


Краснодарский край
Муниципальное образование город Новороссийск

Муниципальное бюджетное общеобразовательное учреждение
средняя общеобразовательная школа № 24 станицы Раевской
муниципального образования город Новороссийск

Утверждено

решением педагогического совета
от 31.08 2020 года протокол № 1
Председатель  Н.А. Голеницкая
Подпись руководителя ОУ Ф.И.О

**ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ
ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА
ТЕХНОЛОГИЧЕСКОЙ НАПРАВЛЕННОСТИ**

«Основы программирования на языке Python на пример
программирования беспилотного летательного аппарата»

Уровень программы: ознакомительный

Направленность: техническая

Срок реализации: 68 часов

Возрастная категория: 12-18 лет

Составитель:
Макутов Тимур Ремзиевич
Педагог дополнительного образования

Новороссийск, 2020

Оглавление

| | |
|--|----|
| Оглавление | 2 |
| 1. Пояснительная записка | 3 |
| 2. Цель и задачи программы | 3 |
| 3. Содержание программы курса..... | 4 |
| 4. Прогнозируемые результаты и способы их проверки | 5 |
| 5. Календарный учебный график | 6 |
| 6. Условия реализации программы | 10 |
| 7. Формы аттестации | 12 |
| 8. Перечень рекомендуемых источников | 12 |

1. Пояснительная записка

Актуальность: в настоящее время процесс информатизации проявляется во всех сферах человеческой деятельности. Создание, внедрение, эксплуатация, а также совершенствование информационных технологий немислимо без участия квалифицированных специалистов, что обосновывает внедрение данного курса в учебный процесс.

Программа учебного курса «Основы программирования на языке Python на примере программирования беспилотного летательного аппарата» на подготовку творческой, технической подкованной личности, которая обладает логическим и алгоритмическим мышлением. Личности, которая способна анализировать и решать задачи, работая в команде.

В рамках курса обучающиеся смогут освоить полный цикл разработки ПО: от идеи до готового продукта. Научатся применять передовые инструменты написания кода и системы командной разработки проектов.

Программа составлена на основе Общеобразовательная общеразвивающая программы технической направленности «Основы программирования на языке Python на примере программирования беспилотного летательного аппарата» г. Москва 2019 г.

2. Цель и задачи программы

Целью программы является обучение через Case-технологии профессиональным навыкам программирования (так называемым Hard-skills), а также формирование универсальных компетенций (т.н. Soft-skills, социальные, интеллектуальные и волевые навыки: коммуникабельность, умение работать в команде, креативность, пунктуальность и уравновешенность).

Задачи программы (обучающие):

- изучить основы программирования: сформировать алгоритмическое мышление, научиться строить блок-схемы, ознакомиться с основными операторами языка программирования Python;
- более углублённо изучить функции и библиотеки Python на примере постепенно усложняющихся задач;

- привить навыки проектной деятельности;
- освоить инструменты командной разработки (Task-трекеры) и системы контроля версий (Git);
- применить полученные знания для программирования БПЛА;
- попрактиковать навыки пилотирования БПЛА;

Задачи программы (развивающие).

Способствовать:

- формированию заинтересованности обучающегося к техническим знаниям;
- развитию алгоритмического мышления, памяти;
- проявлению инициативы, изобретательности;
- увеличению эрудированности;
- развитию социальных навыков для взаимодействия с командой разработчиков.

А также ознакомить обучающегося с достижениями отечественной науки и техники для формирования здорового патриотизма.

Задачи программы (воспитательные):

- воспитать трудолюбие, уважение к труду;
- способствовать формированию положительной мотивации к трудовой деятельности;

3. Содержание программы курса

Программа предполагает постепенное расширение знаний и их углубление, а также приобретение умений в области проектирования, конструирования и изготовления творческого продукта.

В основе образовательного процесса лежит проектный подход. Основная форма подачи теории – интерактивные лекции и пошаговые мастер-классы в группах до 10 человек. Практические задания планируется выполнять как индивидуально, так и в рамках командной разработки. Занятия проводятся в виде бесед, семинаров и лекций с применением мультимедийных материалов (презентации, видеоролики, макеты приложений и т.п.).

4. Прогнозируемые результаты и способы их проверки

Личностные результаты:

- развитие любознательности и сообразительности при выполнении разнообразных заданий проблемного и эвристического характера;
- развитие самостоятельности суждений, независимости и нестандартности мышления;
- формирование коммуникативной компетентности в общении и сотрудничестве с другими обучающимися.

Мета-предметные результаты:

- умение планировать последовательность шагов алгоритма для достижения цели;
- умение ставить цель и планировать её достижение;
- умение вносить коррективы в действия в случае расхождения результата решения задачи на основе оценки и учёта характера сделанных ошибок;
- умение в сотрудничестве ставить новые учебные задачи;
- способность проявлять познавательную инициативу во время работы в команде;
- умение оценивать получающийся результат творческой деятельности и соотносить его с изначальным замыслом, а также вносить коррекции в сам замысел или в проектируемое ПО.

Познаваемые универсальные учебные действия:

- умение осуществлять поиск информации с применением информационных и коммуникационных технологий;
- умение ориентироваться в разнообразии способов решения задач;
- умение синтезировать, составлять целое из частей (в том числе, самостоятельно достраивать с восполнением недостающих компонентов).

Коммуникативные универсальные учебные действия:

- умение аргументировать свою точку зрения;
- умение выслушивать собеседника и вести с ним диалог;
- умение осуществлять постановку вопросов при поиске и сборе информации;
- умение с достаточной полнотой и точностью выражать свои мысли в соответствии с задачами и условиями коммуникации.

Предметные результаты.

В результате освоения программы обучающиеся должны знать:

- основные алгоритмические конструкции;
- принципы построения блок-схем;
- принципы структурного программирования на языке Python;
- что такое БПЛА и их назначение.

уметь:

- составлять алгоритмы для решения прикладных задач;
- реализовывать алгоритмы на компьютере в виде программ,

написанных на языке Python;

- применять различные библиотеки для решения поставленных задач (в том числе, применять графические библиотеки на подобии Tkinter для построения графических интерфейсов);

– отлаживать и тестировать программы, написанные на языке Python;

- пользоваться системами контроля версий (Git);
- настраивать и пилотировать БПЛА;
- представлять свой проект.

владеть:

– основной терминологией в области алгоритмизации и программирования;

- основными навыками программирования на языке Python;
- знаниями по устройству и применению БПЛА.

5. Календарный учебный график

| | Название раздела, темы | Содержание занятий |
|----|--|--|
| 1. | Введение в образовательную программу, техника безопасности (1 ч) | Теория: введение в образовательную программу. Ознакомление обучающихся с программой, приёмами и формами работы. Вводный инструктаж по ТБ. |
| 2. | Основы языка Python (4 ч) | Теория: история языка Python, сфера применения языка, различие в версиях, особенности синтаксиса. Объявление и использование переменных в Python. Использование строк, массивов, кортежей и |

| | | |
|-----|---|---|
| | | словарей в Python. Использование условий, циклов и ветвлений в Python. Практика: запуск интерпретатора. Различия интерпретатора и компилятора. Написание простейших демонстрационных программ. Мини-программы внутри программы. Выражения в вызовах функций. Имена переменных. Упражнения по написанию программ с использованием переменных, условий и циклов. Генерация случайных чисел. Группировка циклов в блоки. Операции сравнения. |
| 3. | Кейс 1. «Угадай число» | При решении данного кейса обучающиеся осваивают основы программирования на языке Python посредством создания игры, в которой пользователь угадывает число, заданное компьютером. Программа затрагивает много ключевых моментов программирования: конвертирование типов данных, запись и чтение файлов, использование алгоритма деления отрезка пополам, обработка полученных данных и представление их в виде графиков. |
| 3.1 | Введение в искусственный интеллект (4 ч) | Теория: алгоритмы поиска числа в массиве. Варианты сортировок. Поиск дихотомией. Работа с переменными, работа с функциями. Практика: упражнения по поиску чисел в массиве. Упражнения на сортировку чисел. Алгоритмы поиска числа. Исследование скорости работы алгоритмов. |
| 3.2 | Подготовка к публичному выступлению для защиты результатов (4 ч) | Теория: создание удобной и понятной презентации. Практика: подготовка презентации для защиты. Подготовка речи для защиты. |
| 4. | Кейс 2. «Спаси остров» | Кейс позволяет обучающимся поработать на языке Python со словарями и списками; изучить, как делать множественное присваивание, добавление элементов в список и их удаление, создать уникальный дизайн будущей игры. |
| 4.1 | Работа на языке Python со словарями и списками, множественное присваивание, добавление элементов в список и их удаление (2 ч) | Теория: знакомство с кейсом, представление поставленной проблемы. Доступ к элементам по индексам. Получение слова из словаря. Отображение игрового поля игрока. Получение предположений игрока. Проверка допустимости предположений игрока. Практика: мозговой штурм. Анализ проблемы, генерация и обсуждение методов её решения. |

| | | |
|-----|---|---|
| | | Создание прототипа программы. Отработка методик. |
| 4.2 | Планирование дизайна и механики игры (2 ч) | Теория: понятие «механика игры», ограничения, правила. Практика: упражнения. Проверка наличия буквы в секретном слове. Проверка — не победил ли игрок. Обработка ошибочных предположений. Проверка — не проиграл ли игрок. Завершение или перезагрузка игры. Создание главного меню игры, реализация подсчёта очков. |
| 4.3 | Визуализация программы в виде блок-схемы (2 ч) | Теория: проектирование проекта с помощью блок-схем. Практика: создание блок-схем. Ветвление в блок-схемах. Заканчиваем или начинаем игру с начала. Следующая попытка. Обратная связь с игроком. |
| 4.4 | Тестирование и доработка написанной программы (2 ч) | Практика: тестирование созданной игры-программы, доработка и расширение возможностей. |
| 4.5 | Подготовка к публичному выступлению для защиты результатов (2 ч) | Практика: подготовка презентации и речи для защиты. Презентация созданной программы. |
| 5. | Кейс 3. «Спаси остров 2» | Реализация игры «Спаси Остров 2» с использованием GUI-библиотеки Tkinter. |
| 5.1 | Постановка проблемы, генерация путей решения (2 ч) | Теория: знакомство с кейсом, представление поставленной проблемы. Практика: мозговой штурм. Анализ проблемы, генерация и обсуждение методов её решения. |
| 5.2 | Переработка псевдографического интерфейса игры в графический оконный с применением библиотеки Tkinter (4 ч) | Практика: реализация игры в виде графического оконного приложения, подготовка графических ассетов (вода, остров, деревья и т.п.). |
| 5.3 | Тестирование написанной программы и её доработка (2 ч) | Практика: связывание программной логики с получившимся графическим интерфейсом, тестирование работы получившейся игры |
| 5.4 | Подготовка к публичному выступлению для защиты результатов (2 ч) | Практика: подготовка презентации и речи для защиты. |
| 6. | Кейс 4. «Калькулятор» | При решении данного кейса учащиеся создают простое приложение-калькулятор: выполняют программную часть на языке программирования Python и создают интерфейс для пользователя при помощи GUI-библиотеки Tkinter. |
| 6.1 | Постановка проблемы, генерация путей решения | Теория: знакомство с кейсом, представление поставленной проблемы. |

| | | |
|-----|--|---|
| | (2 ч) | Практика: мозговой штурм. Анализ проблемы, генерация и обсуждение методов её решения. |
| 6.2 | Создание простейшего калькулятора с помощью библиотеки Tkinter (4 ч) | Практика: создание внешнего вида калькулятора. |
| 6.3 | Тестирование написанной программы и её доработка (1 ч) | Практика: связывание программной логики с получившимся графическим интерфейсом калькулятора. |
| 6.4 | Подготовка к публичному выступлению для защиты результатов (1 ч) | Практика: подготовка презентации и речи для защиты. |
| 7. | Распределённая система контроля версий Git | Системы контроля версий – важная составляющая процесса разработки, без которой невозможно представить ни одного серьёзного проекта. Учащемся предстоит ознакомиться с инструментарием и базовыми понятиями. |
| 7.1 | Клонирование репозитория (2 ч) | Практика: обучение клонированию репозитория Git. |
| 7.2 | Внесение изменений в рабочий каталог (1 ч) | Практика: внесение изменений в рабочий проект, формирование коммитов. |
| 7.3 | Обновление удалённого репозитория (1 ч) | Практика: публикация (push) коммитов, решение конфликтов перед их публикацией. |
| 7.4 | Вытягивание (1 ч) | Практика: загрузка актуальных изменений со стороны сервера. |
| 8. | Кейс 5. «Программирование автономных квадрокоптеров» | Роевое взаимодействие роботов является актуальной задачей в современной робототехнике. Квадрокоптеры можно считать летающей робототехникой. Шоу квадрокоптеров, выполнение задания БПЛА – такие задачи решаются с помощью роевого взаимодействия. Данный кейс посвящён созданию шоу коптеров из 3-х БПЛА, выполняющих полёт в автономном режиме. Обучающиеся получат первые навыки программирования технической системы на языке Python, познакомятся с алгоритмами позиционирования устройств на улице и в помещении, а также узнают о принципах работы оптического распознавания объектов. |
| 7.1 | Техника безопасности (1 ч) | Теория: знакомство с кейсом, представление поставленной проблемы, правила техники безопасности. Изучение конструкции квадрокоптеров. Практика: полёты на квадрокоптерах в ручном режиме. |
| 7.2 | Программирование взлёта и посадки БПЛА (2 ч) | Теория: основы программирования квадрокоптеров на языке Python. |

| | | |
|-----|---|---|
| | | Практика: тестирование написанного кода в режимах взлёта и посадки. |
| 7.3 | Выполнение команд «разворот», «изменение высоты», «изменение позиции» (3 ч) | Теория: теоретические основы выполнения разворота, изменения высоты и позиции на квадрокоптерах. Практика: тестирование программного кода в режимах разворота, изменения высоты и позиции. |
| 7.4 | Выполнение группового полёта вручную (1 ч) | Практика: выполнение группового полёта на квадрокоптере в ручном режиме. |
| 7.5 | Выполнение позиционирования по меткам (6 ч) | Теория: основы позиционирования indoor и outdoor квадрокоптеров. Практика: тестирование режима позиционирования по ArUco - маркерам. |
| 7.6 | Программирование группового полёта (6 ч) | Теория: основы группового полёта квадрокоптеров. Изучение типов группового поведения роботов. Практика: программирование роя квадрокоптеров для группового полёта. |
| 7.7 | Программирование роевого взаимодействия (4 ч) | Теория: основы программирования роя квадрокоптеров. Практика: Выполнение группового полета в автоматическом режиме. |
| | Итого: 68 часов | |

6. Условия реализации программы

Кадровые условия реализации программы

Требования к кадровым ресурсам:

- укомплектованность образовательного учреждения педагогическими, руководящими и иными работниками;
- уровень квалификации педагогических, руководящих и иных работников образовательного учреждения;
- непрерывность профессионального развития педагогических и руководящих работников образовательного учреждения, реализующего основную образовательную программу.

Компетенции педагогического работника, реализующего основную образовательную программу:

- обеспечивать условия для успешной деятельности, позитивной мотивации, а также само мотивирования обучающихся;

- осуществлять самостоятельный поиск и анализ информации с помощью современных информационно-поисковых технологий;
- организовывать и сопровождать учебно-исследовательскую и проектную деятельность обучающихся, выполнение ими индивидуального проекта;

- интерпретировать результаты достижений обучающихся;
- навык программирования на языке Python;
- использовать библиотеку Tkinter;
- навык создания компьютерных игр и приложений;
- проектирование интерфейса пользователей;

поиск и интеграция библиотек программного кода с открытых источников типа GitHub в собственный проект;

- навык работы в специализированном ПО для создания презентаций. Материально-технические условия реализации программы

Аппаратное и техническое обеспечение:

Рабочее место обучающегося:

- ноутбук: производительность процессора (по тесту PassMark — CPU BenchMark <http://www.cpubenchmark.net/>): не менее 2000 единиц; объём оперативной памяти: не менее 4 Гб; объём накопителя SSD/eMMC не менее 128 Гб (или соответствующий по характеристикам персональный компьютер с монитором, клавиатурой и колонками);

- мышь.

Рабочее место наставника:

- ноутбук: процессор Intel Core i5-4590/AMD) FX 8350 — аналогичная или более новая модель, графический процессор NVIDIA GeForce 970, AMD Radeon R9 290 — аналогичная или более новая модель, объём оперативной памяти: не менее 4 Гб, видеовыход HDMI 1.4, DisplayPort1.2 или более новая модель (или соответствующий по характеристикам персональный компьютер с монитором, клавиатурой и колонками);

- компьютеры должны быть подключены к единой сети Wi-Fi с доступом в интернет;

- презентационное оборудование (проектор с экраном) с возможностью подключения к компьютеру — 1 комплект;

- флипчарт с комплектом листов/маркерная доска, соответствующий набор письменных принадлежностей — 1 шт.;

- квадрокоптер DJI Ryze Tello — не менее 3 шт.;

- поле меток;
- Wi-Fi роутер. Программное обеспечение;
- офисное программное обеспечение;
- компилятор Python 3.5;
- веб-браузер;

7. Формы аттестации

Формы подведения итогов реализации общеобразовательной программы

Подведение итогов реализуется в рамках защиты группового проекта.

Формы демонстрации результатов обучения

Представление результатов образовательной деятельности пройдёт в форме публичной презентации решений кейсов командами и последующих ответов выступающих на вопросы наставника и других команд.

Формы диагностики результатов обучения

Беседа, тестирование, опрос.

8. Перечень рекомендуемых источников

1. Гин, А.А. Приёмы педагогической техники: свобода выбора, открытость, деятельность, обратная связь, идеальность: Пособие для учителей / А.А. Гин.
 - Гомель: ИПП «Сож», 1999. — 88 с.
2. Бреннан, К. Креативное программирование / К. Бреннан, К. Болкх, М. Чунг.
 - Гарвардская Высшая школа образования, 2017.
3. Лутц, М. Программирование на Python. Т. 1 / М. Лутц. — М.: Символ, 2016.
 - 992 с.
4. Лутц, М. Программирование на Python. Т. 2 / М. Лутц. — М.: Символ, 2016.
 - 992 с.
5. Понфиленок, О.В. Клевер. Конструирование и программирование квадрокоптеров / О.В. Понфиленок, А.И. Шлыков, А.А. Коригодский. — Москва, 2016.

6. Бриггс, Джейсон. Python для детей. Самоучитель по программированию / Джейсон Бриггс. — МИФ. Детство, 2018. — 320 с.
7. <https://github.com/dji-sdk/Tello-Python>
8. <https://dl-cdn.ryzero.com/downloads/tello/0222/Tello+Scratch+Readme.pdf>.

