**Муниципальное казенное общеобразовательное учреждение**

**«Средняя общеобразовательная школа №2 г. Олонца»**

**Согласовано: Принято: Утверждено:**

Заседание МС № 1 Педсовет № 1 Приказ № 337

от 30.08.2017 от 31.08.2017 от 31.08.2017

 **Директор школы**

 **\_\_\_\_\_\_\_\_ Жатикова Н.Н.**

**РАБОЧАЯ ПРОГРАММА
дополнительного образования**

 **«Юный программист»**

направленность научно- техническая

**для 5-6 классов**

**основного общего образования**

**Срок освоения: 1 учебный год**

Разработал: учитель информатики

Юрченко Наталья Андреевна

**2017 год**

**Общая характеристика курса дополнительного образования «Мир увлекательного программирования»**

Данная программа разработана для организации дополнительного образования учащихся 5-6 классов.

Курс построен таким образом, чтобы помочь учащимся заинтересоваться программированием и найти ответы на вопросы, с которыми им приходится сталкиваться в повседневной жизни при работе с большим объемом информации; научиться общаться с компьютером, который ничего не сможет сделать, если человек не напишет для него соответствующую программу.

Целесообразность изучения пропедевтики программирования в игровой, увлекательной форме, используя среду программирования Scratch обусловлена следующими факторами:

1. В основе Скретч лежит графический язык программирования, который позволяет контролировать действия и взаимодействия между различными типами данных. В среде используется метафора кирпичиков Лего, из которых даже самые маленькие дети могут собрать простейшие конструкции. Но, начав с малого, можно дальше развивать и расширять свое умение строить и программировать.
2. Существенной ролью изучения программирования и алгоритмизации в развитии мышления, формировании научного мировоззрения школьников именно этой возрастной группы.
3. В результате изучения данного курса обучающиеся развивают логическое мышление, получают представление об информационных моделях; учатся применять приобретенные знания при решении различных задач и оценивать полученные результаты; овладевают умениями решения учебных задач; развивают интуицию; получают представление об основных информационных процессах в реальных ситуациях.
4. Занятия по программе «Юный программист» подготовит их к более успешному усвоению базового курса «Информатика и ИКТ» в старших классах, обеспечат осознание значения информатики в повседневной жизни человека, понимание роли информационных процессов в современном мире.

**Сроки реализации программы:**

Программа направлена на общеинтеллектуальное развитие обучающихся. На реализацию программы отводится 17 часов.

**Результаты освоения программы**

Личностные результаты

– 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 компетенности 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.

Метапредметные результаты

– 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 ее 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.

Предметные результаты

– 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.

**Содержание курса**

**Раздел 1**. Знакомство с программной средой Scratch – 2 ч.

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 дальней 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.** Алгоритмы и исполнители – 14 ч.

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.

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 исполнителям 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 одинотипных действий. Принцип суперкомпьютерных технологий. Таймер для вычисления времени выполнения программы. Уменьшение показаний таймера при использовании параллельных вычислений.

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.

**Список литературы:**

1. Босова Л.Л., Сорокина Т.Е. Методика применения интерактивных сред для обучения младших школьников программированию. // Информатика и образование. 2014. № 7.
2. Сорокина Т.Е. Развитие алгоритмического мышления школьников с использованием среды программирования SCRATCH: Мат. Конф./Междунар. научно-практич. конф. 1 апреля 2013 г. в 6 частях. Часть III. Мин-во обр и науки. М.: АР-Консалт, 2013. С. 39–40.
3. Сорокина Т.Е. Пропедевтика программирования в курсе информатики 5–6 классов на базе среды SCRATCH: Мат. Конф./XII открытая всерос. конф. «Преподавание информационных технологий в Российской Федерации»: типография Издательства Казанского университета. 420008, г. Казань, ул. Профессора Нужина, 1/37, 2014.

**Тематическое планирование.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Тема занятия | **Основные виды учебной деятельности** | Кол. часов | **Форма обучения** |
| 1 | Устройство компьютера. Правила техники безопасности | Соблюдать требования безопасности и гигиены в работе со средствами ИКТ. Выполнять требования к организации компьютерного рабочего места, Участвовать в викторине «Что мы знаем о ПК?» | 1 | Коллективная |
| 2 | Знакомство со средой Scratch. Внешний вид среды, поля. Анимация. Исполнитель Scratch, цвет и размер пера. | Знакомиться с основными элементами интерфейса программы Скретч.Знакомиться с основными группами команд. Мини-проект «Аквариум**»** | 1 | КоллективнаяИндивидуальная |
| 3 | Понятие алгоритма. Линейный алгоритм. Рисование линий и квадратов исполнителем Scratch. | Составлять линейные алгоритмы. Знакомство с командами группы «Перо». Изменять параметры команд. Изменение костюма спрайта. Рисуем квадраты и линии. | 1 | КоллективнаяИндивидуальная |
| 4 | Конечный цикл. Исполнитель Scratch рисует квадраты, линии. Копирование фрагментов программы | Сохранять готовые программы. Рисовать блок схемы циклического алгоритма. Использование команды поворота на 90 градусов. Оптимизировать линейный алгоритм за счёт использования циклов. | 1 | Коллективная. Индивидуальная |
| 5 | Циклический алгоритм. Цикл в цикле. Вложенные и внешние циклы. | Использовать операции цикла в цикле для решения учебных задач; тренировка навыка использования поворота на прямой угол (90º) по часовой и против часовой стрелки; формирование навыка оптимизации алгоритма за счёт использования конструкции «цикл в цикле» в программной среде Scratch; | 1 | КоллективнаяИндивидуальная |
| 6 | Цикл в цикле. Повторение пунктирной линии с поворотом. Блок-схема цикла. | Совершенствование навыка использования конструкции «цикл в цикле» при решении учебных задач; создание и реализация алгоритма рисования квадрата несплошными линиями; совершенствование навыка оптимизации алгоритма за счёт использования конструкции «цикл в цикле» в программной среде Scratch  | 1 | КоллективнаяИндивидуальная |
| 7 | Бесконечный цикл. Анимация исполнителя Scratch на основе готовых костюмов. | Использовать операцию копирования внешности исполнителя путём копирования костюмаСоздание анимации за счёт смены костюма исполнителя. Интерактивное взаимодействие с исполнителем | 1 | Коллективная Индивидуальная |
| 8 | Сцена как исполнитель. Создаем модель таймера. | Использовать сцену в качестве исполнителя программной среды, Создавать отдельные программы для сцены. | 1 | КоллективнаяИндивидуальная |
| 9 | Бесконечный цикл. Одна программа для исполнителя Scratch, но разные костюмы. | Использование бесконечного цикла; смена костюмов спрайта | 1 | КоллективнаяИндивидуальная |
| 10 | Одинаковые программы для несколько исполнителей. Параллельное выполнение действий для ускорения процесса выполнения программы. | Создание анимации с использованием команд поворота на указанный угол и создание новых спрайтов путем дублирования**.** | 1 | Коллективная Индивидуальная |
| 11 | Разбиение программы на части для параллельного выполнения исполнителями. Таймер. Уменьшение показаний таймера при параллельных вычислениях. | Программировать синхронную работу разных исполнителей; таймер; изменение параметров таймера. | 1 | КоллективнаяИндивидуальная |
| 12 | Два исполнителя со своими программами. Мини-проект «Часы». | Создание программ с использованием таймерами; планирование действий разных исполнителей во времени для решения задачи; мини-проект «Часы**»** | 1 | КоллективнаяИндивидуальная |
| 13 | Алгоритмы с ветвлением. Условие ЕСЛИ. Два исполнителя. | Познакомиться с ветвлениями в алгоритмах; полная и сокращённая форма условия «если»; рисовать блок схемы с ветвлением; использовать команды для организации ветвления в программной среде Scratch; сенсоры касания. | 1 | Коллективная Индивидуальная |
| 14 | Цикл при условии. Мини-проект «Шарики в лабиринте» | Использование различных сенсоров касания и циклов с условием. Создавать мини-проект «Шарики в лабиринте» | 1 | КоллективнаяИндивидуальная |
| 15 | Цикл при условии. Исполнитель определяет цвета. | Использование датчика случайных чисел и цикла с условием в творческих проектах.  | 1 | КоллективнаяИндивидуальная |
| 16 | Цикл при условии. Исполнители в разных слоях. Мини-проект «Самолет сквозь облака». | Слои в Scratch; перемещение исполнителей в разные слои; использовать цикл с условием. Создавать анимацию «Самолёт сквозь облака» | 1 | КоллективнаяИндивидуальная |
| 17 | Перемещение исполнителя из одного слоя в другой. Действия исполнителей в разных слоях. Мини-проект «Дорога». | Исполнители в разных слоях, использовать цикл с условием; программировать действия исполнителей в разных слоях. Создать мини-проект «Дорога» | 1 | КоллективнаяИндивидуальная |
| **Всего:** | **17** |