

**Муниципальное бюджетное общеобразовательное учреждение
«Каспийская гимназия №11»
«Центр цифрового образования детей «it-Куб»**

**Дополнительная общеобразовательная
(общеразвивающая) программа**

«Программирование на языке Python»

Категория обучающихся: обучающиеся 6-7, 8-9 классов средней школы

Срок освоения программы: 72 часа

Авторы программы:

Педагог дополнительного образования
Центр цифрового образования детей IT-CUBE. ВОЛОГДА
Александрова Юлианна Ромэовна

Утверждено на Внеочередном Педсовете
Протокол № 8

Директор МБОУ
«Каспийская гимназия №11»



/ Тагирова Ж.У

Каспийск, 2020 год

Раздел №1. Комплекс основных характеристик программы

Пояснительная записка

Дополнительная общеобразовательная программа «Программирование на языке Python» (стартовый уровень) является общеразвивающей программой *технической* направленности.

Актуальность программы. 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, не только на пользовательском уровне, но и на уровне начинающего программиста. В школьном курсе информатики программирование нередко представлено лишь на элементарном уровне. Следствием этого является формальное восприятие обучающимися основ современного программирования и неумение применять полученные знания на практике.

Педагогическая целесообразность программы заключается в том, что дети приобретут практические навыки, которые станут основой для дальнейшего изучения основ программирования. Методы, применяемые в процессе обучения, такие как проблемное обучение, проектная деятельность, способствуют формированию мотивации обучающихся к углубленному изучению программирования, как одной из компьютерных наук. У детей формируется познавательный интерес, самостоятельность мышления, стремление к самопознанию.

Осваивая данную программу, обучающиеся будут овладевать навыками, которые будут востребованы в ближайшие десятилетия в специальностях, многие из которых включены в Атлас профессий будущего. Практически для каждой перспективной профессии будут полезны знания и навыки, получаемые в процессе обучения по программе.

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 12 лет. Наполняемость групп: 5 – 12 человек.

Объем и срок освоения программы. Дополнительная общеобразовательная общеразвивающая программа «Программирование на языке Python» (стартовый уровень) рассчитана на 1 год обучения. Дети,

освоившие стартовый уровень, могут продолжить обучение на базовом уровне программы «Программирование на языке Python» (базовый уровень) без входного контроля.

Режим занятий: 72 академических часа в год, 2 академических часа в неделю. Программой предусмотрена возможность обучения детей по индивидуальному образовательному маршруту (приложения 1, 2).

В каникулярное время занятия проводятся в соответствии с календарным учебным графиком, допускается изменение форм занятий, проведение воспитательных мероприятий.

Цель программы: развитие алгоритмического и логического мышления посредством языка программирования «Python».

Задачи:

Обучающие:

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

Развивающие:

- совершенствовать аналитические навыки;
- формировать навык алгоритмического и логического мышления;
- совершенствовать навык поиска информации в сети Интернет, анализа выбранной информации на соответствие запросу, использование 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

К концу реализации программы обучающиеся будут *знать*:

- основные классические алгоритмы и способы их реализации;
- синтаксис языка программирования Python;
- простые и сложные структуры данных, а также конструкции для работы с ними;
- некоторые модули (turtle, random, tkinter и др.).
- основные элементы программирования: ввод (вывод) информации в память (из памяти), данные, операции с данными, условное выполнение, циклы, подпрограммы

К концу реализации программы обучающиеся будут *уметь*:

- разбивать решение задачи на подзадачи, то есть составлять алгоритм;
- объяснять и использовать на практике как простые, так и сложные структуры данных и конструкции для работы с ними;
- искать и обрабатывать ошибки в коде;
- писать грамотный, красивый код;
- анализировать как свой, так и чужой код;
- импортировать модули в программу;
- работать с информацией: находить, оценивать и использовать информацию из различных источников, необходимую для решения профессиональных задач (в том числе на основе системного подхода);
- грамотно строить коммуникацию, исходя из целей и ситуации;
- работать с вычислительной техникой.

К концу реализации программы обучающиеся будут *иметь представление*:

- о возможностях языка программирования Python;
- о структурном программировании.

:

Учебно-тематический план
72 академических часа в год

№ п/п	Название разделов, тем	Количество часов		
		Всего	Теория	Практика
1	Введение в язык программирования Python	16	8	8
1.1	Устройство языка Python. Среда разработки IDLE. Сохранение Python-программ	2	1	1
1.2	Вывод данных на языке Python. Ввод и отладка программ в среде	2	1	1
1.3	Переменные	2	1	1
1.4	Выражения. Типы данных	2	1	1
1.5	Строки и списки	2	1	1
1.6	Кортежи и словари	2	1	1
1.7	Рисование с помощью «Черепашьей графики»	4	2	2
2	Основы языка Python	32	16	16
2.1	Условный оператор	2	1	1
2.2	Объединение условий	2	1	1
2.3	Использование цикла for	2	1	1
2.4	Использование цикла while	2	1	1
2.5	Решение задач на циклические алгоритмы	2	1	1
2.6	Применение функций	2	1	1
2.7	Использование встроенных функций	2	1	1
2.8	Применение модулей	2	1	1
2.9	Классы и объекты	2	1	1
2.10	События	2	1	1
2.11	Работа с файлами	2	1	1
2.12	«Черепашья графика»	4	2	2
2.13	Графика с модулем tkinter	6	3	3
3	Игры на Python	20	4	16
3.1	Игра «Прыг-скок!»	6	1	5
3.2	Игра «Охотник за пузырями»	6	1	5
3.3	Игра «Человечек спешит к выходу»	8	2	6
4	Аттестация	4	-	4
	Итого:	72	28	44

Содержание

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 отрабатывается 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 отрабатываются 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 отработку 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 «Черепаший 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 отработку 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». Импорт 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 подлодки. Программирование подлодки: перемещение, поворот, торможение при касании границы холста. Отрисовка пузырей разного размера и их различное положение на холсте с помощью модуля random. 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 рандомном 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. Аттестация

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

Раздел №2. Комплекс организационно-педагогических условий

Условия реализации программы

Материально-техническое обеспечение программы

- столы, стулья (по росту и количеству детей);
- демонстрационная магнитная доска;
- технические средства обучения (ТСО) (мультимедийное устройство);
- ноутбуки (10 ученических + 1 учительский) с выходом в сеть Интернет и с установленным ПО;

Формы подведения итогов реализации программы

Педагогический мониторинг позволяет систематически отслеживать результативность реализации программы. Мониторинг включает в себя традиционные формы контроля: промежуточную и итоговую аттестацию результатов обучения детей.

Промежуточная аттестация проводится в конце первого полугодия.

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

Аттестация обучающихся может проходить на итоговом занятии в форме презентации своего проекта.

Методические материалы

Особенности организации образовательного процесса: очная форма.

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.

Формы организации образовательного процесса:

- *фронтальная* – подача материала всей учебной группе обучающихся;
- *индивидуальная* – самостоятельная работа обучающихся 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 организации учебного занятия:

- *вводное занятие* – педагог знакомит обучающихся с техникой безопасности, особенностями организации деятельности и предлагаемым планом работы на текущий год;
- *ознакомительное занятие* – педагог знакомит обучающихся 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 обучающихся;
- *занятие-проект* – на таком занятии обучающиеся получают полную свободу в выборе направления работы, не ограниченного определенной тематикой. Обучающиеся, 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 обучающихся 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.

Алгоритм учебного занятия:

1. Мотивация обучающихся.
2. Актуализация имеющихся знаний.
3. Теоретический блок нового материала.
4. Закрепление материала.
5. Перерыв.
6. Теоретический блок нового материала.
7. Закрепление материала.
8. Рефлексия.

Список литературы

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.

Приложение 1.

Карта индивидуального сопровождения ребёнка

Наименование объединения

1. Сведения о ребёнке

- Ф.И.О. _____
- Краткая характеристика ребёнка, сильные стороны, интересы ребёнка _____

3. Цель сопровождения _____

4. Запрос родителей _____

5. Возможные риски _____

6. Мероприятия (примерный перечень мероприятий с указанием конкретных сроков и распределением обязанностей)

Наименование мероприятия	Сроки	Ответственные	Промежуточные результаты	Перспективные задачи развития/рекомендации
Психолого-педагогическая диагностика	1 раз в год	Педагог - психолог		
Психолого-педагогическое просвещение родителей: Информирование о результатах диагностических исследований 1. Совместные творческие работы родителей и ребенка 2. Открытое занятие для родителей		Педагог-психолог Педагог		
Индивидуальная работа по ИОМ	1 раз в неделю	Педагог		
Оформление выставки творческих работ	Раз в квартал	Педагог		
Организация участия в конкурсах различного уровня	Раз в квартал	Педагог		

Маршрутный лист обучающегося

ФИО _____

Объединение _____

Руководитель _____

Ожидаемые результаты: _____

Критерии оценки ожидаемых результатов: _____

Индивидуальный маршрут:

№	Тема	Кол-во часов	Сроки	Методы изучения темы	Результат	Подпись руководителя

Рефлексия индивидуальной образовательной деятельности:

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

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

В дальнейшем мне бы хотелось изучить (научиться, освоить)
