

В. И. Филиппов,
Академия социального управления, г. Москва

ПРОПЕДЕВТИКА ПРОГРАММИРОВАНИЯ ВО ВНЕУРОЧНОЙ ДЕЯТЕЛЬНОСТИ С УЧАЩИМИСЯ V—IX КЛАССОВ

Аннотация

В статье предложена модель раннего обучения программированию, которая может быть осуществлена во внеурочной деятельности с учащимися V—IX классов, дана краткая характеристика используемых программных продуктов.

Ключевые слова: программирование, внеурочная деятельность, проектная деятельность.

Контактная информация

Филиппов Владимир Ильич, аспирант кафедры информационно-коммуникационных технологий Академии социального управления, г. Москва; *адрес:* 129281, г. Москва, Староватулинский проезд, д. 8; *телефон:* (495) 472-32-08, доб. 149; *e-mail:* vf95@rambler.ru

V. I. Filippov,
Academy of Public Administration, Moscow

PROPAEDEUTIC OF PROGRAMMING IN EXTRACURRICULAR ACTIVITY WITH STUDENTS OF 5–9TH GRADES

Abstract

The article describes the model of early training in programming which can be carried out in extracurricular activity with students of 5–9th grades. A brief description of using software is given.

Keywords: programming, extracurricular activity, project activity.

По результатам анализа проведения в 2016 году Государственной итоговой аттестации по информатике и ИКТ на территории Московской области в форме ОГЭ выявлена проблема заметного снижения успешности выполнения заданий, связанных с разделом «Алгоритмизация и программирование» (задания 6, 8–10, 14, 16). Крайне низка результативность выполнения заданий 20.1 и 20.2. Каждый третий участник экзамена не приступил к выполнению задания или неправильно написал программу для формального исполнителя или программу по обработке числовой последовательности на алгоритмическом языке [3]. Основная причина подобного результата, на наш взгляд, заключается в том, что изучению данных разделов уделяется недостаточное внимание в курсе информатики основной школы.

Одним из возможных путей решения данной проблемы могла бы стать организация раннего изучения программирования во внеурочной деятельности с учащимися V—IX классов.

Мы предлагаем модель организации внеурочной деятельности по алгоритмизации и программированию с учащимися основной школы, состоящую из взаимосвязанных модульных курсов. **Время, отводимое на изучение одного модуля, может быть 17 или 34 академических часа.** Это связано с особенностями организации учебно-воспитательного процесса в образовательных организациях. В первом случае модуль рассчитан на полугодие, во втором случае — на год при условии выделения на курс одного часа в неделю. Это также дает возможность масштабирования курса. В случае выделения на изучение курса двух академических часов в неделю модуль объемом 17 часов будет изучаться в течение одной четверти, а модуль объемом 34 часа — в течение полугодия.

Структура модульного курса по раннему изучению программирования представлена в таблице 1.

Таблица 1

Структура модульного курса по раннему изучению программирования

№ п/п	Примерное название модуля	Используемое ПО	Кол-во часов на модуль	Классы
1	Рисуем в среде Scratch	Scratch, Scratch online	17	V—VI
2	Создаем игру в среде Scratch	Scratch, Scratch online	17–34	V—VI
3	Создаем игру в конструкторе игр	Construct, Game Maker, Unity 3D, Kodu	17–34	VI—VII
4	Создаем мобильное приложение в App Inventor 2	App Inventor 2	34	VI—VII
5	Создаем анимацию в Pascal	Pascal ABC	17	VII—VIII
6	Программируем на Python	Python 3.x	34	VIII—IX

Модель модульного курса основана на следующих принципах:

- возможность построения индивидуальных траекторий для учащегося или группы обучающихся — педагог сам определяет порядок изучения модулей, учитывая индивидуальные особенности обучающихся;

- использование проектной деятельности — включение в каждый раздел заданий для проектной деятельности обучающихся;
- масштабируемость — возможность расширения объема часов за счет использования педагогом дополнительных материалов (в том числе для организации проектной деятельности);
- открытость — возможность добавления модулей при условии соблюдения принципа единства структуры и учебных целей.

Каждый модуль курса внеурочной деятельности должен включать в себя следующие структурные элементы:

1. Цель изучения модуля.
2. Задачи изучения модуля.
3. Инвариантное содержание модуля.
4. Методические рекомендации по изучению модуля.
5. Задания для обучающихся.
6. Контрольные мероприятия.
7. Планируемые результаты обучения:
 - предметные;
 - метапредметные;
 - личностные.
8. Перечень рекомендуемого оборудования и программного обеспечения.
9. Перечень информационных ресурсов (литературы и ресурсов сети Интернет), необходимых для реализации данного модуля.

При разработке модуля с целью формирования у обучающихся универсальных учебных действий необходимо придерживаться следующих рекомендаций.

При организации занятий по модулю используются такие методы, как проблемная лекция, эвристическая беседа, учебная дискуссия, самостоятельная работа учащихся с литературой, метод проектов, групповые обсуждения, кейс-метод.

Все занятия внутри модуля можно разделить на три блока:

- 1) вводное занятие (1–2 академических часа);
- 2) изучение возможностей программного обеспечения (от 10 до 28 академических часов);
- 3) разработка и защита индивидуальных проектов или решение творческих задач (2–4 академических часа).

Первый блок. На первом вводном занятии обучающиеся знакомятся с основными понятиями и объектами, изучаемыми в модуле, основными возможностями изучаемого программного обеспечения. Необходимо использовать методы эвристической беседы или лекции с элементами проблемного обучения. Это будет способствовать формированию у обучающихся УУД, связанных с оценкой используемых средств ИКТ.

Второй блок. При изучении возможностей программного обеспечения рекомендуется придерживаться схемы организации и проведения занятия, представленной в таблице 2.

Таким образом, при проведении занятий второго микромодуля у обучающихся формируется комплекс универсальных учебных действий.

Третий блок позволяет подвести итоги изучения модуля и направлен на формирование у обучающихся комплекса УУД на среднем и высоком уровнях. Одним из эффективных способов формирования УУД у учащихся V–IX классов является учебно-исследовательская и проектная деятельность. Процесс создания полноценного проекта способствует реализации компетентного подхода в обучении, формированию универсальных учебных действий. Анализ результатов выполнения учащимися проектов позволяет оценить сформированность у них универсальных учебных действий.

При подготовке к защите проекта учащиеся готовят его описание, в котором должны найти отражение следующие действия:

1. Определите цели и задачи проекта.
2. Опишите процесс разработки вашей программы.
3. Укажите, какие алгоритмические структуры вы использовали при создании программы.
4. Опишите алгоритм, используемый в вашей программе. Постройте блок-схему алгоритма.
5. Укажите, какие доработки вы внесли в программу по сравнению с предложенным заданием; какую цель при этом ставили. Проанализируйте результативность доработки.
6. Опишите способы использования разработанной вами программы.

В процессе защиты проводится демонстрация проекта с целью определения корректности работы программы.

Таблица 2

Схема организации занятия по изучению возможностей программного обеспечения

№ п/п	Этап занятия	Форма организации и проведения	УУД, формируемые на данном этапе	Примерное время (мин)
1	Актуализация ранее полученных знаний, постановка проблемы, решаемой на занятии	Эвристическая беседа	Выделять	2
2	Изучение теории	Кейс-метод, лекция с элементами проблемного обучения	Читать Объяснять	10
3	Разбор задания базового уровня сложности	Кейс-метод, лекция с элементами проблемного обучения	Описывать Формализовывать	5
4	Самостоятельное решение учащимися разноуровневых заданий, подготовка мини-проектов	Разноуровневые задания по теме занятия	Прогнозировать Формализовывать Моделировать Создавать Корректировать Использовать	20
5	Подведение итогов занятия, рефлексия	Беседа	Оценивать	3

При оценивании индивидуально-групповых проектов, выполненных в рамках модуля курса, можно использовать критерии, приведенные в таблице 3.

При оценивании выставляются:

0 баллов — действие не выполнено;

1 балл — присутствуют отдельные элементы;

2 балла — действие выполнено с отдельными недостатками;

3 балла — действие выполнено верно.

По результатам оценивания формируется итоговый балл, позволяющий оценить общую сформированность УУД у учащегося.

Результат оценивания в соответствии с выделенными критериями заносится в общую таблицу сформированности УУД. Оценка работы учащегося производится на протяжении всех занятий по модулю и курсу, что позволяет отслеживать изменение в уровне развития УУД у обучающихся. Модульная система построения курса, в свою очередь, дает возможность поэтапно проследить динамику развития каждого вида УУД.

Рассмотрим рекомендуемое для реализации модели программное обеспечение.

Для первого знакомства с процессом создания игр и основными алгоритмическими конструкциями идеально подойдет среда программирования **Blockly**.

Организация code.org разработала на базе Blockly ряд курсов для начального обучения программированию. Это бесплатное и открытое программное обеспечение, доступное онлайн и не требующее установки на компьютер. Возможна работа на планшетных компьютерах и смартфонах. Среда, размещенная на сайте <https://studio.code.org/>, может быть рекомендована на начальном этапе обучения программированию. Она имеет мультиязычный интерфейс, включая русский язык. После прохождения курсов можно получить именной сертификат. В стартовом пакете, который сейчас русифицирован, есть несколько приложений, например, «Лабиринт» — для обучения азам управляющих конструкций, графическая «Черепашка» и т. д. Важная особенность графической среды — освобождение от заботы о синтаксисе, что позволяет начинающему программисту сосредоточиться на логике программы. Существует возможность зарегистрироваться в роли учителя или ученика. Учитель может отслеживать степень прохождения учащимися курса. Построен вариант учебного курса на базе Blockly (20 этапов с заданиями на все алгоритмические конструкции), с помощью которого можно вести занятия с детьми. Также есть возможность изучения отдельных модулей. Модуль «Программирование с Анной и Эльзой» содержит 20 головоломок, решая которые ученик знакомится

Таблица 3

Критерии оценки УУД по итогам анализа описаний и защиты проектов

№ п/п	Критерий оценивания проекта	УУД, формирование которого проверяется	Балл
1	Определены цели и задачи проекта	Планировать и <i>прогнозировать</i> результаты своей деятельности по разработке программ или устройств	
2	Определены алгоритмические структуры, используемые в программе	<i>Называть</i> алгоритмы, используемые при разработке программы	
3	Описан алгоритм работы программы. Построена блок-схема алгоритма	<i>Формализовывать</i> построенную информационную модель программы (устройства) согласно поставленным задачам	
4	Разработанная программа соответствует заданию и заявленному алгоритму	Осознанно <i>читать</i> требования, предъявляемые к программе или устройству <i>Выделять</i> учебную информацию, соответствующую теме урока, критерии для оценивания информации и достоинства и недостатки отобранной информации	
5	Представленная программа работает в соответствии с описанным алгоритмом	<i>Создавать</i> программу или устройство согласно поставленным условиям	
6	Предложены элементы, улучшающие работу программы	<i>Моделировать</i> процесс своей деятельности для разработки программы или устройства в соответствии с поставленными целями	
7	Реализованы элементы, улучшающие работу программы	<i>Корректировать</i> свою деятельность по конструированию программы или устройства	
8	Проведен анализ результатов доработки или Присутствует анализ работы программы	<i>Оценивать</i> качество программы или устройства интуитивно и в соответствии с требованиями, принципами, целями учебной деятельности	
9	Описаны способы использования разработанной программы	<i>Использовать</i> разработанную программу или устройство для решения задач повседневной жизни	
10	Присутствует описание проекта в соответствии с предложенным планом	<i>Описывать</i> разработанную программу или устройство	
11	Подготовлена и проведена презентация проекта	<i>Объяснять</i> требования, предъявляемые к программе или устройству, необходимость и значимость их выполнения	
ИТОГО за проект (максимум 33 балла)			

с формальным исполнением алгоритмов и основными алгоритмическими конструкциями. Модули «*Порхающий код*» и «*Лаборатория игр*» содержат по 10 уровней. Проходя первые восемь уровней, учащиеся знакомятся с процессом создания игр, на 9-м и 10-м уровнях создают собственную игру — сначала по предложенному, а затем и по собственному сценарию. Созданная игра будет доступна всем желающим через сеть Интернет.

Scratch — объектно-ориентированный язык с возможностью создавать многопоточные программы. С практической точки зрения это простой в изучении, красивый, мощный инструмент программирования. Писать программу можно сразу, через десять минут знакомства с основами языка. Scratch не поощряет плохого стиля программирования. Напротив, «правильные» программы на Scratch писать проще, чем «неправильные». Графика, анимация, музыка, видеоэффекты и в то же время классическое событийно управляемое объектно-ориентированное и параллельное программирование — вот неполный перечень возможностей данного языка. Как показывает опыт применения, учащиеся после первых двух уроков могут приступить к разработке собственной компьютерной игры. В помощь учителю при разработке занятий имеется большой выбор методических материалов, размещенных в сети Интернет. Также предусмотрена публикация готовых проектов на официальном сайте программы. Помимо всего прочего Scratch позволяет создавать игры с различными сюжетами. Доступно много книг и пособий по этому языку, в которых есть подробные описания игр. Также можно познакомиться с проектами, реализованными участниками сообщества Scratch [2].

Для модуля курса «Создаем игру в конструкторе игр» можно выбрать один из следующих программных продуктов: Construct, Game Maker, Unity 3D, Kodu. Отметим, что все эти продукты имеют русскоязычный интерфейс и доступные системные требования.

Construct — это конструктор 2D-игр, он имеет открытый исходный код, и, следовательно, его можно абсолютно бесплатно использовать для разработки игр для Windows. Программа основана на системе событий и не требует программирования. Игры собираются путем расстановки игровых объектов в визуальном редакторе уровня. Редактор событий показывает список действий, которые должны происходить при взаимодействии игровых объектов или выполнении определенных условий в игре.

Создание игры в **Game Maker** не требует предварительного знакомства с каким-либо из языков программирования. Интерфейс программы объединяет в себе редакторы спрайтов, объектов, комнат, скриптов, а также таймлайнов (последовательностей действий с привязкой по времени) и путей (маршрутов) движения. Игра в Game Maker строится как набор игровых объектов. За их внешний вид отвечают спрайты, а поведение задается путем описания реакций на события. Для этого можно использовать графическое представление программ (близкое к блок-схемам) в виде последовательности иконок-действий. Программирование с помощью действий происходит в режиме drag-and-drop. Game Maker имеет возможность разработки игр для мобильных устройств.

Unity 3D представляет собой инструмент для разработки двух- и трехмерных приложений и игр, работающий под операционными системами Windows и OS X. Созданные с помощью Unity 3D приложения работают под операционными системами Windows, OS X, Windows Phone, Android, Apple iOS, Linux, а также на игровых приставках Wii,

PlayStation 3, PlayStation 4, Xbox 360, Xbox One. Есть возможность создавать приложения для запуска в браузерах с помощью специального подключаемого модуля Unity Web Player, а также с помощью реализации технологии WebGL.

Kodu — это визуальный язык программирования от компании Microsoft, который был разработан специально для создания простых игр пользователями, не имеющими навыков разработки программ, а также для знакомства с основами объектно-ориентированного программирования и алгоритмизации. В Kodu вы можете создавать визуальные сцены и игровые миры, добавлять объекты и 3D-модели, а затем настраивать их поведение и взаимодействие, управлять ими с клавиатуры или геймпада. В вашем арсенале имеются инструменты для добавления моделей, создания как окружающего ландшафта, так и активных зон, окна построения алгоритмов, а также большое количество примеров и обучающих материалов. В программе два режима работы: редактирование и проигрывание сцены. В первом режиме можно менять игровой мир, добавлять элементы, программировать поведение объектов. В нем отображается панель инструментов для работы с игровой зоной, при выборе инструмента в правой части экрана отображается набор опций, которые можно использовать, нажимая соответствующие клавиши. Во втором режиме в созданном мире можно играть. Настройка поведения объектов происходит в окне программирования, здесь есть знакомые программистам конструкции if...then...else и др., а также меню настройки всевозможных параметров. Весь «код» выглядит как визуальные блоки, которые можно перемещать, связывать между собой и менять их последовательность.

App Inventor 2 — среда визуальной разработки, позволяющая легко создать приложение для Android-устройств, даже не обладая знаниями в области программирования. Работает эта среда прямо из браузера, скачивать и устанавливать ничего не нужно. Полученный результат можно просматривать на Android-устройстве. Готовые приложения можно размещать в Play Market. App Inventor 2 поддерживает русский язык. В онлайн-редакторе MIT App Inventor 2 приложения строятся на базе стандартных компонентов, которые являются основным элементом разработки Android-приложений. Вопросы методики использования данного программного продукта рассмотрены в статье [4].

Завершить курс обучения программированию можно модулем, посвященным языку **Python**. К достоинствам языка можно отнести: кроссплатформенность, бесплатность среды, простой синтаксис. Разнообразные возможности позволяют записывать программы очень кратко, но в то же время понятно. По простоте освоения язык сравним с Бейсиком, но куда более богат возможностями и значительно более современен. В Python имеется богатая стандартная библиотека, есть возможность разработки промышленных приложений (для работы с сетью, GUI, базами данных и т. д.).

С методическими материалами по вопросам использования описанных выше программных продуктов, можно познакомиться в рамках работы Сетевой методической мастерской «Организация пропедевтики обучения программированию во внеурочной деятельности с учащимися 5–8-х классов» [1].

В заключение необходимо отметить, что раннее обучение программированию позволит заложить у учащихся основы алгоритмического мышления и создаст необходимую базу для успешного освоения программирования

в основной школе, повысит результаты учащихся в процессе итоговой аттестации в основной и старшей школе.

Список использованных источников

1. Материалы Сетевой методической мастерской «Организация пропедевтики обучения программированию во внеурочной деятельности с учащимися 5–8-х классов». <http://net-edu.ru/node/107079>
2. Сетевое сообщество программистов на Scratch. <https://scratch.mit.edu>

3. Филиппов В. И. Анализ результатов государственной итоговой аттестации по информатике и ИКТ на территории Московской области в 2016 году // Анализ результатов государственной итоговой аттестации по основным образовательным программам на территории Московской области в 2016 году. М.: АСОУ, 2016.

4. Шутикова М. И., Филиппов В. И. Использование среды APP INVENTOR при обучении программированию в процессе внеурочной деятельности в 6–7-х классах // Академический вестник. 2016 г., ноябрь. М.: АСОУ, 2016.

НОВОСТИ

С помощью Kinect построили трехмерную модель черепа тираннозавра

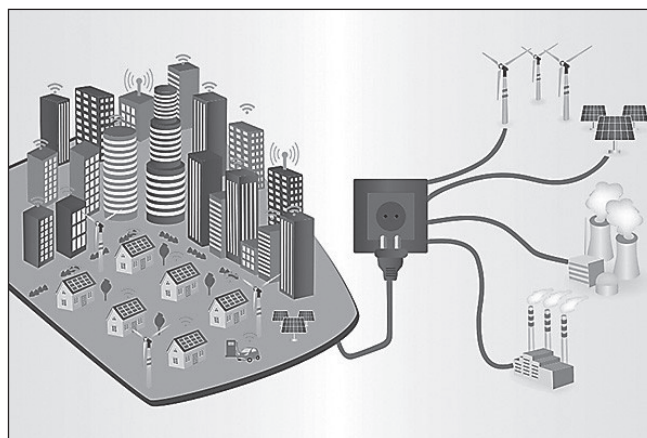
Группа судебно-медицинских экспертов-стоматологов добилась разрешения провести трехмерное сканирование знаменитого черепа тираннозавра, хранящегося в музее естественной истории в Чикаго. В его челюсти имеются отверстия, происхождение которых до сих пор остается неясным; предполагалось, что это следы укуса другого тираннозавра или результат инфекции. Однако для стоматологических сканеров челюсть оказалась слишком велика, и эксперты обратились к ученым из МТИ, создавшим прототип системы для 3D-сканирования с высоким разрешением на основе контроллера Microsoft Kinect с камерами и датчиком глубины. Профессиональные 3D-сканеры стоят десятки тысяч долларов и обеспечивают разрешение по глубине в 50–100 микрометров. С помощью Kinect можно добиться разрешения 500 микрометров, и стоит он всего около ста долларов.

А такого разрешения палеонтологам достаточно, чтобы рассмотреть отверстия в челюсти тираннозавра.



В МТИ предлагают использовать термостаты и электромобили в качестве «виртуального аккумулятора»

Чтобы справляться со скачками нагрузки в электросети, поставщики электроэнергии могли бы пользоваться батареями, но существующие технологии аккумуляторов не дают экономических преимуществ по сравнению с применением резервных электростанций. В МТИ предлагают рассматривать совокупность техники умного дома и подключающихся к зарядным станциям электромобилей в качестве виртуальной батареи. Идею исследователи объясняют на примере офисного здания: поставщик электроэнергии мог бы удаленно менять установленную на термостате температуру в пределах плюс-минус полградуса. Если повысить ее, здание будет потреблять больше энергии, и это соответствует «зарядке» виртуальной батареи, а если понизить, то отдаче запасенной энергии в сеть. Менять свойства «батареи» можно также, варьируя время и темпы зарядки электромобилей, припаркованных у офиса.



Искусственный интеллект победил чемпиона Dota 2

Разработанная в компании OpenAI система искусственного интеллекта победила 11 августа на ежегодном турнире The International 2017 чемпиона Dota 2 Даниила Ишутина.

Ишутин (Dendi), один из наиболее высокооплачиваемых профессиональных геймеров в мире, проиграл в первом матче и сдался во втором. После проигрыша он отметил, что своим стилем игры бот немного напоминает человека, но все же отличается от него. В OpenAI отметили, что перед матчем в течение нескольких недель обучали свою программу.

Компания OpenAI была основана Илоном Маском совместно с главой венчурного фонда Y Combinator Сэмом Альтманом в 2015 году. Цель компании — создание безопасного для людей искусственного интеллекта. Маск, создатель компании Tesla, неоднократно высказывался о потенциальных угрозах, которые несет человечеству искусственный интеллект.

The International — самый крупный турнир по компьютерной многопользовательской командной игре Dota 2.

(По материалам международного компьютерного еженедельника «Computerworld Россия»)