

Муниципальное бюджетное образовательное учреждение
дополнительного образования
«Станция юных техников «Поиск»

РАССМОТРЕНО

Методическим советом

МБОУ ДО «СЮТ «Поиск»

Протокол № 3

23.05. 2022

УТВЕРЖДАЮ

Директор

МБОУ ДО «СЮТ «Поиск»

Вяткина Т.В.

2022



ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ
ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА

«Программирование на языке Python»

техническая направленность

стартовый, базовый уровень

Возраст обучающихся: 14-17 лет

Срок реализации: 2года

Составитель:

Зинченко В.И.,

педагог

дополнительного образования

Кемерово
2022

Оглавление

Пояснительная записка	3
Цель и задачи программы	6
Цель и задачи программы стартового уровня обучения	7
Учебный план программы стартового уровня обучения	8
Содержание программы стартового уровня обучения	9
Цель и задачи программы базового уровня обучения	11
Учебный план программы базового уровня обучения	12
Содержание программы базового уровня обучения	15
Планируемые результаты	19
Комплекс организационно-педагогических условий	21
Список литературы	25

КОМПЛЕКС ОСНОВНЫХ ХАРАКТЕРИСТИК ДОПОЛНИТЕЛЬНОЙ ОБЩЕОБРАЗОВАТЕЛЬНОЙ ОБЩЕРАЗВИВАЮЩЕЙ ПРОГРАММЫ

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Дополнительная общеобразовательная общеразвивающая программа «Программирование на языке Python» имеет **техническую направленность**.

Программа составлена в соответствии с требованиями, предъявляемыми к данному виду учебно-методических и программно-методических документов, и регламентируется следующими нормативно правовыми документами: Федеральным законом Российской Федерации от 26.07.2019г. №273-ФЗ «Об образовании в Российской Федерации», Распоряжение Министерства просвещения РФ от 10 ноября 2021 г. № ТВ-1984/04 «Об утверждении методических рекомендаций по созданию и функционированию центров цифрового образования «IT-куб» в рамках региональных проектов, обеспечивающих достижение целей, показателей и результата федерального проекта «Цифровая образовательная среда» национального проекта «Образование»; приказа Министерства образования и науки Российской Федерации от 19.11.2018 г. N 196 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам»; Приказ Министерства просвещения Российской Федерации от 03.09.2019 г. № 467 «Об утверждении Целевой модели развития региональных систем дополнительного образования детей»; Постановление Главного государственного санитарного врача РФ от 28.09.2020 г. № 28 «Об утверждении СанПиН 2.4.3648-20 «Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи»; Уставом ОУ МБОУ ДО «Станция юных техников «Поиск»; «Положением о дистанционном обучении учащихся МБОУ ДО «Станция юных техников «Поиск» от 6.04.2020.

В настоящее время мы переживаем большие изменения в развитии общества. В современную жизнь человека всё больше внедряются компьютеры и информационные технологии. Всё большее значение приобретает умение человека грамотно обращаться с компьютером, причём зачастую не на пользовательском уровне, а на уровне начинающего программиста.

Изучение основных принципов программирования невозможно без регулярной практики написания программ на каком-либо языке. Для обучения был выбран язык Python. Данный выбор обусловлен тем, что синтаксис языка достаточно прост и интуитивно понятен, а это понижает порог вхождения и позволяет сосредоточиться на логических и алгоритмических аспектах программирования, а не на выучивании тонкостей синтаксиса. При этом Python является очень востребованным языком; он отлично подходит для знакомства с различными современными парадигмами программирования и активно применяется в самых разных областях от разработки веб-приложений до

машинного обучения.

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

Знания и умения, приобретённые в результате освоения курса, могут быть использованы учащимися 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.

Новизна и актуальность

Новизна программы заключается в том, что Python дает более широкие возможности в области программирования, чем Pascal, который входит в школьный курс информатики. На языке Python можно легко и быстро создавать простые компьютерные игры, трёхмерные модели и программировать роботов.

Актуальность программы. В рамках изучения программы, обучающиеся постоянно будут сталкиваться с необходимостью самостоятельной работы над заданиями: школьники учатся решать задачи без помощи преподавателя. Для этого в содержании курса фигурируют задания, в которых для решения задачи необходимо найти какую-то информацию в сети Интернет; может потребоваться устранение ошибки, которую не так просто обнаружить; условие сформулировано недостаточно прозрачно и ученику необходимо самостоятельно формализовать его (или задать правильные вопросы преподавателю). Все эти знания, умения и практические навыки решения актуальных задач, полученные на занятиях, готовят школьников к самостоятельной проектно-исследовательской деятельности с применением современных технологий.

Отличительной особенностью программы является принцип её организации, она организована по принципу дифференциации по уровням сложности и состоит из двух уровней стартового и базового, объединяющих в себе следующие логически завершённые разделы «Введение в программирование» - позволяет обеспечить начальную подготовку детей в области программирования и формирует положительную мотивацию к языкам программирования; «Базовые конструкции языка Python» и «Решение прикладных задач в Python» - предполагает освоение специализированных знаний в языке программирования Python.

Адресат программы. Программа предназначена для обучающихся, не имеющих предварительной подготовки по основным темам и разделам программы, адресована обучающимся 14-17 лет и имеющим минимальный необходимый уровень входных компетенций: уверенный пользователь ПК, проявляют интерес к программированию. Наполняемость групп стартового и базового уровня обучения не более 12 человек.

Обучающиеся, имеющие начальные знания по основным разделам программы, могут быть зачислены сразу на базовый уровень обучения, минуя

стартовый.

Система набора осуществляется на основании результатов собеседования, просмотра работ, наличия базовых знаний в определённой области и т.д. Набор учащихся в объединение осуществляется на добровольной основе. Зачисление в группы производится на основании заполнения родителями (законными представителями) заявления о зачислении в учебное объединение.

Срок реализации программы и объем учебных часов. Программа реализуется на базе ИТ-Куба в учебном кабинете с необходимым оборудованием, техническим и ресурсным обеспечением в соответствии с перечнем, указанным в методических рекомендациях по созданию и функционированию центров цифрового образования «ИТ-Куб» в рамках федерального проекта «Цифровая образовательная среда» национального проекта «Образование». Предполагает два уровня обучения – стартовый и базовый, каждый соответствует одному году обучения и рассчитана на 360 часов, из них 144 часа – стартовый уровень обучения, 216 – базовый уровень обучения.

Формы организации образовательного процесса.

- **индивидуальная работа** (самостоятельное выполнение заданий);
- **групповая**, работа с наглядными пособиями по подгруппам;
- **массовая**, обсуждение результата, исправление ошибок.

Виды занятий.

Стартовый уровень обучения.

Беседы, видео просмотры, инструктаж, выполнение практической работы, интеллектуальная игра, соревнование, проект.

Базовый уровень обучения.

Выставка, чемпионат, проект, соревнование.

Режим занятий. Стартовый уровень обучения рассчитан на 144 часа, 2 часа 2 раза в неделю. Базовый уровень обучения рассчитан на 216 часов, 2 раза в неделю по 3 часа. Продолжительность учебного занятия 45 минут. Между учебными занятиями предполагается перерыв 10 минут. Учебные занятия строятся с учетом психолого-педагогических и физических возрастных особенностей детей.

Цели и задачи программы

Цель программы: изучение основ программирования на языке Python, основных приёмов написания программ на современном языке программирования, развитие алгоритмического мышления учащихся, творческих способностей, аналитических и логических компетенций.

Задачи:

- познакомить обучающихся с конструкцией языка программирования Python и работой в интегрированной среде разработки на языке Python;
- формировать у обучающихся навыки разработки эффективных алгоритмов и программ на основе изучения языка программирования Python;
- развивать у обучающихся навыки алгоритмического и логического мышления, грамотной разработки программ, поиска информации в сети Интернет, анализа выбранной информации на соответствие запросу, использования информации при решении задач;
- способствовать формированию творческой личности, обладающей информационными компетенциями, владеющей базовыми понятиями теории алгоритмов, умеющей разрабатывать эффективные алгоритмы и реализовывать их в виде программы, написанной на языке программирования Python;
- развивать инженерное мышление, изобретательность, умение самостоятельно решать поставленную задачу;
- содействовать профессиональному самоопределению учащихся;
- воспитывать упорство в достижении поставленной цели и результата;
- воспитывать навыки самоорганизации; самостоятельной и командной работы.

УЧЕБНЫЙ ПЛАН ПРОГРАММЫ

Уровень обучения	Объем учебных часов	Всего учебных недель	Режим работы
стартовый	144	36	4 часа в неделю: 2 раза в неделю по 2 часа
базовый	216	36	6 часов в неделю: 2 раза в неделю по 3 часа

Цель и задачи программы стартового уровня обучения

Цель уровня: формирование познавательной активности обучающихся в области объектно-ориентированного и функционального программирования и алгоритмизации, приобретение необходимых навыков работы с информацией, а также получение первичного опыта работы с базовыми конструкциями языка программирования на основе работы в интегрированных средах разработки на языке Python.

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

Обучающие:

- обучить обучающихся базовым навыкам работы с основными конструкциями языка программирования;
- познакомить обучающихся с комплексом базовых понятий и принципов функционального и объектно-ориентированного программирования (изучение структур данных, базовые принципы их обработки);
- способствовать получению у обучающихся опыта работы в интегрированной среде разработки на языке Python;
- формировать у обучающихся необходимые навыки работы с информацией (поиск, анализ, использование информации в сети Интернет).

Развивающие:

- развивать у обучающихся интерес к программированию и техническим видам творчества;
- создавать условия для развития поисковой активности, исследовательского мышления;
- развивать самостоятельность и творческий подход к решению задач;
- развивать логическое, техническое мышление;
- создать условия для творческой самореализации личности ребёнка посредством получения навыков разработки эффективных алгоритмов;
- формировать и развивать познавательную потребность в освоении технических знаний.

Воспитательные:

- формировать навык работы в группе;
- развивать коммуникативную культуру обучающихся, как внутри проектных групп, так и в коллективе в целом;
- создавать творческую атмосферу сотрудничества, обеспечивающую развитие личности, социализацию и эмоциональное благополучие каждого ребёнка;
- формировать стремление к получению качественного законченного результата;
- формировать активную жизненную позицию, гражданско-патриотическую ответственность.

Учебный план программы стартового уровня обучения

№	Название раздела, темы	Всего			Формы контроля
		Всего	Теория	Практика	
1.	Введение в программирование	6	8	18	
1.1	Понятия кода, интерпретатора, программы	8	2	6	Беседа. Опрос
1.2	Интегрированные среды, исполнение кода и отладка	10	2	8	Беседа. Опрос Практическая работа
1.3	Переменные, основные операторы	6	2	4	Беседа. Опрос Практическая работа
1.4	Базовые типы данных, ветвления	2	2		Беседа. Опрос Самостоятельная работа
2.	Базовые конструкции в Python	40	6	34	
2.1	Циклы, срезы, списочные выражения	12	2	10	Практическая работа
2.2	Методы списков и строк. Функции	8	2	6	Практическая работа
2.3	Решение задач по пройденным темам. Практическая работа	10	2	8	Самостоятельная работа
2.4	Контрольная работа по темам	10		10	Контрольная работа
3.	Решение прикладных задач в Python	76	20	56	
3.1	Функции (углублённое рассмотрение), другие структуры данных, библиотеки Python, введение в ООП	40	8	32	Беседа. Опрос
3.2	Решение задач	20	8	12	Контрольная работа
3.3	Проверочные работы и итоговая контрольная работа	16	4	12	Контрольная работа
	Всего:	144	35	85	

Содержание программы стартового уровня обучения

Раздел 1. Введение в программирование

Тема 1.1. Понятия кода, интерпретатора, программы

Теория: понятия кода, интерпретатора, программы. Простейшие программы с выводом на экран. Знакомство с виртуальной средой взаимодействия – регистрация, организация личного кабинета, поиск и выкладывание материалов; знакомство с системой автоматизированной проверки задач и системой Яндекс.

Практика: решение задач.

Тема 1.2. Интегрированные среды, исполнение кода и отладка

Теория: интегрированные среды, исполнение кода. Основные понятия программирования: исполнитель, система команд, алгоритм, программа, среда разработки.

Практика: решение задач.

Тема 1.3. Переменные, основные операторы

Теория: условный оператор. Переменные и арифметика. Погружение в условия. Ввод-вывод в программе, условный оператор, оператор цикла с предусловием. Простейшие программы с использованием условного оператора if, оператора циклов while и операторов ввода-вывода. Технология разработки программы.

Практика: решение задач.

Тема 1.4. Базовые типы данных, ветвления

Теория: знакомство со списками, строками, множествами и кортежами в Python.

Практика: разработка алгоритмов и программ, определение работоспособности разработанной программы.

Раздел 2. Базовые конструкции в Python

Тема 2.1. Циклы, срезы, списочные выражения

Теория: понятие о языке высокого уровня Python. Структура программы, переменные и константы, работа с числовыми переменными, арифметические операторы в Python. Основные управляющие конструкции алгоритмов с ветвлением в Python. Знакомство со срезами и диапазонами. Равенство и совпадение объектов. Устройство циклов for. Основные управляющие конструкции циклического алгоритма в Python. Простейшие циклы и циклы с переменными.

Практика: решение задач по теме «Срезы и диапазоны».

Тема 2.2. Методы списков и строк. Функции – 8 часов.

Теория: списочные выражения. Методы split и join. Другие методы списков и строк. Знакомство с функциями. Области видимости переменных. Работа со списками, строками, множествами и кортежами в Python. Понятие итератора, подпрограммы, процедуры, функции. Функции в Python.

Практика: решение задач.

Тема 2.3. Решение задач по пройденным темам

Теория: повторение пройденных тем.

Практика: решение задач и выполнение практических и творческих заданий по пройденным темам.

Тема 2.4. Контрольная работа по темам

Практика: решение задач контрольной работы на методы списков и строк, по темам «Методы split и join», «Функции», «Области видимости переменных». Анализ результатов.

Раздел 3. Решение прикладных задач в Python

Тема 3.1. Функции (углублённое рассмотрение), другие структуры данных, библиотеки Python, введение в ООП

Теория: функции, функциональная парадигма программирования. Понятие ассоциативного массива. Словари и множества, модульный принцип компоновки программы. Подключение и использование модулей стандартной библиотеки Python, дополнительные библиотеки. Понятие репозитория различных пакетов Python. Основы ООП.

Практика: работа с документацией в стандартной библиотеке, с внешними библиотеками Python и утилитой pip.

Тема 3.2. Решение задач на пройденные темы

Прикладные задачи.

Практика: решение задач по темам «Функции и функциональная парадигма программирования», «Словари и множества», «Стандартная библиотека Python», «Дополнительные библиотеки Python», «Введение в ООП». Самостоятельная работа: работа с конспектом, который описывает способы решения задач, разработка проекта по индивидуальному заданию, составление отчета о выполнении индивидуальной или совместной работы.

Тематика самостоятельных работ:

- задачи, в которых по заданному алгоритму необходимо написать программу;
- задачи, в которых необходимо составить алгоритм решения и написать по нему программу;
- задачи, для решения которых необходимо найти некоторую информацию в Интернете.

Выполненные самостоятельные работы загружаются в автоматизированную систему проверки задач на Яндекс.

Тема 3.3. Проверочные работы и итоговая контрольная работа

Теория: повторение изученного материала, подготовка к контрольной работе. Разбор контрольной работы.

Практика: выполнение проверочных работ. Итоговая контрольная работа по изученным темам. Разработка проекта по индивидуальному заданию, составление отчёта о выполнении индивидуальной или совместной работы.

Учебный план программы базового уровня

Цель уровня: развитие полученных на первом году обучения навыков использования сложных структур данных и программирования в различных интегрированных средах разработки на языке Python, углубление и структурирование знаний основ 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:

- способствовать закреплению у обучающихся опыта объектно-ориентированного и функционального программирования в различных интегрированных средах разработки на языке Python;
- отработка навыков программирования, углубление и структурирование знаний основ современных языков программирования;
- формировать у обучающихся навыки разработки эффективных алгоритмов и программ на основе изученного языка программирования Python;
- обучать обучающихся написанию грамотного кода, умению находить и обрабатывать ошибки в коде.

Развивающие:

- формировать навык логического и алгоритмического мышления;
- развивать самостоятельность и творческий подход к решению задач с использованием средств вычислительной техники;
- формировать общие представления об информационной картине мира, об информации и информационных процессах как элементах реальной действительности;
- развивать навыки работы с различными источниками информации, необходимой для решения учебных задач;
- формировать навык планировать свои действия с учётом фактора времени, а также предвидеть результаты своей работы и оптимальные пути их достижения; – развитие навыка самостоятельного изучения новых технологий.

Воспитательные:

- способствовать воспитанию отношений делового сотрудничества, взаимоуважения;
- формировать целеустремленность, настойчивость в достижении поставленных целей;
- формировать ответственность, равнодушие, взаимоуважение и толерантность в процессе группового взаимодействия;
- развивать коммуникативные отношения внутри проектных групп и в коллективе в целом.

Учебный план программы базового уровня обучения

№ п/п	Название раздела, темы	Количество часов			Формы контроля
		Всего	Теория	Практика	
1	Программирование на Python	42	17	25	
1.1	QT 1. Что такое QT и PyQT	3	3	-	Беседа. Опрос
1.2	QT 2. QtDesigner	3	3	-	Практическая работа
1.3	QT 3. Файлы в Python. Типы файлов и работа с ними. Внутреннее устройство файлов	6	3	3	Практическая работа
1.4	QT 4. Диалоги, работа с изображениями	6	1	5	Самостоятельная работа
1.5	QT 5. Работа с простыми таблицами (csv). Работа с табличными данными в PyQT	6	1	5	Практическая работа
1.6	QT 6. Введение в БД, работа с SQL-таблицами и отображение данных в PyQT. Часть 1	3	-	3	Практическая работа
1.7	QT 7. Введение в БД, работа с SQL-таблицами и отображение данных в PyQT. Часть 2	3	-	3	Практическая работа
1.8	QT 8. Обработка событий. Сборка независимого приложения	9	3	6	Опрос. Беседа. Минипроект
1.9	Защита проекта QT	3	3	-	Защита проекта
2	PyGame	69	39	30	
2.1	Введение в репозитории. Подключение PyCharm. Работа с удаленными репозиториями	6	6	-	Беседа. Опрос
2.2	PyGame 1. Введение	6	3	3	Беседа. Опрос
2.3	PyGame 2. Игровой цикл. События	6	3	3	Беседа. Опрос
2.4	PyGame 3. Основные команды при одиночной работе с Git	6	3	3	Беседа. Опрос
2.5	PyGame 4. Клеточное поле	6	3	3	Беседа. Опрос
2.6	PyGame 5. Классические игры на клеточном поле	6	3	3	Беседа. Опрос

2.7	PyGame 6. Изображения. Спрайты	6	3	3	Практическая работа
2.8	PyGame 7. Столкновения и другие взаимодействия	6	3	3	Практическая работа
2.9	PyGame 8. Игра в целом	6	3	3	Практическая работа
2.10	PyGame 9. Украшения игр	6	3	3	Практическая работа
2.11	Цели и подходы к тестированию. Создание "самодельных" тестов (без библиотек)	6	3	3	Тест
2.12	Защита проекта PyGame	3	3	-	Защита проектов
	Проекты WEB и API	105	45	60	
3.1	WEB. Работа с популярными форматами файлов (json, xml)	3	3	-	Беседа. Опрос
3.2	WEB. Знакомство с API	6	3	3	Беседа. Опрос
3.3	WEB. Понятие исключения, обработка исключений. Собственные исключения	6	3	3	Беседа. Опрос
3.4	WEB. Работа с командной строкой (скрипты, аргументы). Периодические задачи (модуль schedule)	6	3	3	Самостоятельная работа
3.5	WEB. Библиотека argparse. Задачи на создание скриптов с ее помощью	9	3	6	Самостоятельная работа
3.6	WEB. Работа с протоколом HTTP	6	3	3	Самостоятельная работа
3.7	WEB. Решение задач на API Яндекс-карт	9	3	6	Самостоятельная работа
3.8	WEB. Введение во flask. Обработка html-форм	6	3	3	Практическая работа
3.9	WEB. Шаблоны. flask--wtf	6	3	3	Практическая работа
3.10	WEB. Знакомство с flasksqlalchemy	6	3	3	Практическая работа
3.11	WEB. Flask-sqlalchemy	6	3	3	Практическая работа
3.12	WEB. REST-API. Понятие. Делаем простое REST-API	6	3	3	Практическая работа
3.13	WEB. REST-API. Flask-restfull	6	3	3	Практическая

					работа
3.14	WEB. Разворачиваем проект в облаке. Дорешка	9	3	6	Практическая работа
3.15	WEB. Алиса. Часть 1	6	-	3	Практическая работа
3.16	WEB. Алиса. Часть 2	6	-	3	Практическая работа
3.17	WEB. Защита проекта WebServer + API	3	3	-	Защита проекта
	Общее количество часов	216	101	115	

Содержание учебного плана программы базового уровня обучения

Тема 1.1 QT 1. Что такое QT и PyQt

Теория: Обзор рынка фреймворков для разработки настольных приложений на Python. История, предназначение, преимущества и недостатки фреймворка Qt. Подключение и первые шаги с PyQt.

Практика: решение задач.

Тема 1.2 QT 2. QtDesigner

Теория: Изучение основных элементов графического интерфейса и настроек QtDesigner, способы интеграции графического и текстового стиля создания настольных приложений.

Практика: решение задач, связанных с созданием простейшего однооконного приложения.

Тема 1.3 QT 3. Файлы в Python. Типы файлов и работа с ними. Внутреннее устройство файлов

Теория: Принципы работы с файловой системой средствами Python. Чтение и запись информации в файл приложением, созданным при помощи QtDesigner.

Практика: решение задач, связанных с работой с файлами из настольного приложения.

Тема 1.4 QT 4. Диалоги, работа с изображениями

Теория: Изучение набора встроенных в QtDesigner диалоговых окон. Способы чтения, модификации и внедрения графических файлов в настольное приложение.

Практика: решение задач по обработке изображений.

Тема 1.5 QT 5. Работа с простыми таблицами (csv). Работа с табличными данными в PyQt

Теория: Формат данных csv, его преимущества и недостатки. Способы чтения и записи файлов в данном формате. Работа с табличными данными в PyQt в рамках собственного настольного приложения.

Практика: решение задач по работе с таблицами.

Тема 1.6 QT 6. Введение в БД, работа с SQL-таблицами и отображение данных в PyQt. Часть 1

Теория: Предназначение и разновидности баз данных. Реляционные базы данных и язык запросов SQL. Обращение к базе данных из создаваемого при помощи PyQt настольного приложения.

Практика: решение задач по работе с БД и SQL.

Тема 1.7 QT 7. Введение в БД, работа с SQL-таблицами и отображение данных в PyQt. Часть 2

Теория: Технология написания более сложных SQL запросов. Объектнореляционное отображение.

Практика: решение задач на работу с БД из PyQt.

Тема 1.8 QT 8. Обработка событий. Сборка независимого приложения

Теория: Петля событий настольного приложения. Обработка событий стандартных устройств ввода (мышь, клавиатура). Обработка событий, генерируемых модулями программы. Сборка интерактивного приложения.

Практика: работа над проектом.

Тема 1.9 Защита проекта QT

Практика: защита проекта (настольное приложение).

Модуль 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 с 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.

Тема 2.12 Защита проекта PyGame

Практика: защита проекта (игра).

Модуль 3. Проекты WEB и API

Тема 3.1 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 3.2 WEB. Знакомство с API

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

Практика: решение задач.

Тема 3.3 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 3.4 WEB. Работа с командной строкой (скрипты, аргументы). Периодические задачи (модуль shedule)

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

Практика: решение задач.

Тема 3.5 WEB. Библиотека argparse. Задачи на создание скриптов с ее помощью

Теория: Парсинг аргументов командной строки средствами argparse.

Практика: решение задач.

Тема 3.6 WEB. Работа с протоколом HTTP

Теория: Протокол HTTP и клиент-серверная архитектура. Обмен данными с использованием данного протокола.

Практика: решение задач - самостоятельная работа.

Тема 3.7 WEB. Решение задач на API Яндекс-карт

Теория: Картографические сервисы Яндекса для сайтов, мобильных приложений и логистики. Работа с ними средствами Python применительно к WEB-проектам.

Практика: решение задач.

Тема 3.8 WEB. Введение во flask. Обработка html-форм

Теория: Фреймворк для создания WEB-приложений Flask, структура и базовая механика. Создание и обработка HTML-форм с его помощью.

Практика: решение задач - самостоятельная работа.

Тема 3.9 WEB. Шаблоны. flask--wtf

Теория: Расширения фреймвора Flask. Предназначение и использование расширения Flask-WTF.

Практика: решение задач.

Тема 3.10 WEB. Знакомство с flask-sqlalchemy

Теория: Предназначение и использование расширения (фреймворка Flask) Flask-SQLAlchemy.

Практика: решение задач.

Тема 3.11 WEB. Flask-sqlalchemy

Теория: Нюансы Flask-SQLAlchemy.

Практика: решение задач — контрольная работа.

Тема 3.12 WEB. REST-API. Понятие. Делаем простое REST-API

Теория: REST подход к организации взаимодействия компонентов распределенного приложения. REST-API.

Практика: решение задач.

Тема 3.13 WEB. REST-API. Flask-restful

Теория: Расширение (фреймворка Flask) Flask-RESTful для быстрого построения REST-API.

Практика: работа над проектом — создание своего web-сервиса.

Тема 3.14 WEB. Разворачиваем проект в облаке. Дорешка

Теория: Размещение web-проекта в сети интернет.

Практика: работа над проектом.

Тема 3.15 WEB. Алиса. Часть 1

Теория: История развития и основные технологии, лежащие в основе голосовых помощников. Яндекс. Диалоги. Алиса.

Практика: решение задач.

Тема 3.16 WEB. Алиса. Часть 2

Теория: Создание навыка для Алисы.

Практика: решение задач.

Тема 3.17 WEB. Защита проекта WebServer + API

Практика: защита своего web-проекта.

ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ

Планируемые результаты

Личностные результаты:

- формирование умения самостоятельной деятельности;
- формирование умения работать в команде;
- формирование коммуникативных навыков;
- формирование навыков анализа и самоанализа;
- формирование целеустремлённости и усидчивости в процессе творческой, исследовательской работы и учебной деятельности.

Метапредметные результаты:

- формирование умения ориентироваться в системе знаний;
- формирование умения выбирать наиболее эффективные способы решения задач на компьютере в зависимости от конкретных условий;
- формирование приёмов проектной деятельности, включая умения видеть проблему, формулировать тему и цель проекта, составлять план своей деятельности, осуществлять действия по реализации плана, результат деятельности соотносить с целью, классифицировать, наблюдать, проводить эксперименты, делать выводы и заключения, доказывать, защищать свои идеи, оценивать результаты своей работы;
- формирование умения распределять время;
- формирование умений успешной самопрезентации.

Предметные результаты:

- формирование понятий «алгоритм», «программа»;
- формирование понятий об основных конструкциях языка программирования Python, таких как оператор ветвления if, операторы цикла while, for, вспомогательные алгоритмы;
- формирование понятий о структурах данных языка программирования Python;
- формирование основных приёмов составления программ на языке программирования Python;
- формирование алгоритмического и логического стилей мышления.

**КОМПЛЕКС ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИХ УСЛОВИЙ.
КАЛЕНДАРНЫЙ УЧЕБНЫЙ ГРАФИК**

№	Год обучения	Дата начала занятий	Дата окончания занятий	Количество учебных недель	Количество учебных дней	Количество учебных часов	Режим занятий	Сроки проведения промежуточной/ итоговой аттестации
1	Первый год (стартовый уровень)	15.09.2022	31.05.2023	36	2	4	2 занятия по 2 часа в неделю с перерывом на перемену 10 минут	Промежуточная аттестация – декабрь 2022 Итоговая аттестация май 2023
2	Первый год (базовый уровень)	15.09.2022	31.05.2023	36	2	6	2 занятия по 3 часа в неделю с перерывом на перемену 10 минут	Промежуточная аттестация – декабрь 2022 Итоговая аттестация май 2023

УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ

Материально-техническое обеспечение

Условия реализации программы: учебный кабинет, оснащенный оборудованием (стандарт).

Перечень необходимого оборудования и расходных материалов (количество единиц оборудования и материалов указано из расчета на 12 человек)

- компьютеры и ноутбуки, на которых установлено соответствующее программное обеспечение: на каждого учащегося и преподавателя - 13 шт. или 1 шт. на малую группу (должны быть подключены к единой Wi-Fi сети с доступом в интернет);

- презентационное оборудование – 2 шт.;

- интерактивная панель – 1 шт.;

- наушники полноразмерные - 12 шт.

Кадровое обеспечение

Программа реализуется педагогом дополнительного образования, имеющим опыт работы в учреждении дополнительного образования технической направленности, образование – не ниже средне-профессионального, профильное или педагогическое.

ФОРМЫ КОНТРОЛЯ И ОЦЕНОЧНЫЕ МАТЕРИАЛЫ

Формы контроля и подведения итогов реализации программы

На занятиях используются: входной и текущий контроль, промежуточная и итоговая аттестация.

Входной контроль осуществляется через наблюдение за деятельностью учащихся, предполагает собеседование с учащимися, в ходе которого определяется наличие у них минимального необходимого уровня входных компетенций: уверенный пользователь ПК, проявляет интерес к прикладному программированию, конструированию, мехатронике, информационным технологиям в целом.

Текущий контроль осуществляется регулярно во время проведения каждого лабораторного занятия, заключается в ответе учащихся на контрольные вопросы, демонстрации полученных программ, фронтальных опросах, проводимых учителем. Отмечается активность участия учащихся в мероприятиях, степень самостоятельности при работе над практическими заданиями, самостоятельный поиск и разработка интересных тем для доклада (или мини- проекта).

Промежуточная и итоговая аттестация предполагает написание программы для решения одной из задач, контрольные работы и задания по изученным темам для определения уровня знаний учеников, разработку и реализацию проектов, представление и защиту индивидуальных и групповых проектов, публичное выступление с демонстрацией результатов работы, участие в профильных конкурсах и мероприятиях.

Оценочные материалы

Критерии оценивания контрольных заданий

Набранный балл	Оценка	Критерий
5	Высокий уровень	Получен полный и развёрнутый ответ на вопрос, приведены иллюстрирующие ответ примеры, получены ответы на дополнительные вопросы преподавателя
4	Средний уровень	Получен полный и развёрнутый ответ на вопрос, приведены иллюстрирующие ответ примеры, но не получены ответы на дополнительные вопросы преподавателя
3	Низкий уровень	Получен неполный ответ на вопрос, не приведены иллюстрирующие ответ примеры, получены неполные ответы на дополнительные вопросы преподавателя

Материалы для организации и проведения учебно-исследовательской и проектной деятельности школьников

Проект по программированию представляет собой проект, результатом которого является программа для решения той или иной задачи. Особенностью является то, что одна и та же задача в зависимости от уровня проработки может быть решена как начинающим, так и опытным программистом.

При выполнении проекта по программированию учащиеся имеют следующие возможности: выработать умение самостоятельно формулировать цели и задачи проекта, планировать свою деятельность, повысить уровень программирования на языке Python, получить умение представлять результаты своей деятельности.

Проект может разрабатываться индивидуально или группой учащихся. Если задача достаточно сложная, то проект может быть разбит на подзадачи, подпроекты. Каждую подзадачу будут выполнять различные группы участников проекта.

Например, одна группа занимается разработкой алгоритма, другая группа — непосредственно написанием и отладкой кода на языке Python, третья — подготовкой к презентации проекта.

Лист оценивания проекта

Критерий оценивания	1-я группа	2-я группа
Актуальность темы		
Соответствие содержания проекта заявленной теме		
Техническая сложность разработанной		

программы		
Оригинальность алгоритма		
Дизайн интерфейса		
Степень разработанности программы		
Применение программы для решения аналогичных задач		
Итоговое количество баллов		

МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ

Методические материалы

Особенности организации образовательного процесса: очно, очно-заочно, заочно, дистанционно, в условиях сетевого взаимодействия и др.;

– методы обучения (словесный, наглядный практический; объяснительно-иллюстративный, репродуктивный, частично-поисковый, исследовательский проблемный; игровой, дискуссионный, проектный и др.) и воспитания (убеждение, поощрение, упражнение, стимулирование, мотивация и др.);

– формы организации образовательного процесса: индивидуальная, индивидуально-групповая и групповая;

– формы организации учебного занятия: беседа, встреча с интересными людьми, выставка, защита проектов, игра, конкурс, конференция, лабораторное занятие, лекция, мастер-класс, «мозговой штурм», наблюдение, открытое занятие, практическое занятие, презентация, семинар, соревнование, фестиваль, чемпионат, эксперимент;

– педагогические технологии: технология индивидуализации обучения, технология группового обучения, технология программированного обучения, технология проблемного обучения, технология дистанционного обучения, технология проектной деятельности.

МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ

Раздел программы	Формы занятий	Приёмы и методы организации образовательного процесса	Дидактический материал	Техническое оснащение занятий	Формы подведения итогов

Вводное занятие	Лекция, презентация, игра, инструктаж	Словесно-наглядный, проблемное изложение, поиск ответов на поставленные вопросы	Презентация, инструкции, подборка профильных мероприятий	Оборудование ИТ-Куба	Собеседование
Введение в программирование	Лекция, групповая, индивидуальная, практическая работа, проекты	Словесно-наглядный, поисковый, практический, проблемный	Подборка презентационных материалов и прикладных задач	Оборудование ИТ-Куба	Беседа, выполнение мини-проекта, практические работы
Базовые конструкции в Python	Лекция, групповая, индивидуальная, практическая работа, решение задач, проекты	Словесно-наглядный, поисковый, практический, проблемный	Подборка презентационных материалов и прикладных задач	Оборудование ИТ-Куба	Практические задания, беседа, выполнение мини-проекта, контрольная работа
Решение прикладных задач в Python	Лекция, групповая, индивидуальная, практическая работа, решение задач, проекты	Словесно-наглядный, поисковый, практический, проблемный	Подборка презентационных материалов и прикладных задач	Оборудование ИТ-Куба	Практические задания, беседа, выполнение мини-проекта, контрольная работа

СПИСОК ЛИТЕРАТУРЫ ДЛЯ ПЕДАГОГА

1. Банкрашков, А.В. Программирование для детей на языке Python / А.В. Банкрашков. - М.: АСТ, 2018. - 288 с.
2. Буйлова Л.Н. Технология разработки и оценки качества дополнительных общеобразовательных общеразвивающих программ: новое время – новые подходы: Методическое пособие. – М., 2015.
3. Буйначев, С. К. Основы программирования на языке Python [Электронный ресурс]: учебное пособие / С. К. Буйначев, Боклаг Н. Ю. – Электрон. текстовые дан. – Издательство Уральского университета, 2014.
4. Васильев, Александр Николаевич Python на примерах. Практический курс по программированию. Руководство / Васильев Александр Николаевич. - М.: Наука и техника, 2017. - 752 с.
5. Вордерман, К. Программирование на Python. Иллюстрированное руководство для детей / К. Вордерман, К. Стили, К. Квигли. - М.: Манн, Иванов и Фербер, 2017. - 346 с.
6. Гуриков, С.Р. Основы алгоритмизации и программирования на Python. Учебное пособие. Гриф МО РФ / С.Р. Гуриков. - М.: Инфра-М, Форум, 2018. - 707 с.
7. Златопольский, Д.М. Основы программирования на языке Python. – М.: ДМК Пресс, 2017. – 284 с.
8. Многообразие образовательных практик учреждений дополнительного образования детей: Материалы научно-практической конференции «Развитие системы дополнительного образования детей» в 2-х частях. – М., 2012.
9. Педагогика дополнительного образования. Психолого-педагогическое сопровождение детей. Учебник для академического бакалавриата / ред. Байбородова Л. В. М.: Юрайт, 2019. 364 с.
10. Свейгарт, Эл. Автоматизация рутинных задач с помощью Python: практическое руководство для начинающих. Пер. с англ. — М.: Вильямс, 2016. — 592 с.
11. Федоров, Д. Ю. Программирование на языке высокого уровня Python: учебное пособие для прикладного бакалавриата / Д. Ю. Федоров. – 2-е изд., перераб. и доп. – Москва: Издательство Юрайт, 2019. – 161 с. – (Бакалавр. Прикладной курс).
12. Шмпандар, Алекс Дж. Искусственный интеллект в компьютерных играх: как обучить виртуальные персонажи реагировать на внешние воздействия.: Пер. с англ. — М.: ООО "ИД. Вильямс", 2007. — 768 с.: ил.