

**РАБОЧАЯ ПРОГРАММА**

учебного курса «Программирование на языке Python»

10 - 11 класс

## ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Направленность программы — техническая. Она заключается в развитии технического и творческого мышления у обучающихся среднего школьного возраста через разработку алгоритмов в цифровой среде, взаимодействие с приложениями, сервисами и инструментами вне зависимости от платформы или интерфейса, формирование навыка определения различных видов ошибок (логическая, синтаксическая, ошибка среды и ошибка взаимодействия) и принципах их исправления в процессе разработки с помощью процедур отладки. Также развитие у детей умения составлять план деятельности, стремления к познавательным активностям, исследовательских, прикладных способностей, формирование навыков сотрудничества и коллективной работы в цифровой среде.

Развитие навыков программирования является уникальным способом формирования интереса учащихся к техническим направлениям деятельности, а также совершенствования алгоритмического, креативного мышления и навыков проектной деятельности.

Python — это язык программирования общего назначения, распространяемый с открытыми исходными текстами. Он оптимизирован для создания качественного программного обеспечения. Язык Python используется сотнями тысяч разработчиков по всему миру в таких областях, как создание веб-сценариев, системное программирование, создание пользовательских интерфейсов, настройка программных продуктов под пользователя, численное программирование и в других. ...

Программа «Программирование на Python» представляет собой углубленный курс по программированию, дающий представление о понятиях структурного программирования (данных, операциях, переменных, ветвлениях в программе, циклах и функциях, массивах, файлах). Программа направлена на развитие логического и пространственного мышления обучающегося, способствует раскрытию творческого потенциала личности, приобретению практических умений и навыков в области компьютерных технологий, способствует интеллектуальному развитию ребенка.

Главной целью является приобретение обучающимся теоретических знаний и практических навыков и компетенций, достаточных для дальнейшего успешного обучения в системе непрерывного образования: школа — допобразование — ВУЗ — предприятие. Сущностью программы можно определить как развитие мышления и формирование мировоззрения в

условиях преподавания интегрированного курса информатики, математики, программирования обучающихся среднего и старшего звена. Изучение всех этих предметов должно быть не целью, а средством познания мира, давать возможность обучающимся проникать в сущность изучаемых проблем. В теории интеграции в данном случае вкладывается понятие взаимосвязи, взаимопроникновения математики и информатики, что предполагает качественное изменение в параметрах нового объекта. В школьном курсе информатики вопросы программирования рассматриваются лишь в ознакомительном плане и на это выделяется недостаточное количество часов, как следствие – формальное восприятие обучающимися основ современного программирования. Образовательная программа «Программирование на Python» направлена на устранение данного пробела.

Отличительные особенности программы и новизна. Отличительной особенностью курса является его направленность на формирование у учащихся навыков поиска собственного решения поставленной практической задачи, представленную в виде адаптированного кейса из реальной жизни, составления алгоритма решения и его реализации с помощью средств программирования. Для этого, учащиеся проходят через следующие этапы: определяют и составляют последовательность действий, ведущую к цели, и записывают ее с помощью формального языка. Полученные знания учащиеся применяют при создании собственных проектов.

Программа рассчитана на 2 года обучения по 1 часу в неделю, итого 68 часов.

Цель курса – способствовать формированию алгоритмического и критического мышления, навыков проектной деятельности через использование языка программирования Python как инструмента для реализации идей в области своих интересов.

## **СОДЕРЖАНИЕ УЧЕБНОГО КУРСА «ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ PYTHON»**

### **Раздел 1. Введение в Python**

#### **1.1. Инструкции и структура программы.**

Техника безопасности. История создания. Области применения и перспективы. Инструкции и структура программы. Основы Алгоритмизации. Достоинства языка.

#### **1.2. Установка Python. Ввод и вывод данных.**

Установка Python. Доступ к документации. Основы ввода и вывода данных. Первая программа на языке Python. Основы Алгоритмизации.

Практическая работа: Решение задач для развития логики и понимания работы алгоритма. Установка программы Python. Написание первой программы вывод на экран.

## **Раздел 2. Типы данных и операции**

### **а. Переменные. Операторы.**

Переменные. Оператор присваивания. Имена переменных и ключевые слова. Выражения. Операции. Порядок выполнения операций. Математические функции. Композиция. Ввод и вывод. Ввод данных с клавиатуры. Вывод данных на экран.

Практическая работа: Решение задач на элементарные действия с числами. Создание программы простейший калькулятор.

### **б. Типы данных. Базовые функции для работы с различными типами данных.**

Типы данных. Преобразование типов. Различия типов данных. Базовые функции (abs, round, int, math)

Практическая работа: Решение математических задач с использованием функции import math.

## **Раздел 3. Инструкции и синтаксис**

### **3.1. Условные операторы if, elif, else.**

Логический тип данных. Логические выражения и операторы. Сложные условные выражения (логические операции and, or, not). Условный оператор. Альтернативное выполнение. Примеры решения задач с условным оператором. Множественное ветвление. Реализация ветвления в языке Python.

### **3.2. Циклы while, for. Обработка исключений. Случайные числа.**

Понятие цикла. Тело цикла. Условия выполнения тела цикла. Оператор цикла с условием. Оператор цикла while. Бесконечные циклы. Альтернативная ветка цикла while. Обновление переменной. Краткая форма записи обновления. Примеры использования циклов. Оператор цикла с параметром for. Операторы управления циклом. Пример задачи с использованием цикла for. Вложенные циклы. Циклы в циклах. Случайные числа. Функция randrange. Функция random. Примеры решения задач с циклом.

Практическая работа: Числа Фибоначчи. Решение задачи с циклом for. Создание игры угадай число. Повторение пройденного.

## **Раздел 4. Функции и модули в программировании**

### **4.1. Встроенные и пользовательские функции.**

Создание функций. Параметры и аргументы. Локальные и глобальные переменные. Поток выполнения. Функции, возвращающие результат.

Анонимные функции, инструкция lambda. Примеры решения задач с использованием функций. Рекурсивные функции. Вычисление факториала.

Практическая работа: Создание игры русская рулетка. Создание всех ранее созданных программ с использованием функций.

#### **4.2. Строки - последовательности символов.**

Составной тип данных - строка. Доступ по индексу. Длина строки и отрицательные индексы. Преобразование типов. Применение цикла для обхода строки. Срезы строк. Строки нельзя изменить. Сравнение строк. Оператор in. Модуль string. Операторы для всех типов последовательностей (строки, списки, кортежи). Примеры решения задач со строками.

Практическая работа: Написание программ. Работа со строками.

### **Раздел 5. Сложные типы данных**

#### **5.1. Списки. Срезы списков.**

Списки. Тип список (list). Индексы. Обход списка. Проверка вхождения в список. Добавление в список. Суммирование или изменение списка. Операторы для списков. Срезы списков. Удаление списка. Клонирование списков. Списочные параметры. Функция range. Списки: примеры решения задач.

Практическая работа: Написание программ. Работа со списками.

#### **5.2. Матрицы**

Матрицы. Вложенные списки. Матрицы. Строки и списки. Генераторы списков в Python.

Установка Python. Доступ к документации. Ввод и вывод данных. Первая программа на языке Python.

Практическая работа: Создание программ. Практическое закрепление полученных знаний.

#### **5.3. Кортежи**

Кортежи. Присваивание кортежей. Кортежи как возвращаемые значения.

Практическая работа: Создание программ. Практическое закрепление полученных знаний.

#### **5.4. Словари**

Введение в словари. Тип словарь (dict). Словарные операции. Словарные методы.

Практическая работа: Создание программ. Практическое закрепление полученных знаний.

#### **5.5. Множества в языке Python**

Множества в языке Python. Множества. Множественный тип данных. Описание множеств. Операции, допустимые над множествами: объединение,

пересечение, разность, включение. Оператор определения принадлежности элемента множеству.

Практическая работа: Создание программ. Практическое закрепление полученных знаний.

### **Раздел 6. Написание и отладка программ. Самостоятельная работа**

Практическая работа: Стиль программирования. Отладка программ. Проектная работа по курсу «Программирование на языке Python».

## **ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОСВОЕНИЯ ПРОГРАММЫ УЧЕБНОГО КУРСА «ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ PYTHON»**

Личностные :

- умение работать в команде: работа в общем ритме, эффективное распределение задач и др.
- формирование высокого познавательного интереса учащихся
- формирование критического мышления
- проявление технического мышления, познавательной деятельности, творческой инициативы, самостоятельности

Метапредметные:

- умение ориентироваться в информационном пространстве, продуктивно использовать техническую литературу для поиска сложных решений
- способность творчески решать технические задачи
- готовность и способность применения теоретических знаний по физике, информатике для решения задач в реальном мире
- способность правильно организовывать рабочее место и время для достижения поставленных целей
- способность правильно организовывать рабочее место и время для достижения поставленных целей
- Знание основ ТРИЗ, навыки публичного выступления и презентации результатов, навык генерации идей

Предметные:

- знание основ и принципов программирования
- знание и понимание основных алгоритмических конструкций
- знание основ и овладение практическими базисными знаниями программирования в Python
- знание основ и овладение практическими базисными навыками разработки игр в PyGame

## ТЕМАТИЧЕСКОЕ ПЛАНИРОВАНИЕ

	Наименование раздела, темы	Количество часов
<b>1</b>	<b>Введение в Python</b>	<b>4</b>
	Инструкции и структура программы.	2
	Установка Python. Ввод и вывод данных.	2
<b>2</b>	<b>Типы данных и операции</b>	<b>12</b>
	Переменные. Операторы.	4
	Типы данных. Преобразование типов. Различия типов данных. Базовые функции (abs, round, int, math)	8
<b>3</b>	<b>Инструкции и синтаксис</b>	<b>14</b>
	Условный оператор if, Альтернативный условный оператор elif, else.	4
	Циклы while. Цикл for. Обработка включений. Функция random. Случайные числа.	10
<b>4</b>	<b>Функции и модули в программировании</b>	<b>18</b>
	Встроенные и пользовательские функции. Создание функций. Функции, возвращающие результат.	6
	Строки, последовательность символов. Доступ по индексу. Длина строки и отрицательные индексы. Преобразование типов. Применение цикла для обхода строки. Срезы строк. Сравнение строк.	12
<b>5</b>	<b>Сложные типы данных</b>	<b>14</b>
	Списки. Срезы списков. Решение задач со списками	8
	Матрицы	2
	Словари	2
	Множества в языке Python	2
<b>6</b>	<b>Написание и отладка программ. Самостоятельная работа</b>	<b>6</b>
	<b>ИТОГО</b>	<b>68</b>