мере направления или координатах действующего персонажа. Тем не менее этот крайне аскетичный набор выразительных средств позволяет наглядно продемонстрировать работу большинства алгоритмических конструкций структурного программирования и способ многократного выполнения участка программы с помощью рекурсии.

Если задействовать творческий подход и фантазию, в среде рассматриваемого исполнителя можно написать эффектные программы, например, для рисования греческого орнамента или других повторяющихся узоров. В 1990-х годах среди школьников проводились олимпиады по программированию «Кенгурёнка Ру», на которых предлагалось решить довольно сложные задачи [7].

Помимо удачного выбора СКИ среда «Кенгурёнок Ру» привлекает возможностью пошагового выполнения программы. В режиме пошагового выполнения можно одновременно наблюдать текущую команду программы (в области редактора) и красочно анимированное действие исполнителя в ответ на эту команду (на рабочем поле). Для экономии времени имеется возможность быстрого выполнения программы с немедленным отображением результата.

К сожалению, работа с учебной средой «Кенгурёнок Ру» под управлением современных операционных систем сопряжена с рядом трудностей. Прежде всего, это необходимость использования эмулятора операционной системы DOS. Кроме того, можно отметить, что способ управления средой с клавиатуры не отличается интуитивной понятностью, неочевидна смена режимов работы.

Имея в виду уникальные методические качества учебной среды «Кенгурёнок Ру», было принято решение воспроизвести ее на современной программной платформе, обращая внимание на сохранение достоинств и по возможности избавление от недостатков.

### 3. Воспроизведение среды «Кенгурёнок Ру»

При воспроизведении учебной среды «Кенгурёнок Ру» преследовались следующие цели:

1. Новая среда должна быть программно совместима со средой «Кенгурёнок Ру», т. е. программы, разработанные для «Кенгурёнка», должны без изменений выполняться в новой среде и приводить к идентичному результату.
2. Интерфейс новой среды должен воспроизводить положительные качества среды «Кенгурёнок Ру», однако в нем допускаются изменения, не ухудшающие эксплуатационные свойства.
3. Средства реализации новой среды должны обеспечивать ее использование на максимально широком спектре программно-аппаратных платформ.
4. Новая реализация должна исключать претензии в нарушении авторских прав третьих сторон.
5. В новой реализации должны быть учтены пожелания пользователей среды «Кенгурёнок Ру», касающиеся внесения улучшений.

Для достижения *первой цели* в новой среде сохранены СКИ, размеры рабочего поля и начальное положение исполнителя.

Для достижения *второй цели* явным образом обозначены режимы работы новой среды и реализовано управление интерфейсом как с клавиатуры, так и с помощью манипулятора «мышь» или сенсорного экрана.

Для реализации новой среды были выбраны средства клиентских веб-технологий — HTML, CSS и JavaScript [8]. Базовых возможностей перечисленных средств оказалось достаточно для воплощения всех замыслов, сторонние библиотеки не использовались. Язык JavaScript был задействован в пределах стандарта ECMA Script v.3 [12]. Рабочая область исполнителя была реализована с помощью элемента canvas, введенного стандартом HTML5 [1]. В результате новая среда функционирует в популярных веб-браузерах Chrome, Opera, Firefox на различных операционных системах и устройствах, что является достижением *третьей цели*.

Для достижения *четвертой цели* в новой среде в качестве исполнителя использован нейтральный персонаж — «Букашка», изображение которого, помимо прочих качеств, благодаря преобладающей симметрии упростило реализацию анимации.

*Пятая цель* достигнута реализацией счетчиков выполненных исполнителем действий, что позволяет

ввести в образовательный процесс элемент соревновательности, и обеспечением возможности использования в новой среде национальных языков.

В результате проделанной работы был создан **программный продукт, состоящий из следующих файлов, определяющих его структуру:**

- index.html — шаблон интерфейса учебной среды, консолидирующий все ее составные части и являющийся «точкой входа» разработанного веб-приложения (учебная среда запускается путем загрузки этого файла в веб-браузере);
- index.css — таблица стилей, определяющая визуальное оформление учебной среды;
- img/sprite.gif — набор графических изображений персонажа, действующего в среде, в различных фазах движения и состояниях;
- script/index.js — сжатый программный код учебной среды на языке программирования JavaScript;
- script/i18n.js — сжатый программный код модуля интернационализации, позволяющего использовать в учебной среде различные национальные языки. По умолчанию среда пытается установить язык, соответствующий языку интерфейса веб-браузера. Имеется возможность принудительно указать желаемый язык при запуске среды с помощью параметра lang. Например, строка запуска: `index.html?lang=en` приведет к использованию английского языка интерфейса;
- script/i18n\_ru.js, script/i18n\_en.js, script/i18n\_en-US — таблицы синтаксического дерева языка программирования учебной среды и текстовых сообщений на национальных языках. Чтобы добавить в учебную среду новый национальный язык, достаточно создать по имеющимся образцам файл `script/i18n_???.js` в кодировке UTF-8 (где ??? — обозначение национального языка в свойствах интерфейса веб-браузера) и подключить его к учебной среде, добавив в файл `index.html` строку вида: `<script src="script/i18n_???.js"></script>` непосредственно перед строкой `<script src="script/index.js"></script>`

**ГРИС «Букашка» может эксплуатироваться в локальном или сетевом варианте** (рис. 2).

**В локальном варианте** достаточно распаковать дистрибутивный архив на носитель целевого устройства и создать ярлык или закладку в веб-браузере, ведущую к стартовому файлу `index.html`.

**Сетевой вариант** требует организации веб-сервера, размещения на нем файлов ГРИС «Букашка», предоставления к серверу сетевого доступа с устройств учащихся и создания на них ярлыков с URL к ресурсу веб-сервера, соответствующего местоположению ГРИС.

Серверный вариант установки сложнее, однако при наличии на веб-сервере исполняющей системы PHP появляется возможность разработки HTML-страниц с примерами программ, которые могут быть быстро загружены в ГРИС без необходимости ручного ввода. Для этого в дистрибутивном пакете имеется вспомогательный PHP-сценарий `program.php`. Чтобы реализовать эту возможность, надо на HTML-странице разместить форму следующего вида:

```
<form method="post" action="https://iharsw.
login.by/gris/program.php">
<input type="hidden" name="lang" value="ru">
<h2>Рисование лестницы</h2>
<textarea name="program" readonly="readonly"
cols="40" rows="9">
пока впереди не край, повторять
шаг
поворот
поворот
поворот
шаг
поворот
конец цикла
</textarea><br>
<input type="submit" value="Выполнить">
</form>
```

Вместо «`https://iharsw.login.by/`» надо указать правильный сетевой путь к собственному HTTP-серверу, а в элементе `textarea` записать команды демонстрируемой программы. Этот фрагмент создаст на HTML-странице

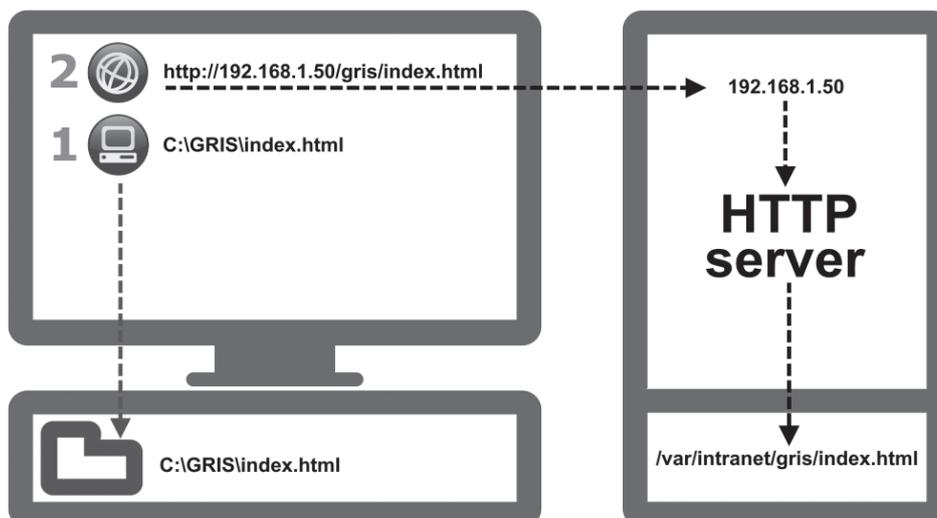


Рис. 2. Варианты установки ГРИС «Букашка»: 1 — локальный, 2 — сетевой.

форму с текстом программы и кнопкой «Выполнить», при нажатии на которую в веб-браузер будет загружено приложение ГРИС «Букашка» с готовой к выполнению программой (рис. 3).

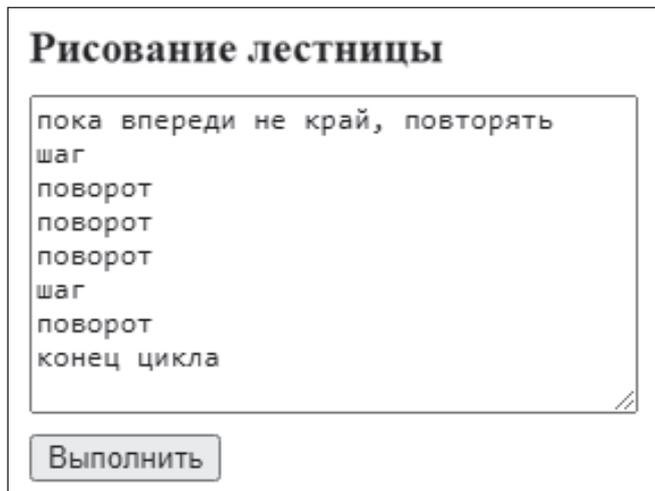


Рис. 3. Форма с текстом программы для быстрого выполнения в ГРИС «Букашка».

Указанным способом можно создавать веб-сайты с примерами программ для ГРИС «Букашка», иллюстрирующими рассматриваемый на занятии вопрос или демонстрирующими лучшие работы учащихся.

#### 4. Интерфейс ГРИС «Букашка»

Интерфейс разработанной учебной среды состоит из пяти основных элементов (рис. 4):

1. **Индикатор текущего режима работы** — информирует о том, в каком режиме находится учебная среда. Всего реализовано три режима:
  - 1) режим непосредственного выполнения команд;
  - 2) режим программирования исполнителя;
  - 3) режим выполнения программы.

В последнем режиме имеются подрежимы отладки (выполнения программы по шагам) и быстрого получения результата. Переключение между режимами работы — последовательное, однонаправленное, закольцованное: 1, 2, 3, 1, 2, ... .

2. **Строка главного меню учебной среды** — содержит перечень действий (пункты меню), допустимых для выполнения в текущем режиме. Пункты меню можно активировать щелчком мыши, прикосновением к сенсорному экрану или с помощью клавиатуры. Перед названием действия отображается обозначение клавиши, с помощью которой его можно активировать. Например, переход к очередному режиму работы может быть выполнен с помощью последнего пункта в строке главного меню, название которого зависит от целевого режима, но которому поставлена в соответствие одна и та же клавиша Tab.



Рис. 4. Элементы интерфейса ГРИС «Букашка».

3. **Область редактора текста программы** — прямоугольная область в левой части окна учебной среды, в которой размещается текст программы. Активно работать с текстом программы (создавать, вносить корректировки и удалять команды) можно только в режиме № 2 «Программирование», а в остальных режимах отображаемый текст недоступен для изменений.
4. **Рабочая область исполнителя** — прямоугольная область в правой части окна учебной среды, на которой действует исполнитель. Для удобства на ней отображается сетка, по которой происходит перемещение и рисование. Это позволяет визуально контролировать процесс построения изображения и отслеживать ситуации выхода за границы рабочей области.
5. **Область оповещения** — прямоугольная область в нижней части окна учебной среды, в которой отображаются подсказки, информационные сообщения и сообщения об ошибках.

## 5. Сценарий использования ГРИС «Букашка»

Обучение основам алгоритмизации с применением ГРИС «Букашка» может проводиться на занятиях, которые построены по следующему сценарию.

### Занятие 1. Вводное.

Учитель предлагает учащимся вспомнить, какие бытовые приборы работают по программе и какие элементарные действия они при этом выполняют.

*Пример 1.* Микроволновая печь: включить/выключить вращение столика, включить/выключить генератор излучения, включить/выключить нагрев гриля.

*Пример 2.* Стиральная машина: набрать/слить воду, включить/выключить подогрев воды, включить/выключить вращение барабана.

Декларируется, что такие бытовые приборы могут быть названы *исполнителями*, а элементарные действия — *системой команд исполнителя*. После этого можно познакомить учащихся с виртуальным исполнителем «Букашка» и его СКИ. Учащиеся могут попробовать нарисовать какое-нибудь изображение в режиме непосредственного выполнения команд, используя пункт меню *ESC сброс* для очистки рабочей области и возврата исполнителя в исходное положение в случае, если проведена неправильная линия или исполнитель вышел за границы рабочей области.

Необходимо обратить внимание учащихся на ограниченность СКИ и асимметричность команды «поворот»: для поворота исполнителя на четверть окружности против часовой стрелки требуется одно действие, а для поворота по часовой стрелке — три действия. Учитель поясняет, что такие ограничения часто встречаются при программировании реальных устройств.

### Занятие 2. Введение в программирование.

Учитель демонстрирует учащимся работу ГРИС «Букашка» в режиме программирования на примере линейной программы, которая строит простое изображение с помощью команд, которые уже знакомы учащимся из опыта их непосредственного выполнения

на предыдущем занятии. Объясняет разницу между СКИ (неупорядоченным множеством команд) и программой (упорядоченной последовательностью команд).

Учащиеся осваивают работу с редактором текста программы, способ быстрого ввода команд с помощью пунктов меню, очистку окна редактора с помощью пункта меню *ESC сброс*. Учитель объясняет, как перейти в режим выполнения программы и получить результат ее работы. Демонстрирует, как среда ГРИС реагирует на синтаксические ошибки в программе.

### Занятие 3. Алгоритмическая конструкция «следование».

Учитель объясняет, что участок программ, состоящий исключительно из императивных команд, которые всегда выполняются одна за другой в одном и том же порядке, является реализацией алгоритмической конструкции «следование».

Учащимся предлагается составить программу для рисования простого изображения, ввести ее в ГРИС и выполнить.

### Занятие 4. Алгоритмическая конструкция «подпрограмма».

Учитель приводит пример программы, в которой часто встречаются последовательности одних и тех же команд. Так, для поворота на четверть окружности направо исполнителю надо выполнить подряд три команды «поворот», каждая из которых приводит к повороту исполнителя на четверть окружности налево. Другой пример — последовательность команд, необходимых для рисования одной ступени в программе рисования лестницы. Педагог поясняет, что такие повторяющиеся последовательности команд можно «вынести за скобки» и оформить в виде подпрограмм, которые в ГРИС «Букашка» называются *процедурами*.

Учитель демонстрирует, как после определения процедуры исходные последовательности команд можно заменить одной командой вызова соответствующей процедуры. Учащиеся должны понять, что использование подпрограмм позволяет сократить размер программы и сделать ее более понятной.

### Занятие 5. Прием программирования «рекурсия».

Учитель предлагает учащимся написать программу, рисующую лестницу с помощью процедуры «ступень», разработанной на предыдущем занятии. Обращает внимание на сложность выполнения этого задания, если количество ступеней велико или заранее неизвестно.

После того как учащиеся осознают задачу, учитель рассказывает о *рекурсии* — возможности вызова подпрограммы из нее самой, что приведет к повторному выполнению содержащихся в ней команд. Составляется программа рисования лестницы из бесконечного количества ступеней с помощью рекурсии. Демонстрируется выполнение программы в обычном и пошаговом режимах. Обозначается проблема корректного завершения рекурсивной программы, но решение этой проблемы откладывается до следующего занятия.

### Занятие 6. Алгоритмическая конструкция «ветвление».

Учитель напоминает о необходимости решения проблемы корректного завершения рекурсивной про-

граммы и рассказывает о том, что существует возможность выполнения команд программы в зависимости от результата проверки некоторого условия, а не в раз и навсегда определенном порядке.

В программе рисования лестницы вызов подпрограммы «ступень» помещается в условную конструкцию «если впереди не край, то ... конец ветвления».

### Занятие 7. Алгоритмическая конструкция «цикл».

Учитель объясняет, что такое *стек*, как он используется для хранения контекста при вызовах процедуры и то, что использование рекурсии для повторного выполнения участка программы может быть дорогим решением. Сказанное демонстрируется многократным выполнением команды «конец процедуры», освобождающей стековую память после того, как исполнитель рекурсивной программы рисования лестницы достигнет края рабочей области. Учитель сообщает о возможности организации многократного выполнения группы команд с помощью циклической конструкции. Рекурсия в программе рисования лестницы заменяется блоком «пока впереди не край, повторять ... конец цикла».

### Занятие 8. Итоговое.

Учитель подводит итог изучения основ алгоритмизации, напоминает изученные алгоритмические конструкции и приемы программирования. Останавливается на понятии исполнителя и его СКИ, сообщает, что веб-браузер является исполнителем для построения ГРИС «Букашка», а в роли СКИ выступает язык программирования JavaScript. Ученики должны понять связь между изученной учебной моделью и моделью «настоящего» программирования.

## 6. Заключение

В результате многочисленных исследований было обнаружено, что наиболее эффективным методическим средством обучения основам алгоритмизации является использование учебных виртуальных миров с действующими в них исполнителями. Концепция учебных исполнителей дает возможность постигать предмет в упрощенном контексте, что позволяет снизить порог вхождения в предметную область и требования к имеющимся знаниям и жизненному опыту учащихся.

Наиболее простыми моделями исполнителей являются ГРИС, которые наглядно демонстрируют работу основных алгоритмических конструкций: следования, ветвления, цикла, процедуры. Предельной простотой отличается учебная среда «Кенгурёнок Ру», разработанная в начале 1990-х годов для поддержки проекта «Пилотные школы». Она может применяться в обучении детей младшего возраста, но ее использование на современных компьютерных устройствах сопряжено с рядом технических сложностей.

Среда «Кенгурёнок Ру» была воспроизведена с помощью клиентских веб-технологий в виде ГРИС «Букашка» и работает на большинстве веб-браузеров. При реализации новой среды удалось сохранить основные достоинства и уникальные методические качества среды «Кенгурёнок Ру», произвести адаптацию к современным программно-аппаратным платформам и внести некоторые улучшения.

Ознакомиться с работой веб-приложения ГРИС «Букашка» можно на сайте автора по ссылке:

<https://iharsw.login.by/gris/>

Дистрибутив веб-приложения ГРИС «Букашка» может быть получен с сайта «Учительский портал» по ссылке: <https://www.uchportal.ru/load/22-1-0-91492> Там же можно обсудить вопросы, связанные с его реализацией и использованием.

Исходные тексты веб-приложения распространяются по лицензии GPL v.3 и могут быть получены из репозитория GitHub:

<https://github.com/R0bur/drawing-player>

### Список использованных источников

1. Гоше Х. Д. HTML5. Для профессионалов. СПб.: Питер, 2013. 496 с.
2. Звенигородский Г. А. Первые уроки программирования. М.: Наука, 1985. 208 с.
3. Звенигородский Г. А., Глаголева Н. Г., Земцов П. А., Налимов Е. В., Цикоза В. А. Программная система «Школьная» и ее реализация на персональных ЭВМ // Микропроцессорные средства и системы. 1984. № 1. С. 50–55.
4. Киселев Б. Проект «Пилотные школы» // Информатика и образование. 1990. № 5. С. 124–125.
5. Куширенко А., Семёнов А. Программы для школьного компьютера // Информатика и образование. 1986. № 1. С. 72.
6. Куширенко А., Эникетов М. КуМир — новое семейство учебных программ // Информатика и образование. 1993. № 1. С. 21–25.
7. Паклина Е. П. Задачи повышенной сложности с решениями в среде графического исполнителя Кенгурёнок // Фестиваль педагогических идей «Открытый урок». <https://urok.1sept.ru/статьи/101993/>
8. Пьюривал С. Основы разработки веб-приложений. СПб.: Питер, 2015. 272 с.
9. Роберт И. В. Современные информационные технологии в образовании: дидактические проблемы; перспективы использования. М.: Школа-Пресс, 1994. 65 с.
10. Родыгин Е. Ф. Методические рекомендации обучения программированию в школе // Вестник Марийского государственного университета. 2011. № 7. С. 20–22.
11. Федеральный государственный образовательный стандарт начального общего образования. <https://fgos.ru/>
12. Флэнаган Д. JavaScript. Подробное руководство. СПб.: Символ-Плюс, 2004. 960 с.
13. Solomon C., Harvey B., Kahn K., Lieberman H., Miller M. L., Minsky M., Papert A., Silverman B. History of Logo // Proceedings of the ACM on Programming Languages. Vol. 4. Is. HOPL (June 2020). Article No.: 79. P. 1–66. DOI: 10.1145/3386329