

Принято решением
педагогического совета

Протокол № _____

от « _____ » _____ 2019г.

Утверждаю

И.о. директор МКОУ «Хосрехская СОШ им. Ахмедова Т.А.»

_____ Ахмедов Ш.А.

« _____ » _____ 2019г.



МКОУ Хосрехская СОШ им. Ахмедова Т.А.

Дополнительная общеразвивающая программа

**«Основы программирования на языке
Python»**

на 2019/ 2020 учебный год

Направленность: социально-педагогическое

Уровень: базовый

Срок реализации программы 2 года

Возраст обучающихся 9-12 лет

Автор-составитель:

Учитель начальных классов

Ахмедова Л.М.

**Кулинский район с. Хосрех
(месторасположение)**

I. Пояснительная записка	3
II. Учебно-тематический план	8
III. Содержание учебно-тематического плана	10
IV. Материально-технические условия реализации программы	22
V. Список литературы	23

I. Пояснительная записка

Актуальность: в настоящее время процесс информатизации проявляется во всех сферах человеческой деятельности. Использование современных информационных технологий является необходимым условием успешного развития как отдельных отраслей, так и государства в целом. Создание, внедрение, эксплуатация, а также совершенствование информационных технологий немислимо без участия квалифицированных и увлечённых специалистов, в связи с этим внедрение курса «Основы программирования на языке Python на примере программирования беспилотного летательного аппарата» в учебный процесс актуально.

Программа учебного курса «Основы программирования на языке Python на примере программирования беспилотного летательного аппарата» направлена на подготовку творческой, технически грамотной, гармонично развитой личности, обладающей логическим мышлением, способной анализировать и решать задачи в команде в области информационных и аэротехнологий, решать ситуационные кейсовые задания, основанные на групповых проектах.

Занятия по данному курсу рассчитаны на общенаучную подготовку обучающихся, развитие их мышления, логики, математических способностей, исследовательских навыков.

Учебный курс «Основы программирования на языке Python на примере программирования беспилотного летательного аппарата» направлен на изучение основ программирования на языке Python и программирование автономных квадрокоптеров.

В рамках курса «Основы программирования на языке Python на примере программирования беспилотного летательного аппарата» обучающиеся смогут познакомиться с физическими, техническими и математическими понятиями. Приобретённые знания будут применимы в творческих проектах.

Учебный курс «Основы программирования на языке Python на примере программирования беспилотного летательного аппарата» представляет собой самостоятельный модуль и содержит необходимые темы из курса информатики и физики.

Цель программы: освоение Hard- и Soft-компетенций обучающимися в области программирования и аэротехнологий через использование кейс-технологий.

Задачи:

Обучающие:

- изучить базовые понятия: алгоритм, блок-схема, переменная, цикл, условия, вычисляемая функция;
- сформировать навыки выполнения технологической цепочки разработки программ средствами языка программирования Python;
- изучить основные конструкции языка программирования Python, позволяющие работать с простыми и составными типами данных (строками, списками, кортежами, словарями, множествами);
- научить применять навыки программирования на конкретной учебной ситуации (программирование беспилотных летательных аппаратов на учебную задачу);
- развить навык пилотирования беспилотных летательных аппаратов (БПЛА) на практике;
- привить навыки проектной деятельности.

Развивающие:

- способствовать расширению словарного запаса;
- способствовать развитию памяти, внимания, технического мышления, изобретательности;
- способствовать развитию алгоритмического мышления;

- способствовать формированию интереса к техническим знаниям;
- способствовать формированию умения практического применения полученных знаний;
- сформировать умение формулировать, аргументировать и отстаивать своё мнение;
- сформировать умение выступать публично с докладами, презентациями и т. п.

Воспитательные:

- воспитывать аккуратность и дисциплинированность при выполнении работы;
- способствовать формированию положительной мотивации к трудовой деятельности;
- способствовать формированию опыта совместного и индивидуального творчества при выполнении командных заданий;
- воспитывать трудолюбие, уважение к труду;
- формировать чувство коллективизма и взаимопомощи;
- воспитывать чувство патриотизма, гражданственности, гордости за достижения отечественной науки и техники.

Прогнозируемые результаты и способы их проверки

Личностные результаты:

- критическое отношение к информации и избирательность её восприятия;
- осмысление мотивов своих действий при выполнении заданий;
- развитие любознательности, сообразительности при выполнении разнообразных заданий проблемного и эвристического характера;
- развитие внимательности, настойчивости, целеустремлённости, умения преодолевать трудности;
- развитие самостоятельности суждений, независимости и нестандартности мышления;
- освоение социальных норм, правил поведения, ролей и форм социальной жизни в группах и сообществах;
- формирование коммуникативной компетентности в общении и сотрудничестве с другими обучающимися.

Метапредметные результаты:

Регулятивные универсальные учебные действия:

- умение принимать и сохранять учебную задачу;
- умение планировать последовательность шагов алгоритма для достижения цели;
- умение ставить цель (создание творческой работы), планировать достижение этой цели;
- умение осуществлять итоговый и пошаговый контроль по результату;
- способность адекватно воспринимать оценку наставника и других обучающихся;
- умение различать способ и результат действия;
- умение вносить коррективы в действия в случае расхождения результата решения задачи на основе её оценки и учёта характера сделанных ошибок;
- умение в сотрудничестве ставить новые учебные задачи;
- способность проявлять познавательную инициативу в учебном сотрудничестве;
- умение осваивать способы решения проблем творческого характера в жизненных ситуациях;
- умение оценивать получающийся творческий продукт и соотносить его с изначальным замыслом, выполнять по необходимости коррекции либо продукта, либо замысла.

Познавательные универсальные учебные действия:

- умение осуществлять поиск информации в индивидуальных информационных архивах обучающегося, информационной среде образовательного учреждения, федеральных хранилищах информационных образовательных ресурсов;
- умение использовать средства информационных и коммуникационных технологий для решения коммуникативных, познавательных и творческих задач;
- умение ориентироваться в разнообразии способов решения задач;
- умение осуществлять анализ объектов с выделением существенных и несущественных признаков;
- умение проводить сравнение, классификацию по заданным критериям;
- умение строить логические рассуждения в форме связи простых суждений об объекте;

- умение устанавливать аналогии, причинно-следственные связи;
- умение моделировать, преобразовывать объект из чувственной формы в модель, где выделены существенные характеристики объекта (пространственно-графическая или знаково-символическая);
- умение синтезировать, составлять целое из частей, в том числе самостоятельно достраивать с восполнением недостающих компонентов.

Коммуникативные универсальные учебные действия:

- умение аргументировать свою точку зрения на выбор оснований и критериев при выделении признаков, сравнении и классификации объектов;
- умение выслушивать собеседника и вести диалог;
- способность признавать возможность существования различных точек зрения и права каждого иметь свою;
- умение планировать учебное сотрудничество с наставником и другими обучающимися: определять цели, функции участников, способы взаимодействия;
- умение осуществлять постановку вопросов: инициативное сотрудничество в поиске и сборе информации;
- умение разрешать конфликты: выявление, идентификация проблемы, поиск и оценка альтернативных способов разрешения конфликта, принятие решения и его реализация;
- умение с достаточной полнотой и точностью выражать свои мысли в соответствии с задачами и условиями коммуникации;
- владение монологической и диалогической формами речи.

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 предназначение.

уметь:

- составлять алгоритмы для решения прикладных задач;
- реализовывать алгоритмы на компьютере в виде программ, написанных на языке Python;
- применять библиотеку Tkinter;
- отлаживать и тестировать программы, написанные на языке Python;
- настраивать БПЛА;
- представлять свой проект.

владеть:

- основной терминологией в области алгоритмизации и программирования;
- основными навыками программирования на языке Python;
- знаниями по устройству и применению беспилотников.

Формы подведения итогов реализации дополнительной программы

Подведение итогов реализуется в рамках следующих мероприятий: тестирование по программированию на языке Python, защита результатов выполнения кейса № 4, групповые соревнования.

Формы демонстрации результатов обучения

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

Формы диагностики результатов обучения

Беседа, тестирование, опрос.

II. Учебно-тематический план

№ п/п	Название раздела, темы	Количество часов			Формы аттестации/ контроля
		Всего	Теория	Практика	
1.	Введение в образовательную программу, техника безопасности	1	1	-	Тестирование
2.	Основы языка Python. Примеры на языке Python с разбором конструкций: циклы, условия, ветвления, массивы, типы данных	2	1	1	Тестирование
3.	Кейс 1. «Угадай число»	4	2	2	Демонстрация решений кейса
3.1	Введение в искусственный интеллект. Примеры на языке Python с искусственным интеллектом по угадыванию чисел, метод дихотомии. Управление искусственным интеллектом	3	1	2	
3.2	Подготовка к публичному выступлению для защиты результатов. Демонстрация отчёта в группе и защита результатов работы	1	0,5	0,5	
4.	Кейс 2. «Спаси остров»	5	1	4	Демонстрация решений кейса
4.1	Работа на языке Python со словарями и списками, множественное присваивание, добавление элементов в список и их удаление	2	1	1	
4.2	Планирование дизайна и механики игры. Создание главного меню игры, подсчёта очков	2	0	2	
4.3	Тестирование написанной программы и доработка. Подготовка к публичному выступлению для защиты результатов. Демонстрация результатов работы	1	0	1	
5.	Кейс 3. «Калькулятор»	5	1	4	Демонстрация решений кейса
5.1	Постановка проблемы, генерация путей решения	1	0,5	0,5	
5.2	Создание простейшего калькулятора с помощью библиотеки Tkinter	2	-	2	
5.3	Тестирование написанной программы и доработка	1	-	1	
5.4	Подготовка к публичному выступлению для защиты результатов. Демонстрация результатов работы	1	0,5	0,5	
6.	Кейс 4. Программирование автономных квадрокоптеров	17	5	12	Демонстрация решений кейса

6.1	Техника безопасности при полётах. Проведение полётов в ручном режиме	1	0,5	0,5	
6.2	Программирование взлёта и посадки беспилотного летательного аппарата	2	0,5	1,5	
6.3	Выполнение команд «разворот», «изменение высоты», «изменение позиции»	3	1	2	
6.4	Выполнение группового полёта вручную	1	0	1	
6.5	Выполнение позиционирования по меткам	4	1	3	
6.6	Программирование группового полёта	3	1	2	
6.7	Программирование роевого взаимодействия	3	1	2	
Итого:		34	11	23	

III. Содержание учебно-тематического плана

№	Темы занятий	Содержание занятий
1.	Вводное занятие. Введение в предмет, 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. Примеры на языке Python с разбором конструкций: циклы, условия, ветвления, массивы, типы данных (2 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 «Угадай число»	
3.1	Введение в искусственный интеллект. Примеры на языке Python с искусственным интеллектом по угадыванию чисел, метод дихотомии. Управление искусственным интеллектом (3 ч)	<p>Теория: алгоритмы поиска числа в массиве. Варианты сортировок. Поиск дихотомией. Работа с переменными, работа с функциями.</p> <p>Практика: упражнения по поиску чисел в массиве. Упражнения на сортировку чисел. Алгоритмы поиска числа. Исследование скорости работы алгоритмов.</p>
3.2	Подготовка к публичному выступлению для защиты результатов. Демонстрация отчёта в группе и защита результатов работы (1 ч)	<p>Теория: создание удобной и понятной презентации.</p> <p>Практика: подготовка презентации для защиты. Подготовка речи для защиты.</p>
4	Кейс «Спаси остров»	
4.1	Работа на языке Python со словарями и списками, множественное присваивание, добавление элементов в список и их удаление (2 ч)	<p>Теория: знакомство с кейсом, представление поставленной проблемы.</p> <p>Доступ к элементам по индексам. Получение слова из словаря. Отображение игрового поля игрока. Получение предположений игрока. Проверка допустимости предположений игрока.</p> <p>Практика: мозговой штурм. Анализ проблемы, генерация и обсуждение методов её решения. Создание прототипа программы. Отработка методик.</p>
4.2	Планирование дизайна и механики игры. Создание главного меню игры, подсчёта очков (1 ч)	<p>Теория: понятие «механика игры», ограничения, правила.</p> <p>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.</p>
4.3	Тестирование написанной программы и доработка	Практика: тестирование созданной игры-программы, доработка и расширение возможностей.

	(1 ч)	
4.4	Подготовка к публичному выступлению для защиты результатов. Демонстрация результатов работы (1 ч)	Практика: подготовка презентации и речи для защиты. Презентация созданной программы.
5	Кейс «Калькулятор»	
5.1	Оформление проектной идеи. Формирование программы работ (0,5 ч)	Теория: знакомство с кейсом, представление поставленной проблемы. Практика: мозговой штурм. Анализ проблемы, генерация и обсуждение методов её решения.
5.2	Программа для работы калькулятора (1 ч)	Практика: написание программы для будущего калькулятора.
5.3	Создание внешнего вида калькулятора (1 ч)	Практика: создание внешнего вида калькулятора.
5.4	Тестирование написанной программы и доработка (1 ч)	Практика: тестирование созданной программы, доработка и расширение возможностей.
5.5	Подготовка к публичному выступлению для защиты результатов (1 ч)	Практика: подготовка презентации и речи для защиты.
5.6	Демонстрация результатов работы (0,5 ч)	Практика: презентация созданной программы.
6	Кейс «Программирование автономных квадрокоптеров»	
6.1	Техника безопасности при полётах. Проведение полётов в ручном режиме (1 ч)	Теория: знакомство с кейсом, представление поставленной проблемы, правила техники безопасности. Изучение конструкции квадрокоптеров. Практика: полёты на квадрокоптерах в ручном режиме.
6.2	Программирование взлёта и посадки беспилотного летательного аппарата (2 ч)	Теория: основы программирования квадрокоптеров на языке Python. Практика: тестирование написанного кода в режимах взлёта и посадки.
6.3	Выполнение команд «разворот», «изменение высоты»,	Теория: теоретические основы выполнения разворота, изменения высоты и позиции на квадро-

	«изменение позиции» (3 ч)	коптерах. Практика: тестирование программного кода в режимах разворота, изменения высоты и позиции.
6.4	Выполнение группового полёта вручную (1 ч)	Практика: выполнение группового полёта на квадрокоптере в ручном режиме.
6.5	Выполнение позиционирования по меткам (4 ч)	Теория: основы позиционирования indoor и outdoor квадрокоптеров. Практика: тестирование режима позиционирования по ArUco - маркерам.
6.6	Программирование группового полёта (3 ч)	Теория: основы группового полёта квадрокоптеров. Изучение типов группового поведения роботов. Практика: программирование роя квадрокоптеров для группового полёта.
6.7	Программирование роевого взаимодействия (3 ч)	Теория: основы программирования роя квадрокоптеров. Практика: Выполнение группового полета в автоматическом режиме.

IV. Содержание тем программы

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.

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.

Кейс 3. «Калькулятор»

При решении данного кейса учащиеся создают первое простое приложение калькулятор: выполняют программную часть на языке программирования Python и создают интерфейс для пользователя при помощи библиотеки Tkinter.

Кейс 4. Программирование автономных квадрокоптеров

Роевое взаимодействие роботов является актуальной задачей в современной робототехнике. Квадрокоптеры можно считать летающей робототехникой. Шоу квадрокоптеров, выполнение задания боевыми беспилотными летательными аппаратами - такие задачи решаются с помощью применения алгоритмов роевого взаимодействия.

Данный кейс посвящен созданию шоу коптеров из 3х бпла выполняющих полет в автономном режиме. Обучающиеся получают 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.

Кадровые условия реализации программы

Комплектование образовательной организации педагогическими, руководящими и иными работниками, соответствующими квалификационным характеристикам по соответствующей должности.

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 самомотивирования 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.

V. Материально-технические условия реализации программ- МЫ

Аппаратное и техническое обеспечение:

- Рабочее место обучающегося:
ноутбук: производительность процессора (по тесту PassMark - CPU BenchMark<http://www.cpubenchmark.net/>): не менее 2000 единиц; объем оперативной памяти: не менее 4 Гб; объем накопителя SSD/eMMC: не менее 128 Гб (или соответствующий по характеристикам персональный компьютер с монитором, клавиатурой и колонками).
- рабочее место преподавателя:
ноутбук: процессор Intel Core i5-4590/AMD FX 8350 аналогичная или более новая модель, графический процессор NVIDIA GeForce GTX 970, AMD Radeon R9 290 аналогичная или более новая модель, объем оперативной памяти: не менее 4 Гб, видеовыход HDMI 1.4, DisplayPort 1.2 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:

- компилятор Python 3.5;
- веб-браузер;
- пакет офисного ПО;
- текстовый редактор.

VII. Примерный календарный учебный график на 2019/2020 учебный год

Период обучения — сентябрь-май.

Количество учебных недель — 34.

Количество часов — 34.

Режим проведения занятий: 1 раз в неделю.

№ п/п	Месяц	Форма занятия	Кол-во часов	Тема занятия	Форма контроля
1.	сентябрь	Л/ПР	1	Введение в образовательную программу, техника безопасности	Тестирование
2.	сентябрь	Л/ПР	2	Основы языка Python. Примеры на языке Python с разбором конструкций: циклы, условия, ветвления, массивы, типы данных	Тестирование
3.	сентябрь	Л/ПР	1	Кейс 1. «Угадай число»	Беседа
4.	сентябрь	Л/ПР	1	Кейс 1. «Угадай число»	Беседа
5.	октябрь	Л/ПР	1	Кейс 1. «Угадай число»	Беседа
6.	октябрь	Л/ПР	1	Кейс 1. «Угадай число»	Демонстрация решений кейса
7.	октябрь	Л/ПР	1	Кейс 2. «Спаси остров»	Беседа
8.	октябрь	Л/ПР	1	Кейс 2. «Спаси остров»	Беседа
9.	ноябрь	Л/ПР	1	Кейс 2. «Спаси остров»	Беседа
10.	ноябрь	Л/ПР	1	Кейс 2. «Спаси остров»	Беседа
11.	ноябрь	Л/ПР	1	Кейс 2. «Спаси остров»	Демонстрация решений кейса
12.	ноябрь	Л/ПР	1	Кейс 3. «Калькулятор»	Беседа
13.	декабрь	Л/ПР	1	Кейс 3. «Калькулятор»	Беседа
14.	декабрь	Л/ПР	1	Кейс 3. «Калькулятор»	Беседа
15.	декабрь	Л/ПР	1	Кейс 3. «Калькулятор»	Беседа
16.	декабрь	Л/ПР	1	Кейс 3. «Калькулятор»	Демонстрация решений кейса

17.	январь	Л/ПР	1	Кейс 4. «Программирование автономных квадрокоптеров»	Беседа
18.	январь	Л/ПР	1	Кейс 4. «Программирование автономных квадрокоптеров»	Беседа
19.	январь	Л/ПР	1	Кейс 4. «Программирование автономных квадрокоптеров»	Беседа
20.	январь	Л/ПР	1	Кейс 4. «Программирование автономных квадрокоптеров»	Беседа
21.	февраль	Л/ПР	1	Кейс 4. «Программирование автономных квадрокоптеров»	Беседа
22.	февраль	Л/ПР	1	Кейс 4. «Программирование автономных квадрокоптеров»	Беседа
23.	февраль	Л/ПР	1	Кейс 4. «Программирование автономных квадрокоптеров»	Беседа
24.	февраль	Л/ПР	1	Кейс 4. «Программирование автономных квадрокоптеров»	Беседа
25.	март	Л/ПР	1	Кейс 4. «Программирование автономных квадрокоптеров»	Беседа
26.	март	Л/ПР	1	Кейс 4. «Программирование автономных квадрокоптеров»	Беседа
27.	март	Л/ПР	1	Кейс 4. «Программирование автономных квадрокоптеров»	Беседа
28.	март	Л/ПР	1	Кейс 4. «Программирование автономных квадрокоптеров»	Беседа
29.	апрель	Л/ПР	1	Кейс 4. «Программирование автономных квадрокоптеров»	Беседа
30.	апрель	Л/ПР	1	Кейс 4. «Программирование автономных квадрокоптеров»	Беседа

31.	апрель	Л/ПР	1	Кейс 4. «Программирование автономных квадрокоптеров»	Беседа
32.	апрель	Л/ПР	1	Кейс 4. «Программирование автономных квадрокоптеров»	Беседа
33.	апрель	Л/ПР	1	Кейс 4. «Программирование автономных квадрокоптеров»	Беседа
34.	май	Л/ПР	1	Кейс 4. «Программирование автономных квадрокоптеров»	Беседа
35.	май	Л/ПР	1	Кейс 4. «Программирование автономных квадрокоптеров»	Беседа
36.	май	Л/ПР	1	Кейс 4. «Программирование автономных квадрокоптеров»	Демонстрация решений кейса

VIII. Список литературы и методического материала

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.ryzerobotics.com/downloads/tello/0222/Tello+Scratch+Readme.pdf>.

ПРИЛОЖЕНИЕ

Кейс «Угадай число»

Текст кейса

“Программирование - вторая грамотность”- говорил великий академик и первый советский программист Андрей Петрович Ершов. И он был несомненно прав, ведь, для того, чтобы продолжить успешное изучение принципов работы компьютера и научиться решать интересные инженерно-технические задачи, изобретая полезные и неповторимые в современном мире вещи, необходимо научиться автоматизировать их работу при помощи программ, написанных на разных языках программирования. Для того, чтобы овладеть базовыми навыками программирования можно создать игру.

Описание

Компьютер загадывает секретное число от 0 до 100 и просит пользователя угадать это число. После каждой попытки угадать, компьютер будет сообщать пользователю, было его число больше или меньше загаданного. Пользователь выиграет, если угадает число за шесть ходов.

Программа затрагивает много ключевых моментов программирования: конвертирования типов данных, запись и чтение файлов, использование алгоритма деления отрезка пополам, обработку полученных данных и представление их в виде графиков.

Категория кейса: вводный, рассчитан на учащихся 8 класса

Вопросы к кейсу:

1. Какие виды алгоритмов вам известны?
2. Какие жанры компьютерных игр вы могли бы назвать?
3. Какие знания в программировании вам понадобятся для решения данного?

Место кейса в структуре программы: рекомендуется к выполнению после изучения основ языка программирования Python, основных алгоритмических понятий и конструкций.

Количество учебных часов: 8 часов

Учебно-тематическое планирование (занятие - 2 часа):

Занятие 1	Занятие 2	Занятие 3
-----------	-----------	-----------

Цель: Постановка проблемы, генерация путей решения		Цель: Разработка алгоритма поиска числа методом деления отрезка пополам		Цель: Представление данных в виде графиков в Python	
Что делается: Представление поставленной проблемы группе детей. Анализ проблемы, генерация и обсуждение методов ее решения.	Компетенции: Умение искать информацию в различных источниках. Умение генерировать идеи предложенными методами.	Что делается: Упражнения по поиску чисел в массиве. Упражнения на сортировку чисел. Алгоритмы поиска числа. Исследование скорости работы алгоритмов.	Компетенции: Умение программировать на языке Python.	Что делается: Преобразование текстовых данных в графики на Python через библиотеку matplotlib. Разбор элементов графика, управление цветом, опциями и свойствами элементов.	Компетенции: Умение программировать на языке Python.
Занятие 4					
Цель: Подготовка к публичному выступлению для защиты результатов. Демонстрация игры в группе и защита результатов					
Что делается: Подготовка речи и презентации для публичной демонстрации результатов работы в кейсе. Публичная презентация результатов работы. Рефлексия.	Компетенции: Умение анализировать результаты работы. Грамотное представление результатов своей деятельности. Базовые навыки ораторства, публичных выступлений, аргументирование точки зрения.				

Метод работы с кейсом: продвижение по шагам жизненного цикла игры

Минимально необходимый уровень входных компетенций:

- ✓ Работа с компьютером на уровне начинающего пользователя
- ✓ Знание математики на уровне выпускника начальной школы

Предполагаемые образовательные результаты учащихся:

Артефакты: готовый рабочий код

Формируемые навыки (soft skills):

- ✓ Развитие аналитического и алгоритмического мышления;
- ✓ Умение аргументировать свою точку зрения и отстаивать ее;

Формируемые навыки (hard skills):

- ✓ Понятие алгоритма, последовательного выполнения действий;
- ✓ Понятие программы, подпрограммы, цикла, условия;
- ✓ Умение создавать программы и игры в Python;
- ✓ Умение создавать презентации.

Процедуры и формы выявления образовательного результата:

Представление результатов образовательной деятельности пройдет в форме

публичной презентации решений кейса командами и последующих ответов выступающих на вопросы наставника и других команд.

Педагогический сценарий (руководство для наставника):

Кейс представляет собой разработку игрового приложения для решения проблемной ситуации. Кейс включает в себя:

- ✓ Введение в проблему при помощи беседы с учащимися (приведение жизненных примеров);
- ✓ Групповое обсуждение проблемы, поиск путей решения, введение в группу мысли о решении проблемы через разработку игрового приложения;
- ✓ Введение в программирование на языке Python;
- ✓ Составление технического задания на разработку;
- ✓ Сбор статистики и анализ результатов работы алгоритма;
- ✓ Подготовка к публичной презентации и защите проекта и защита проекта с демонстрацией игрового приложения;
- ✓ Подведение итогов, рефлексия.

Необходимое оборудование:

- ✓ Персональный компьютер с Windows 7 / 8 / 10
- ✓ Доступ в интернет
- ✓ Компилятор Python 3

Кейс «Спаси остров»

Описание

«Спаси остров» – это игра для двоих, в которой один игрок загадывает слово и рисует на странице отдельные пустые клетки для каждой буквы. А второй игрок пытается угадать буквы, которые могут быть в данном слове, а затем и всё слово целиком.

Если второй игрок правильно угадывает букву, первый игрок вписывает её в соответствующую пустую клетку. А если ошибается, первый игрок “льёт воду на остров”, который представляет из себя поле из квадрата размером 4 на 4 см, заполняя его постепенно водой. Чтобы победить, второй игрок должен угадать все буквы в слове до того, как остров уйдет под воду.

Категория кейса: Вводный, рассчитан на учащихся 8 класса

Вопросы к кейсу:

1. Для каких задач служит тип цикла for?
2. В каких еще ситуациях можно использовать тип данных - список?
3. Как делать множественное присваивание?
4. Какие ещё известные компьютерные игры вы могли бы и хотели реализовать?

Место кейса в структуре программы: базовый, рекомендуется к выполнению после изучения основ программирования и кейса “ Угадай число”

Количество учебных часов на которые рассчитан кейс: 10 часов

Учебно-тематическое планирование (занятие - 2 часа):

Занятие 1		Занятие 2		Занятие 3	
Цель: Постановка проблемы, генерация путей решения		Цель: Изучение дизайна и механики игры		Цель: Визуализация программы в виде блок-схемы	
Что делается: Представление поставленной проблемы группе детей. Анализ проблемы,	Компетенции: Умение искать информацию в различных источниках. Умение генерировать	Что делается: Знакомство с игровыми механиками игры, изучение ограничений и	Компетенции: Работа с переменными величинами. Проверка наличия буквы в секретном слове.	Что делается: Создание блок-схем. Ветвление в блок-схемах.	Компетенции: Умение генерировать идеи. Умение

генерация и обсуждение методов ее решения	идеи предложенными методами.	правил..	Проверка – не победил ли игрок. Обработка ошибочных предположений. Проверка – не проиграл ли игрок. Умение программировать в Python.	логически разбирать проект на блоки.
---	------------------------------	----------	--	--------------------------------------

Занятие 4	Занятие 5		
Цель: Тестирование и доработка.	Цель: Подготовка к публичному выступлению для защиты результатов. Демонстрация игры в группе и защита результатов		
Что делается: Тестирование созданной игры. Проверка на граничные условия. Выявление багов и их исправление.	Компетенции: Умение программировать на языке Python. Умение проводить тестирование созданных продуктов.	Что делается: Подготовка речи и презентации в для публичной демонстрации результатов работы в кейсе. Публичная презентация результатов работы. Ответы на вопросы. Рефлексия.	Компетенции: Умение анализировать результаты работы. Грамотное представление результатов своей деятельности. Базовые навыки ораторства, публичных выступлений, аргументирование точки зрения.

Метод работы с кейсом: продвижение по шагам жизненного цикла игры
Минимально необходимый уровень входных компетенций: рекомендуется к выполнению после изучения основ языка программирования Python, основных алгоритмических понятий и конструкций.

Предполагаемые образовательные результаты учащихся:

Артефакты: готовая игра

Формируемые навыки (soft skills):

- ✓ Развитие аналитического и алгоритмического мышления;
- ✓ Умение аргументировать свою точку зрения и отстаивать ее.

Формируемые навыки (hard skills):

- ✓ Понятие алгоритма, последовательного выполнения действий;
- ✓ Понятие программы, подпрограммы, цикла, условия;
- ✓ Умение создавать программы и игры в Python;
- ✓ Умение создавать презентации.

Количество учебных часов на которые рассчитан кейс (может варьироваться в зависимости от уровня подготовки, условий, и т.д.): **10 часов**

Процедуры и формы выявления образовательного результата:

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

Педагогический сценарий (руководство для наставника):

Кейс представляет собой разработку игрового приложения для решения проблемной ситуации. Кейс включает в себя:

- ✓ Введение в проблему при помощи беседы с обучающимися (приведение жизненных примеров);
- ✓ Групповое обсуждение проблемы, поиск путей решения, введение в группу мысли о решении проблемы через разработку игрового приложения;
- ✓ Введение в программирование игровых приложений в Python;
- ✓ Составление технического задания на разработку игрового приложения;
- ✓ Создание уровней, программирование механики игры;
- ✓ Проведение тестирования разработанного игрового приложения и его доработка;
- ✓ Подготовка к публичной презентации и защите проекта и защита проекта с демонстрацией игрового приложения;
- ✓ Подведение итогов, рефлексия.

Минимально необходимый уровень входных компетенций:

- ✓ Работа с компьютером на уровне начинающего пользователя
- ✓ Знание математики на уровне выпускника начальной школы

Необходимое оборудование:

- ✓ Персональный компьютер с Windows 7 / 8 / 10
- ✓ Доступ в интернет
- ✓ Компилятор Python 3.5

Кейс «Калькулятор»

Описание:

Для ускорения счета часто используют такие электронные устройства как калькулятор. Данное приложение входит в базовый состав компонентов любого мобильного устройства, но, несмотря на широкий функционал, пользоваться такими приложениями не всегда является удобным. Многие его функции являются избыточными, тогда как других не хватает.

Необходимо разработать собственный программный продукт типа «Калькулятор», со своим собственным интерфейсом.

Категория кейса: Вводный.

Вопросы к кейсу:

1. Проанализируйте правила функционирования Калькулятора.
2. Для чего служат библиотеки?
3. Для каких задач применяется библиотека Tkinter?

Место кейса в структуре программы: вводный, рассчитан на учащихся 8 класса

Количество учебных часов: 10 часов

Учебно-тематическое планирование (занятие - 2 часа):

Занятие 1	Занятие 2	Занятие 3
Цель: Постановка проблемы, генерация путей решения	Цель: Создание программы для работы калькулятора	Цель: Создание внешнего вида калькулятора

<p>Что делается: Проводим анализ проблемной ситуации, генерация и обсуждение методов ее решения и возможности достижения конечного результата. Проводим разделение на группы..</p>	<p>Компетенции: Развитие критического мышления. Умение поиска и представления информации. Коммуникативность. Умение генерировать идеи, слушать и слышать собеседника, аргументированно отстаивать свою точку зрения. Развитие критического мышления, коммуникативных навыков, навыков командной работы.</p>	<p>Что делается: Написание программной части для будущего калькулятора</p>	<p>Компетенции: Развитие алгоритмического мышления. Навыки работы с языком программирования Python</p>	<p>Что делается: Создание внешнего вида калькулятора с использованием библиотеки Tkinter</p>	<p>Компетенции: Развитие алгоритмического мышления. Навыки работы с языком программирования Python.</p>
---	--	---	---	---	--

Занятие 4	Занятие 5
<p>Цель: Тестирование и доработка</p>	<p>Цель: Подготовка к публичному выступлению для защиты результатов. Демонстрация игры в группе и защита результатов.</p>
<p>Что делается: Тестирование работы калькулятора. Выявление багов и их исправление. решения проблемы; - оформляют проектную идею и презентуют ее.</p>	<p>Компетенции: Умение программировать на языке Python. Умение проводить тестирование созданных продуктов.</p>
<p>Что делается: Подготовка речи и презентации в для публичной демонстрации результатов работы в кейсе. Публичная презентация результатов работы. Ответы на вопросы. Рефлексия.</p>	<p>Компетенции: Умение анализировать результаты работы. Грамотное представление результатов своей деятельности. Базовые навыки ораторства, публичных выступлений, аргументированные точки зрения.</p>

Метод работы с кейсом: продвижение по шагам жизненного цикла игры

Минимально необходимый уровень входных компетенций:

- ✓ Работа с компьютером на уровне начинающего пользователя
- ✓ Знание математики на уровне выпускника начальной школы

Предполагаемые образовательные результаты учащихся:

Артефакты: готовый продукт калькулятор

Формируемые навыки (soft skills):

- ✓ Развитие аналитического и алгоритмического мышления;
- ✓ Понятие алгоритма, последовательного выполнения действий;
- ✓ Понятие программы, подпрограммы, цикла, условия;
- ✓ Умение аргументировать свою точку зрения и отстаивать ее;
- ✓ Умение создавать презентации.

Формируемые навыки (hard skills):

- ✓ Умение создавать программы и игры в Python;

- ✓ Умение создавать презентации.

Количество учебных часов на которые рассчитан кейс: 10 часов

Процедуры и формы выявления образовательного результата:

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

Педагогический сценарий (руководство для наставника):

Кейс представляет собой разработку игрового приложения для решения проблемной ситуации. Кейс включает в себя:

- ✓ Введение в проблему при помощи беседы с обучающимися (приведение жизненных примеров);
- ✓ Групповое обсуждение проблемы, поиск путей решения, введение в группу мысли о решении проблемы через разработку игрового приложения;
- ✓ Введение в программирование игровых приложений в Python;
- ✓ Составление технического задания на разработку игрового приложения;
- ✓ Создание уровней, программирование механики игры;
- ✓ Проведение тестирования разработанного игрового приложения и его доработка;
- ✓ Подготовка к публичной презентации и защите проекта и защита проекта с демонстрацией игрового приложения;
- ✓ Подведение итогов, рефлексия.

Необходимое оборудование:

- ✓ Персональный компьютер с Windows 7 / 8 / 10 или Linux Debian 9.6
- ✓ Доступ в интернет
- ✓ Компилятор Python 3.5

