

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ
федеральное государственное бюджетное образовательное учреждение высшего
образования
КРАСНОЯРСКИЙ ГОСУДАРСТВЕННЫЙ ПЕДАГОГИЧЕСКИЙ УНИВЕРСИТЕТ
им.В.П.АСТАФЬЕВА
(КГПУ им.В.П.Астафьева)

Институт/факультет

Институт математики, физики и информатики
(полное наименование института/факультета/филиала)

Выпускающая кафедра

Базовая кафедра информатики и
информационных технологий в образовании
(полное наименование кафедры)

Мелешко Диана Геннадьевна

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

Тема Обучение основам программирования в визуальном редакторе
Blockly Code Editor во внеурочной деятельности основной школы.

Направление подготовки

44.03.01 Педагогическое образование
(код и наименование направления)

Профиль

Информатика

(наименование профиля для бакалавриата)

ДОПУСКАЮ К ЗАЩИТЕ

Зав.кафедрой

д.п.н., профессор Пак Н.И.

(ученая степень, ученое звание, фамилия, инициалы)

(дата, подпись)

Руководитель

к.п.н., доцент базовой кафедры ИИТвО, Кулакова И.А.

(ученая степень, ученое звание, фамилия, инициалы)

Дата защиты _____

Обучающийся _____

(фамилия, инициалы)

(дата, подпись)

Оценка _____

(прописью)

Красноярск 2017

Содержание

Введение.....	2
ГЛАВА I. Теоретические аспекты обучения программированию в школе	5
1.1. Обучение основам программирования в основной школе.....	5
1.2. Визуальный редактор Blockly Code Editor.....	9
1.3. Сравнение визуального и текстового языка программирования (Blockly и Паскаль)	15
ГЛАВА II. Программа внеурочной деятельности по информатике для основной школы	31
2.1. Основные требования к организации внеурочной деятельности по ФГОС	32
2.2. Тематическое планирование и содержание программы внеурочной деятельности «Основы программирования в визуальном редакторе Blockly Code Editor»	35
Заключение	54
Список используемой литературы	56

Введение

В условиях стремительного вхождения в нашу повседневную жизнь информационных и коммуникационных технологий особую значимость приобретает подготовка подрастающего поколения в области информатики и ИКТ.

Одним из проблемных вопросов, требующих особого внимания в обучении информатике остается вопрос о системе обучения программированию. С одной стороны, активно развивающиеся информационные технологии «выталкивают» программирование. Но с другой стороны – ЕГЭ, ОГЭ, олимпиады, различные конкурсы предусматривают наличие у учащихся хороших навыков формализации различных задач, уверенного владения приемами программирования, что обуславливает проблематику данной работы.

Как заинтересовать учащихся программированием, как научить, понимать и решать задачи на начальном уровне изучения программирования остаются весьма актуальными вопросами.

В последнее время в России обратили внимание на перспективный язык программирования Blockly. Его популярность началась с западных стран, где в настоящее время реализуется множество онлайн-проектов обучения азам программирования школьников и дошкольников. Этот язык, отлично может быть встроен в российскую классно-урочную систему, поэтому есть надежда, что он будет востребован и в нашем образовании.

С декабря 2014 года в школах проходит всероссийская акция "Час Кода", которая направлена на формирование и поддержку интереса молодежи к изучению информатики и программирования, а также на повышение престижности ИТ-специальностей в глазах молодых людей.

По мнению разработчиков языка одна из целей создания языка: лишить программирование кастовости и закрытости, сделать его более естественным и доступным для как можно более широкого круга лиц.

Blockly является образовательным проектом для будущих программистов. После программирования в нем, обучающиеся будут лучше подготовлены к изучению обычных текстовых языков программирования.

Цель исследования: Разработка программы внеурочной деятельности по основам программирования в визуальном редакторе Blockly Code Editor для обучающихся основной школы.

Объект исследования: Процесс обучения программированию в основной школе.

Предмет исследования: Содержание и средства внеурочной деятельности по начальному обучению программированию.

В соответствии с поставленной целью были выделены следующие

Задачи:

1. Изучить теоретические аспекты обучения программированию в основной школе.
2. Рассмотреть особенности визуального редактора Blockly Code Editor.
3. Провести сравнительный анализ решения задач средствами Blockly Code Editor и Паскаль.
4. Проанализировать основные требования к организации внеурочной деятельности по ФГОС при планировании внеурочного курса.
5. Разработать тематическое планирование и содержание программы внеурочной деятельности по теме «Основы программирования в Blockly Code Editor»

Для решения поставленных задач использовались теоретические методы исследования: анализ, обобщение, сравнение, классификация

Работа состоит из введения, двух глав, пяти параграфов, заключения и списка литературы. Выполнена на 59 страницах, содержит 2 таблицы и 4 рисунка.

ГЛАВА I. Теоретические аспекты обучения программированию в школе

1.1. Обучение основам программирования в основной школе

Курс информатики основной школы является частью непрерывного курса информатики, который включает в себя также пропедевтический курс в начальной школе и обучение информатике в старших классах (на базовом или профильном уровне). К концу обучения начальной школы (в соответствии с ФГОС начального общего образования) обучающиеся должны обладать ИКТ — компетентностью, достаточной для дальнейшего обучения. В основной школе, начиная с 5-го класса, они закрепляют полученные технические навыки и развивают их в рамках применения при изучении всех предметов. Образовательное учреждение, исходя из конкретных условий, может начинать изучение курса информатики с 5 класса за счет часов школьного учебного плана, выстраивая непрерывный курс информатики в 5–9 классах, обеспечивая его преемственность с курсом информатики начальной школы.[7]

Курс информатики по ФГОС обязательно:

- в 5 – 6 классах, по усмотрению школы – от 17 до 34 учебных часов в год из расчета 0,5 - 1 учебных часов в неделю.
- в 7 – 8 классе, 34 учебных часов в год из расчета 1 учебный час в неделю;
- в 9 классе, 68 учебных часов в год из расчета 2 учебных часа в неделю;

Одной из дидактических задач образовательного учреждения является интеллектуальное развитие учащегося, важной составляющей которого является алгоритмическое мышление.

Учащиеся на начальной стадии обучения должны ознакомиться с различными языками программирования, понять необходимость изучения алгоритмического языка программирования. В начале изучения учащиеся должны ознакомиться с основными алгоритмическими конструкциями, командами языка, правилами описания объектов языка программирования, структурой программы и правилами написания. Учащимся необходимо сначала сформировать навыки написания простейших программ с использованием алгоритмических конструкций и основных объектов языка программирования, а затем перейти к изучению простых и далее более сложных методов программирования.

При построении обучения учащихся теме «Алгоритмизация и программирование» каждый учитель информатики сталкивается с огромным количеством вопросов: как построить изложение материала, какие использовать методические разработки, в какой форме проводить занятия, какие составить практические задания, какой материал использовать учащимся при изучении и другие. Все эти вопросы возникают из-за отсутствия четко и в полном объеме изложенных учебно-методических материалов для изучения данной темы.

Перед началом обучения учителю необходимо выбрать язык программирования с учетом интересов учащихся, их направленности и структуры образовательного процесса в школе. В последнее время уже наблюдается тенденция перехода обучения от алгоритмических языков к объектно-ориентированным языкам программирования, что показывает об изменении общего подхода к преподаванию программирования в школе. Так в учебниках Н. Д. Угриновича «Информатика и информационные технологии. Учебник для 10-11 классов» и «Информатика и ИКТ. Базовый курс: Учебник для 9 класса» изучается тема «Алгоритмизация и программирование» на основе объектно-ориентированного языка программирования Visual Basic.

По мнению многих авторов, самым простым в изучении алгоритмическим языком является Turbo Pascal, после которого легко перейти к изучению среды объектно-ориентированного языка программирования Delphi. Для более углубленного изучения может быть использована система PascalABC, в ней реализован диалект языка Паскаль, который позволяет решать довольно сложные задачи с использованием различных средств, таких, как модули и объекты. К системе также прилагается специально разработанный электронный задачник, используемый для практического обучения решению задач.

Каждый школьный учебник по информатике включает в себя различные разделы, связанные с изучением информационно-коммуникационных технологий и основ информатики.

В различных учебниках тема «Алгоритмизация и программирование» не рассмотрена достаточно полно, глубоко и доступно, и предполагается разное количество часов на изучение данной темы, а изложенный материал в научных пособиях часто не поддается изучению учащимся. Поэтому задача учителя в школе состоит в разработке такой методики, которая максимально упростит развитие способности программировать, что очень важно для большинства людей в современном техническом мире. Учителю приходится пользоваться личными разработками уроков, использовать ранее наработанный опыт и учебники, которые прямо или косвенно содержат материал для изучения выбранного языка программирования.

Из количества 102 часов в учебный год, для 8 – 9 классов отводится 19 часов на изучение темы «Алгоритмы и исполнители», причем подразумевается изучение формальных исполнителей алгоритмов. В данное количество часов не входят часы на изучение темы «Алгоритмизация и программирование». Предполагается, что учитель будет использовать язык программирования во время решения задач при изучении других тем.

Таким образом, объём часов на изучение темы «Алгоритмизация и программирование» не дает возможности в полной мере изучить данную тему в школьном курсе. Кроме того, изучение программирования в курсе «Информатика» является самым сложным периодом обучения в данной науке. Для детей 11-14 лет научно изложенные учебники редко привлекательны, только очень пытливые умы старательно разбирают непонятные слова, а затем уже и суть основного материала. Задачей учителя в школе стоит разработка такой методики, которая максимально упростит постижение способности программировать, что очень важно для большинства людей в современном техническом мире.

1.2. Визуальный редактор Blockly Code Editor

Blockly - визуальный язык программирования, созданный в 2012 году компанией Google. Разработчики: Нил Фрейзер, при участии Эллен Спертус и Марка Фридмана. Нил Фрейзер называет Blockly редактором визуального программирования.

Редактор позволяет создавать программы вообще без ввода каких-либо символов или текста. Разработка осуществляется из широкого набора типовых лего-блоков, логическая композиция и соединение которых позволяет реализовывать заданную алгоритмическую функциональность программы. Интерфейс редактора представлен на рисунке 1. Технически этот визуальный язык реализован на Java Script и позволяет составлять такие визуальные программы прямо в своем браузере, просто перетаскивая и komponуя в логические цепочки функциональные блоки, после чего такая программа может быть скомпилирована в более традиционный целевой язык, такой как JavaScript, Dart или Python. Google Blockly – это Open Source проект, который выпущен под лицензией Apache License 2.0. Чтобы подчеркнуть то, что пока это только «демонстрация концепции и не более того», данный релиз не распространяется в традиционном виде отдельного файла (тарболла) с исходниками, а доступен лишь через синхронизацию с SVN - репозиторием разработчиков.

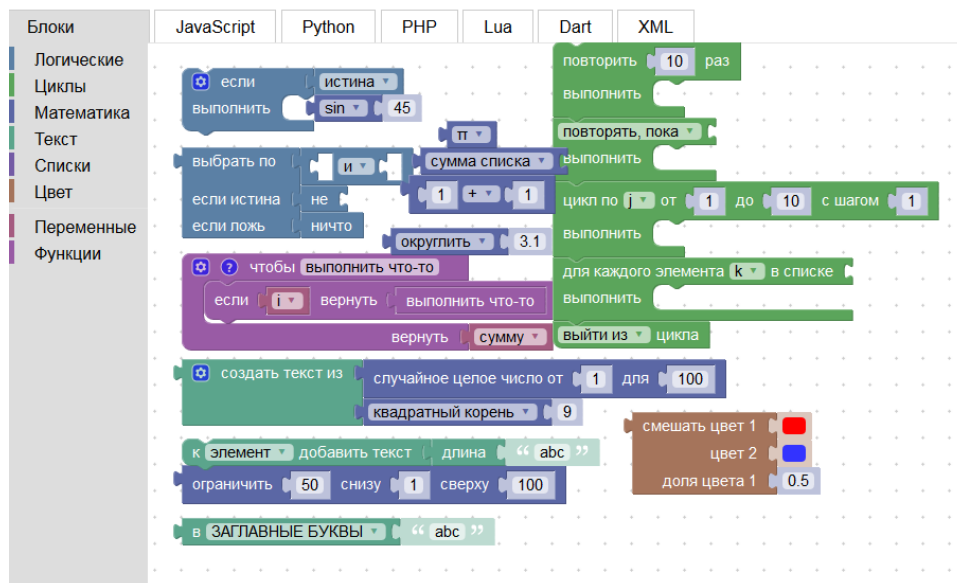


Рис. 1. Веб – интерфейс Blockly

Blockly является образовательным проектом для будущих программистов. На официальном сайте представлена серия заданий, обучающих основам программирования.[20] Данные задания могут быть использованы как в образовательных учреждениях, так и для самостоятельного обучения. Они предназначены для тех, кто не имеет опыта программирования. После выполнения всех заданий, обучающиеся будут лучше подготовлены к изучению обычных текстовых языков программирования.

Этот язык, по виду похожий на Scratch, буквально сразу после своего появления стал популярен в образовании. Он начал использоваться для обучения основам программирования школьников, и даже студентов. Популярность обусловлена рядом его преимуществ над Scratch и другими подобными языками: бесплатная и свободная лицензия, простой и понятный web - интерфейс, адаптирован для работы на планшетах, имеется возможность кросс - компиляции в другие языки. Популярность Blockly началась с западных стран, где в настоящее время реализуется множество онлайн-проектов обучения азам программирования школьников и

дошкольников. В последнее время и в России обратили внимание на этот бесспорно перспективный язык. Например, в декабре 2014 года в школах с большим успехом прошла всероссийская акция "Час Кода". Акция направлена на формирование и поддержку интереса молодежи к изучению информатики и программирования, а также на повышение престижности ИТ-специальностей в глазах молодых людей. Кроме всего прочего, этот язык, в отличие от Scratch, отлично встраивается в российскую классно-урочную систему, поэтому есть надежда, что он будет востребован и в российском образовании. Возможно, со временем он заменит в школах систему программирования Кумир.

Главные особенности Blockly

Blockly заставляет школьников учиться самостоятельно, играя, общаясь и обмениваясь кодом. Сообщество насчитывает несколько десятков тысяч школьников, которые создают свои игры, мультики, интерактивные истории, модели, ставят эксперименты и просто играют в игры, созданные ровесниками и исследуют код чужих проектов для того, чтобы самим сделать также или лучше.

Blockly увлекает и требует свободы. На первый взгляд Blockly очень похож на Scratch – те же визуальные блоки, тот же принцип соединения.

Однако это только на первый взгляд. Приглядевшись, сразу виден аскетичный интерфейс Blockly.

Blockly полноценно можно использовать в обучении школьников, среда программирования позволяет создавать учебные задания, компоновать их в фактически готовые занятия, проходя которые, школьник постепенно осваивает какую-то тему.

На странице Blockly в сети Google+ появляются все новые и новые задания. Сейчас доступны несколько таких заданий, специально разработанных авторами для обучения:

- Головоломка
- Лабиринт
- Черепашка
- Графический калькулятор
- Код
- Калькулятор посадочных мест в самолёте
- Фабрика блоков

Все перечисленное дает основания надеяться, что Blockly будет идеальным вариантом для использования, на уроках, когда школьники смогут получать серии заданий, подобных описанным, и “проходить их” в индивидуальном ритме.

Все работает в браузере, устанавливать специальное программное обеспечение на компьютер не надо, все бесплатно и русифицировано.

Blockly можно использовать именно в начале, причем независимо от того, как дальше будет строиться программа по информатике – будет ли упор на программирование или нет.

Разработчики пытались создать продукт для очень серьезных целей, и для этого внесли в него возможность кросс - компиляции в традиционные языки программирования, это создает условия для реализации самых разных подходов по дальнейшему развитию проекта. [31] Например, разработчики могут использовать Blockly для быстрого создания отдельных частей

больших и сложных программ, которые затем, после компиляции, например, в Python, могут дорабатываться уже на Python'e , или даже попросту вставляться в уже готовый проект в целях его расширения. Такое сочетание простоты и сложности вполне может быть очень интересно для обучения школьников.

Обучение школьников с помощью Blockly

Чтобы подчеркнуть новизну этого языка, его создатели выделяют в нём два принципиальных момента, подсвечивающих его на фоне множества других VPL - языков.

Во-первых, это сильный акцент на визуальной составляющей разработки; по аналогии с высокоуровневыми языками предложено называть подобный тип языков «сильно-визуальными» (все лего - подобные языки). Такой подход исключительно востребован для новых сенсорных устройств, где ввод традиционного текста затруднён, кроме того, такая форма программирования, как показывает опыт, имеет гораздо более легкий порог вхождения, в частности программы на Google Blockly составляют даже дети. С учетом самых серьёзных намерений по его развитию со стороны Google этот язык имеет очень интересные перспективы на будущее.

Второй принципиальный момент в нем – это возможность компиляции на более традиционный язык программирования на выбор, что создает условия для реализации самых разных подходов по дальнейшему программированию. Например возможности расширения Gmail для создания фильтрации своих писем на Google Blockly. Google считает, что это типовой случай в современном программировании, например при написании некоего расширения для MS Office, когда 90% этого расширения – это некая «полезная логика», а 10% – просто код сопряжения этого расширения с API хост - программы. Google считает, что в таком случае очень удобно быстро «собрать» собственно прикладную логику приложения на Google Blockly, а

весь более низкоуровневый код работы с API дописать уже на Python или Dart.[3]

При этом текущее временное ограничение языка такое: он не предназначен для создания больших приложений. Пока область его применения – небольшие сервисные скрипты, содержащие сложную логику и многократно вложенные логические и циклические конструкции, которые, по мнению специалистов Google, реально проще составлять именно на сверхнаглядном Blockly. Отдельно подчеркивается, что в следующих версиях этого визуального языка планируется существенно расширить его семантику, что по заверениям Google сделает его возможности сопоставимыми с традиционными языками программирования. Отсюда важный вывод и отличие от других VPL'ов: не следует рассматривать этот язык лишь как обучающий – в будущем планируется создание полноценного мультиплатформенного языка программирования на его базе. Именно по причине будущего расширения этого языка принято решение не использовать подход упаковки логики в уже распространенные условно-символические программные схемы (data - flow metaphor languages), так как эксперименты показали, что такой устаревший способ представления тяжело поддается расширению и применим исключительно к узкоспециализированным областям программирования.

1.3. Сравнение визуального и текстового языка программирования (Blockly и Паскаль)

Некоторые авторы считают, что школьники должны начинать с визуальных языков программирования, таких как Скретч (Scratch), а затем переходить к текстовым, например, Паскаль.

Не важно, визуальный язык программирования или текстовый выбирает учитель для обучения, важно следующее:

- позволяет ли обучающемуся определенный язык добиться того, что он хочет, и так, чтобы это было одновременно эффективно и понятно;
- соответствует ли язык программирования уровню знаний обучающегося, опыту и способу мышления.[26]

Одна из особенностей текстовых языков – то, что они по своей природе линейны. Текстовые языки не очень хорошо показывают связи между различными частями кода.

Визуальные языки обычно поддерживают нелинейную структуру. Современное программирование, особенно разработка игр, аппаратное обеспечение и Интернет вещей, зачастую работают на основе событий и выполняются одновременно. Код – это не книга, он не выполняется последовательно от начала до конца.

В Blockly можно организовать код, чтобы переместить спрайт следующим образом:

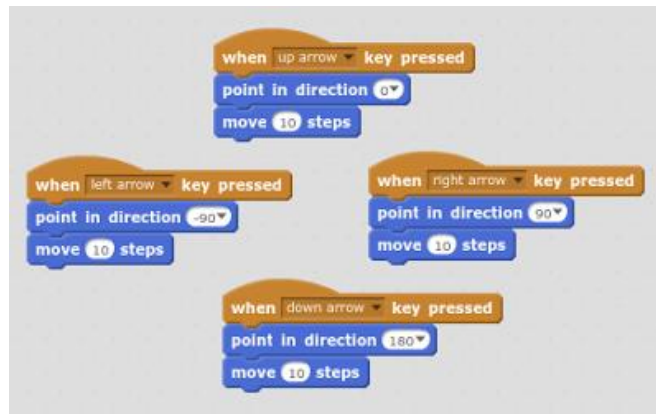


Рис. 2. Код перемещения спрайта

Это ведет непосредственно к фактическому движению спрайта. В Паскале понадобилось бы поместить команды в произвольном порядке, и можно потерять дополнительную информацию.

Для Blockly параллелизм – это естественная концепция.

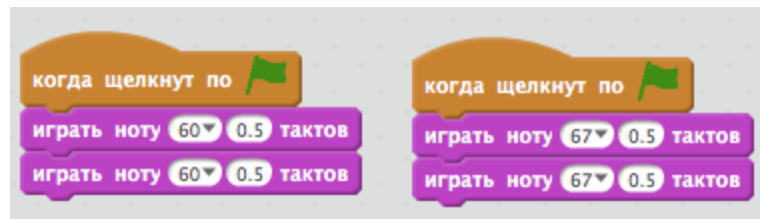


Рис. 3. Параллелизм Blockly

Однако, Blockly по-прежнему в основном линейный. Может быть трудно понять модели поведения, которые зависят от одного передающего сообщение спрайта, и нескольких других, которые его получают.

Wonder Workshop использовал другой подход к языку Wonder для роботов Dash&Dot. В нем используется многопоточность. Поток обычно применяют для описания бизнес-процессов точным (и иногда исполняемым) способом. Блоки состояния часто используются в системах управления. Это полезное средство для описания поведения систем, которые ведут себя по-разному в зависимости от текущего состояния.

Очень интересен подход к программированию в Minecraft, там используют смесь текстовых команд и визуальных блоков в 3D-пространстве. Код фактически является частью мира, а ведь для многих Minecraft столь же реален, как и настоящий мир вокруг.

Паскаль - это один из самых известных языков программирования. Изучают его, как правило, в старших классах или же на первых курсах вузов. Считается базой для многих других языков программирования

Важнейшим шагом в истории языка Паскаль является выход свободных реализаций. Они включили в себя все лучшее от многих других диалектов. В них присутствует отличная совместимость с большинством операционных систем. Знакомство с языком Паскаль необходимо начать с его особенностей. Главная из них - строжайшая типизация и присутствие структурного программирования. Паскаль - первейших из подобного рода языков. Как считал разработчик, язык должен поспособствовать дисциплинированному программированию, так как в нем максимально исключены различные синтаксические ошибки, а также присутствует интуитивно понятный интерфейс.[14]

Рассмотрим типовые учебные задачи, реализуемые на Паскале и Блокли.

Несколько типовых задач на Паскаль:

```
{ Составить программу проверки есть ли в тексте буква "s" }  
program z1;  
uses crt;  
var t : string; i : integer; ot :boolean;  
begin clrscr;  
writeln('введите текст:');readln(t);  
for i:=1 to length(t) do  
if t[i]='s' then ot:=true;  
if ot=true then write('da')
```

```

else write('нет');
readln;
end.

```

{ Определить лежит ли точка а на прямой $y=kx+l$ }

```

uses crt;
program z2;
var x,y,l,k : integer;
begin clrscr;
write('x=');readln(x);
write('y=');readln(y);
write('k=');readln(k);
write('l=');readln(l);
if y=k*x+l then write('Да')
else write('Нет');
readln;
end.

```

{ Даны координаты вершин треугольника ABC A(x1;y1), B(x2;y2), C(x3;y3) яв-ся ли треугольник равнобедренным }

```

program z3;
uses crt;
var x1,x2,x3,y1,y2,y3,a,b,c : real;
begin clrscr;
write('x1=');readln(x1); write('y1=');readln(y1);
write('x2=');readln(x2); write('y2=');readln(y2);
write('x3=');readln(x3); write('y3=');readln(y3);
a:=sqrt(sqr(x1-x2)+sqr(y1-y2));
b:=sqrt(sqr(x2-x3)+sqr(y2-y3));
c:=sqrt(sqr(x1-x3)+sqr(y1-y3));
if (a=b)or(a=c)or(b=c) then write('равнобедренный')
else write('неравнобедренный');

```

```
readln;
```

```
end.
```

{ Составить программу для определения лежит ли точка (x3;y3), на прямой проходящей через точки (x1;y1),C(x2;y2) }

```
program z4;
```

```
uses crt;
```

```
var x1,x2,x3,y1,y2,y3 : real;
```

```
begin clrscr;
```

```
write('x1=');readln(x1);
```

```
write('y1=');readln(y1);
```

```
write('x2=');readln(x2);
```

```
write('y2=');readln(y2);
```

```
write('x3=');readln(x3);
```

```
write('y3=');readln(y3);
```

```
if (x3-x1)*(y2-y1)-(y3-y1)*(x2-x1)=0
```

```
then write('лежит')
```

```
else write('нележит');
```

```
readln;
```

```
end.
```

Blockly — визуальный язык программирования с веб-интерфейсом.

Создание программы осуществляется путём соединения блоков.

Сама Google предлагает типичную задачу для событийно-ориентированного семейства языков – прохождении лабиринта, для решения которой уже развернута соответствующая рабочая web - среда. На рисунке 4 приведен листинг варианта решения прохождения лабиринта.



Рис. 4. Листинг прохождения лабиринта

Приведем сравнение решения одинаковых задач на двух языках программирования. Представление кода программы позволяет наглядно увидеть соответствие операторов языка Паскаль и блоков языка Блокли. Например, оператору ввода `read` соответствует блок «запросить ... с подсказкой», оператору вывода `write` соответствует блок «напечатать», более полное сравнение операторов и блоков представлено в таблице 1 в конце параграфа.

Задача 1: Вывести на экран сообщение «Hello World!».

Решение: Эта задача включает в себя лишь демонстрацию использования оператора вывода `write` (или `writeln`), который будет единственным в теле программы.

Паскаль:

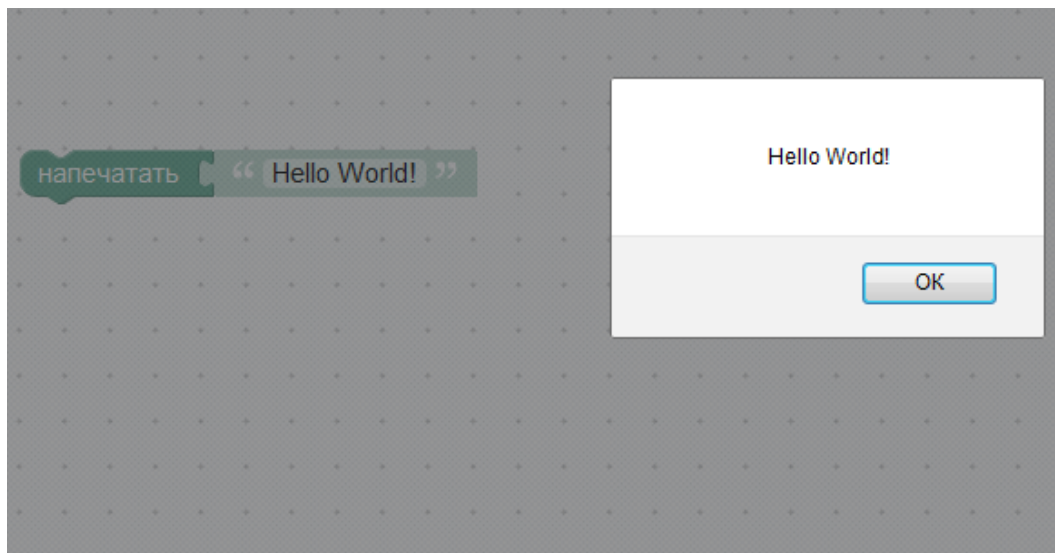
Program HelloWorld;

begin

writeln('Hello World!')

end.

Blockly:



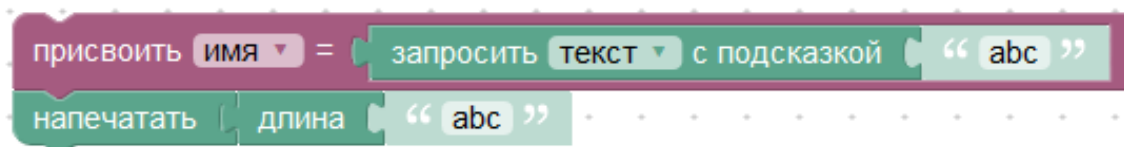
Задача 2: Составить программу, которая определяет количество СИМВОЛОВ В СЛОВЕ.

Решение: В данной программе используется оператор *length*, который определяет размер строки.

Паскаль:

```
var a:string;  
begin  
  writeln('введите строку');  
  readln(a);  
  writeln(length(a));  
  readln;  
end.
```

Blockly:



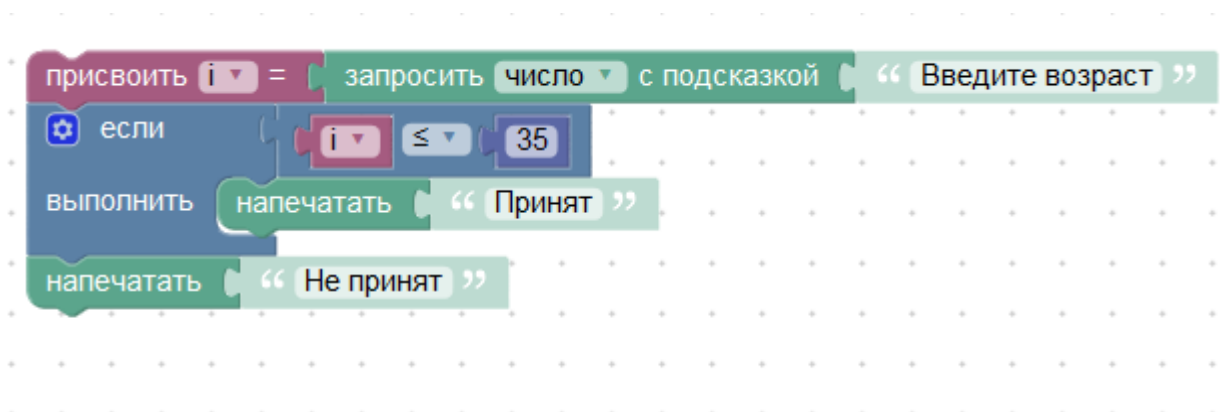
Задача 3: На работу в совместное предприятие приглашаются специалисты, возраст которых не превышает 35 лет. Определи, будет ли принят на работу человек, возраст которого N лет?

Решение: Для решения задач, в которых проверяется, какое либо условие требуется использовать алгоритм ветвление. В данной задаче есть условие, и оно звучит, что возраст человека не должен превышать 35 лет, запишем это условие $N \leq 35$. Следовательно: мы должны введенное число проверять на это условие и выводить.

Паскаль:

```
var N:integer;  
  
begin  
write('Введите возраст человека');  
readln(N);  
if N<=35 then writeln('Принят')  
else writeln('Не принят');  
readln;  
end.
```

Blockly:



Задача 4: Даны два числа. Вывести на экран то из них, которое больше.

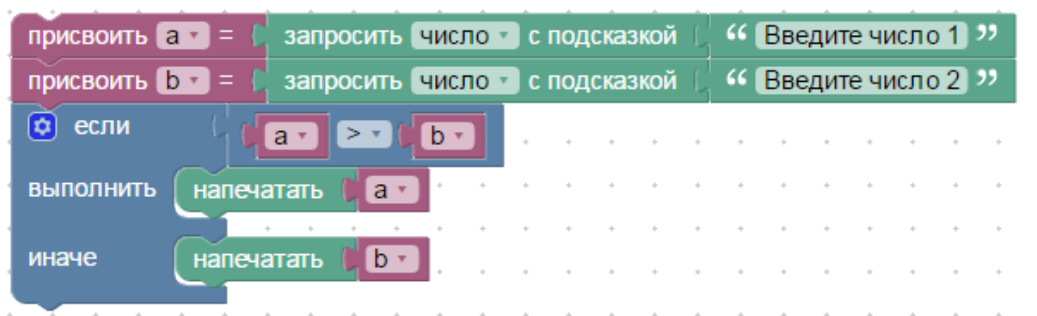
Решение: Собственно, это самая простая задача, с помощью которой можно продемонстрировать использование условного оператора if. Мы

вводим с клавиатуры числа в переменные a и b , затем в операторе `if` проверяем булевское выражение « $a > b$ »: если оно истинно, то выполняется `then`-блок оператора, если ложно – `else`-блок. Соответственно, если a больше b (условие в заголовке истинно), то в `then`-блоке мы выводим a , а если a не больше b (условие в заголовке ложно), то выводим b (хотя сюда попадает и случай, когда $a = b$, что, впрочем, не нарушает решения).

Паскаль:

```
var
  a, b: integer;
begin
  readln(a, b);
  if a > b then writeln(a) else writeln(b)
  readln;
end.
```

Blockly:



Задача 5: Программа, запрашивает число и выводит его квадратный корень.

Решение: В этой программе используется условный оператор `if` и цикл `repeat until`. Запросы продолжаются до тех пор, пока пользователь не введет положительное число.

Паскаль:

```
var x:integer;
begin
```

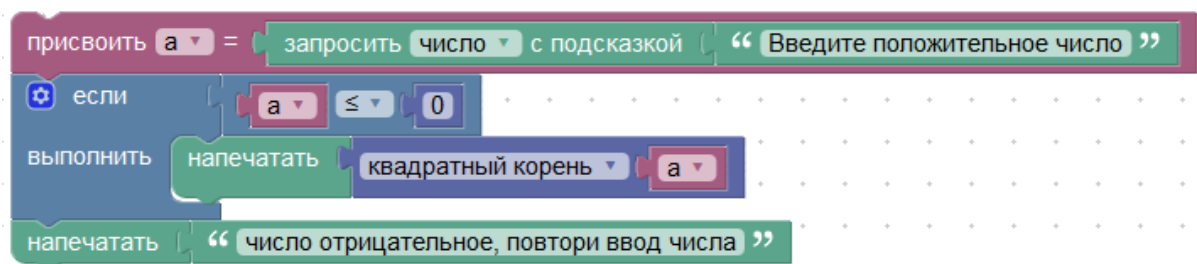


```

repeat
writeln('Введите положительное число: ');
readln(x);
if x >= 0 then
writeln ('Корень квадратный из ',x,' = ',sqrt(x)) else writeln ('число
отрицательное, повтори ввод числа');
until x > 0;
end.

```

Blockly:



Задача 6: Программа, которая вычисляет произведение двух чисел.

Решение: В этой программе вводим три переменных, в этом нам поможет оператор присваивания *writeln*, его функция - это присваивания значения переменной после каких либо действий или значения другой переменной.

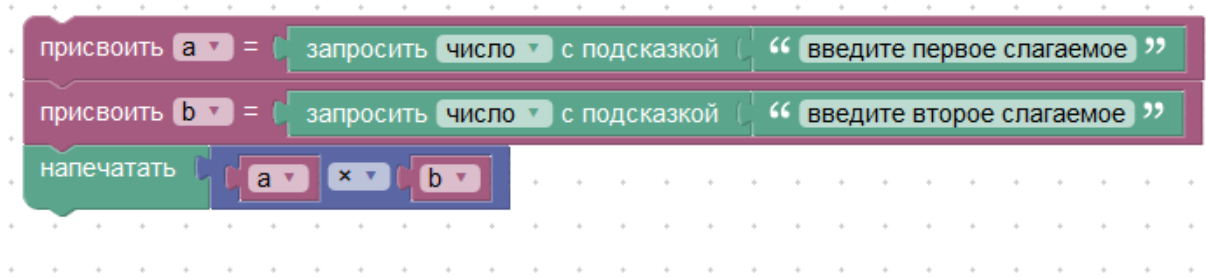
Паскаль:

```

var a, b, c:integer;
begin
writeln ('Введите первое число');
readln (a);
writeln ('Введите второе число');
readln (b);
c:=a*b;
writeln ('Произведение равно ',c);
end.

```

Blockly:



Задача 7: Найти произведение целых положительных нечетных чисел, меньших 200 и кратных 5.

Паскаль:

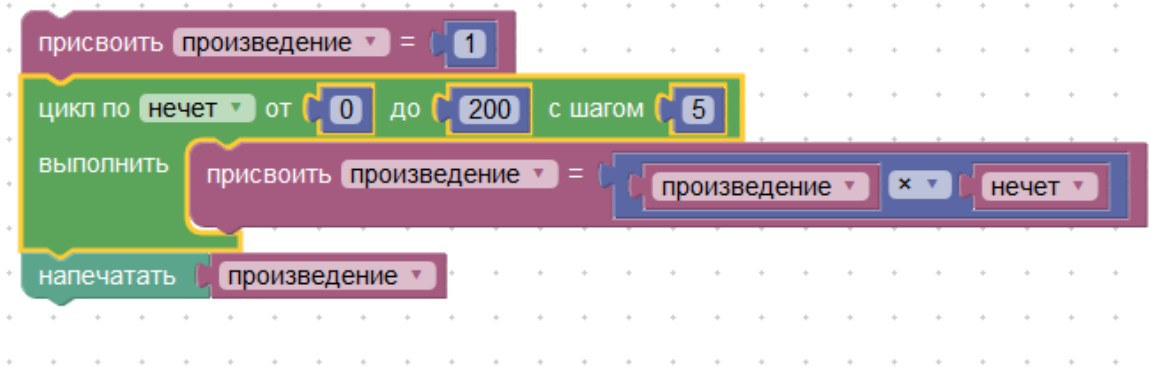
1 вариант решения

```
uses crt;  
var a,s:integer;  
begin  
clrscr;  
a:=5;  
s:=1;  
while a<200 do  
begin  
a:=a+5;  
s:=s*a;  
end;  
writeln('S=',s);  
readln  
end.
```

2 вариант решения

```
uses crt;  
var a,s:integer;  
begin  
clrscr;  
a:=5;  
s:=1;  
repeat  
a:=a+5;  
s:=s+a;  
until a=200;  
writeln('S=',s);  
readln  
end.
```

Blockly:



Задача 8: Написать программу нахождения суммы первых N четных натуральных чисел

Паскаль:

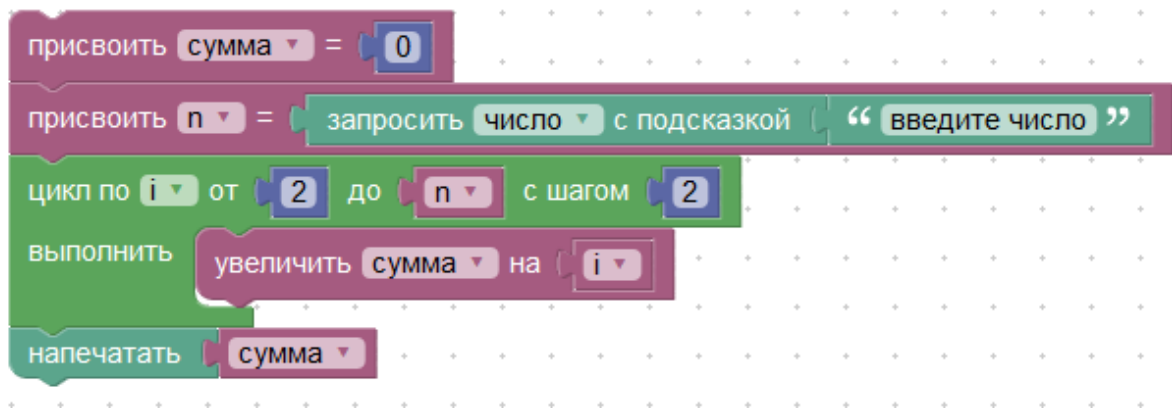
1 вариант решения

```
uses crt;  
var i,n,sum:integer;  
begin clrscr;  
writeln("Enter n");  
readln(n)  
sum:=0;  
i:=2;  
while i<=n do  
begin  
sum:=sum+i; i:=i+2;  
end;  
writeln("result=",sum);  
readln;  
end.
```

Blockly:

2 вариант решения

```
uses crt;  
var i,n,sum:integer;  
begin clrscr;  
writeln("Enter n");  
readln(n)  
sum:=0;  
for i:=2 to n do  
If I mod 2=0 then sum:=sum+i;  
writeln("result=",sum);  
readln;  
end.
```



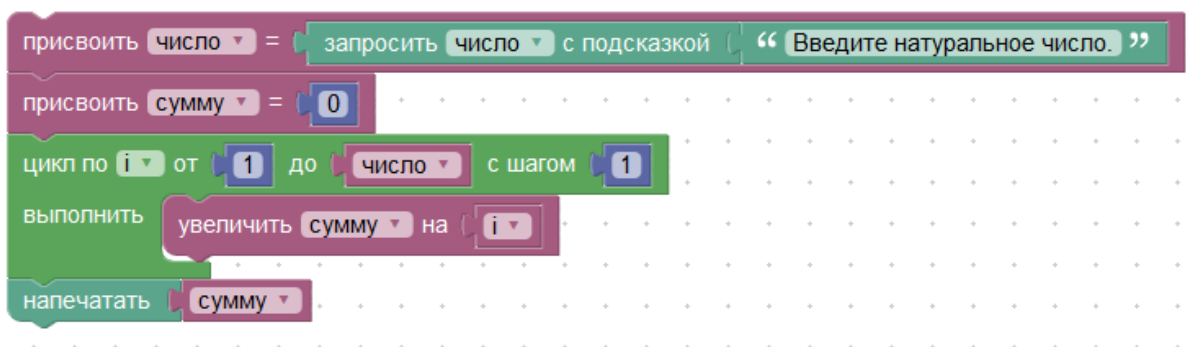
Задача 9: Напишите программу, пользователь вводит натуральное число N. Посчитать сумму всех чисел от 1 до N:

Паскаль:

```

var i,s,n:integer;
begin
  write('Введите n=');
  readln(n);
  s:=0;
  for i:=1 to n do
    s:=s+i;      {или inc(s,i);}
  writeln('Сумма s=',s);
end.
  
```

Blockly:



Задача 10: Ввести произвольный текст. В данном тексте определить, сколько раз встречается буквосочетание "он"

Паскаль:

```

var
Count: integer;
Str: string;
i: byte;
begin
write('Текст: ');
readln(Str);
for i:=1 to length(Str)-1 do
if (Str[i] = 'о') and (Str[i+1] = 'н') then inc(Count);
writeln('"он" в тексте встречается ', Count, ' раз(a)!');
end.

```

Blockly:

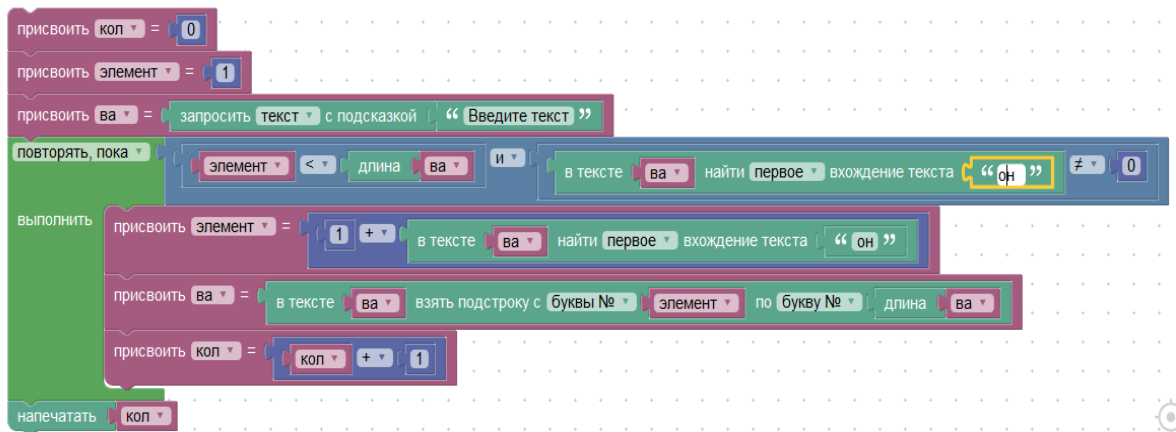


Таблица 1. Операторы языка Pascal и блоки Blockly

Pascal	Blockly
sqrt...	Корень квадратный

var..	присвоить
readln	запросить ... с подсказкой
writeln	напечатать
writelnelse writeln	Если Выполнять иначе
For ... to ... do	Цикл ... от ... до ... с шагом
if ...and....then	В тексте ... найти... вхождение текста...

При одинаковом решении задачи, написание программы на Blockly доступно в более раннем возрасте и упрощено за счет следующих факторов:

- названия блоков на русском языке;
- нельзя «стыковать» неподходящие блоки друг к другу, следовательно меньшее число ошибок при написании программы;
- блоки сгруппированы по назначению, что выделено различными цветами.

В то же время Blockly осуществляет частичный контроль над типами, не позволяя в элементарных ситуациях использовать вставки несоответствующего типа, а язык Паскаль строго типизирован.

Выводы по I главе:

При рассмотрении теоретических аспектов изучения программирования в основной школе была выявлена недостаточность объёма часов на изучение данной темы в курсе информатике.

Во втором параграфе были рассмотрены особенности визуального редактора Blockly Code Editor, который позволяет создавать программы без

ввода каких-либо символов или текста. Разработка программы осуществляется из широкого набора типовых лего-блоков, логическая композиция и соединение которых позволяет реализовывать заданную алгоритмическую функциональность программы.

В третьем параграфе проведено сравнение визуальных и текстовых языков программирования и листингов программ, реализуемых на Pascal и Blockly. Отмечено соответствие операторов и блоков этих языков и то, что написание программ на Blockly упрощено и доступно в более раннем возрасте.

ГЛАВА II. Программа внеурочной деятельности по информатике для основной школы

В теоретической части данной работы было отмечено, что в настоящее время существует проблема нехватки времени при изучении разделов «Алгоритмизация» и «Программирование» в курсе информатики основной школы. Согласно существующим примерным программам на изучение данных разделов отводится порядка 19 часов в 8 и 9 классах. Также необходимо отметить, что данные разделы достаточно трудны для освоения школьников и требуют предварительной подготовки.

Одним из возможных путей решения проблемы может быть раннее изучение программирования в 5-8 классах. В это время учащиеся проявляют повышенный интерес к компьютерам, и видится возможным развитие интереса к программированию и формированию алгоритмического мышления через использование в обучении простой и понятной среды программирования, например, такой как Blockly Code Editor.

Сделать это возможно несколькими путями. Первый и наиболее вероятный – через разработку программы внеурочной деятельности для учащихся. Второй вариант основан на том, что программирование – это технологическая деятельность и тем самым возможно включить ее элементы в курс технологии на более раннем этапе обучения.

2.1. Основные требования к организации внеурочной деятельности по ФГОС

Современный уровень информатизации образования характеризуется активным использованием средств информационных и коммуникационных технологий (ИКТ) во всех сферах жизнедеятельности человека. Это требует соответствующей подготовки школьников в области информатики.

Внеурочная деятельность, как и деятельность обучающихся в рамках уроков направлена на достижение результатов освоения основной образовательной программы. Но в первую очередь – это достижение личностных и метапредметных результатов. Это определяет и специфику внеурочной деятельности, в ходе которой обучающийся не только и даже не столько должен узнать, сколько научиться действовать, чувствовать, принимать решения и др.

При организации внеурочной деятельности в общеобразовательных учреждениях целесообразно использовать разнообразные формы организации деятельности обучающихся (экскурсии, кружковые и секционные занятия, клубные заседания, круглые столы, конференции, диспуты, школьные научные общества, олимпиады, соревнования, поисковые и научные исследования, общественно полезные практики и т.д.), которые отличны от организационных форм в урочной системе обучения.

Несомненно, внеурочная работа тесно связана с дополнительным образованием детей, когда дело касается создания условий для развития творческих интересов детей и включения их в художественную, техническую, эколого-биологическую, спортивную и другую деятельность.

При организации в образовательном учреждении наполняемость каждого класса не должна превышать 25 человек. При наличии необходимых условий и средств возможно деление классов на группы.

Воспитательные результаты внеурочной деятельности школьников распределяются по трем уровням.

Первый уровень результатов – приобретение школьником социальных знаний (об общественных нормах, об устройстве общества, о социально одобряемых и неодобряемых формах поведения в обществе), первичного понимания социальной реальности и повседневной жизни.

Для достижения данного уровня результатов особое значение имеет взаимодействие ученика со своими учителями (в основном и дополнительном образовании) как значимыми для него носителями положительного социального значения и повседневного опыта.

Второй уровень результатов – получение школьником опыта переживания и позитивного отношения к базовым ценностям общества (человек, семья, Отечество, природа, мир, знания, труд, культура), ценностного отношения к социальной реальности в целом.

Для достижения данного уровня результатов особое значение имеет взаимодействие школьников между собой на уровне класса, школы, то есть в защищенной, дружественной просоциальной среде. Именно в такой близкой социальной среде ребенок получает (или не получает) первое практическое подтверждение приобретенных социальных знаний, начинает их ценить (или отвергает).

Третий уровень результатов – получение школьником опыта самостоятельного общественного действия. Только в самостоятельном общественном действии юный человек действительно становится (а не просто узнает о том, как стать) социальным деятелем, гражданином, свободным человеком.

Для достижения данного уровня результатов особое значение имеет взаимодействие школьника с социальными субъектами за пределами школы, в открытой общественной среде.

Достижение трех уровней результатов внеурочной деятельности увеличивает вероятность появления эффектов воспитания и социализации детей. У учеников могут быть сформированы коммуникативная, этическая, социальная, гражданская компетенции и социокультурная идентичность.

Нормативные, инструктивные и методологические документы, обеспечивающие организацию учебной деятельности

Переход на Федеральный государственный образовательный стандарт (ФГОС) основного общего образования в 2015-2016 учебном году осуществляется в общеобразовательных организациях при наличии необходимых условий.

Внеурочная деятельность в общеобразовательных организациях определяется рядом нормативных документов и с учетом методических рекомендаций. Основными нормативными документами федерального уровня являются: Федеральный государственный образовательный стандарт основного общего образования (приказ Министерства образования и науки Российской Федерации от 17.12. 2010 г. № 1897) и Письмо ДОО Министерства образования и науки России от 12.05.2011 № 03 – 296 «Об организации внеурочной деятельности при введении федерального государственного стандарта общего образования».[15]

Об утверждении Федерального компонента государственного образовательного стандарта начального общего, основного общего и среднего (полного) общего образования / Приказ Министерства образования и науки Российской Федерации от 05.03.2004 г. № 1089.[16]

2.2. Тематическое планирование и содержание программы внеурочной деятельности «Основы программирования в визуальном редакторе Blockly Code Editor»

Величина недельной образовательной нагрузки (количество занятий), реализуемой через внеурочную деятельность, определяется за пределами количества часов, отведенных на освоение обучающимися учебного плана, но не более 10 часов. Для недопущения перегрузки обучающихся допускается перенос образовательной нагрузки, реализуемой через внеурочную деятельность, на периоды каникул, но не более 1/2 количества часов. Внеурочная деятельность в каникулярное время может реализовываться в рамках тематических программ (лагерь с дневным пребыванием на базе общеобразовательной организации или на базе загородных детских центров, в походах, поездках и т. д.).

В зависимости от задач на каждом этапе реализации примерной образовательной программы количество часов, отводимых на внеурочную деятельность, может изменяться.

Программа внеурочной деятельности рассчитана на 34 часа (1 час в неделю).

Целью курса является формирование и развитие алгоритмического мышления учащихся.

К задачам курса можно отнести:

- 1) формирование представлений учащихся об основных алгоритмических конструкциях;
- 2) формирование представления о программировании как виде профессиональной деятельности;
- 3) формирование и развитие навыков проектной деятельности;

4) развитие творческих способностей учащихся в процессе разработки и реализации проекта. Изучение программ-конструкторов компьютерных игр будет способствовать достижению у учащихся целой группы личностных и метапредметных результатов.

К личностным результатам можно отнести:

1) развитие навыков сотрудничества со сверстниками, и взрослыми в образовательной, учебно-исследовательской, проектной и других видах деятельности;

2) формирование осознанной выбор будущей профессии и возможностей реализации собственных жизненных планов.

К метапредметным результатам относятся:

1) умение самостоятельно определять цели деятельности и составлять планы деятельности; самостоятельно осуществлять, контролировать и корректировать деятельность; использовать все возможные ресурсы для достижения поставленных целей и реализации планов деятельности; выбирать успешные стратегии в различных ситуациях;

2) владение навыками познавательной, учебно-исследовательской и проектной деятельности, навыками разрешения проблем; способность и готовность к самостоятельному поиску методов решения практических задач, применению различных методов познания;

3) готовность и способность к самостоятельной информационно-познавательной деятельности, включая умение ориентироваться в различных источниках информации, критически оценивать и интерпретировать информацию, получаемую из различных источников;

4) умение использовать средства информационных и коммуникационных технологий в решении когнитивных,

коммуникативных и организационных задач с соблюдением требований эргономики, техники безопасности, гигиены, ресурсосбережения, правовых и этических норм, норм информационной безопасности.

Программой предусмотрены *методы обучения*: объяснительно-иллюстративные, частично-поисковые (вариативные задания), творческие, практические.

Программа реализуется на основе следующих *принципов*:

- *Обучение в активной познавательной деятельности.* Все темы учащиеся изучают на практике, выполняя различные творческие задания, общаясь в парах, группах и друг с другом.
- *Индивидуальное обучение.* Обучение учащихся работе на компьютере дает возможность организовать деятельность учащихся с индивидуальной скоростью и в индивидуальном объеме. Данный принцип реализован через организацию практикума по освоению навыков работы на компьютере.
- *Принцип природосообразности.* Основной вид деятельности школьников – игра, через нее дети познают окружающий мир, поэтому в занятия включены игровые элементы, способствующие успешному освоению курса.
- *Преемственность.* Программа курса построена так, что каждая последующая тема логически связана с предыдущей. Данный принцип учащимся помогает понять важность уже изученного материала и значимость каждого отдельного занятия.
- *Целостность и непрерывность,* означающие, что данная ступень является важным звеном единой общешкольной подготовки по информатике и информационным технологиям. В рамках данной ступени подготовки продолжается осуществление вводного, ознакомительного обучения школьников, предваряющего более

глубокое изучение предмета в 6-9 (основной курс) и 10-11 (профильные курсы) классах.

- *Практико-ориентированность*, обеспечивающая отбор содержания, направленного на решение простейших практических задач планирования деятельности, поиска нужной информации, инструментирования всех видов деятельности на базе общепринятых средств информационной деятельности, реализующих основные пользовательские возможности информационных технологий.
- *Принцип дидактической спирали* как важнейший фактор структуризации в методике обучения информатике: вначале общее знакомство с понятием с учетом имеющегося опыта обучаемых, затем его последующее развитие и обогащение, создающее предпосылки для научного обобщения в старших классах.
- *Принцип развивающего обучения*. Обучение ориентировано не только на получение новых знаний в области информатики и информационных технологий, но и на активизацию мыслительных процессов, формирование и развитие у школьников обобщенных способов деятельности, формирование навыков самостоятельной работы.

Формы проведения занятий: беседы, игры, практические занятия, самостоятельная работа, викторины и проекты.

Используются такие педагогические технологии как обучение в малых группах сотрудничества, индивидуализация и дифференциация обучения, дискуссии, мозговые атаки, круглые столы, проектные методы обучения, технологии использования в обучении игровых методов, информационно-коммуникационные технологии, ситуационный анализ, рефлексия.

Использование метода проектов позволяет обеспечить условия для развития у ребят навыков самостоятельной постановки задач и выбора

оптимального варианта их решения, самостоятельного достижения цели, анализа полученных результатов с точки зрения решения поставленной задачи.

Программой предусмотрены методы обучения: объяснительно-иллюстративные, частично-поисковые (вариативные задания), творческие, практические.

Содержание курса внеурочной деятельности:

1. Введение в компьютерное программирование – 3 часа
2. Алгоритмы. Основные обозначения блок-схем – 3 часа

Аналитическая: обобщение полученной информации об устройствах компьютера, выбор необходимой алгоритмической конструкции для решения поставленной задачи.

Практическая: использование различных устройств для ввода, вывода и хранения информации, создание описание и проверка алгоритма

3. Основные приемы программирования и создания проекта в визуальном редакторе Blockly – 3 часа
4. Учимся программировать на языке Blockly. – 22 часа

Аналитическая: сопоставление алгоритмических конструкций в виде блок-схем с записью в программе Blockly.

Практическая: создание и отладка программного алгоритма в среде Blockly.

5. Создание личного проекта. Защита проектов. – 3 часа

Аналитическая: Обоснование выбора темы проекта.

Практическая: Реализация и защита проекта.

В таблице 2 представлено учебно-тематическое планирование с указанием тем и количеством часов.

Таблица 2. Учебно-тематический план.

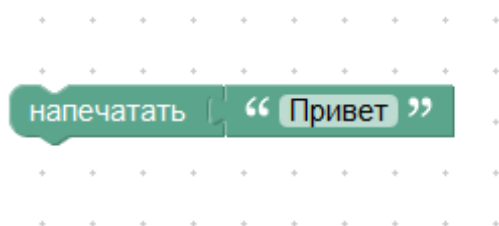
№ п.п	Тема	Кол-во часов
	Введение – 3 часа	3
1	Вводное занятие	1
2	Визуальные языки программирования	1
3	Blockly Code Editor	1
	Алгоритмы – 3 часа	3
4	Линейный алгоритм	1
5	Разветвляющийся алгоритм	1
6	Циклические алгоритмы	1
	Основные приемы программирования и создания проекта в визуальном редакторе Blockly Code Editor – 3 часа	3
7	Текстовые, списки, циклы	1
8	Логические, математические	1
9	Переменные и функции	1
Учимся программировать на языке Blockly – 22 часов		
10-13	<i>Задачи с использованием линейного алгоритма</i>	4
<p>Первые программы: Программа – приветствие. Здравствуй, МИР! Первые уровни прохождения лабиринта.</p>		
<p>Практические работы: «Программа - знакомство, которая спрашивает имя и возраст пользователя.» «Программа, которая выводит количество символов в имени пользователя.» «Программа, которая вычисляет произведение двух чисел.» «Программа, запрашивает число и выводит его квадратный корень.»</p>		
14-20	<i>Задачи с использованием линейного и разветвляющегося алгоритма</i>	7
<p>Изучаем программы: Флора и фауна. Страны мира. Калькулятор посадочных мест в самолёте.</p>		
<p>Практические работы: «Программа сообщает, что имя пользователя длинное, если оно больше 9 символов.» «Программа для вычисления выражения: $23*(44+37)-100$».</p>		

<p>«Программа сообщает положительное или отрицательное число выйдет при вычислении выражения: $23*47-17*213+125*3$»</p> <p>«Определить, встречается или нет в введенном пользователем тексте слово "привет".»</p> <p>«Напечатать, сколько раз в введенном пользователем тексте встречается слово "один".»</p>		
21-30	Задачи с использованием линейного, разветвляющего и циклического алгоритма	10
<p>Изучаем программы: График функции. Черепашка. Лабиринт средние уровни.</p>		
<p>Практические работы:</p> <p>«Составить программу рисования квадрата, используя блок "повторить"»</p> <p>«Составить программу рисования правильного шестиугольника»</p> <p>«Составить программу рисования девятиконечной звезды, используя блок "повторить"»</p> <p>«Составить программу вычисления произведения цифр числа, введенного пользователем»</p> <p>«Написать программу вычисления значения факториала $F=N!$ для заданного числа N. Факториал N - это произведение всех чисел от 1 до N. Например: $5! = 1 \times 2 \times 3 \times 4 \times 5$»</p> <p>«Написать программу нахождения суммы первых N четных натуральных чисел»</p> <p>«Найти произведение целых положительных нечетных чисел, меньших 200 и кратных 5»</p>		
31	«Час кода»	1
	Создание личного проекта. Защита проектов.	3
32-33	Создание собственных проектов.	2
34	Защита проектов.	1
	Всего	34 часов

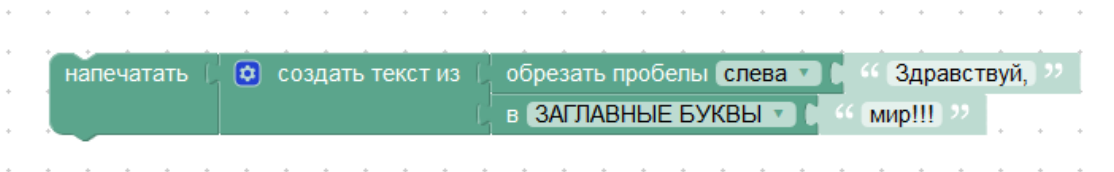
Содержание практикума:

Задачи линейного алгоритма – 4 часа

Задача 1. Программа – приветствие.



Задача 2. Здравствуй, МИР!



Задача 3. Первые уровни прохождения лабиринта.

Уровень 1

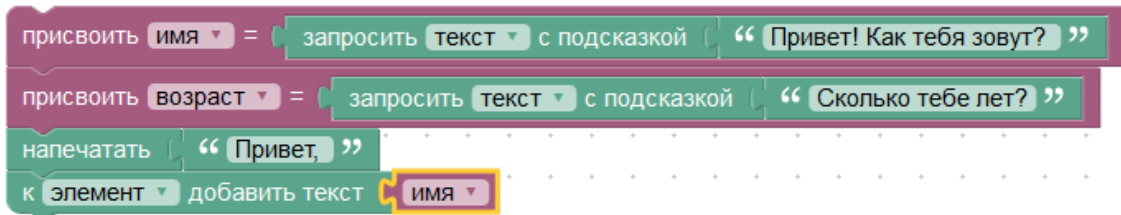


Уровень 2



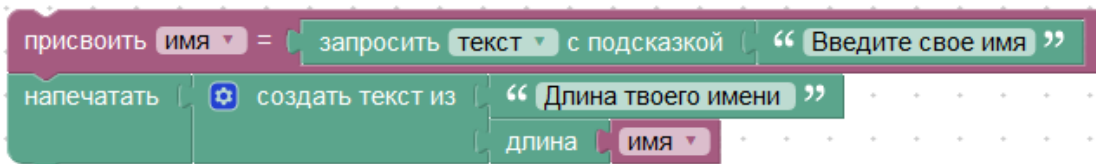
Практические работы

Задание 1. Программа - знакомство, которая спрашивает имя и возраст пользователя



```
присвоить ИМЯ = запросить текст с подсказкой "Привет! Как тебя зовут?"
присвоить возраст = запросить текст с подсказкой "Сколько тебе лет?"
напечатать "Привет,"
к элемент добавить текст ИМЯ
```

Задание 2. Программа, которая выводит количество символов в имени пользователя



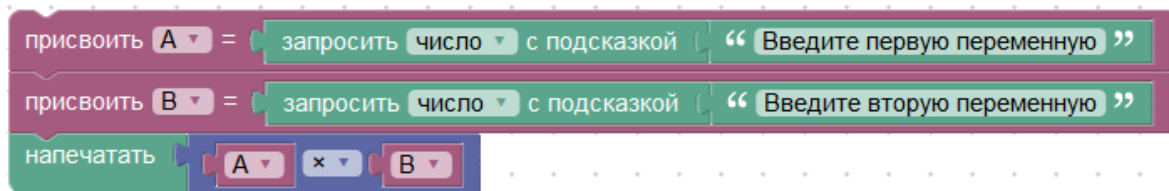
```
присвоить ИМЯ = запросить текст с подсказкой "Введите свое имя"
напечатать создать текст из "Длина твоего имени"
длина ИМЯ
```

Задание 3. Программа, которая вычисляет произведение двух чисел



```
присвоить a = запросить число с подсказкой "Напишите число для извлечения квадратного корня"
напечатать квадратный корень a
```

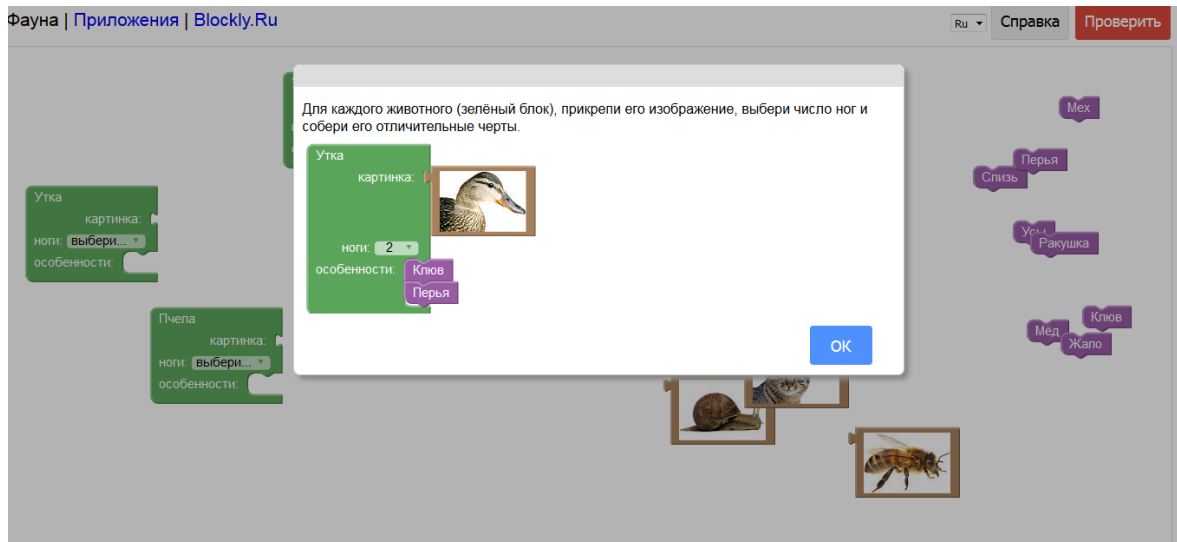
Задание 4. Программа, запрашивает число и выводит его квадратный корень



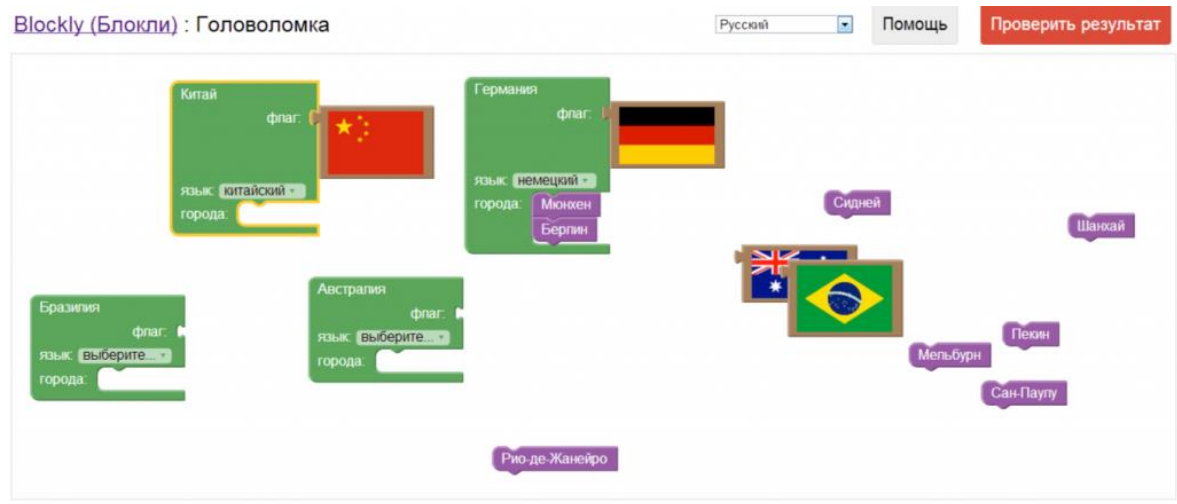
```
присвоить A = запросить число с подсказкой "Введите первую переменную"
присвоить B = запросить число с подсказкой "Введите вторую переменную"
напечатать A * B
```

Задачи с использованием линейного и разветвляющего алгоритма – 7 часов

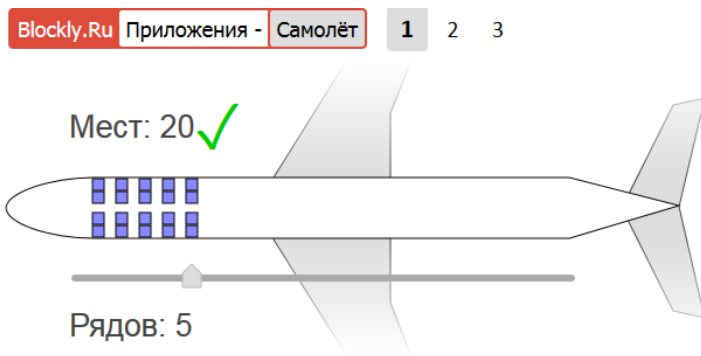
Задача 1. Флора и фауна.



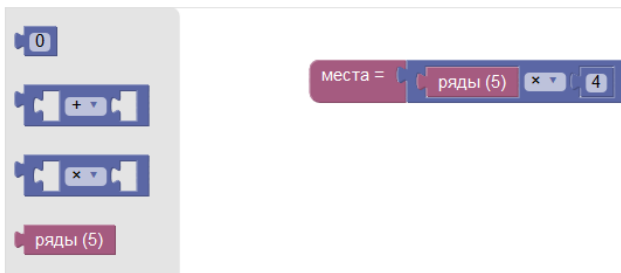
Задача 2. Страны мира.



Задача 3. Калькулятор посадочных мест в самолёте.

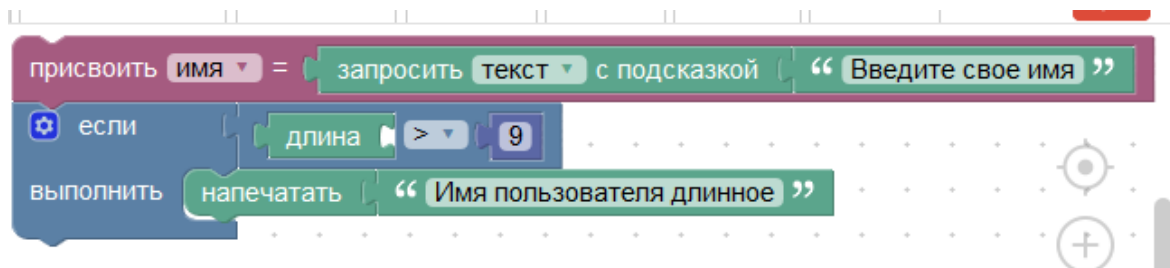


В самолёте несколько рядов с пассажирскими местами. В каждом ряду 4 места.
 Постройте формулу в области ниже, которая поможет рассчитать общее количество мест в самолёте (как на рисунке выше).

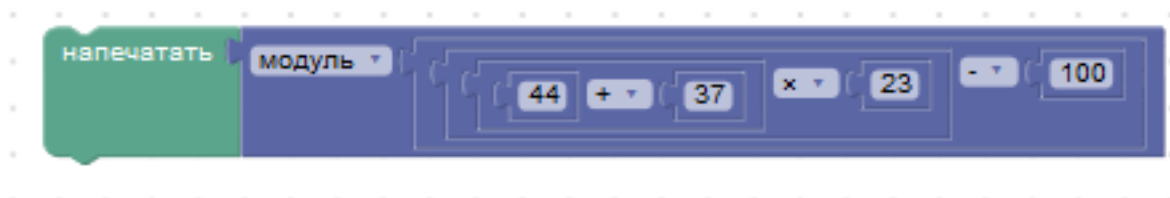


Практические работы:

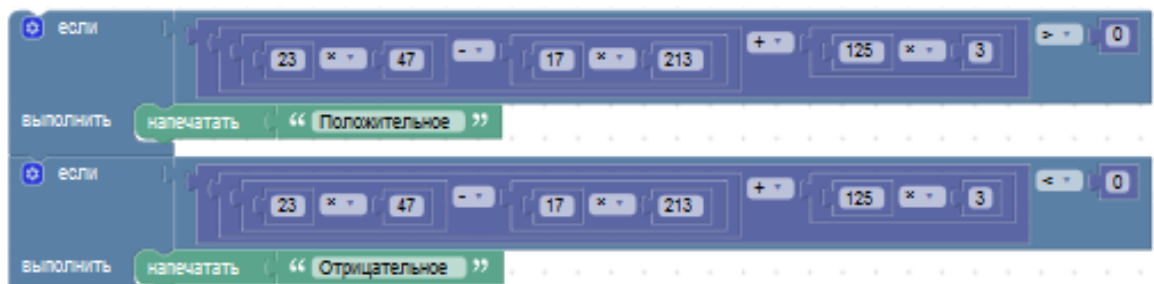
Задание 1. Программа сообщает, что имя пользователя длинное, если оно больше 9 символов.



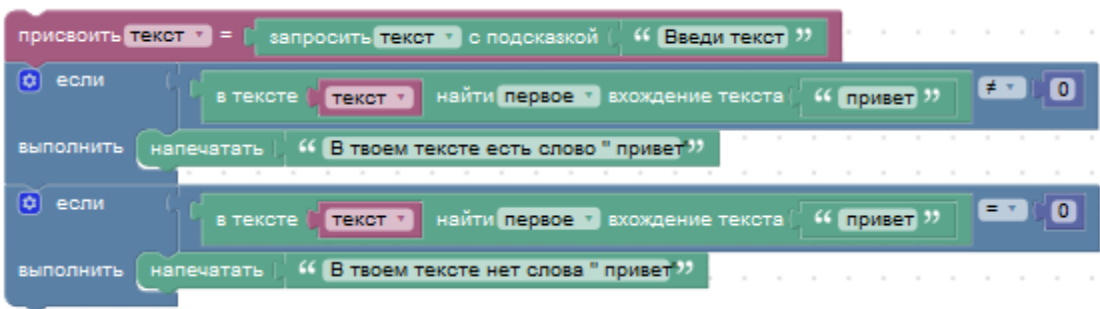
Задание 2. Программа для вычисления выражения: $|23*(44+37)-100|$



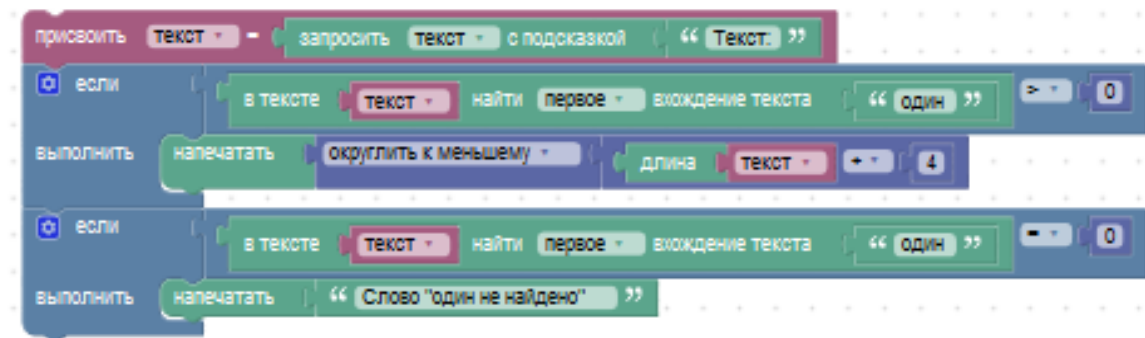
Задание 3. Программа сообщает положительное или отрицательное число выйдет при вычислении выражения: $23*47-17*213+125*3$



Задание 4. Определить, встречается или нет в введенном пользователем тексте слово "привет".



Задание 5. Напечатать, сколько раз в введенном пользователем тексте встречается слово "один".



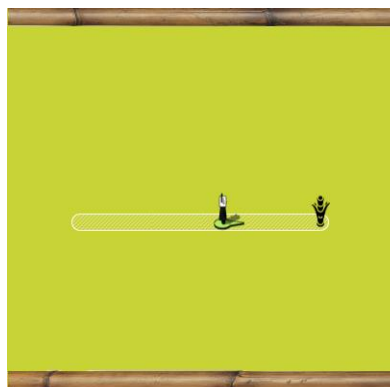
Задачи с использованием линейного, разветвляющего и циклического алгоритма – 10 часов

Задача 1. График функции.

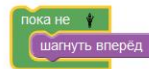
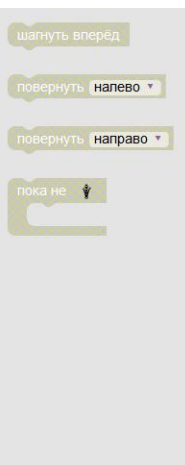
Задача 2. Черепаха.

Задача 3. Лабиринт средние уровни.

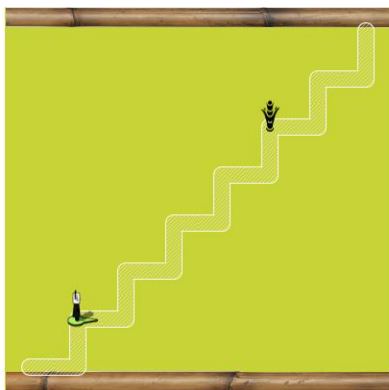
Уровень 3.



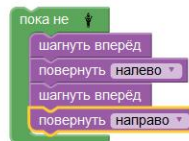
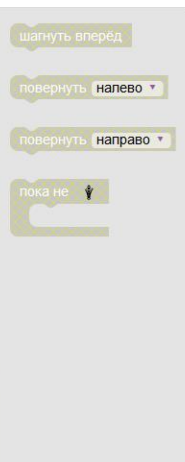
У вас осталось 0 блока (блоков).



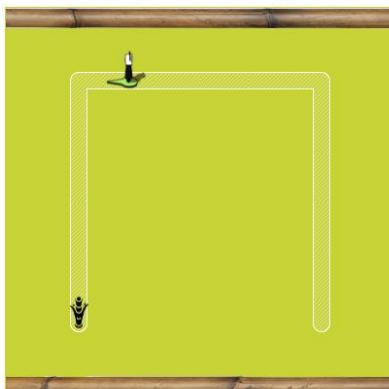
Уровень 4.



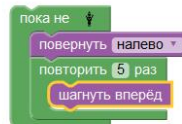
У вас осталось 0 блока (блоков).



Уровень 5.



У вас осталось 0 блока (блоков).

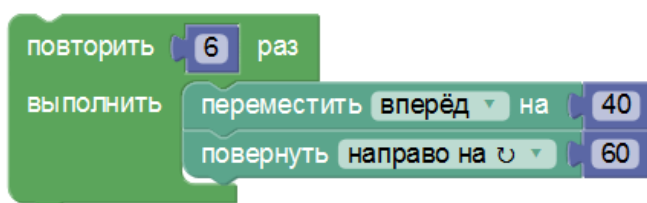


Практические работы

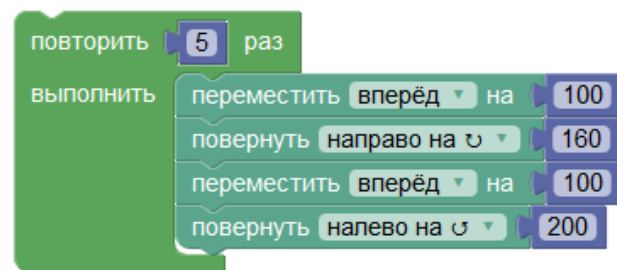
Задание 1. Составить программу рисования квадрата, используя блок "повторить"



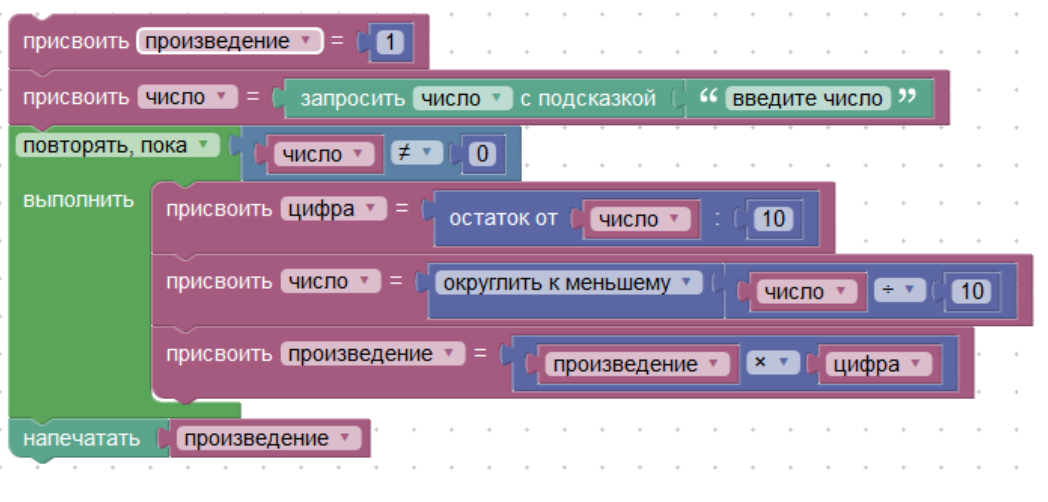
Задание 2. Составить программу рисования правильного шестиугольника



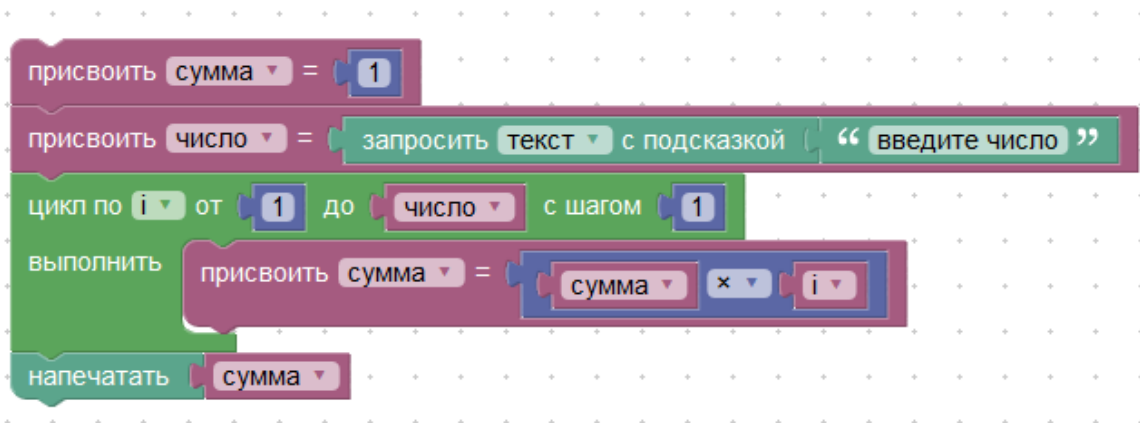
Задание 3. Составить программу рисования девятиконечной звезды, используя блок "повторить"



Задание 4. Составить программу вычисления произведения цифр числа, введенного пользователем

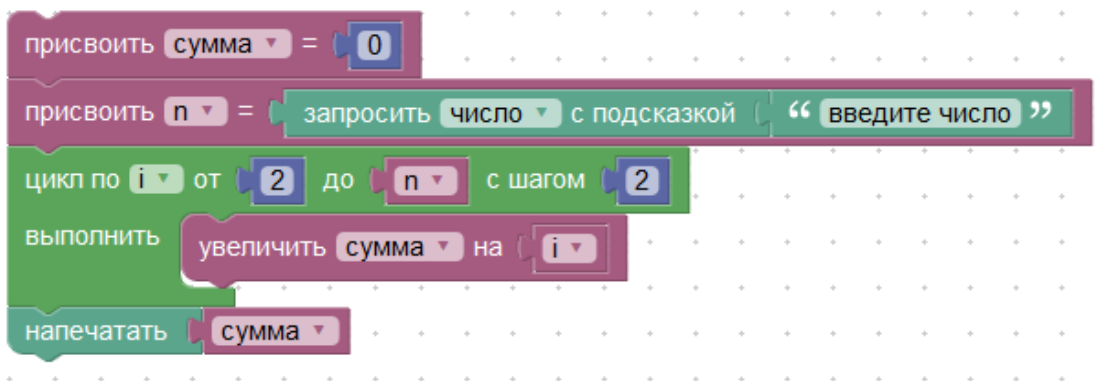


Задание 5. Написать программу вычисления значения факториала $F=N!$ для заданного числа N . Факториал N - это произведение всех чисел от 1 до N .
Например: $5! = 1 \times 2 \times 3 \times 4 \times 5$



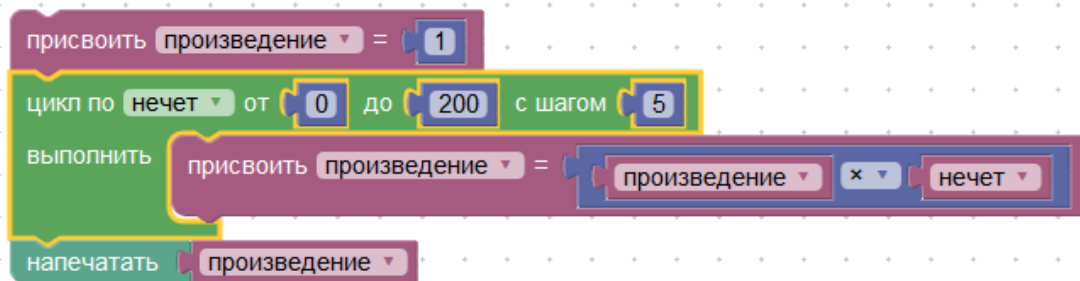
```
присвоить сумма = 1
присвоить число = запросить текст с подсказкой "введите число"
цикл по i от 1 до число с шагом 1
  выполнить
    присвоить сумма = сумма * i
напечатать сумма
```

Задание 6. Написать программу нахождения суммы первых N четных натуральных чисел



```
присвоить сумма = 0
присвоить n = запросить число с подсказкой "введите число"
цикл по i от 2 до n с шагом 2
  выполнить
    увеличить сумма на i
напечатать сумма
```

Задание 7. Найти произведение целых положительных нечетных чисел, меньших 200 и кратных 5



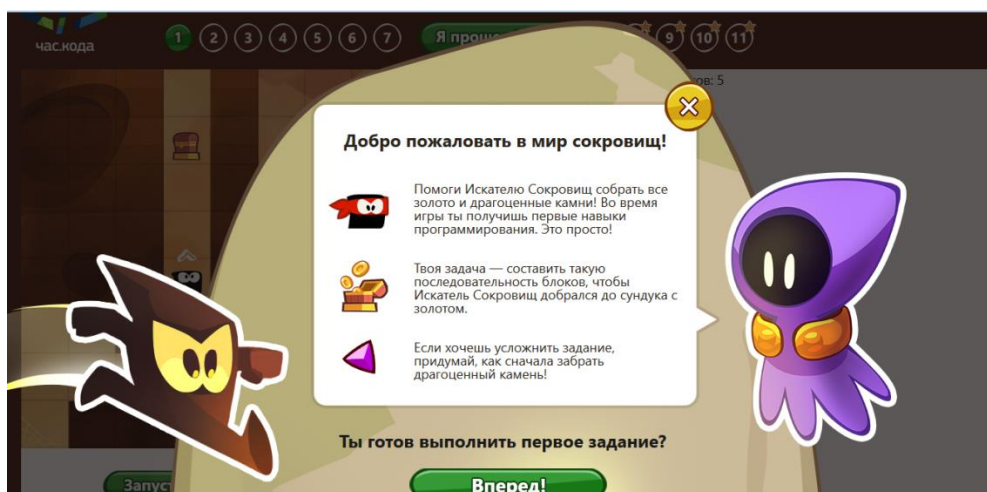
```
присвоить произведение = 1
цикл по нечет от 0 до 200 с шагом 5
  выполнить
    присвоить произведение = произведение * нечет
напечатать произведение
```

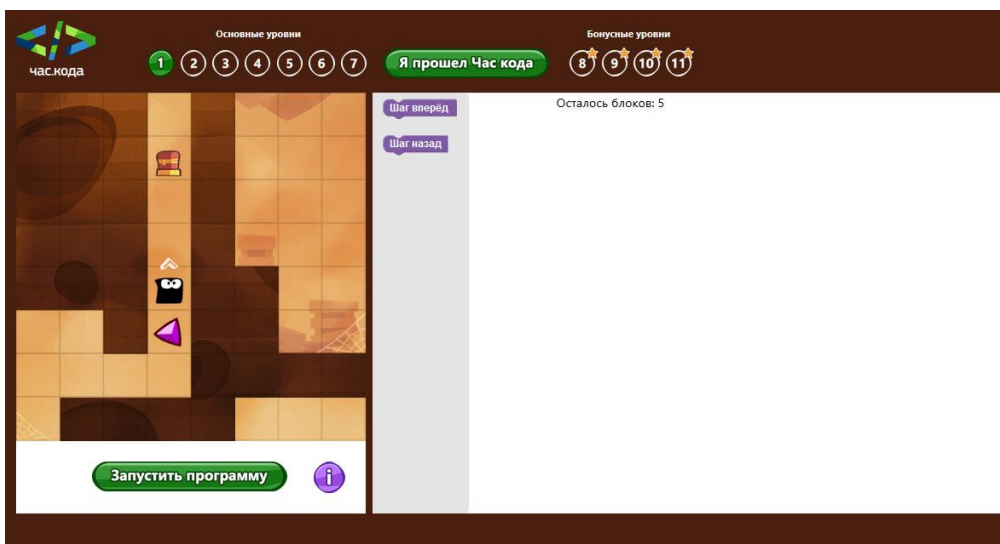
Дидактический материал к уроку «Час кода» - 1 час

«Час кода» - Акция проводится Минобрнауки России совместно с Минкомсвязи России при участии ведущих ИТ-компаний: ВКонтакте, Лаборатория Касперского, Майкрософт, 1С, Дневник.ру, Акронис.

Чтобы вдохновить учащихся осваивать программирование, во всем мире проводится акция «Час Кода», в рамках которой проводят необычные уроки, занятия и мероприятия, направленные на формирование и поддержку интереса молодых людей к изучению информационных технологий и информатики.

Благодаря игровой форме и наличию известных компьютерных героев тренажёр будет одинаково интересен учащимся разных возрастов. Работа на тренажёре строится по принципу решения забавных головоломок и помогает подкрепить интерес учащихся, возникший во время проведения первого блока занятия (видео, лекции и обсуждения), а также даёт возможность сразу, на практике, проверить свои силы в программировании и логике





Выводы по II главе:

В параграфе 2.1 рассмотрены основные требования к организации внеурочной деятельности по ФГОС. Внеурочная деятельность направлена на достижение результатов освоения соответствующей образовательной программы, воспитательных результатов и формирование у обучающихся мотивации к получению фундаментального образования.

В параграфе 2.2 представлено тематическое планирование и содержание программы внеурочной деятельности «Изучение основ программирования в визуальном редакторе Blockly Code Editor», рассчитанной на 34 часа. В содержании данной программы выделено 5

разделов: «Введение»; «Алгоритмы»; «Основные приемы программирования и создания проекта в визуальном редакторе Blockly Code Editor»; «Учимся программировать на языке Blockly»; «Создание личного проекта». В работе так же представлены примеры решения всех предложенных учебных задач.

Заключение

Одним из проблемных вопросов в обучении информатике остается вопрос о системе обучения программированию. Как заинтересовать учащихся программированием, как научить, понимать и решать задачи на начальном уровне изучения программирования остаются весьма актуальными вопросами.

В ходе решения первой задачи было рассмотрено: с какого класса начинается изучение курса информатики; количество часов выделенное для изучения этого курса; сколько из общего количества часов выделено на программирование. При рассмотрении было выявлено, что объём часов на изучение программирования не дает возможности в полной мере изучить данную тему в школьном курсе.

В ходе решения второй задачи были выявлены особенности визуального редактора Blockly Code Editor. В этом редакторе можно создавать программы, соединяя визуальные блоки друг с другом в соответствии с их формой, создание программы таким образом напоминает собирание паззла.

В ходе решения третьей задачи было проведено сравнение визуальных и текстовых языков программирования. Сравнились решения одинаковых задач на двух языках программирования, представление кода программы позволяло наглядно увидеть соответствие операторов языка Паскаль и визуальных блоков языка Blockly. Было отмечено, что написание программ на Blockly упрощено и доступно в более раннем возрасте.

В ходе решения четвертой задачи были рассмотрены основные требования организации внеурочной деятельности по ФГОС. Внеурочная деятельность направлена на формирование у обучающихся мотивации к получению фундаментального образования, достижение результатов

освоения соответствующей образовательной программы. В ходе освоения программы обучающийся не только и даже не столько должен узнать, сколько научиться действовать, чувствовать, принимать решения.

В ходе решения шестой задачи разработано планирование и содержание программы внеурочной деятельности по теме «Основы программирования в визуальном редакторе Blockly Code Editor». В качестве содержания отобраны следующие разделы программы: «Введение»; «Алгоритмы»; «Основные приемы программирования и создания проекта в визуальном редакторе Blockly Code Editor»; «Учимся программировать на языке Blockly»; «Создание личного проекта». В планировании выделено содержание практикума с подробным решением каждой задачи.

Список используемой литературы

1. Аркадьев, А.Г., Днепров, Э.Д. Сборник нормативных документов. Информатика и ИКТ – М.: Дрофа, 2007.
2. Атре Ш. Структурный подход к организации баз данных. – М.: Финансы и статистика, 2008.
3. Библиотека on-line [Электронный ресурс]. - Режим доступа: <http://citforum.ru/news/28549/> (дата обращения: 01.05.2017).
4. Благодатских В.А. Экономика, разработка и использование программного обеспечения ЭВМ. – М: Финансы и статистика, 1999.
5. Википедия, свободная энциклопедия [Электронный ресурс]. - Режим доступа: <https://ru.wikipedia.org/wiki/%D0%91%D0%BB%D0%BE%D0%BA%D0%BB%D0%B8> (дата обращения: 01.05.2017).
6. Додонова Е. А. 2010-2017. Материалы к урокам информатики, алгоритмики и программирования. [Электронный ресурс]. - Режим доступа: <http://forinformatics.blogspot.ru/2015/01/blog-post.html> (дата обращения: 23.04.2017).
7. Документация по государственным стандартам [Электронный ресурс]. - Режим доступа: WWW. STANDARD.RU (дата обращения: 28.05.2017).
8. Иванюк Т. В. Применение информационных технологий в учебно-исследовательской деятельности. – 2001.
9. ИТ [Электронный ресурс]. - Режим доступа: <https://dev.by/lenta/main/google-blockly-apofeoz-vizualnogo-programmirovaniya> (дата обращения: 11.05.2017).
10. Компьютерные технологии обработки информации./Под.ред. С.В.Назарова. - М.: Финансы и статистика, 1995
11. Коутс Р., Влейминк И. Интерфейс «человек-компьютер»: Пер. с англ. М.: Мир, 2000.
12. Левицкова Л.А., Информатика в школе [Электронный ресурс]. - Режим доступа: <http://krivaksin.ru/izuchenie-informatiki-v-usloviyah-fgos/> (дата обращения: 20.04.2017).

13. Липаев В.В. Системное проектирование сложных программных систем – М: СИНТЕГ, 1998
14. Малахов В. А. Компьютеры Программирование [Электронный ресурс]. - Режим доступа: <http://fb.ru/article/250639/что-такое-paskal-programmirovanie-na-paskale> (дата обращения: 25.05.2017).
15. Методические рекомендации общеобразовательным организациям по сопровождению образовательной деятельности в условиях введения ФГОС основного общего образования. [Электронный ресурс]. - Режим доступа: <http://docs.cntd.ru/document/420296562> (дата обращения: 03.06.2017).
16. Министерство образования и науки РФ. [Электронный ресурс]. – Режим доступа: <http://минобрнауки.рф/%D0%B4%D0%BE%D0%BA%D1%83%D0%BC%D0%B5%D0%BD%D1%82%D1%8B/7378> (дата обращения: 1.05.2017).
17. Нельзина О.В. Проблемы обучения программированию по курсу информатики в системе «школа-вуз» // Вопросы Интернет - образования. - 2006. - №13.
18. Необычный урок программирования [Электронный ресурс]. - Режим доступа: <http://www.coderussia.ru/hoc2014.html> (дата обращения: 25.05.2017).
19. Новые педагогические и информационные технологии в системе образования: Учебное пособие / Е. С. Полат, М. Ю. Бухаркина, М. В. Моисеева, А. Е. Петров; Под ред. Е.С.Полат.-М.: Издательский центр «Академия», 2003.
20. Программирование на Blockly [Электронный ресурс]. - Режим доступа: <http://blockly.ru/about.html> (дата обращения: 11.05.2017).
21. Проектирование информационных систем: учеб./под ред. Ю. Ф. Тельнова. М., 2009
22. Ройс У. Управление проектами по созданию программного обеспечения. Пер. с англ. – М.: ЛОРИ, 2005

23. Смирнова Г.Н. и др. Проектирование экономических информационных систем: Учебник / Под ред. Ю.Ф. Тельнова. — М.: Финансы и статистика, 2009 — 512 с.
24. Сухоруков Д.В., Сорокина Л.А. Активизация познавательной деятельности учащихся образовательных школ // Инновационные проекты и программы в образовании. 2015. [Электронный ресурс]. - Режим доступа: <http://cyberleninka.ru/article/n/aktivizatsiya-roznavatelnoy-deyatelnosti-uchaschihsya-obrazovatelnyh-shkol> (дата обращения: 21.05.2017).
25. Ткалич С. К. Медиаобразование: интеграция дидактики, информационной эстетики и локализации мультимедийного продукта на основе национально-культурных маркеров //Современные наукоёмкие технологии. – 2015. – №. 11. – С. 105-107.
26. Т.Гарднер, Выбор между визуальным и текстовым программированием для детей.[Электронный ресурс].-Режим доступа:<http://www.pvsm.ru/sravnenie/227924>(дата обращения: 30.04.2017).
27. Угринович, Н.Д. Информатика и ИКТ. Базовый курс. Учебник для 9 класса / Н.Д. Угринович. – М.: БИНОМ. Лаборатория знаний, 2007.
28. Форум программистов и сисадминов. .[Электронный ресурс].-Режим доступа: <http://www.cyberforum.ru/>(дата обращения: 3.04.2017).
29. Фридман А.Л Основы объектно-ориентированной разработки программных систем – М: Финансы и статистика, 2007
30. Шафер Д.Ф., Фартрел Т., Шафер Л.И. Управление программными проектами: достижение оптимального качества при минимуме затрат.: Пер. с англ. – М.: Вильямс, 2008.
31. Эксклюзивные ИТ-новости, обзоры и интервью [Электронный ресурс]. - Режим доступа: <http://bloggerator.org/page/sovremennoe-vizualnoe-programmirovanie-google-blockly-vpl> (дата обращения: 28.05.2017).