

АНО ДО «Центр развития «ЭВРИКА»

ЭВРИКА
центр
развития

«Согласовано»

Проректор по учебной работе КемГУ



Р.М. Котов
2018

«Утверждено»

Директор АНО ДО «Центр развития
«Эврика»

Асякина
Л.К. Асякина
2018



РАБОЧАЯ ПРОГРАММА ВНЕУРОЧНОЙ ДЕЯТЕЛЬНОСТИ

«Школа программирования (на языке Scratch)»

1-4 класс

Автор-составитель:

директор АНО ДО «Центр развития «Эврика»

Асякина Л.К.

Кемерово – 2018

Содержание

1. Результаты освоения курса	3
2. Содержание курса внеурочной деятельности с указанием форм организации и видов деятельности	4
3. Учебный план	8

1. Результаты освоения курса внеурочной деятельности

Основные личностные результаты, формируемые в процессе освоения программы– это:

формирование ответственного отношения к учению, способности довести до конца начатое дело на примере завершённых творческих учебных проектов;

формирование способности к саморазвитию и самообразованию средствами информационных технологий на основе, приобретённой благодаря иллюстративной среде программирования мотивации к обучению и познанию;

развитие опыта участия в социально значимых проектах, повышение уровня самооценки, благодаря реализованным проектам;

формирование коммуникативной компетентности в общении и сотрудничестве со сверстниками в процессе образовательной, учебно-исследовательской и проектной деятельности, участия в конкурсах и конференциях различного уровня;

формирование целостного мировоззрения, соответствующего современному уровню развития информационных технологий;

формирование осознанного позитивного отношения к другому человеку, его мнению, результату его деятельности;

развитие эстетического сознания через творческую деятельность на базе иллюстрированной среды программирования.

К основным метапредметным результатам (осваиваемым обучающимися межпредметным понятиям и универсальным учебным действиям, способности их использования как в учебной, так и в познавательной и социальной практике), формируемые в процессе освоения программы, можно отнести:

умение самостоятельно ставить и формулировать для себя новые задачи, развивать мотивы своей познавательной деятельности;

умение самостоятельно планировать пути решения поставленной проблемы для получения эффективного результата, понимая, что в программировании длинная программа не значит лучшая программа;

умение оценивать правильность решения учебно-исследовательской задачи;

умение корректировать свои действия, вносить изменения в программу и отлаживать её в соответствии с изменяющимися условиями;

владение основами самоконтроля, принятия решений;

умение создавать, применять и преобразовывать знаки и символы, модели и схемы для решения учебно-исследовательских и проектных работ;

ИКТ-компетенцию;

умение сотрудничества и совместной деятельности со сверстниками в процессе проектной и учебно-исследовательской деятельности.

Основные предметные результаты, формируемые в процессе изучения программы направлены на:

осознание значения математики и информатики в повседневной жизни человека;

формирование представлений об основных предметных понятиях — «информация», «алгоритм», «модель» и их свойствах;

развитие логических способностей и алгоритмического мышления, умений составить и записать алгоритм для конкретного исполнителя, знакомство с основными алгоритмическими структурами — линейной, условной и циклической;

развитие представлений о числах, числовых системах;

овладение символьным языком алгебры, умение составлять и использовать сложные алгебраические выражения для моделирования учебных проектов, моделировать реальные ситуации на языке алгебры;

развитие пространственных представлений, навыков геометрических построений и моделирования таких процессов, развитие изобразительных умений с помощью средств ИКТ;

формирование информационной и алгоритмической культуры, развитие основных навыков использования компьютерных устройств и программ;

формирование умения соблюдать нормы информационной этики и права.

2. Содержание курса внеурочной деятельности с указанием форм организации и видов деятельности

Scratch – это творческая среда, разработанная специально для развития мышления, творческих и исследовательских способностей детей и подростков. Среда Scratch появилась в 2007 году под руководством профессора Митчелла Резника в исследовательской группе под названием Lifelong Kindergarten research group, которая существует при Массачусетском технологическом институте.

По поводу целей проекта Митчелл Резник сказал: «Это следующий шаг в генерации контента (материалов) пользователем. Нашей целью было расширить диапазон того, что дети могут создавать, совместно использовать и изучать. Работая над проектом в Scratch, дети учатся думать креативно (созидательно) и решать проблемы систематично – а это умения, которые являются критическими для достижения успеха в XXI веке».

Программа Scratch имеет понятный интерфейс, встроенный графический редактор, меню готовых программ (кирпичиков), широкие возможности работы с мультимедийными объектами.

Актуальность данной образовательной программы состоит в том, что мультимедийная среда Scratch позволяет сформировать у детей интерес к программированию, отвечает всем современным требованиям объектно-ориентированного программирования. Среда Scratch позволяет формировать

навыки программирования, раскрыть технологию программирования. Изучение языка значительно облегчает последующий переход к изучению других языков программирования. Преимуществом Scratch, среди подобных сред программирования, является наличие версий для различных операционных систем, к тому же программа является свободно распространяемой, что немало важно для образовательных учреждений России. Именно в настоящее время имеет смысл рассматривать программы с открытым кодом, что позволяет сформировать у учащихся более широкое представление о возможностях работы с цифровой техникой.

Аспект новизны заключается в том, что Scratch не просто язык программирования, а еще и интерактивная среда, где результаты действий визуализированы, что делает работу с программой понятной, интересной и увлекательной.

Особенность среды Scratch, позволяющая создавать в программе мультфильмы, анимацию и даже простейшие игры, делает образовательную программу «Увлекательное программирование» практически значимой для современного подростка, т.к. дает возможность увидеть практическое назначение алгоритмов и программ, что будет способствовать развитию интереса к профессиям, связанным с программированием.

Педагогическая целесообразность данной образовательной программы состоит в том, что изучая программирование в среде Scratch, у обучающихся формируется не только логическое мышление, но и навыки работы с мультимедиа; создаются условия для активного, поискового учения, предоставляются широкие возможности для разнообразного программирования.

Основной вид деятельности: игра. Также на занятиях практикуется учебная, познавательная и творческая деятельность.

Цели программы:

Развитие логического мышления, творческого и познавательного потенциала подростка;

Задачи:

Образовательные:

Овладение базовыми понятиями объектно-ориентированного программирования и применение их при создании проектов в визуальной среде программирования Scratch;

Приобщение обучающихся к новым технологиям, способным помочь им в реализации собственного творческого потенциала;

Развитие познавательной деятельности учащихся в области новых информационных технологий;

Совершенствование навыков работы на компьютере и повышение интереса к программированию.

Воспитательные:

Формирование культуры и навыки сетевого взаимодействия;

Способствование развитию творческих способностей и эстетического вкуса подростков;

Способствование развитию коммуникативных умений и навыков обучающихся.

Развивающие:

Способствование развитию логического мышления, памяти и умению анализировать;

Создание условия для повышения самооценки обучающегося, реализации его как личности;

Формирование потребности в саморазвитии;

Способствование развитию познавательной самостоятельности.

В результате обучения школьники получают представление о:

свободно распространяемых программах;

функциональном устройстве программной среды Scratch и основных структурных элементах пользовательского интерфейса;

назначении и использовании основных блоков команд, состояний, программ;

правилах сохранения документа и необходимости присвоения правильного имени;

возможности и способах отладки написанной программы;

сущности понятий «спрайт», «сцена», «скрипт»;

исполнителях и системах их команд, возможности непосредственного управления исполнителем;

наличии заготовок для персонажей и сцен в соответствующих библиотеках, иерархическом устройстве библиотек и возможности импортирования их элементов;

возможности использования встроенного растрового редактора, наличии и назначении основных инструментов;

использовании других программ (например, LibreOfficeDraw) для создания собственных изображений;

алгоритме как формальном описании последовательности действий исполнителя, приводящих от исходных данных к конечному результату;

использовании схематического описания алгоритма;

программном управлении исполнителем и линейных алгоритмах;

написании программ для исполнителей, создающих геометрические фигуры на экране в процессе своего перемещения;

необходимости программного прерывания;

использовании циклических команд при необходимости повторений однотипных действий;

видах циклических алгоритмов и их применении;

достижении эффекта перемещения путем использования циклов;

возможности распараллеливания однотипных действий за счёт использования нескольких исполнителей;

организации интерактивности программ;
возможности взаимодействия исполнителей между собой, в различных слоях изображения;
видах и формах разветвленных алгоритмов, включая циклы с условием; управлении событиями.
использовании метода проектов для моделирования объектов и систем;
возможности описания реальных задач средствами программной среды;
создании анимационных, игровых, обучающих проектов, а также систем тестирования в программной среде Scratch.

Школьники будут уметь:

самостоятельно устанавливать программную среду на домашний компьютер;

изменять некоторые стандартные установки пользовательского интерфейса (например, язык отображения информации);

использовать различные способы отладки программ, включая пошаговую отладку;

уверенно использовать инструменты встроенного графического редактора, включая работу с фрагментами изображения и создание градиентов;

создавать собственные изображения в других программах (например, LibreOfficeDraw) и импортировать их в программную среду Scratch;

использовать графические примитивы векторного редактора LibreOfficeDraw для создания объектов;

создавать изображения из пунктирных и штрих-пунктирных линий с изменением цвета и толщины линии;

упрощать программы за счёт использования циклических команд и применять их;

составлять простые параллельные алгоритмы;

создавать программы и игры с использованием интерактивных технологий;

моделировать ситуации с использованием необходимых форм ветвления алгоритма, включая цикл по условию;

передавать сообщения исполнителям для выполнения последовательности команд (включая разные типы исполнителей).

планировать и создавать анимации по определенному сюжету;

создавать игры, используя интерактивные возможности программной среды Scratch;

планировать и создавать обучающие программы для иллюстрации пройденного материала других предметных областей;

продумывать и описывать интерактивное взаимодействие для создания простейших тренажеров;

подходить творчески к построению моделей различных объектов и систем.

Полученные по окончании программы знания и умения могут способствовать развитию интереса к профессиям, связанным с программированием, анимацией, мультипликацией.

На занятиях обращается внимание на соблюдение требований безопасности труда, пожарной безопасности и личной гигиены.

3. Учебный план

Раздел 1. Знакомство с программной средой Scratch

Свободное программное обеспечение. Авторы программной среды Scratch. Параметры для скачивания и установки программной среды на домашний компьютер.

Основные элементы пользовательского интерфейса программной среды Scratch. Внешний вид рабочего окна. Блочная структура систематизации информации. Функциональные блоки. Блоки команд, состояний, программ, запуска, действий и исполнителей. Установка русского языка для Scratch.

Создание и сохранение документа. Понятия спрайта, сцены, скрипта. Очистка экрана.

Основной персонаж как исполнитель программ. Система команд исполнителя (СКИ). Блочная структура программы. Непосредственное управление исполнителем.

Библиотека персонажей. Сцена и разнообразие сцен, исходя из библиотеки данных. Систематизация данных библиотек персонажей и сцен. Иерархия в организации хранения костюмов персонажа и фонов для сцен. Импорт костюма, импорт фона.

Аналитическая деятельность:

выделять аппаратное и программное обеспечение компьютера;
определять технические устройства для ввода и вывода информации;
понимать иерархическую организацию библиотеки данных программной среды;

выделять путь к элементам библиотеки;

выделять фрагменты изображения для дальнейшей работы с ними;

планировать работу по созданию сложных изображений путем копирования и масштабирования простых;

выбирать наиболее подходящий инструмент графического редактора для создания фрагмента изображения;

различать верхний и нижний цвета изображения;

придумывать и создавать различные градиенты для заливки замкнутой области;

планировать создание симметричных изображений.

Практическая деятельность:

выбирать и запускать программную среду Scratch;

работать с основными элементами пользовательского интерфейса программной среды;
изменять размер и перемещать окно программы, выбирать необходимый режим окна;
вводить имя файла с помощью клавиатуры;
выбирать необходимый файл из нужной папки библиотеки программы;
создавать, копировать, переименовывать, перемещать, копировать и удалять файлы;
соблюдать требования техники безопасности при работе в компьютерном классе.

Раздел 2. Компьютерная графика

Компьютерная графика. Векторные и растровые графические редакторы. Встроенный растровый графический редактор. Основные инструменты графического редактора — кисточка, ластик, заливка (цветом или градиентом), рисование линий, прямоугольников, квадратов, эллипсов и окружностей, выбор фрагмента изображения и отражение его по горизонтали или вертикали, использование инструмента печать для копирования выделенной области изображения, работа с текстом. Масштаб фрагмента изображения. Палитра цветов, установка цвета переднего плана и фона, выбор цвета из изображения с помощью инструмента пипетка. Изменение центра костюма. Изменение размера костюма.

Основные возможности изменения внешнего вида исполнителя: 1) использование встроенной библиотеки данных путём импорта её элемента; 2) редактирование выбранного элемента с помощью инструментов встроенного растрового графического редактора; 3) создание собственных изображений в других программах (например, LibreOfficeDraw) и импортирование их в программную среду Scratch.

Знакомство с основными графическими примитивами векторного редактора LibreOfficeDraw. Возможность создания геометрических фигур без внутренней заливки, но с текстовым блоком внутри. Стрелки, их направление.

Аналитическая деятельность:

выделять фрагменты изображения для дальнейшей работы с ними;
планировать работу по созданию сложных изображений путем копирования и масштабирования простых;
выбирать наиболее подходящий инструмент графического редактора для создания фрагмента изображения;
различать верхний и нижний цвета изображения;
придумывать и создавать различные градиенты для заливки замкнутой области;
планировать создание симметричных изображений.

Практическая деятельность:

использовать простейшие растровые и векторные редакторы для создания и редактирования изображений;
изменять центр изображения;

вносить изменения в изображения из встроенной библиотеки;
создавать сложные графические объекты путем копирования и модификации простых объектов и их фрагментов,
использовать возможности работы с цветом.

Раздел 3. Алгоритмы и исполнители

Алгоритм. Понятие алгоритма как формального описания последовательности действий исполнителя, приводящих от исходных данных к конечному результату. Схематическая запись алгоритма. Использование геометрических фигур для схематической записи алгоритма. Создание блок-схем в свободном векторном редакторе LibreOfficeDraw.

Линейные алгоритмы

Основные признаки линейного алгоритма. Схематическое описание линейного алгоритма. Геометрические примитивы, используемые для описания линейного алгоритма.

Программное управление исполнителем. Создание программ для перемещения исполнителя по экранному полю. Понятие поворота исполнителя в определенное направление. Прямой угол. Поворот исполнителя на прямой угол по часовой стрелке и против часовой стрелки.

Создание программ для рисования линий. Изменение цвета и толщины рисуемой линии. Особенности пунктирной линии. Написание программы для исполнителя, чтобы он оставлял пунктирную линию при перемещении по экранному полю.

Прямоугольник, квадрат — основные черты. Написание программ для движения исполнителя вдоль сторон квадрата, прямоугольника. Внесение изменений в программу рисования квадрата, если необходимо получить другой размер стороны квадрата.

Прерывание программы.

Циклические алгоритмы

Многократное повторение команд как организация цикла. Особенности использования цикла в программе. Упрощение программы путём сокращения количества команд при переходе от линейных алгоритмов к циклическим.

Схематическая запись циклического алгоритма.

Типы циклических алгоритмов. Основные конструкции программной среды, используемые для написания программ исполнителем с применением циклов.

Конечный цикл. Сокращение программы для исполнителя, рисующего линии, квадраты, прямоугольники при использовании цикла. Программа исполнителя для рисования нескольких однотипных геометрических фигур, например, нескольких квадратов из одной вершины, но с различным значением стороны.

Конструкции программной среды спрятаться/показаться. Выполнение программы исполнителем, не показанным на поле выполнения программы.

Написание и отладка программ с применением конструкции цикл в цикле.

Бесконечный цикл. Повторяющаяся смена внешности исполнителя для имитации движения персонажа. Использование бесконечного цикла для создания анимации.

Получение различного эффекта воспроизведения программы при изменении костюма исполнителя Scratch.

Параллелизм в программной среде

Использование нескольких исполнителей. Копирование программы одного исполнителя другим. Выполнение одинаковых программ разными исполнителями с использованием различных начальных условий. Параллельное выполнение однотипных действий. Принцип суперкомпьютерных технологий. Таймер для вычисления времени выполнения программы. Уменьшение показаний таймера при использовании параллельных вычислений.

Интерактивность программ. Возможность организации диалога между исполнителями. Операторы для слияния текстовых выражений.

Взаимодействие исполнителей путём касания друг друга или цвета. Использование сенсоров при взаимодействии исполнителей. Задержка выполнения программы.

Работа исполнителей в разных слоях изображения.

Ветвление в алгоритмах

Использование ветвления при написании программ. Короткая форма. Полная форма условного оператора. Конструкции ветвления для моделирования ситуации.

Цикл пока. Повторение команд исполнителя при выполнении определенного условия.

Последовательное выполнение фрагментов программы разными исполнителями

Типы исполнителей программной среды Scratch. Системы команд исполнителей. Различные системы команд для разных типов исполнителей.

Управление событиями. Передача сообщений исполнителям для выполнения определенной последовательности команд.

Передача управления между различными типами исполнителей.

Аналитическая деятельность:

придумывать задачи для исполнителей программной среды;

выделять ситуации, для описания которых можно использовать линейный алгоритм, алгоритм с ветвлениями, повторениями;

определять эффективный способ решения поставленной задачи;

находить параллельности в выполняемых действиях и программировать их с помощью нескольких исполнителей;

планировать последовательность событий для заданного проекта.

Практическая деятельность:

составлять и отлаживать программный код;

использовать конструкции программной среды для создания линейных, разветвленных и циклических алгоритмов;

организовывать параллельные вычисления;

организовывать последовательность событий программы, передачу управления от одних исполнителей другим.

Раздел 4. Проектная деятельность и моделирование процессов и систем

Мультимедийный проект. Описание сюжетных событий. Анимация. Создание эффекта анимации с помощью последовательной смены изображений. Имитационные модели. Интерактивные проекты. Игры.

Аналитическая деятельность:

создавать план появления событий для отражения определенной темы;

выбирать иллюстративный материал из встроенной библиотеки;

выбирать метод анимации для конкретной задачи;

планировать последовательность событий для создания эффекта анимации по выбранному сценарию.

Практическая деятельность:

использовать возможности программной среды Scratch для создания мультимедийных проектов;

создавать имитационные модели, интерактивные проекты и игры средствами программной среды.

Методическое обеспечение:

Для реализации программы используются следующие **методы обучения:**

По источнику полученных знаний: словесные, наглядные, практические.

По способу организации познавательной деятельности:

Развивающего обучения (проблемный, проектный, творческий, частично-поисковый, исследовательский, программированный)

Дифференцированного обучения (уровневые, индивидуальные задания)

Игровые (конкурсы, игры-конструкторы, турниры с использованием мультимедиа, дидактические)

Средства:

Дидактические материалы (опорные конспекты, проекты примеры, раздаточный материал для практических работ)

Методические разработки (презентации, видеоуроки, flash-ролики)

Сетевые ресурсы Scratch

Видеохостинг Youtub (видеоуроки «работа в среде Scratch»)

Учебно-тематический план

Аппаратное обеспечение:

Процессор не ниже Pentium II

Оперативная память не менее 512 Мб

Дисковое пространство не меньше 800 Мб

Монитор с 16-битной видеокартой

Разрешение монитора не ниже 800x600

Программное обеспечение:

Операционная система: Windows 7 или Windows 8

Open Office

Компьютерные программы: Scrath