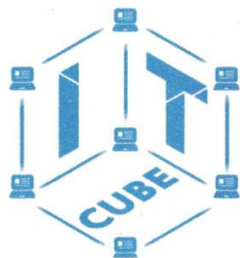


**МИНИСТЕРСТВО ОБРАЗОВАНИЯ ПЕНЗЕНСКОЙ ОБЛАСТИ
ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ПРОФЕССИОНАЛЬНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ПЕНЗЕНСКОЙ ОБЛАСТИ
«КУЗНЕЦКИЙ КОЛЛЕДЖ ЭЛЕКТРОННЫХ ТЕХНОЛОГИЙ»**

Центр цифрового образования детей «IT-куб»



**ЦЕНТР ЦИФРОВОГО
ОБРАЗОВАНИЯ ДЕТЕЙ "IT-КУБ"
г.Кузнецк, Пензенская область**



УТВЕРЖДАЮ
Директор ГАПОУ ПО «ККЭТ»
Т.А. Хархун
Приказ № 1 от 29 августа 2023 г.

**Дополнительная общеразвивающая программа
«Программирование на языке Python»
(144 учебных часа)**

Составитель: Г.А. Иванова.

2023 г.

СОДЕРЖАНИЕ

1	Паспорт программы	3
2	Содержание программы	6
3	Условие реализации программ	10

1 ПАСПОРТ ПРОГРАММЫ

Дополнительная общеобразовательная общеразвивающая программа «Учимся программировать на языке Python» относится к программам стартового уровня, имеет техническую направленность.

Программа разработана на основе:

- Федерального Закона РФ «Об образовании в Российской Федерации» от 29.12.2012 года № 273-ФЗ,

- Концепции развития дополнительного образования, утвержденной распоряжением Правительства Российской Федерации от 4 сентября 2014 г. № 1726-р,

- Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам, утвержденным приказом Министерства просвещения Российской Федерации № 196 от 09.11.2018г,

- Постановления Главного государственного санитарного врача РФ от 04.07.2014 № 41 «Об утверждении СанПиН 2.4.4.3172-14 «Санитарноэпидемиологические требования к устройству, содержанию и организации режима работы образовательных организаций дополнительного образования детей».

Актуальность дополнительной общеобразовательной общеразвивающей программы соотносится с тенденциями развития дополнительного образования и согласно Концепции развития дополнительного образования способствует формированию и развитию имеющихся навыков операционного и логического стиля мышления, исследовательских и творческих качеств. Актуальность программы отвечает современным потребностям детей и родителей в получении необходимых теоретических знаний и навыков в сфере программирования для личностного развития и успешной адаптации к требованиям жизни в современном информационном обществе.

Методы, применяемые в процессе обучения, такие как проблемное обучение, проектная деятельность, способствуют формированию мотивации учащихся к углубленному изучению программирования, как одной из компьютерных наук. У детей формируется познавательный интерес, самостоятельность мышления, стремление к самопознанию.

Отличительная особенность программы состоит в том, что она позволяет привлечь детей среднего школьного возраста к изучению программирования при помощи языка Python, так как он обладает следующими достоинствами:

- Python – это текстовый язык программирования. Он универсален, пригоден для создания самых разных программ, от текстовых процессоров до веб-браузеров;

- Python – простой и удобный язык. По сравнению со многими другими языками читать и составлять программы на Python совсем не сложно;

- В Python есть библиотеки готовых процедур для использования в своих программах. Это позволяет создавать сложные программы быстро;

- Python используется как язык программирования крупными корпорациями, такими как Google.

Адресат программы: дети от 11 до 16 лет. Состав группы 10-12 человек.

Дети в этом возрасте активны, им все интересно. Они гибки в своем поведении, восприимчивы и открыты для сотрудничества, уже достаточно зрелы интеллектуально. Именно в подростковом возрасте начинает устанавливаться определенный круг интересов, который постепенно приобретает известную устойчивость.

Начинает развиваться мышление на уровне формальных операций. Конкретные предметы и события больше не являются обязательными для мышления. Теперь подростки способны: решать абстрактные задачи, делать логические выводы из своих рассуждений, строить прогнозы и планы.

Качественно улучшается внимание: увеличивается объем, устойчивость, возможность распределения и переключения. Запоминание и воспроизведение приобретает все больше смысловой характер. Увеличивается объем памяти, избирательность и точность запоминания.

Форма обучения: очная.

Объем и срок освоения программы: срок реализации программы - 1 год. Общая продолжительность образовательного процесса составляет 144 часа.

Режим: Занятия проводятся 2 раза в неделю по 2 часа.

Цель программы: способствовать развитию алгоритмического и логического мышления посредством языка программирования Python.

Задачи программы:

— *Обучающие:*

- сформировать представление об основных элементах программирования;
- познакомить с синтаксисом языка программирования Python;
- сформировать навыки работы в интегрированной среде разработки на языке Python;
- способствовать приобретению навыков разработки эффективных алгоритмов и программ на основе изучения языка программирования Python.

— *Развивающие:*

- совершенствовать аналитические навыки;
- способствовать формированию алгоритмического и логического мышления;
- совершенствовать навык поиска информации в сети Интернет, анализа выбранной информации на соответствие запросу, использования информации при решении задач;
- развивать умение планировать свои действия с учётом фактора времени.

— *Воспитательные:*

- воспитывать в детях усидчивость, аккуратность, умение доводить начатое дело до конца;
- формировать коммуникативные навыки.

Планируемые результаты освоения программы:

Предметные результаты:

- иметь представление об основных классических алгоритмах и способах их реализации;
- знать синтаксис языка программирования Python;
- иметь представление о величине и ее характеристиках;
- знать простые и сложные структуры данных, а также конструкции для работы с ними;
- иметь представление о некоторых модулях (turtle, random, tkinter и др.);
- уметь анализировать как свой, так и чужой код;
- иметь представление о функциях, владеть навыками работы с ними.

Метапредметные результаты:

- учащиеся способны осуществлять рефлексивную деятельность, оценивать собственные результаты, корректировать дальнейшую деятельность по программированию;
- сформировано алгоритмическое и логическое мышление;
- обладают навыками поиска информации в сети Интернет, анализа выбранной информации на соответствие запросу, использования информации при решении задач;
- сформировано умение планировать деятельность с учетом фактора времени.

Личностные результаты:

- учащиеся проявляют усидчивость, аккуратность, умеют доводить начатое дело до конца;
- сформировано умение работать в коллективе.

Формы и виды учебной деятельности

Формы организации учебных занятий:

- фронтальная – подача материала всей учебной группе учащихся;
- индивидуальная – самостоятельная работа учащихся с оказанием педагогом помощи при возникновении затруднения;
- групповая – предоставление учащимся возможности самостоятельно построить свою деятельность, ощутить помощь со стороны друг друга, учесть возможности каждого на конкретном этапе деятельности.

Формы проведения занятий:

- вводное занятие – педагог знакомит учащихся с техникой безопасности, особенностями организации деятельности и предлагаемым планом работы на текущий год;

- ознакомительное занятие – педагог знакомит учащихся с новыми методами работы в зависимости от темы занятия;

- тематическое занятие – на котором детям предлагается работать над моделированием по определенной теме. Занятие содействует развитию творческого воображения учащихся;

- занятие-проект – на таком занятии учащиеся получают полную свободу в выборе направления работы, не ограниченного определенной тематикой. Учащиеся, участвующие в работе по выполнению предложенного задания, рассказывают о выполненной работе, о ходе выполнения задания, о назначении выполненного проекта;

- конкурсное игровое занятие – строится в виде соревнования для повышения активности учащихся и их коммуникации между собой;

- комбинированное занятие – проводится для решения нескольких учебных задач;

- итоговое занятие – служит подведению итогов работы за учебный год, может проходить в виде мини-выставок, просмотров творческих работ и презентаций.

Формы контроля результатов освоения программы.

Формы аттестации: наблюдение, опрос, защита проекта, самостоятельная работа, конкурс работ.

Итоговый контроль проводится в конце года с целью определения степени достижения результатов обучения и получения сведений для совершенствования программы и методов обучения.

2 УЧЕБНЫЙ ПЛАН

№ п/п	Название разделов, тем	Количество часов			Форма аттестации/контроля
		Всего	Теория	Практика	
1	Введение в язык программирования Python	28	12	16	
1.1	Устройство языка Python. Среда разработки IDLE. Сохранение Python-программ	2	1	1	Наблюдение
1.2	Вывод данных на языке Python. Ввод и отладка программ в среде	2	1	1	Наблюдение
1.3	Переменные	4	2	2	Опрос
1.4	Выражения. Типы данных	4	2	2	Опрос
1.5	Строки и списки	4	2	2	Самостоятельная работа
1.6	Кортежи и словари	4	2	2	Самостоятельная работа
1.7	Рисование с помощью «Черепашей графики»	8	2	6	Конкурс работ
2	Основы языка Python	64	18	46	
2.1	Условный оператор	4	2	2	Опрос
2.2	Объединение условий	4	1	3	Самостоятельная работа
2.3	Использование цикла for	4	1	3	Опрос
2.4	Использование цикла while	4	1	3	Опрос
2.5	Решение задач на циклические алгоритмы	4	1	3	Самостоятельная работа
2.6	Применение функций	4	1	3	Самостоятельная работа
2.7	Использование встроенных функций	4	1	3	Опрос
2.8	Применение модулей	4	1	3	Самостоятельная работа
2.9	Классы и объекты	4	1	3	Опрос
2.10	События	4	1	3	Конкурс работ
2.11	Работа с файлами	4	1	3	Самостоятельная работа
2.12	«Черепашья графика»	8	2	6	Конкурс работ
2.13	Графика с модулем tkinter	12	4	8	Защита проекта
3	Игры на Python	38	12	26	
3.1	Игра «Прыг-скок!»	12	4	8	Защита проекта
3.2	Игра «Охотник за пузырями»	12	4	8	Защита проекта
3.3	Игра «Человечек спешит к выходу»	14	4	10	Защита проекта
4	Итоговый проект	14	2	12	Защита проекта
	Итого:	144	44	100	

Содержание программы

Раздел 1. Введение в язык программирования Python

Тема 1.1. Устройство языка Python. Среда разработки IDLE. Сохранение Python-программ

Теория. Техника безопасности на занятии. Понятие «алгоритм», «исполнитель», «язык программирования», «программа», «интерпретатор». История языка программирования Python и его возможности. Виды окон в IDLE: окно программы и окно консоли. Сравнение этих окон и их возможностей.

Практика. Сохранение и запуск python-программ в среде разработки IDLE.

Тема 1.2. Вывод данных на языке Python. Ввод и отладка программ в среде

Теория. Правила синтаксиса Python: правило начала, правило порядка, правило регистра. Понятие функции. Функция print().

Практика. Проект «Символьная графика». Создание определённого рисунка с помощью символов. Отработка функции print().

Тема 1.3. Переменные

Теория. Понятие «переменная». Правила именования переменных в языке Python. Оператор присваивания. Функция input().

Практика. Проект «Аватар». В данном проекте обрабатывается функция input(), с помощью которой становится возможным ввести свои данные в программу и отобразить их.

Тема 1.4. Выражения. Типы данных

Теория. Арифметические операции с помощью математических операторов +, -, *, /. Порядок выполнения операций. Понятие «выражение», «типы данных». Функции int() и str().

Практика. Проект «Сумматор». При написании данной программы обрабатываются математические операторы и функции int() и str().

Тема 1.5. Строки и списки

Теория. Понятие «строка». Создание строк. Переменные внутри строк. Операции со строками. Понятие «список». Создание списков. Добавление/удаление элементов в/из список/списка. Операции со списками.

Практика. Решение задач на обработку операций со строками и списками.

Тема 1.6. Кортежи и словари

Теория. Понятие «кортеж». Создание кортежа. Операции с кортежем. Понятие «словарь». Создание словаря.

Практика. Проект «Любимые вещи»: создание списка любимых развлечений и любимых лакомств.

Тема 1.7. Рисование с помощью «Черепашьей графики»

Теория. Понятие «модуль». Модуль turtle. Импортирование модуля. Создание холста. Перемещение черепашки.

Практика. Проект «Я рисую»: рисование изображения, состоящего из линий с помощью команд модуля на холсте.

Раздел 2. Основы языка Python

Тема 2.1. Условный оператор

Теория. Понятие «условный оператор», «вложенные команды», «оператор сравнения». Конструкция if и её синтаксис. Операторы сравнения: <, >, >=, <=, !=, ==. Структура программы. Конструкция if-else. Команды if и elif.

Практика. Решение задач на обработку условного оператора и операторов сравнения.

Тема 2.2. Объединение условий

Теория. Логические операторы: and, or, not. Порядок выполнения операций. Переменные без значения – None.

Практика. Проект «Калькулятор»: создание приложения по определенным условиям.

Тема 2.3. Использование цикла for

Теория. Понятие «цикл», «цикл со счётчиком». Конструкция for и её синтаксис.

Практика. Проект «Таблица умножения»: создание приложения по определенным условиям.

Тема 2.4. Использование цикла while

Теория. Понятие «цикл с предусловием». Конструкция while и её синтаксис. Зацикливание и выход из цикла с помощью команды break.

Практика. Проект «Бомба взорвалась!». Написание программы по определенным условиям.

Тема 2.5. Решение задач на циклические алгоритмы

Теория. Виды циклов и их конструкции.

Практика. Решение задач на применение циклов for и while.

Тема 2.6. Применение функций

Теория. Понятие «функция», «параметр функции», «значение функции». Строение функции: имя, аргумент, тело. Создание и вызов функции. Переменные и область видимости.

Практика. Решение задач на отработку понятия «функция», её строение и синтаксис.

Тема 2.7. Использование встроенных функций

Теория. Функции: abs, bool, dir, eval, exec, float, int, len, max, min, range, sum.

Практика. Решение задач на отработку функций.

Тема 2.8. Применение модулей

Теория. Понятие «модуль». Импортирование модуля в программу. Полезные модули: random, time, pickle.

Практика. Решение задач на применение модулей.

Тема 2.9. Классы и объекты

Теория. Понятие «объект». Концепция объектов. Понятие «класс». Использование классов в Python. Инициализация объектов.

Практика. Решение задач на отработку умения создавать классы и объекты.

Тема 2.10. События

Теория. Понятие «событие», «подписка на событие», «обработка события».

Практика. Проект «Поймай черепашку». Пишется приложение-игра, в котором создается холст и на нём прорисовывается черепашка, при нажатии на которую происходит ее перемещение в новое место на холсте.

Тема 2.11. Работа с файлами

Теория. Понятие «файл». Классификация файлов в зависимости от видов информации. Создание текстового файла. Открытие файла в Python. Запись в файл.

Практика. Проект «Блокнот»: создание простейшего текстового редактора в виде приложения.

Тема 2.12. «Черепашня графика»

Теория. Рисование фигур с использованием управляющих конструкций.

Практика. Проект «Фантастический мир»: рисование фигур, параметры которых задаются пользователем при запуске программы.

Тема 2.13. Графика с модулем tkinter

Теория. Модуль tkinter и его возможности.

Практика. Проект «Анимация»: отрисовывание объекта на холсте и программирование анимации.

Раздел 3. Игры на Python

Тема 3.1. Игра «Прыг-скок!»

Теория. Обсуждение сюжета игры: объекты и их роль. Создание игрового холста. Создание главного цикла игры. Отрисовка мяча. Программирование мяча: перемещение, отскоки от границ игрового холста, первоначальное направление движения мяча. Отрисовка ракетки. Программирование ракетки: управление, установления границ. Обработка события столкновения мяча и ракетки. Обработка события «проигрыш». Доработка игры: задержка перед началом игры; экран «Конец игры»; ускорение мяча; счет в игре.

Практика. Приложение-игра «Прыг-скок!»: неуправляемый пользователем объект «мяч» и управляемые пользователем объект «ракетка».

Тема 3.2. Игра «Охотник за пузырями»

Теория. Обсуждение сюжета игры: объекты и их роль. Создание игрового холста. Создание главного цикла игры. Отрисовка подложки. Программирование подложки: перемещение, поворот, торможение при касании границы холста. Отрисовка пузырей разного размера и их различное положение на холсте с помощью модуля `gandom`. Программирование пузырей: движение. Обработка события касания объектов «подводная лодка» и «пузырь». Доработка игры: задержка перед началом игры; счет в игре и его отображение; создание лимита времени; экран «Конец игры».

Практика. Приложение-игра «Охотник за пузырями»: управляемый пользователем объект «подводная лодка» и неуправляемый объект «пузырь», создаваемые в случайном количестве. Задача пользователя: управлять лодкой и протыкать пузыри, зарабатывая при этом баллы.

Тема 3.3. Игра «Человечек спешит к выходу»

Теория. Обсуждение сюжета игры: объекты и их роль. Создание игрового холста. Создание главного цикла игры. Отрисовка человечка (три изображения для анимации: влево, вправо, на месте). Программирование человечка: анимация, управление. Отрисовка платформ трёх видов: маленькой длины, средней и большой. Размещение платформ на холсте игры. Отрисовка двери: закрытая и открытая. Обработка событий столкновений человечка с объектами: «дверь» и «платформа». Доработка игры: экран «Вы победили!»; экран «Конец игры»; создание резерва жизней для игрока; создание второго уровня с движущимися платформами; создание третьего уровня с противниками на движущихся платформах.

Практика. Приложение-игра «Человечек спешит к выходу». Сюжет игры: Человечек очутился в плену у злодея, и ваша задача – помочь человечку спастись, добравшись до выхода на верхнем этаже. Человечек выглядит как фигурка, которая может двигаться вправо, влево, а также прыгать. На этажах расположены платформы, на которые человечку предстоит запрыгивать. Цель игры – добраться до двери выхода.

Раздел 4. Итоговый проект.

Практика. Разработка собственного приложения-игры с сюжетом. Условия игры: обязательно должен присутствовать управляемый пользователем персонаж, должна быть прорисована и запрограммирована игровая обстановка.

3 УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ

Комплекс условий реализации программы:

Аппаратное и техническое обеспечение:

а) Рабочее место учащегося

- компьютер или ноутбук с выходом в сеть Интернет;

- установленный на каждый ПК интерпретатор языка программирования Python 3 с установленным IDE PyCharm.

б) Рабочее место наставника

- компьютер или ноутбук с выходом в сеть Интернет;

- установленный на ПК интерпретатор языка программирования Python 3 с установленным IDE PyCharm;

- технические средства обучения (ТСО) (мультимедийное устройство).

Методы обучения:

- объяснительно-иллюстративный (беседы, объяснения, дискуссии);

- репродуктивный (деятельность обучаемых носит алгоритмический характер, выполняется по инструкциям, предписаниям, правилам в аналогичных, сходных с показанным образцом ситуациях);

- метод проблемного изложения;

- эвристический (метод обучения заключается в организации активного поиска решения выдвинутых в обучении (или самостоятельно сформулированных) познавательных задач в ходе подготовки и реализации творческих проектов);

- исследовательский.

Педагогические технологии: проектная технология, здоровьесберегающие технологии, технология проблемного обучения. Проектная технология дает возможность самостоятельно конструировать свои знания, ориентироваться в информационном пространстве, развивается критическое и творческое мышление. Если проектная технология является спланированной и постоянной составляющей частью образовательного процесса, то будут созданы условия для формирования и развития внутренней мотивации учащихся к более качественному овладению знаниями, повышения мыслительной активности и приобретения навыков логического мышления. Проблемное обучение — это тип развивающего обучения, содержание которого представлено системой проблемных задач различного уровня сложности, в процессе решения которых учащиеся овладевают новыми знаниями и способами действия, а через это происходит формирование творческих способностей: продуктивного мышления, воображения, познавательной мотивации, интеллектуальных эмоций.

Кадровые условия реализации программы

Программу реализует педагог дополнительного образования, который обладает навыками программирования на языке Python, владеет проектным мышлением и умеет организовать групповую проектную деятельность учащихся и руководить ею.

Перечень рекомендуемых источников

1. Бриггс, Джейсон, Python для детей. Самоучитель по программированию / Джейсон Бриггс ; пер. с англ. Станислава Ломакина ; [науч. ред. Д. Абрамова]. — М. : Манн, Иванов и Фербер, 2017. — 320 с.

2. Великович Л., Цветкова М. Программирование для начинающих. – М.: Бином, 2008.

3. Программирование для детей. Перевод с английского Станислава Ломакина, Москва, «Манн, Иванов и Фербер», 2015 г.

4. Сайт pythonworld.ru — «Python 3 для начинающих».

5. Сайт pythontutor.ru — «Питонтьютор».

Приложение

Рефлексия индивидуальной образовательной деятельности:

1. Полученные результаты _____ соответствуют (указывается в какой степени) поставленным целям
2. Мне удалось _____
3. Я создал (достиг, участвовал и т.п.) _____
4. Я научился _____
5. Самооценка результатов на основании критериев

№	Полученные образовательные продукты	Критерии оценки		
		1 критерий. Новизна и актуальность	2 критерий. Практическая значимость	3 критерий. Культура оформления материалов

В _____ дальнейшем _____ мне _____ бы _____ хотелось _____ изучить _____ (научиться, освоить)
