

**Муниципальное автономное общеобразовательное учреждение
средняя общеобразовательная школа №1**

ОБСУЖДЕНО
педагогическим советом
МАОУ СОШ №1
Протокол №11
от 14 июня 2024 г.



**Дополнительная общеобразовательная общеразвивающая программа
технической направленности
«Программирование на Python»**

Возраст обучающихся: 12-17 лет
Срок реализации: 1 год

Автор-составитель:
Харин Е.С.,
педагог дополнительного
образования

г. Североуральск

Оглавление

1. Комплекс основных характеристик	3
1.1. Пояснительная записка.....	3
1.2.Цель и задачи программы.....	7
1.3.Планируемые результаты.....	9
1.4.Содержание общеразвивающей программы.....	11
2. Комплекс организационно-педагогических условий.....	18
2.1.Календарный учебный график.....	18
2.2.Условия реализации программы.....	18
2.3.Форма аттестации и оценочные материалы.....	20
Список литературы.....	27
Приложения.....	28

1. Комплекс основных характеристик

1.1 Пояснительная записка

Направленность: техническая

Актуальность программы обусловлены высоким интересом подростков к IT-сфере. Важнейшей характеристикой подрастающего поколения является активность в информационном пространстве, интернет-коммуникации. Python является очень востребованным языком программирования

- он отлично подходит для знакомства с различными современными парадигмами программирования и активно применяется в самых разных областях от разработки веб-приложений до машинного обучения.

В рамках изучения программы, обучающиеся постоянно будут сталкиваться с необходимостью самостоятельной работы над заданиями: обучающиеся учатся решать задачи без помощи преподавателя. Для этого в содержании курса фигурируют задания, в которых для решения задачи необходимо найти информацию самостоятельно; может потребоваться устранение ошибки, которую не так просто обнаружить. Все эти знания, умения и практические навыки решения актуальных задач, полученные на занятиях, готовят обучающихся к самостоятельной проектно-исследовательской деятельности с применением современных технологий.

Отличительная особенность программы «Программирование на Python» в том, что она является практико-ориентированной. Освоение подростками IT-навыков происходит в процессе практической и самостоятельной работы. Это позволяет обучающимся получать не только теоретические знания в области программирования, но и уверенно овладевать информационными технологиями, что поможет им самоопределиться и выстроить траекторию личностного роста

в современном информационном обществе.

Научившись программировать на языке Python, обучающиеся получают мощный и удобный инструмент для решения как учебных, так и прикладных задач. Вместе с тем чистота и ясность его конструкций позволит обучающимся потом с легкостью выучить любой другой язык программирования.

Знания и умения, приобретенные в результате освоения курса, могут быть использованы обучающимися при сдаче ЕГЭ, при участии в олимпиадах по программированию, при решении задач по физике, химии, биологии, лингвистике и другим наукам, а также они являются фундаментом для дальнейшего совершенствования мастерства программирования.

Адресат программы: программа предназначена для детей в возрасте 12–17 лет, проявляющих интерес к информационным технологиям.

Формы занятий групповые, количество обучающихся в группе от 10 до 12 человек. Группы формируются по возрасту: 12–14 и 15–17 лет. Состав групп постоянный.

Возрастные особенности группы содержание программы учитывает возрастные и психологические особенности детей 11–17 лет. При формировании групп выделены следующие возрастные периоды 12–14 лет, основываются на психологических особенностях младшего подросткового возраста и 15–17 лет соответственно базируются на психологических особенностях развития старшего подросткового возраста.

Для подростков 12–14 лет к значимым типам деятельности относится проектная деятельность: встреча замысла и результата как авторское действие подростка, проявление себя в общественно значимых ролях. Планирование содержания данной программы разворачивается от конечного результата, которого должен достичь подросток. Содержание программы обуславливает процесс получения итогового продукта в определенном цикле двух лет. Содержание развития – это образовательный маршрут по подготовке подростка к самопрезентации.

Ведущая деятельность подростков 15-17 лет – учебно-профессиональная. Организация образования сводится к подготовке и осуществлению профессиональной пробы в комплексном варианте: проживание инженерной деятельности. Содержание программы включает последовательное осуществление различных видов деятельности: выдвижение идеи; проявление продуктивного мышления, исследование, эксперимент, обобщение, финальный проект. Итоговый результат носит опережающий характер, а учебные действия обусловлены изобретательностью.

Режим занятий, объем общеразвивающей программы: продолжительность одного академического часа - 45 минут. Перерыв между учебными занятиями – 10 минут. Общее количество часов в неделю – 4 часа. Занятия проводятся 2 раза в неделю по 2 часа.

Срок освоения общеразвивающей программы определяется содержанием программы и составляет 1 год.

Объем общеразвивающей программы: 144 часа.

Формы обучения: Очная, возможна реализация очно с применением электронного обучения и дистанционных образовательных технологий (Закон №273-ФЗ, гл.2, ст.17, п.2.).

Платформы трансляции материала и организации взаимодействия: Сферум, Telemost.Yandex, ВКонтакте, индивидуальный сайт педагога и др.

По уровню освоения программа общеразвивающая, одноуровневая – базовый уровень. Она обеспечивает возможность обучения детей с первичными навыками программирования в том числе, одарённых детей.

Зачисление детей на базовый уровень данной программы производится после завершения обучения по программе «Программирование на Python стартовый уровень» по результатам успешной сдачи итоговой аттестации (защиты итогового проекта), либо после прохождения входного тестирования, минимальный проходной балл 10 (приложение 8).

Обучение по программе «Программирование на Python» базового уровня

нацелено на углубление и структурирование знаний основ современных языков программирования; умение на практике использовать сложные структуры данных; работать со специальными средствами и библиотеками языка Python.

К концу обучения подростки способны самостоятельно определять профессиональные задачи и пути решения; писать грамотный, красивый код; находить и обрабатывать ошибки в коде; разрабатывать эффективные алгоритмы и программы на основе изученного языка программирования Python; способны самостоятельно изучать новые технологии.

1.2 Цель и задачи программы

Цель: углубление и структурирование знаний основных современных языков программирования, получение навыков самостоятельного написания кода и разработки эффективных алгоритмов и программ.

Для успешной реализации поставленной цели необходимо решить ряд педагогических, развивающих и воспитательных **задач:**

Обучающие:

– закрепить опыт объектно-ориентированного и функционального программирования в различных интегрированных средах разработки на языке Python;

– углубить и структурировать знания основ современных языков программирования;

– сформировать навыки разработки эффективных алгоритмов и программ на основе изученного языка программирования Python;

– сформировать навыки работы со специальными средствами и библиотеками языка Python;

– сформировать навыки написания грамотного, красивого кода, уметь находить и обрабатывать ошибки в коде;

– научить на практике использовать сложные структуры данных.

Развивающие:

– способствовать развитию навыков логического и алгоритмического мышления;

– способствовать формированию общих представлений об информационной картине мира, об информации и информационных процессах как элементах

реальной действительности;

– способствовать формированию навыков планировать свои действия с учётом фактора времени, а также предвидеть результаты своей работы и оптимальные пути их достижения;

– способствовать развитию навыков самостоятельного изучения новых технологий.

Воспитательные:

– способствовать развитию делового сотрудничества, взаимоуважения внутри группы к в коллективе в целом;

– способствовать формированию целеустремленности, настойчивости в достижении поставленных целей;

– способствовать созданию условий для развития устойчивой потребности в самообразовании;

– способствовать воспитанию аккуратности при работе с компьютерным оборудованием.

1.3 Планируемые результаты

Дополнительная общеразвивающая программа направлена на достижение школьниками следующих личностных, метапредметных и предметных результатов:

Личностные результаты:

- активно вступает в диалог, ведет диалог с учетом общепринятых норм эффективной коммуникации;
- демонстрирует позитивное отношение к другому человеку, его мнению, результату его деятельности;
- проявляет устойчивый интерес к саморазвитию;
- ответственное отношение к обучению, способность довести до конца начатое дело;
- соблюдает правила техники безопасности при работе с компьютерной техникой;

Метапредметные результаты:

- умеет самостоятельно определять цели своего обучения, ставить и формулировать для себя новые задачи;
- умеет создавать, применять и преобразовывать знаки и символы, модели и схемы для решения учебно-исследовательских и проектных работ;
- умеет планировать последовательность шагов для достижения целей, соотношение своих действий с планируемыми результатами, осуществление контроля своей деятельности в процессе достижения результата;
- умеет работать в области использования информационно-коммуникационных технологий.

Предметные результаты

- умеет разрабатывать и использовать компьютерно-математические модели и использовать на практике сложные структуры данных;
- умеет работать со специальными средствами и библиотеками языка Python;
- умеет выполнять созданные программы, осуществлять их разработку, тестирование и отладку, используя изученный язык программирования;
- знает базовых управляющих конструкций объектно-ориентированного программирования и умение их использовать;
- знает самостоятельно создавать программы анализа данных на выбранном для этого универсальном языке программирования;
- знает определять результат выполнения алгоритма при заданных исходных данных.

1.4 Содержание общеразвивающей программы

УЧЕБНО-ТЕМАТИЧЕСКИЙ ПЛАН

№ п/п	Название раздела, темы	Кол-во часов			Формы аттестации/ контроля
		Всего	Теория	Практика	
Раздел 1. Программирование на Python		26	11	15	
1.1	Вводное занятие. Введение в программу 2-го года обучения Инструктаж по ТБ	2	1	1	Устный опрос
1.2	Повторение. Решение задач на основные конструкции и структуры данных. Решение задач на классы и библиотеки	6	3	3	Устный опрос, решение задач
1.3	Решение задач по теме «Повторение»	4	0	4	Решение задач
1.4	ООП	8	4	4	Устный опрос, решение задач
1.5	Рекурсия. Промежуточный контроль.	6	3	3	Устный опрос, решение задач. Практическое задание.
Раздел 2. PyGame		44	18	26	
2.1	Введение в репозитории. Подключение PyCharm. Работа с удаленными репозиториями	4	1	3	Устный опрос
2.2	PyGame 1. Введение	2	1	1	Устный опрос
2.3	PyGame 2. Игровой цикл. События	4	2	2	Практическое занятие
2.4	PyGame 3. Основные команды при одиночной работе с Git	4	2	2	Самостоятельная работа
2.5	PyGame 4. Клеточное поле	4	1	3	Устный опрос
2.6	PyGame 5. Классические игры на клеточном поле	4	2	2	Практическое задание.
2.7	PyGame 6. Изображения. Спрайты	4	2	2	Устный опрос.
2.8	PyGame 7. Столкновения и другие взаимодействия	4	2	2	Устный опрос.
2.9	PyGame 8. Игра в целом	6	2	4	Практическое задание
2.10	PyGame 9. Украшения игр	4	2	2	Практическое задание

2.11	Цели и подходы к тестированию. Создание "самодельных" тестов (без библиотек)	4	1	3	Практическое задание.
Раздел 3. Машинное зрение на Python		44	22	22	
3.1	Машинное зрение на Robomaster S1	8	4	4	Практическое задание
3.2	Машинное зрение на DJI Tello	6	3	3	Практическое задание
3.3	Машинное зрение на Coex Clever	8	4	4	Практическое занятие
3.4	Введение в OpenCV на Python	6	3	3	Практическое задание
3.5	OpenCV на Python с использованием Raspberry Pi	8	4	4	Практическое задание
3.6	Поиск и отслеживание с использованием OpenCV на Coex Clever	8	4	4	Практическое задание.
Раздел 4. Проекты WEB и API		16	8	8	
4.1	WEB. Работа с популярными форматами файлов (json, xml)	4	2	2	Устный опрос
4.2	WEB. Знакомство с API	4	2	2	Практическое задание
4.3	WEB. Понятие исключения, обработка исключений. Собственные исключения.	4	2	2	Самостоятельная работа
4.4	WEB. Работа с командной строкой (скрипты, аргументы). Периодические задачи (модуль shedule) Промежуточный контроль.	4	2	2	Устный опрос
Раздел 5. Проектная деятельность.		14	5	9	
5.1	Проблематизация.	2	1	1	Практическое задание
5.2	Гипотеза	2	1	1	Практическое задание
5.3	Цели и задачи	2	1	1	Практическое задание
5.4	Диаграмма Ганте	2	1	1	Практическое задание
5.5	Работа с презентациями.	2	1	1	Практическое задание
5.6	Работа над проектами	4	0	4	Защита проекта
Итого		144	64	80	

Содержание учебно-тематического плана

Раздел 1. Программирование на Python

Тема 1.0 Вводное занятие. Инструктажи по технике безопасности с отметкой в журнале. Введение в программу.

Практика: Установка ПО.

Тема 1.1 Повторение. Решение задач на основные конструкции и структуры данных. Решение задач на классы и библиотеки

Теория: Повторение основных конструкций и структур данных. Повторение модулей стандартной библиотеки Python.

Практика: решение задач.

Тема 1.2 Повторение. Решение задач на основные конструкции и структуры данных. Решение задач на классы и библиотеки

Практика: решение задач в рамках самостоятельной работы.

Тема 1.3 Решение задач по теме «Повторение»

Практика: решение задач в рамках самостоятельной работы.

Тема 1.4 ООП

Теория: Изучение объектно-ориентированного программирования, создание объектов, взаимодействие с ними.

Практика: решение задач.

Тема 1.5 Рекурсия

Теория: Создание рекурсивных функций, вызов функции и самовывоз внутри функции.

Практика: решение задач, промежуточный контроль.

Раздел 2. PyGame

Тема 2.1 Введение в репозитории. Подключение PyCharm. Работа с удаленными репозиториями.

Теория: Разновидности и предназначение систем контроля версий. Репозитории для

хранения и обмена данными. Использование среды разработки PyCharm удаленными репозиториями.

Практика: решение задач по работе с репозиторием.

Тема 2.2 PyGame 1. Введение

Теория: Что такое PyGame, его место в экосистеме и какие игры можно создавать с его помощью.

Практика: решение задач.

Тема 2.3 PyGame 2. Игровой цикл. События

Теория: Создание игрового цикла, обработка событий стандартных устройств ввода.

Практика: решение задач.

Тема 2.4 PyGame 3. Основные команды при одиночной работе с Git

Теория: Система контроля версий Git. Использование git при разработке приложения в одиночку — графический и консольный интерфейс, основные команды.

Практика: решение задач.

Тема 2.5 PyGame 4. Клеточное поле

Теория: Реализация клеточного поля в PyGame. Способы хранения глобальных настроек и обмена данными между игровыми агентами.

Практика: решение задач.

Тема 2.6 PyGame 5. Классические игры на клеточном поле

Теория: Обзор классических игр на клеточном поле и вариантов их реализации при помощи PyGame.

Практика: решение задач по реализации игры на клеточном поле.

Тема 2.7 PyGame 6. Изображения. Спрайты

Теория: Графические возможности PyGame, чтение, изменение и встраивание изображений. Техника работы со спрайтами.

Практика: решение задач по работе с изображениями.

Тема 2.8 PyGame 7. Столкновения и другие взаимодействия

Теория: Обсуждение основных видов взаимодействия между игроками и другими

сущностями игры, а также способов программной реализации такого рода взаимодействий.

Практика: решение задач.

Тема 2.9 PyGame 8. Игра в целом

Теория: Проектирование архитектуры игрового приложения.

Практика: создание своей игры.

Тема 2.10 PyGame 9. Украшения игр

Теория: Нюансы создания интерфейса игры, визуального оформления игровой механики, подключение звуковых эффектов.

Практика: украшение своей игры.

Тема 2.11 Цели и подходы к тестированию. Создание "самодельных" тестов (без библиотек).

Теория: как и зачем осуществляют тестирование в промышленной разработке. Подходы к тестированию программных продуктов. Технология создания тестов стандартными средствами Python.

Практика: тестирование своего игрового приложения.

Раздел 3. Машинное зрение на Python

Тема 3.1 Машинное зрение на Robomaster S1

Теория: Использование библиотеки алгоритмов распознавания карточек. Алгоритмы поиска и отслеживания.

Практика: Практическое задание программирования Robomaster S1.

Тема 3.1 Машинное зрение на DJI Tello

Теория: Поиск карточек, построение карты местности.

Практика: Практическое задание программирования DJI Tello.

Тема 3.1 Машинное зрение на Coex Clever

Теория: Поиск и распознавание маркеров, добавление новых маркеров.

Практика: Практическое задание программирования Coex Clever.

Тема 3.1 Введение в OpenCV на Python

Теория: Подключение и использование библиотеки OpenCV. Функции, объекты

библиотеки.

Практика: Практическое задание программирования распознавания текста.

Тема 3.1 OpenCV на Python с использованием Raspberry Pi

Теория: Чтение и запись данных с камеры на Raspberry Pi, создание алгоритма распознавания объектов.

Практика: Практическое задание программирования Raspberry Pi.

Тема 3.1 Поиск и отслеживание с использованием OpenCV на Coex Clever

Теория: Патрулирование местности и поиск объектов, отслеживание объектов. Алгоритмы повторного поиска.

Практика: Практическое задание программирования Coex Clever и Raspberry Pi.

Раздел 4. Проекты WEB и API

Тема 4.1 WEB. Работа с популярными форматами файлов (json, xml)

Теория: Структура форматов json и csv. Python-модули для работы с ними.

Основные сферы применения, практика применения в WEB.

Практика: решение задач.

Тема 4.2 WEB. Знакомство с API

Теория: Программный интерфейс приложения в разработке для основных существующих платформ. Особенности API в WEB-разработке.

Практика: решение задач.

Тема 4.3 WEB. Понятие исключения, обработка исключений. Собственные исключения.

Теория: Программные ошибки, понятие исключительной ситуации. Механизм исключений в Python, различные практики его использования.

Практика: решение задач - самостоятельная работа.

Тема 4.4 WEB. Работа с командной строкой (скрипты, аргументы).

Периодические задачи (модуль schedule)

Теория: Интерфейс командной строки основных операционных систем. Его применение в backend-разработке и администрировании.

Практика: решение задач, промежуточный контроль.

Раздел 5. Проектная деятельность

Тема 5.1 Проблематизация.

Теория: Выявление и рассмотрение проблем.

Практика: Практическое задание по выявлению проблемы.

Тема 5.2 Гипотеза.

Теория: Построение гипотезы решения проблемы.

Практика: Практическое задание по построению гипотезы.

Тема 5.3 Цели и задачи.

Теория: Написание цели и задачи по S.M.A.R.T. технологии.

Практика: Практическое задание по написанию цели.

Тема 5.4 Диаграмма Ганте

Теория: Построение диаграммы Ганте

Практика: Практическое задание по составлению диаграммы Ганте.

Тема 5.5 Работа с презентациями.

Теория: Оформление презентаций, анимация слайдов, добавление PNG файлов.

Практика: решение задач.

Тема 5.6 Работа над проектами

Теория: Структура форматов json и csv. Python-модули для работы с ними.

Основные сферы применения, практика применения в WEB.

Практика: Защита проекта.

2. Комплекс организационно-педагогических условий

2.1 Календарный учебный график

Год обучения	Дата начала обучения	Дата окончания обучения	Количество учебных недель	Количество учебных дней	Количество учебных часов	Режим занятий
1 год	16 сентября	31 мая	36	72	144	2 занятия по 2 часа в неделю

2.2 Условия реализации программы

Материально-техническое обеспечение

Требования к помещению:

- помещение для занятий, отвечающее требованиям СанПиН для учреждений дополнительного образования;
- качественное освещение.

Оборудование:

- столы, стулья по количеству обучающихся и 1 рабочее место для педагога;
- ноутбуки для каждого обучающегося и преподавателя;
- наушники;
- web-камера;
- набор Robomaster S1;
- Raspberry Pi;
- Coex clever4;
- многофункциональное устройство (принтер, сканер и копир);
- моноблочное интерактивное устройство;
- напольная мобильная стойка для интерактивных досок или универсальное настенное крепление;
- доска магнито-маркерная настенная;

- флипчарт.

Расходные материалы:

- whiteboard маркеры;
- стиратель с диска;
- бумага писчая;
- шариковые ручки;
- permanent маркеры.

Информационное обеспечение:

- операционная система (желательно Windows);
- поддерживаемые браузеры (для работы LMS): Yandex Browser, Firefox, Opera, Edge;
- среда Wing IDE 101 (версии 6 или выше);
- среда PyCharm Community Edition;
- пакет PyQt4 (на Qt5);
- пакет библиотек со SciPy: numpy, scipy, matplotlib, ipython + ipythonnotebook, sympy, pandas;
- рекомендуется установить ПО Anaconda.

Методическое обеспечение:

Методические пособия, разработанные преподавателем с учётом конкретных задач, варианты демонстрационных программ, материалы по терминологии ПО, инструкции по настройке оборудования, учебная и техническая литература. Используются педагогические технологии индивидуализации обучения и коллективной деятельности.

Кадровое обеспечение:

Реализовывать программу могут педагоги дополнительного образования, имеющие высшее образование (бакалавриат), среднее профессиональное образование, владеющие педагогическими методами и приемами, методикой преподавания основам программирования на языке Python, обладающие профессиональными знаниями и компетенциями в организации и проведении образовательного деятельности.

2.3 Форма аттестации и оценочные материалы

Мониторинг качества образования – это систематическая и регулярная процедура сбора данных по важным образовательным аспектам. Педагогический мониторинг это - непрерывное, систематическое отслеживание состояния и результатов какой-либо деятельности, системы с целью управления их качеством, повышения эффективности. Объектами мониторинга являются образовательный процесс и его результаты, личностные характеристики всех участников образовательного процесса, их потребности и отношение к образовательному учреждению.

Предметные результаты. Оценка предметных результатов состоит из суммарного учета результатов промежуточной и итоговой аттестации. Результаты входного контроля не учитываются.

Система отслеживания результатов обучающихся выстроена следующим образом:

- промежуточный контроль;
- итоговый контроль.

Текущий контроль осуществляется регулярно в течение учебного года. Контроль теоретических знаний осуществляется с помощью педагогического наблюдения, опросов. Проверка знаний и умений детей в форме педагогического наблюдения осуществляется в процессе выполнения ими практических заданий. В практической деятельности результативность оценивается качеством выполнения работ учащихся, где анализируются положительные и отрицательные стороны работ, корректируются недостатки.

Текущий контроль осуществляется путём определения качества выполнения практических заданий (Приложение 1), отслеживания динамики развития обучающегося. Промежуточный контроль реализуется в форме контрольного тестирования. Пример контрольного тестирования представлен в приложении 2, 3.

Итоговый контроль реализуется в форме защиты индивидуальных или групповых проектов. Примерная тематика итоговых проектов второго года обучения представлена в Приложениях 4,5.

Защита итогового проекта осуществляется путем выступления-презентации обучающимся или командой обучающихся. Тема проекта выбирается самостоятельно. Презентация должна включать в себя тему проекта, его цели и задачи, результаты, средства, которыми были достигнуты полученные результаты. Презентация может быть выполнена любым удобным наглядным показательным способом (видеоролик, презентация и т.п.).

Индивидуальный / групповой проект оценивается формируемой комиссией. Если проект выполнен группой обучающихся, то при оценивании учитывается не только уровень исполнения проекта в целом, но и личный вклад каждого из авторов. Решение принимается коллегиально.

Оценочные материалы для аттестации учащихся

Название	Вид контроля	Краткие указания по использованию
Приложение 2: Тестирование обучающихся	Промежуточный контроль	Может быть использовано для промежуточной оценки (тема 1.5). Максимум 50 баллов.
Приложение 3: Бланк для проведения промежуточного контроля	Промежуточный контроль	Может быть использовано для промежуточной оценки (тема 4.4). Максимум 25 баллов.
Приложение 5: «Лист оценки работы обучающихся в процессе защиты итогового проекта»	Итоговый контроль	Может быть использовано в завершении образовательного процесса для итоговой оценки по заданным критериям работы обучающихся в процессе работы над итоговым проектом (тема 5.6). Максимум 25 баллов.

На основании учета результатов по всем видам контроля максимальное значение получаемых баллов в год – 100 баллов. Сумма баллов результатов промежуточного контроля и защиты итогового проекта переводится в один из уровней освоения образовательной программы согласно таблице 4:

Уровень освоения программы по окончании обучения

Таблица 4

Баллы	Уровень освоения программы
0–39 баллов	Низкий
40–79 баллов	Средний
80-100 баллов	Высокий

Формы проведения итогов по каждой теме общеразвивающей программы соответствуют целям и задачам ДООП.

Личностные и метапредметные результаты отслеживаются посредством наблюдения за динамикой развития обучающегося в процессе освоения программы. По результатам наблюдения заполняются экспертные карты (Приложения 6,7).

Методические материалы

Образовательный процесс осуществляется в очной форме.

В образовательном процессе используются следующие **методы**:

1. объяснительно-иллюстративный;
2. метод проблемного изложения: анализ истории научного изучения проблемы, постановка проблемных вопросов, объяснение основных понятий, определений, терминов, демонстрация опыта, использование наглядности; самостоятельная постановка, формулировка и решение проблемы обучающимися: поиск и отбор аргументов, фактов, доказательств, самостоятельный поиск ответа обучающимися на поставленную проблему;
3. проектно-исследовательский: лабораторные занятия: работа с приборами, препаратами, техническими устройствами, эксперименты, опытническая работа;
4. словесный – беседа, рассказ, объяснение, пояснение, вопросы;
5. кейс-метод;
6. метод проектов;
7. наглядный:

- демонстрация плакатов, схем, таблиц, диаграмм;
- демонстрация возможностей устройства;
- использование технических средств;
- просмотр видеороликов;

8. практический:

- практические задания;
- анализ и решение проблемных ситуаций и т. д.

Выбор методов обучения осуществляется исходя из анализа уровня готовности обучающихся к освоению содержания модуля, степени сложности материала, типа учебного занятия.

Формы обучения:

– **фронтальная** – предполагает работу педагога сразу со всеми обучающимися в едином темпе и с общими задачами. Для реализации обучения используется компьютер педагога с мультимедиа проектором, посредством которых учебный материал демонстрируется на общий экран. Активно используются Интернет-ресурсы;

– **коллективная** – это форма сотрудничества, при котором коллектив обучает каждого своего члена и каждый член коллектива активно участвует в обучении своих товарищей по совместной учебной работе;

– **групповая** – предполагает, что занятия проводятся с подгруппой. Для этого группа распределяется на подгруппы не более 6 человек, работа в которых регулируется педагогом;

– **индивидуальная** – подразумевает взаимодействие преподавателя с одним обучающимся. Как правило данная форма используется в сочетании с фронтальной. Часть занятия (объяснение новой темы) проводится фронтально, затем обучающийся выполняют индивидуальные задания или общие задания в индивидуальном темпе;

– **дистанционная** – взаимодействие педагога и обучающихся между собой на расстоянии, отражающее все присущие учебному процессу компоненты. Для реализации дистанционной формы обучения весь дидактический материал

размещается в свободном доступе в сети Интернет, происходит свободное общение педагога и обучающихся в социальных сетях, по электронной почте, посредством видеоконференции или в общем чате. Кроме того, дистанционное обучение позволяет проводить консультации обучающегося при самостоятельной работе дома.

Формы организации учебного занятия:

В образовательном процессе помимо традиционного учебного занятия используются многообразные формы, которые несут учебную нагрузку и могут использоваться как активные способы освоения детьми образовательной программы, в соответствии с возрастом обучающихся, составом группы, содержанием учебного модуля: беседа, лекция, мастер-класс, практическое занятие, защита проектов, конкурс, викторина, диспут, круглый стол, «мозговой штурм», воркшоп, глоссирование, деловая игра, квиз, экскурсия.

Каждая тема курса начинается с постановки задачи – характеристики предметной области или конкретной программы на языке Python, которую предстоит изучить. С этой целью педагог проводит демонстрацию презентации или показывает саму программу, а также готовые работы, выполненные в ней. Закрепление знаний проводится с помощью практики отработки умений самостоятельно решать поставленные задачи, соответствующие минимальному уровню планируемых результатов обучения.

Основные задания являются обязательными для выполнения всеми обучающимися группы. Задания выполняются на компьютере с использованием интегрированной среды разработки. При этом обучающиеся не только формируют новые теоретические и практические знания, но и приобретают новые технологические навыки.

Методика обучения ориентирована на индивидуальный подход. Такая форма организации обучения стимулирует интерес обучающегося к предмету, активность и самостоятельность обучающихся, способствует объективному контролю глубины и широты знаний, повышению качества усвоения материала обучающимися,

позволяет педагогу получить объективную оценку выбранной им тактики и стратегии работы, методики индивидуального обучения и обучения в группе, выбора предметного содержания.

В ходе обучения проводится промежуточное тестирование по темам для определения уровня знаний обучающихся. Выполнение контрольных заданий способствует активизации учебно-познавательной деятельности и ведёт к закреплению знаний, а также служит индикатором успешности образовательного процесса.

По типу организации взаимодействия педагогов с обучающимися при реализации программы используются личностно ориентированные технологии, технологии сотрудничества.

Методы воспитания: мотивация, убеждение, поощрение, упражнение, стимулирование, создание ситуации успеха и др.

Педагогические технологии: индивидуализации обучения; группового обучения; коллективного взаимообучения; дифференцированного обучения; разноуровневого обучения; проблемного обучения; развивающего обучения; дистанционного обучения; игровой деятельности; коммуникативная технология обучения; коллективной творческой деятельности; решения изобретательских задач; здоровьесберегающая технология.

Реализация программы предполагает использование здоровьесберегающих технологий.

Здоровьесберегающая деятельность реализуется:

- через создание безопасных материально-технических условий;
- включением в занятие динамических пауз, периодической смены деятельности обучающихся;
- контролем соблюдения обучающимися правил работы на ПК;
- через создание благоприятного психологического климата в учебной группе в целом.

Дидактические материалы:

Методические пособия, разрабатываемые преподавателем с учётом конкретных условий. Техническая библиотека объединения, содержащая справочный материал, учебную и техническую литературу. Индивидуальные задания.

Методическое обеспечение учебного процесса включает разработку преподавателем методических пособий, вариантов демонстрационных программ и справочного материала.

Список литературы

Литература, использованная при составлении программы:

1. Изучаем Python: программирование игр, визуализация данных, веб-приложения. 3-е изд. — СПб.: Питер, 2020.
2. Изучаем Python, том 1, 5-е изд.: Пер. с англ. — СПб.: ООО “Диалектика”, 2019.
3. Изучаем Python, том 2, 5-е изд. : Пер. с англ. — СПб. : ООО “Диалектика”, 2020.
4. Как устроен Python. Гид для разработчиков, программистов и интересующихся. — СПб.: Питер, 2019.

Литература для обучающихся и родителей:

Классические задачи Computer Science на языке Python, Дэвид Копец –СПб.: Питер, 2022 – 224 с.;

Современные операционные системы, Таненбаум Эндрю, Бос Херберт – СПб.:Питер, 2022 – 1120 с.;

Python Быстрый старт, Джейми Чан, 352 стр. 2021 г. – СПб.: Питер, 2022 – 224 с.

Электронные ресурсы:

Питонтьютор. Бесплатный курс по программированию с нуля. // [Электронный ресурс] URL: <https://pythontutor.ru/> (дата обращения: 14.04.2023);

Code Basics: обучение базовым аспектам языков программирования от образовательной платформы Hexlet. // [Электронный ресурс] URL: <https://ru.code-basics.com/> (дата обращения: 20.04.2023)

**Лист оценки работы обучающихся
в процессе разработки**

№ группы: _____

Дата: _

№ п/п	ФИ обучающегося	Сложность языка разработки (по шкале от 0 до 5 баллов)	Соответствие написания программного кода поставленной задаче (по шкале от 0 до 5 баллов)	Презентация модели по плану. Степень владения специальными терминами (по шкале от 0 до 5 баллов)	Степень увлеченности процессом и стремления к оригинальности при выполнении заданий (по шкале от 0 до 5 баллов)	Кол-во вопросов и затруднений (шт. за одно занятие)
1.						
2.						
3.						
4.						
5.						
6.						
7.						
8.						
9.						
10.						

**Пример контрольного тестирования (тема 1.5)
максимальное количество баллов 50**

№	Вопросы	Верный ответ
1	<p>Что выведет следующий фрагмент кода? (5 баллов)</p> <pre>x = 4.5 y = 2 print(x // y)</pre> <p>Варианты ответов:</p> <p>1) 2.0 2) 2.25 3) 9.0 4) 20.25 5) 21</p>	2.0
2	<p>Что выведет следующий код, при его исполнении? Используется Python 3.x. (5 баллов)</p> <pre>print(type(1 / 2))</pre> <p>Варианты ответов</p> <p>class 'int' class 'number' class 'float' class 'double' class 'tuple'</p>	class 'float'

3	<p>Что будет напечатано? (5 баллов)</p> <pre>kvps = {"user","bill", "password","hillary" } print(kvps['password'])</pre> <p>Варианты ответов:</p> <ol style="list-style-type: none"> 1) user 2) bill 3) password 4) hillary 5) Ничего. TypeError. 	Ничего. TypeError.
4	<p>Что будет напечатано? (5 баллов)</p> <pre>name = "snow storm" print("%s" % name[6:8])</pre> <p>Варианты ответов:</p> <ol style="list-style-type: none"> 1) st 2) sto 3) to 4) Syntax Error 	to
5	<p>Что напечатает следующий код: (5 баллов)</p> <pre>word = 'foobar' print(word[3:] + word[:3])</pre> <p>Варианты ответов:</p> <ol style="list-style-type: none"> 1) foobar 2) obarof 3) barfoo 4) SyntaxError 	barfoo

6	<p>Что выведет следующая программа? (5 баллов)</p> <pre>x = True y = False z = False if not x or y: print(1) elif not x or not y and z: print(2) elif not x or y or not y and x: print(3) else: print(4)</pre> <p>Варианты ответов:</p> <p>1) 1 2) 2 3) 3 4) 4</p>	3
7	<p>Что выведет следующая программа? (5 баллов)</p> <pre>a = [1,2,3,None,(),[],] print(len(a))</pre> <p>Варианты ответов</p> <p>1) 4 2) 5 3) 6 4) 7</p>	6

8	<p>Имеем следующую последовательность действий, чему равна переменная L2 ?: (5 баллов)</p> <pre>>>> L1 = [2, 3, 4] >>> L2 = L1 >>> L1[0] = 24 >>> L1 [24, 3, 4] >>> L2</pre> <p>Варианты ответов:</p> <ol style="list-style-type: none"> 1) [2,3,4] 2) [24,3,4] 3) [2,3,24] 4) [3,4,2] 	[24,3,4]
9	<p>Что покажет этот код? (5 баллов)</p> <pre>for i in range(5): if i % 2 == 0: continue print(i)</pre> <p>Варианты ответов:</p> <ol style="list-style-type: none"> 1) Ошибку, так как i не присвоена 2) Ошибку из-за неверного вывода 3) Числа: 1, 3 и 5 4) Числа: 0, 2 и 4 5) Числа: 1 и 3 	Числа: 1 и 3

10	<p>Что покажет этот код? (5 баллов)</p> <pre>for j in 'Hi! I\'m mister Robert': if j == '\': print ("Найдено") break else: print ("Готово")</pre> <p>Варианты ответов:</p> <ol style="list-style-type: none">1) Ошибку в коде2) "Найдено" и "Готово"3) "Готово"4) "Найдено"	"Найдено"
----	--	-----------

Пример контрольного тестирования (тема 4.4)
максимальное количество баллов 25

1. Движение шасси Robomaster S1 выполняется командой: (1 балл)
 - chassis_ctrl.move()
 - chassis.move()
 - move()
 - chassis_move()
2. События в Pygame - это (1 балл)
 - события связанные с игрой и они хранятся в списке событий
 - события связанные с оконным дисплеем и их можно запросить командой get()
 - события связанные с компьютером и они перезаписываются
3. Асинхронные функции это (1 балл)
 - функции которые не зависят от друг друга
 - функции которые запускаются одновременно
 - функции которые могут приостанавливаться для выполнения другой функции
 - функции которые выполняются по очереди
4. Асинхронные функции вызываются: (1 балл)
 - await
 - async
 - def
 - progs
5. Raspberry Pi - это: (1 балл)
 - микрокомпьютер программируемый на любом языке
 - микрочип программируемый на Python
 - передатчик кода с компьютера на моторы
6. Напишите программу считывающая сообщения от пользователя и записывающая на локальный хост (10 баллов)

7. Напишите программу которая записывает изображение с камеры Raspberry Pi (10 баллов)



Примерные темы итоговых проектов

1. Разработка приложения «Текстовый редактор».
2. Разработка приложения «Графический редактор».
3. Разработка игры «Гексагон» с графическим интерфейсом.
4. Создание чат бота.
5. Разработка игры для трех игроков с возможностью добавления ботов.
6. Создание WEB-проекта, предоставляющего для команд пользователей совместное рабочее пространство.

Критерии оценки итоговых проектов

№ группы: _____

Дата: _

№ п/п	ФИО обучающегося	Соответствие теме (по шкале от 0 до 5 баллов)	Сложность функционала (по шкале от 0 до 5 баллов)	Качество кода (по шкале от 0 до 5 баллов)	Удобство использования (по шкале от 0 до 5 баллов)	Презентация (по шкале от 0 до 5 баллов)
1.						
2.						
3.						
4.						
5.						
6.						
7.						
8.						
9.						
10.						

Значение личностных результатов обучающегося:

3 балла – качество проявляется систематически

2 балла – качество проявляется ситуативно

1 балл – качество не проявляется

Значение показателя по группе:

1 - 1,7 балла – низкий уровень развития качества в группе

1,8 - 2,5 балла – средний уровень развития качества в группе

2,6 - 3 балла – высокий уровень развития качества в группе

**Входное тестирование
(максимум 20 баллов)**

1. Язык программирования Python подходит для разработки: (1 балл)

- Ничего из этого
- Компьютерных и мобильных приложений
- Все вышеперечисленное
- Аналитика и машинное обучение
- Игр

2. Что хранит в себе переменная? (1 балл)

- Тип
- Имя
- Длину своего значения
- Значение

3. Что обозначает тип данных `int`? (2 балла)

- Вещественное
- Строковое
- Логическое
- Целочисленное

4. Выберите правильную запись оператора присваивания (1 балл)

- $10 = x$
- $y = 7,8$
- $x = 5$
- $a == b + x$

5. Укажите оператор ввода: (1 балл)

- `input()`
- `random()`
- `int()`
- `print()`

6. Сколько возможных значений у переменной типа `bool`? (2 балла)

- Бесконечность не предел
- 10
- 2
- 4

7. Оператор цикла в языке Python: (3 балла)

- `print`
- `while`
- `if`
- `for`

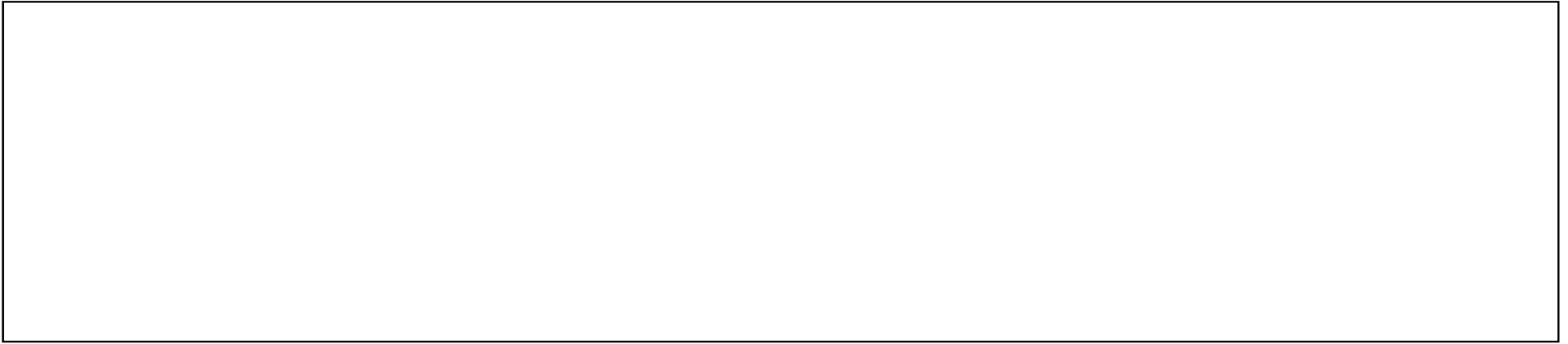
8. Для чего нужен оператор `break`? (2 балла)

- Для поломки компьютера
- Для удаления программы
- Для выхода из цикла
- Для завершения программы

9. Напишите программу которая принимает 1 число и выводит 3 следующих за ним числа (3 балла)



10. Напишите программу которая принимает 10 различных чисел и выводит 2 списка: отрицательных и положительных (4 балла)



Аннотация

Программа «Программирование на Python» имеет техническую направленность. Цель программы: углубление и структурирование знаний основных современных языков программирования, получение навыков самостоятельного написания кода и разработки эффективных алгоритмов и программ.

Данная образовательная программа интегрирует в себе достижения сразу нескольких традиционных направлений, а главной её особенностью является большой блок практических заданий и самостоятельная работа над решением поставленных задач: школьники учатся решать задачи без помощи преподавателя, что способствует развитию у них навыков алгоритмического и логического мышления, умению мыслить самостоятельно и повышает мотивацию учащихся к обучению.

Программа рассчитана на обучающихся 12–17 лет.

По уровню освоения программа общеразвивающая, базового уровня. Объём общеразвивающей программы: 144 часа.