

МУНИЦИПАЛЬНОЕ БЮДЖЕТНОЕ ОБЩЕОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
СРЕДНЯЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ ШКОЛА ПОС. МИЗУР



Утверждаю

О.Н. Калоева

2023 г.

Рабочая программа
по спецкурсу
«Поколение Python – курс для начинающих»
(8-9 класс)

Составитель:
учитель информатики
Ревазова Ф.Ф.

2023г.

Раздел 1. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

1.1. Актуальность программы

По результатам исследования компании Jet Brains (<https://www.jetbrains.com/ru-ru/lp/devecosystem-2019/>) язык 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).

Программа курса разработана на основе авторской дополнительной общеразвивающей программы «Python для начинающих». Авторы: Белых Роман, Гув Тимур, Скобина Ольга

Изучение Python в школе откроет ученикам возможности дальнейшего развития в области IT и поможет профориентации в старших классах, пригодится в олимпиадах по программированию и решению 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.

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

Направленность дополнительной общеобразовательной программы: техническая.

Задачи реализации программы:

Обучающие

1. Изучить основы программирования на языке Python;

Развивающие

2. Научиться применять полученные знания для решения практических задач.

Воспитательные

3. Научиться применять полученные знания для решения практических задач.
4. Повысить уровень самостоятельности в обучении (по четырехступенчатой шкале GROW)

Методические материалы курса состоят из:

1. Презентации с иллюстративным изложением теоретического материала;
2. Упражнений на платформе Stepik с теоретическим и практическим материалом языка Python;
3. Интерактивных проверочных заданий в приложениях Kahoot и Learning Apps;
4. Подвижных игр, направленных на закрепление знаний, полученных на занятии.

5. Инструкций для проведения рефлексии процесса обучения с учениками.

Планируемые результаты обучения:

Личностные

По окончании курса учащийся сможет:

- Программировать на языке Python.
- Использовать инструменты разработки среды Wing.
- Самостоятельно реализовывать проекты, связанные с разработкой игр.

Метапредметные

- Ставить учебные цели.
- Формулировать достигнутый результат.
- Планировать свою самостоятельную учебно-познавательную деятельность; выбирать индивидуальную траекторию достижения учебной цели.
- Определять подходы и методы для достижения поставленной цели.
- Отбирать необходимые средства для достижения поставленной цели.
- Осуществлять самооценку промежуточных и итоговых результатов своей самостоятельной учебно-познавательной деятельности.
- Проводить рефлексии своей учебно-познавательной деятельности.

Предметные

Практическим результатом работы служит финальный проект каждого ученика: 2D игра на движке Pygame, либо серия самостоятельно разработанных мини-проектов в консольном режиме языка Python.

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.

Раздел 2. Содержание программы

В рамках смешанного подхода к обучению каждая из тем, изложенных преподавателем, отрабатывается в уроках платформы Stepik и в практических заданиях. (<https://stepik.org/course/58852/syllabus>)

2.2. Содержание курса

№	Тема	Содержание и виды работы
Раздел 1. Знакомство с Python. Ввод-вывод данных. Работа с целыми числами		
1	Знакомство с Python. Команды input() и print(). Параметры sep, end. Переменные. Комментарии. PEP 8. Работа с целыми числами	Знакомство с учениками. Сбор ожиданий учеников, пояснение программы курса, рефлексия. Регистрация на платформе Stepik. Решение задач. Отработка навыков работы с переменными, использования комментариев в коде. Обсуждение правильной работы с ошибками. Практика взаимодействия в группе. Отработка операций с целыми числами. Обработка цифр числа.
Раздел 2. Программирование ветвлений. Условный оператор		
3	Условный оператор. Логические операции and, or, not. Вложенный и каскадный условный оператор. Типы данных int, float, str. Встроенные функции min(), max(), abs(). Оператор in	Приоритеты и планирование. Решение задач с использованием условного оператора и логических операций. Решение задач с использованием каскадного условного оператора и вложенных ветвлений. Самостоятельное изучение темы в малых группах и создание презентации. Групповая работа над презентацией. Решение задач на платформе.
Раздел 3. Программирование циклов		
7	Цикл for. Функция range(). Частые сценарии при написании циклов. Расширенные операторы присваивания. Цикл с предусловием while. Операторы break, continue, else. Вложенные циклы	Подвижная игра с функцией range(). Решение задач на платформе. Игровое повторение предыдущей темы. Самостоятельное изучение новой темы в командах. Самостоятельное исследование частых сценариев программирования. Решение задач на платформе на отработку частых сценариев. Игра на вопросы с бинарной логикой “данетка”. Решение задач на использование цикла с предусловием. Самостоятельное изучение процедуры обработки цифр натурального числа. Работа в командах над ревью кода. Решение задач на платформе: отработка применения оператора break в циклах. Отработка применения вложенных циклов.

№	Тема	Содержание и виды работы
	Повторение изученного в 8 классе	
Раздел 4. Строковый тип данных		
1	Строковый тип данных: индексация и срезы. Методы строк. Основные методы конвертации регистра. Основные методы поиска и замены. , основные методы классификации символов. Представление строк в памяти компьютера.	Самостоятельное изучение темы “Строковый тип данных”, решение задач на платформе. Устный разбор методов строк и их функционала. Решение задач на платформе: использование методов строк.
Раздел 5. Списки		
2	Списки. Основы работы со списками. Методы списков. Вывод элементов списка. Строковые методы split() и join(). Списочные выражения. Сортировка списков	Повторение темы “Циклы”. Повторение темы “Строки”. Объяснение новой темы “Списки”. Решение задач на платформе на использование списков. Индивидуальная работа с учениками. Самостоятельное изучение теории. Решение задач на отработку методов работы со списками. Подведение общих итогов самостоятельного изучения теории. Объяснение методов split и join. Игра в парах с образцами кода.

2.1. Тематический план

8 класс

№	Название темы	Кол-во часов	Теоретические	Практические
1	Знакомство с Python. Ввод-вывод данных. Работа с целыми числами	7	3	4
2	Программирование ветвлений. Условный оператор	9	4	5
3	Программирование циклов	15	6	9
	Резерв	3		
	Итого	34	13	18

9 класс

№	Название темы	Кол-во часов	Теоретические	Практические
1	Повторение изученного в 8 классе	2	1	1
2	Строковый тип данных	11	5	6
3	Списки	19	9	10
	Резерв	2		
	Итого	34	13	18

2.1. Учебный план

№	Название темы	Кол-во часов	Теоретические	Практические
1	Знакомство с Python. Команды input() и print().	2	1	1
2	Параметры sep, end. Переменные. Комментарии. PEP 8	2	1	1
3	Работа с целыми числами	2	1	1
4	Проверочная работа на ввод-вывод данных	1		1
5	Условный оператор. Логические операции and, or, not	2	1	1
6	Вложенный и каскадный условный оператор	2	1	1
7	Типы данных int, float, str.	2	1	1
8	Встроенные функции min(), max(), abs(). Оператор in.	2	1	1
9	Проверочная работа на условный оператор	1		1
10	Цикл for. Функция range().	2	1	1
11	Частые сценарии при написании циклов.	2	1	1
12	Расширенные операторы присваивания.	2	1	1
13	Цикл с предусловием while	2	1	1
14	Операторы break, continue, else.	2	1	1
15	Поиск ошибок и ревью кода	2		2
16	Вложенные циклы	2	1	1
17	Проверочная работа на циклы	1		1
18	Резерв	3		
	Итого	34	13	18

9 класс				
1	Повторение изученного в 8 классе	2	1	1
2	Строковый тип данных: индексация и срезы	2	1	1
3	Строковый тип данных, основные методы конвертации регистра.	2	1	1
4	Строковый тип данных, основные методы поиска и замены.	2	1	1
5	Строковый тип данных, основные методы классификации символов.	2	1	1
6	Представление строк в памяти компьютера	2	1	1
7	Проверочная работа на строки	1		1
8	Введение в списки.	2	1	1
9	Списки. Встроенные функции <code>len()</code> , <code>sum()</code> , <code>min()</code> , <code>max()</code>	2	1	1
10	Списки. Индексация и срезы	2	1	1
11	Списки. Методы добавления и удаления элементов.	2	1	1
12	Вывод элементов списка.	2	1	1
13	Строковые методы <code>split()</code> и <code>join()</code>	2	1	1
14	Методы списков	2	1	1
15	Списочные выражения.	2	1	1
16	Сортировка списков	2	1	1
17	Проверочная работа на списки	1		1
	Резерв	2		
	Итого	34	13	18

Раздел 3. Формы аттестации и оценочные материалы

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 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.

Раздел 4. Организационно-педагогические условия реализации программы

4.1. 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. **Харрисон, М. Как устроен Python.** [Текст]: практ. курс / М. Харрисон - Питер, 2002 - 272 с.

Литература для обучающихся: Не предусмотрена

Электронные ресурсы:

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

4.2. Материально-технические условия реализации программы

1. Обязательные

- помещение (предпочтительно, изолированное);
- 10—15 рабочих мест: стол, стул, розетка, компьютеры на каждое рабочее место;
- проектор, аудио колонки;
- Интернет-соединение, скорость загрузки не менее 2 Мбит/сек;
- меловая, магнитно-маркерная доска или флипчарт;

- общие условия в соответствии с СанПиН 2.4.4.3172-14

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).