МУНИЦИПАЛЬНОЕ БЮДЖЕТНОЕ
ОБЩЕОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
"СРЕДНЯЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ ШКОЛА №
15" ГОРОДСКОГО ОКРУГА СПАССК-ДАЛЬНИЙ

Подписано цифровой подписью: МУНИЦИПАЛЬНОЕ БЮДЖЕТНОЕ ОБЩЕОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ "СРЕДНЯЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ ШКОЛА № 15" ГОРОДСКОГО ОКРУГА СПАССК-ДАЛЬНИЙ
DN: c=RU, st=Приморский край, street=Красногвардейская 104/6, l=Спасск-Дальний, title=Директор, o=МУНИЦИПАЛЬНОЕ БЮДЖЕТНОЕ ОБЩЕОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ "СРЕДНЯЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ ШКОЛА № 15" ГОРОДСКОГО ОКРУГА СПАССК-ДАЛЬНИЙ, 1.2.643.100.1=120D313032323530038323036333 8, 1.2.643.100.3=120B303432393536313830363 7, 1.2.643.100.4=120A323531303030373532 39, 1.2.643.3.131.1.1=120C323531303033383236333730, email=spkspassk@gmail.com, givenName=Сергей Александрович, sn=Гриднев, cn=МУНИЦИПАЛЬНОЕ БЮДЖЕТНОЕ ОБЩЕОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ "СРЕДНЯЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ ШКОЛА № 15" ГОРОДСКОГО ОКРУГА СПАССК-ДАЛЬНИЙ
Дата: 2022.12.09 17:45:45 +10'00'

Управление образования Администрации городского округа Спасск-Дальний
Муниципальное бюджетное общеобразовательное учреждение
«Средняя общеобразовательная школа № 15» городского округа Спасск-Дальний

ПРИНЯТА
на заседании
педагогического совета
Протокол № 5 от 18.05.2022г.

УТВЕРЖДЕНА
приказом директора
№ 129-а
от 19.05.2022 г.

«Язык программирования Python для начинающих»

Дополнительная общеобразовательная общеразвивающая программа технической направленности

Возраст учащихся: 15-17 лет

Срок реализации программы: 1 год

Постнова Татьяна Владимировна,
Педагог дополнительного образования

Г. Спасск-Дальний

2022 г.

Раздел № 1. ОСНОВНЫЕ ХАРАКТЕРИСТИКИ ПРОГРАММЫ

1.1. Пояснительная записка

Актуальность программы

По результатам исследования компании Jet Brains (https://www.jetbrains.com/ru-ru/lp/devecosystem-2019/) язык программирования Python – самый изучаемый в 2019 году и один из самых востребованных на рынке труда. Python достаточно широко используется при изучении основ алгоритмизации и программирования в школьном курсе информатики. В частности, используется в качестве базового языка для изучения в УМК по информатике авторов Поляков К. Ю., Еремин Е. А., допущенном к использованию в общеобразовательных организациях Министерством просвещения РФ (Приказ Министерства Просвещения РФ от 28 декабря 2018 года № 345).

Изучение языка программирования Python на занятиях по данной программе откроет ученикам возможности дальнейшего развития в области IT и поможет профессиональному самоопределению в старших классах, пригодится в олимпиадах по программированию и решении заданий ЕГЭ.

Курс предполагает смешанный формат обучения. Сочетание групповой работы с учителем в классе и индивидуальной работы в личном кабинете на онлайн-платформе позволяет ученикам выработать не только технические навыки программирования, но и навыки социального взаимодействия при работе над финальным проектом курса, а главное – научиться самостоятельно выстраивать свое профессиональное развитие.

Направленность программы: техническая.

Уровень освоения - стартовый.

Отличительные особенности

Авторский подход к организации образовательного процесса и сетевая форма реализации данного курса по программированию. Реализация

программы предполагает проведение аудиторных занятий с использованием вариативных дистанционных ресурсов, предоставленных Благотворительным фондом развития образования «Айкью Опшн» («Возможность интеллекта») на основе сетевого взаимодействия:

- авторских онлайн уроков, размещенных на платформе Stepik;
- авторских интерактивных заданий (опросов, викторин, дидактических игр и др.), размещенных на бесплатных специализированных сервисах Kahoot и Learning Apps;
- авторских проектных заданий для создания учащимися игр на языке Python.

Адресат программы: учащиеся школы 15-17 лет.

Срок и режим реализации программы: 1 учебный год с интенсивностью проведения занятий 1 раз в неделю 1 учебный час. Продолжительность одного учебного часа – 60 минут. Общий объем часов по программе – 34 часа.

Формы организации образовательной деятельности:

смешанная: очное и онлайн-обучение. В очных занятиях в группах по 10-15 человек используются задания на онлайн-платформе с автоматизированной проверкой. Задания на платформе доступны ученикам для самостоятельного изучения в любое время. В ходе курса запланирован плавный переход к формату “перевернутого класса”: на первых пяти занятиях учеников знакомят с онлайн-платформой и основными инструментами программирования на Python. Начиная с шестого урока предлагаются различные формы самостоятельной активности на уроке, с групповой взаимопроверкой. После десятого урока иногда ученикам предлагается самостоятельно изучить новую тему, опираясь на конспект онлайн-курса, а роль учителя сдвигается к консультации и модерации. Последние пять уроков отведены на почти самостоятельную работу над проектами.

Основные формы организации занятий программы «Python для начинающих»:

- Практические занятия с использованием онлайн-платформы Stepik;

- Работа в IDE «Wing 101»;
- Прохождение опросов в приложениях Kahoot и Learning Apps;
- Домашние практические занятия с использованием онлайн-платформы Stepik, направленные на отработку навыков программирования на языке Python.

1.2 Цель и задачи программы

Цель программы: формирование у обучающихся навыков программирования на языке Python и самонаправленного обучения.

Задачи программы:

Воспитательные:

1. Воспитать уважительное отношение к преподавателям и сверстникам, культуру поведения во время занятий и совместной продуктивной деятельности;
2. Формировать характер программиста, а также его эстетические и нравственные качества;
3. Формировать культуру занятий, направленную на воспитание личностных и социальных качеств;
4. Формировать чувство ответственности.

Развивающие:

1. Формировать интерес к изучению профессии, связанной с программированием;
2. Развивать навыки самостоятельного изучения материала и оценки результатов своей деятельности.
3. Научить применять полученные знания для решения практических задач.
4. Повысить уровень самостоятельности и творческого подхода к решению задач с помощью средств современной вычислительной техники.

Обучающие:

1. Изучить основы программирования на языке Python;
2. Сформировать навыки выполнения технологической цепочки разработки программ средствами языка программирования Python;
3. Формировать навыки работы в системе программирования Python; объяснить основные конструкции языка программирования Python, позволяющие работать с простыми и составными типами данных (строками, списками, кортежами, словарями, множествами); научить применять функции при написании программ на языке программирования Python;
4. Формировать навыки грамотной разработки программ; научить отлаживать и тестировать программы, делать выводы о работе этих программ.
5. Формировать практические навыки решения прикладных задач;

1.3 Содержание программы

Учебный план 1 года обучения

№ п/п	Название раздела, темы	Количество часов			Формы аттестации/ контроля
		Всего	Теория	Практика	
Раздел 1. Знакомство с языком Python. Переменные и выражения					
1	Знакомство с Python. Команды input() и print().	1	1		Решение практических задач на платформе Stepik.
2	Параметры sep, end. Переменные. Комментарии. PEP 8	2	1	1	
3	Работа с целыми числами	1		1	
Раздел 2. Условные предложения					
4	Условный оператор. Логические операции and, or, not	1	1		Тестирование, решение практических

5	Вложенный и каскадный условный оператор	2	1	1	задач на платформе Stepik.
6	Типы данных int, float, str. Встроенные функции min(), max(), abs(). Оператор in.	1		1	
Раздел 3. Циклы					
7	Цикл for. Функция range().	2	1	1	Тестирование, решение практических задач на платформе Stepik.
8	Частые сценарии при написании циклов. Расширенные операторы присваивания.	2	1	1	
9	Цикл с предусловием while	2	1	1	
10	Операторы break, continue, else.	1		1	
11	Вложенные циклы	1		1	
Раздел 4. Строки - последовательности символов					
12	Строковый тип данных: индексация и срезы	2	1	1	Тестирование, решение практических задач на платформе Stepik.
13	Методы строк	2	1	1	
Раздел 5. Сложные типы данных. Списки					
14	Резервное время. Введение в списки.	1	1		Тестирование, решение практических задач на платформе
15	Основы работы со списками. Методы списков	2	1	1	

16	Вывод элементов списка. Строковые методы split() и join()	1		1	Stepik.
17	Методы списков. Списочные выражения	2	1	1	
Раздел 6. Функции					
18	Функции	1		1	Тестирование, решение практических задач на платформе Stepik.
19	Локальные и глобальные переменные. Функции возвращающие значения.	2	1	1	
20	Функции возвращающие значения.	2	1	1	
Раздел 7. Работа над проектом					
21	Работа над проектом	3		3	Работа над проектом, презентация проекта, рефлексия.
	Итого	34	14	20	

Содержание учебного плана 1 года обучения

Раздел 1 Знакомство с языком Python. Переменные и выражения (4 ч)

Теория. Знакомство с планом работы объединения, инструктаж по ТБ. Общие сведения о языке Python. Установка Python на компьютер. Режимы работы Python. Что такое программа. Первая программа. Структура программы на языке Python. Комментарии. Типы данных. Преобразование типов. Переменные. Оператор присваивания. Имена переменных и ключевые слова.

Выражения. Операции. Порядок выполнения операций. Композиция.

Ввод и вывод. Ввод данных с клавиатуры. Вывод данных на экран. Пример скрипта, использующего ввод и вывод данных. Задачи на элементарные действия с числами. Решение задач на элементарные действия с числами.

Практика. Установка программы Python. Регистрация на платформе Stepik. Решение задач. Переменные и выражения. Задачи на элементарные действия с числами.

Раздел 2. Условные предложения (4 ч)

Теория. Логический тип данных. Логические выражения и операторы. Сложные условные выражения (логические операции and, or, not). Условный оператор. Альтернативное выполнение. Примеры решения задач с условным оператором. Множественное ветвление. Реализация ветвления в языке Python.

Практика. Логические выражения. Решение задач по теме "Условные операторы. Множественное ветвление", "Составление программ с ветвлением"

Раздел 3. Циклы (8 часов)

Теория. Понятие цикла. Тело цикла. Условия выполнения тела цикла. Оператор цикла с условием. Оператор цикла while. Бесконечные циклы. Альтернативная ветка цикла while. Обновление переменной. Краткая форма записи обновления. Примеры использования циклов.

Оператор цикла с параметром for. Операторы управления циклом. Пример задачи с использованием цикла for. Вложенные циклы. Циклы в циклах. Случайные числа. Функция randrange. Функция random. Примеры решения задач с циклом.

Практика. Решение задачи с циклом for, while. Реализация циклических алгоритмов.

Раздел 4. Строки - последовательности символов (4 часов)

Теория. Составной тип данных - строка. Доступ по индексу. Длина строки и отрицательные индексы. Преобразование типов. Применение цикла для обхода строки.

Срезы строк. Строки нельзя изменить. Сравнение строк. Оператор in. Модуль string. Операторы для всех типов последовательностей (строки, списки, кортежи). Примеры решения задач со строками.

Практика. Решение задач со строками. Основы работы со списками, методы списков.

Раздел 5. Сложные типы данных. Списки (6 ч)

Теория. Списки. Тип список (list). Индексы. Обход списка. Проверка вхождения в список. Добавление в список. Суммирование или изменение списка. Операторы для списков. Срезы списков. Удаление списка. Клонирование списков. Списочные параметры. Функция range. Списки: примеры решения задач.

Матрицы. Вложенные списки. Матрицы. Строки и списки. Генераторы списков в Python. Кортежи. Присваивание кортежей. Кортежи как возвращаемые значения

Введение в словари. Тип словарь (dict). Словарные операции. Словарные методы.

Множества в языке Python. Множества. Множественный тип данных. Описание множеств. Операции, допустимые над множествами: объединение, пересечение, разность, включение. Оператор определения принадлежности элемента множеству.

Практика. Списки. Решение задач со списками.

Раздел 6. Функции (5 часов)

Теория. Создание функций. Параметры и аргументы. Локальные и глобальные переменные.

Поток выполнения. Функции, возвращающие результат. Анонимные функции, инструкция lambda. Примеры решения задач с использованием функций.

Рекурсивные функции.

Практика. Создание функций. Локальные переменные. Решение задач с использованием функций.

Раздел 7. Работа над проектом (3 ч)

Практика. Самостоятельная работа над проектом.

1.4 Планируемые результаты

Личностные результаты:

Обучающийся будет:

- владеть первичными навыками анализа и критичной оценки получаемой информации;
- готов к общению и сотрудничеству со сверстниками и взрослыми в процессе образовательной, учебно-исследовательской, творческой деятельности;
- способен осуществлять самооценку и самоанализ полученных результатов;
- развивать чувство товарищества, чувство личной ответственности.

У обучающегося будет возможность:

- развить ответственное отношение к информации с учетом правовых и этических аспектов ее распространения;
- развить чувство личной ответственности за качество окружающей информационной среды;
- связать учебное содержание курса с собственным жизненным опытом, понять значимость обучения программированию в современном информационном обществе;
- формировать осознанное, уважительное и доброжелательное отношение к другому человеку, его мнению; готовность и способность вести диалог с другими людьми и достигать в нём взаимопонимания;
- к повышению своего образовательного уровня и продолжению обучения с использованием полученных навыков программирования.

Метапредметные результаты:

Обучающийся будет знать:

- способы сбора информации;

- подходы и методы для достижения поставленной цели;
- как самостоятельно организовывать свое рабочее место (подбирать инструментарий) в соответствии с целью выполнения задания.
- как осуществлять планирование своей деятельности и корректировать выполнение задания в соответствии с планом, условиями выполнения, результатом действий на определенном этапе.

Обучающийся приобретёт:

- возможность научиться слушать собеседника и вести диалог, признавать возможность существования различных точек зрения и права каждого иметь свою, излагать своё мнение, аргументировать свою точку зрения и оценку событий;
- возможность научиться планировать свою самостоятельную учебно-познавательную деятельность; выбирать индивидуальную траекторию достижения учебной цели.
- умения самостоятельно планировать пути достижения целей; соотносить свои действия с планируемыми результатами, осуществлять контроль своей деятельности, определять способы действий в рамках предложенных условий, корректировать свои действия в соответствии с изменяющейся ситуацией; оценивать правильность выполнения учебной задачи;
- навыки владения основами самоконтроля, самооценки, принятия решений и осуществления осознанного выбора в учебной и познавательной деятельности;
- навыки постановки и формулирования проблемы; поиска и выделения необходимой информации и технических средств, для решения поставленной цели.
- навыки осуществлять самооценку промежуточных и итоговых результатов своей самостоятельной учебно-познавательной

деятельности, проводить рефлексию своей учебно-познавательной деятельности.

Предметные результаты:

Обучающийся будет знать:

- как использовать инструменты разработки среды Wing;
- основные конструкции языка программирования Python;
- основы грамотной разработки программ на Python;

Обучающийся будет уметь:

- работать в системе программирования Python;
- применять основы программирования на языке Python, позволяющие работать с простыми и составными типами данных (строками, списками, кортежами, словарями, множествами); применять функции при написании программ на языке программирования Python;
- реализовывать алгоритмы на компьютере в виде программ, написанных на языке Python;
- отлаживать и тестировать программы на языке Python, делать выводы о работе этих программ;
- самостоятельно реализовывать проект, связанный с разработкой игр (2D игра на движке Pygame, либо самостоятельно разработанный мини-проект в консольном режиме языка Python).

Обучающийся будет владеть:

- практическими навыками решения прикладных задач на языке Python;

РАЗДЕЛ № 2. ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИЕ УСЛОВИЯ

2.1 Условия реализации программы

1. Материально-техническое обеспечение:

- Помещение (предпочтительно, изолированное) - кабинет информатики;
- 10—15 рабочих мест: стол, стул, розетка, компьютеры на каждое рабочее место;
- проектор, аудио колонки;

- меловая, магнитно-маркерная доска или флипчарт;
- **Требования к ПО:**
  - Операционная система Windows 7 или моложе / MacOS / Unix-based системы с поддержкой протокола HTML5;
  - Приложения Google Chrome, Gimp, Brackets;
  - интерактивная оболочка (бесплатная IDE Wing101 или аналог).
- Интернет-соединение, скорость загрузки не менее 2 Мбит/сек.

2. Учебно-методическое и информационное обеспечение:

Литература для педагога

1. **Васильев, А. Н. Python на примерах** [Текст]:практ. курс /А. Н. Васильев - Наука и Техника, 2019 - 432 с.
2. **Прохоренок, Н. А. Python 3: самое необходимое** [Текст]:практ. курс / Н. А. Прохоренок, В. А. Дронов - БХВ-Петербург, 2019 - 608 с.
3. **Гэддис, Т. Начинаем программировать на Python** [Текст]: учебник/Т. Гэддис - БХВ-Петербург, 2019 - 768 с.
4. **Седжвик, Р. Программирование на языке Python** /Р. Седжвик, К. Уэйн, Р. Дондеро - Вильямс, 2017 - 736 с.
5. Комплект методических материалов к курсу «Python для начинающих» от Благотворительного фонда развития образования «Айкью Опшн» («iq option») в рамках проекта «Поколение Python

Литература для обучающихся: Не предусмотрена

Электронные ресурсы:

1. Курс Поколение Python: курс для начинающих на платформе Stepik. (https://stepik.org/course/58852/syllabus)

2.2 Оценочные материалы и формы аттестации

Аттестация проводится в форме выполнения индивидуальных и групповых заданий по пройденному материалу. Контроль в указанной форме

осуществляется как промежуточный, так и итоговый. Отметочная форма контроля отсутствуют.

Для заданий на онлайн-тренажере указан необходимый минимум — 70% выполненных заданий, чтобы тема считалась пройденной успешно и был открыт доступ к следующей теме. После каждой темы в онлайн-курсе стоит итоговая работа: от ученика требуется в ограниченное время (три часа) решить набор задач по пройденной теме. В среднем, ученик справляется с решением за 30 минут. Преподаватели могут использовать эти итоговые работы в качестве промежуточных проверочных работ.
В конце курса, по итогам работы над групповыми и индивидуальными проектами проводится обсуждение результатов в коллективе с опорой на чек-лист, исправление ошибок и, тем самым, коррекция и закрепление полученных знаний.

2.3 Методические материалы

Методы обучения:

1. Словесный: объяснение нового материала; рассказ обзорный для раскрытия новой темы; беседы с учащимися в процессе изучения темы.

2. Наглядный: применение демонстрационного материала, наглядных пособий, презентаций по теме.

3. Практический: индивидуальная и совместная продуктивная деятельность, выполнение учащимися определенных заданий, решение задач.

4. Интерактивный: создание специальных заданий, моделирующих реальную жизненную ситуацию, из которой учащимся предлагается найти выход.

Технологии:

1. Технология проблемного диалога. Учащимся не только сообщаются готовые знания, но и организуется такая их деятельность, в процессе которой они сами делают «открытия», узнают что-то новое и используют полученные знания и умения для решения жизненных задач.

2. Технология коллективного взаимообучения («организованный диалог», «сочетательный диалог», «коллективный способ обучения (КСО), «работа учащихся в парах сменного состава») позволяет плодотворно развивать у обучаемых самостоятельность и коммуникативные умения.

3. Элементы здоровьесберегающих технологий являются необходимым условием снижения утомляемости и перегрузки учащихся.

4. Проектная технология предлагает практические творческие задания, требующие от учащихся их применение для решения проблемных заданий, знания материала на данный исторический этап. Овладевая культурой проектирования, школьник приучается творчески мыслить, прогнозировать возможные варианты решения стоящих перед ним задач.

5. Информационно-коммуникационные технологии активизируют творческий потенциал учащихся; способствует развитию логики, внимания, речи, повышению качества знаний; формированию умения пользоваться информацией, выбирать из нее необходимое для принятия решения, работать со всеми видами информации, программным обеспечением, специальными программами и т.д.

2.4 Календарный учебный график

Этапы образовательного процесса		1 год
Продолжительность учебного года, неделя		34
Количество учебных дней		34
Продолжительность учебных периодов	1 полугодие	1.09.2022-25.12.2022
	2 полугодие	11.01.2023-25.05.2023
Возраст детей, лет		15-17
Продолжительность занятия, час		1
Режим занятия		1 раза/нед
Годовая учебная нагрузка, час		34

СПИСОК ЛИТЕРАТУРЫ

Ссылки на печатные источники:

1. *Васильев, А. Н.* Python на примерах: практ. курс /А. Н. Васильев - Наука и Техника, 2019.

2. *Прохоренок, Н. А.* Python 3: самое необходимое: практ. курс / Н. А. Прохоренок, В. А. Дронов - БХВ-Петербург, 2019.
3. *Гэддис, Т.* Начинаем программировать на Python: учебник/Т. Гэддис - БХВ-Петербург, 2019 - 768.
4. *Седжвик, Р.* Программирование на языке Python /Р. Седжвик, К. Уэйн, Р. Дондеро - Вильямс, 2017.

Ссылки на электронные ресурсы:

1. Официальный сайт программы [Электронный ресурс] − Режим доступа: https://docs.python.org/, свободный.
2. Сайт / справочные материалы [Электронный ресурс] − Режим доступа: https://metanit.com/python/, свободный.
3. Сайт / интерактивный сборник задач для практики программирования [Электронный ресурс] − Режим доступа: http://pythontutor.ru/, свободный.
4. Сайт / Адаптивный тренажер Python [Электронный ресурс] − Режим доступа: https://stepik.org/course/431 , свободный.
5. Сайт проекта Open Book Project. Практические примеры на Python Криса Мейерса [Электронный ресурс] − Режим доступа: openbookproject.net , свободный.