

/Приложение №1  
к Приказу от 09.08.2023г. № 1/09082023

ОБЩЕСТВО С ОГРАНИЧЕННОЙ ОТВЕТСТВЕННОСТЬЮ  
«ЦИФРОВОЕ ОБРАЗОВАНИЕ»

СОГЛАСОВАНО  
Директор MAOY COII №4



/ О.Н. Гуржева /

Утверждаю:



Исполнительный директор  
ООО «Цифровое образование»

/ А.Е.Т. Олешинкин /

«09» августа 2023г.

Дополнительная общеобразовательная общеразвивающая  
программа «Основы программирования на Python»

Возраст обучающихся:

6 класс (изучающие математику на углублённом уровне), 7 класс

Срок реализации: 32 академических часа

Автор-составитель:  
Левшина А. В.,  
методист ООО  
«Цифровое  
образование»

2023 г.

## Пояснительная записка

Язык программирования Python на сегодняшний день является наиболее популярной высокоуровневой средой международного олимпиадного программирования. Python – мощный и простой для изучения язык программирования. В нем представлены проработанные высокоуровневые структуры данных и простой, но эффективный подход к объектно-ориентированному программированию.

Данная программа ориентирована на обучающихся 6 классов, изучающих математику на углублённом уровне и учеников 7 классов, интересующихся информатикой, желающих расширить знания в этой области.

Актуальность программы заключается в применении методик преподавания с использованием компьютерных технологий. Обучающимся предлагается изучить материал через решение задач, требующих от них гибкости ума и аналитического мышления.

Отличительными особенностями курса является его направленность на формирование у учащихся навыков поиска собственного решения поставленной задачи, составления алгоритма решения и реализации алгоритма с помощью средств программирования. В рамках предлагаемого курса изучение основ программирования на языке Python — это не столько средство подготовки к будущей профессиональной деятельности, сколько формирование новых общеинтеллектуальных умений и навыков: разделение задачи на этапы решения, построение алгоритма и др.

Новизна данного курса заключается в общедоступности и понятности, так как все материалы представлены в готовом видео-формате и не требуют высокого уровня профессиональной предметной подготовки преподавателя, присутствующего на занятии.

Практическая направленность программы на создание внешних образовательных продуктов — алгоритмов, простейших приложений с графическим интерфейсом на языке Python — способствует выявлению фактов, которые невозможно объяснить на основе имеющихся у школьников знаний. Возникающие при этом познавательные переживания обуславливают сознательное отношение к изучению основных теоретических положений информатики, стимулируют развитие индивидуально-личностных качеств школьников. Активизация познавательного процесса позволяет учащимся более полно выражать свой творческий потенциал и реализовывать собственные идеи в изучаемой области знаний, создаёт предпосылки по применению освоенных навыков программирования в других учебных курсах, а также способствует возникновению дальнейшей мотивации, направленной на освоение профессий, связанных с разработкой программного обеспечения.

Программа способствует созданию дополнительных условий для проявления индивидуальных образовательных интересов учащихся 6 и 7 классов. Курс требует освоения знаний общеобразовательной программы предмета «Математика» на углублённом уровне для учеников 6 классов, на базовом уровне для учеников 7 классов.

С целью контроля и оценки результатов подготовки и учета индивидуальных образовательных достижений обучающихся применяются: поурочный контроль знаний, итоговая диагностика. Аттестация проводится по результатам итогового тестирования.

Данная дополнительная общеразвивающая программа технической направленности, уровень освоения - базовый. Программа рассчитана на 32 часа.

Цель программы — познакомить школьников с современным языком программирования Python, научить алгоритмически решать простейшие задачи с использованием среды программирования, а также дать возможность применить полученные знания на практике через решение разнообразных задач и создание приложения с графическим интерфейсом. Для реализации этой цели необходимо решить следующие задачи:

- развивать алгоритмические способности учащихся и развивать у школьников критическое мышление, логическое мышление, алгоритмическую культуру;
- изучить простейшие структуры данных и синтаксис языка Python;
- развивать умения записи алгоритмов при решении математических задач и переноса этого алгоритма на язык программирования с учетом его синтаксиса;
- создать условия для разработки оконного приложения с графическим интерфейсом;
- способствовать популяризации информатики как науки.

## Планируемые результаты освоения программы

Личностные	<ul style="list-style-type: none"> <li>– сформированность ценностей самостоятельности и инициативы;</li> <li>– готовность обучающихся к саморазвитию, самостоятельности и личностному самоопределению;</li> <li>– наличие мотивации к целенаправленной социально значимой деятельности;</li> <li>– сформированность внутренней позиции личности как особого ценностного отношения к себе, окружающим людям и жизни в целом.</li> </ul>
Метапредметные	<p><b>Регулятивные универсальные учебные действия</b></p> <ul style="list-style-type: none"> <li>– самостоятельно определять цели, задавать параметры и критерии, по которым можно определить, что цель достигнута;</li> </ul>

	<ul style="list-style-type: none"> <li>- оценивать возможные последствия достижения поставленной цели в деятельности, собственной жизни и жизни окружающих людей, основываясь на соображениях этики и морали;</li> <li>- ставить и формулировать собственные задачи в образовательной деятельности;</li> <li>- оценивать ресурсы, в том числе время и другие нематериальные ресурсы, необходимые для достижения поставленной цели;</li> <li>- выбирать путь достижения цели, планировать решение поставленных задач, оптимизируя материальные и нематериальные затраты;</li> <li>- организовывать эффективный поиск ресурсов, необходимых для достижения поставленной цели;</li> <li>- сопоставлять полученный результат деятельности с поставленной заранее целью.</li> </ul> <p><b>Познавательные универсальные учебные действия</b></p> <ul style="list-style-type: none"> <li>- искать и находить обобщенные способы решения задач, в том числе, осуществлять развернутый информационный поиск и ставить на его основе новые (учебные и познавательные) задачи;</li> <li>- критически оценивать и интерпретировать информацию с разных позиций, распознавать и фиксировать противоречия в информационных источниках;</li> <li>- использовать различные модельно-схематические средства для представления существенных связей и отношений, а также противоречий, выявленных в информационных источниках;</li> <li>- находить и приводить критические аргументы в отношении действий и суждений другого; спокойно и разумно относиться к критическим замечаниям в отношении собственного суждения, рассматривать их как ресурс собственного развития;</li> <li>- выходить за рамки учебного предмета и осуществлять целенаправленный поиск возможностей для широкого переноса средств и способов действия;</li> <li>- выстраивать индивидуальную образовательную траекторию, учитывая ограничения со стороны других участников и ресурсные ограничения;</li> <li>- менять и удерживать разные позиции в познавательной деятельности.</li> </ul> <p><b>Коммуникативные универсальные учебные действия</b></p> <ul style="list-style-type: none"> <li>- развернуто, логично и точно излагать свою точку зрения с использованием адекватных языковых средств.</li> </ul>
Предметные (образовательные)	<ul style="list-style-type: none"> <li>- уметь писать простейшие программы;</li> <li>- уметь использовать приобретенные знания и умения в практической деятельности и повседневной жизни;</li> <li>- уметь выполнять вычисления и преобразования с учетом синтаксиса изучаемого языка;</li> <li>- уметь выполнять действия с функциями с учетом синтаксиса изучаемого языка;</li> <li>- уметь строить и исследовать математические модели;</li> <li>- точно и грамотно выражать свои мысли в устной и письменной</li> </ul>

	речи.
--	-------

## Учебный план

Учебный план строится с учетом продолжительности курса 32 часа.

№ п/п	Название раздела	Всего	Теория	Практика	Формы аттестации (контроля)
1.	Среда разработки, типы данных, переменные, функции ввода и вывода, простейшие арифметические функции	6	2	4	тестирование
2.	Условные конструкции и циклы	4	2	2	тестирование
3.	Строки, списки, словари, множества	6	2	4	тестирование
4.	Объявление функций, возвратные функции, работа с функциями	3	1	2	тестирование
5.	Модули, графический модуль turtle	4	1	3	тестирование
6.	Основы ООП	4	1	3	тестирование
7.	Графический модуль tkinter. Создание приложения с графическим интерфейсом	4	1	3	тестирование
8.	Итоговое тестирование	1		1	тестирование
	<b>Итого</b>	<b>32</b>	<b>10</b>	<b>22</b>	

## Содержание учебного плана

**Раздел 1. Среда разработки, типы данных, переменные, функции**

## **ввода и вывода, простейшие арифметические функции. (6 часов)**

### **1.1 Среда разработки VSCode. Переменные. (1 час)**

Знакомство со средой Visual Studio Code, с понятием переменной, простейшими типами данных и операциями над ними.

### **1.2 Вывод данных. Функция print(). (1 час)**

Возможности и особенности работы функции print() и понятие функции в целом.

### **1.3 Типы данных. Функции int() и str() и float(). (1 час)**

Подробнее о типах данных, преобразование типов данных. Зачем нужны, для чего используются и как работают функции int() str() и float().

### **1.4 Ввод данных на компьютере. Функция input(). (1 час)**

Особенности ввода данных с клавиатуры. Работа функции input(), аргумент функции input(), ввод строк и ввод чисел, различия, приёмы.

### **1.5 Операции с числами в Python. (1 час)**

Целочисленная арифметика в Python, практика, применение. Определение последней, первой цифры числа с помощью целочисленного деления.

### **1.6 Работа с функциями и переменными. Решение задач. (1 час)**

Использование функций на практике. Написание программ и решение задач с использованием функций print, input, str, int, float и работы с переменными.

## **Раздел 2. Условные конструкции и циклы. (4 часов)**

### **2.1 Условные конструкции. Оператор if-else. (1 час)**

Знакомство с логическим типом данных, условные выражения, условный оператор if-else.

### **2.2 Вложенные условные конструкции. Оператор elif. (1 час)**

Знакомство с вложенными условными инструкциями. Зачем нужен elif, синтаксис, применение.

### **2.3 Циклы в программировании. Цикл while. (1 час)**

Понятие цикла в программировании, его применение, принцип работы и синтаксис цикла while. Манипуляции с итерациями цикла, применение операторов break и continue.

### **2.4 Цикл for. Функция range(). (1 час)**

Принцип работы и синтаксис цикла for. Отличие от цикла while, функция range и её аргументы.

### **Раздел 3. Строки, списки, словари и множества. (6 часов)**

#### **3.1 Строки. Операции над строками. Обращение по индексу. (1 час)**

Углубление в строки, методы строк. Строка как массив и обращение по индексу.

#### **3.2 Строки. Операции над строками. Срезы. Методы. (1 час)**

Понятие подстроки строки, срезы строк, использование метода replace().

#### **3.3 Списки. Операции со списками. Обращение по индексу. (1 час)**

Что такое список, для чего он нужен, его отличие от обычной записи в переменную. Обращение по индексу, добавление и удаление из списка.

#### **3.4 Списки. Работа с элементами списка. Функции и методы. (1 час)**

Практическая работа со списками, нахождение наибольшего\наименьшего элемента с и без использования функций min(), max().

#### **3.5 Словари. Операции со словарями. Ключи и значения. Методы словарей. (1 час)**

Структура данных словарь, принцип работы, операции со словарём.

#### **3.6 Множества. Операции с множествами. Методы множеств. (1 час)**

Структура данных множество, принцип работы, операции со множествами.

### **Раздел 4. Объявление функций, возвратные функции, работа с функциями. (3 часа)**

#### **4.1 Объявление и использование функций в Python. Аргументы функций. (1 час)**

Создание собственных функций в Python, вызов собственных функций, выбор аргументов при создании функции.

#### **4.2 Объявление и использование функций в Python. Типы функций. Оператор return. (1 час)**

Понятие возвращаемого значения, оператор return.

#### **4.3 Работа с функциями. Простейший калькулятор. (1 час)**

Практика с функциями. Написание простейшего калькулятора с использованием функций.



## **Раздел 5. Модули, графический модуль turtle. (4 часа)**

### **5.1 Библиотеки в программировании. Подключение библиотек. Модуль random(). (1 час)**

Как подключать библиотеки в проект на Python, использование собственных программ как модулей, модуль random() и его функции.

### **5.2 Графический модуль turtle. Функции forward(), left(), exitonclick(). (1 час)**

Подключение и работа с графическим модулем turtle. Использование turtle для визуализации работы простейших программы.

### **5.3 Графический модуль turtle. Функции color(), shape(), begin\_fill(), end\_fill(). (1 час)**

Продвинутое рисование с помощью turtle. Функции изменения формы, цвета и заливки.

### **5.4 Графический модуль turtle. Практика. Проект “Картина”. (1 час)**

Проект “Картина”, основанный на применении собственных функций и функций встроенных в модуль turtle.

## **Раздел 6. Основы объектно-ориентированного программирования. (4 часа)**

### **6.1 Введение в ООП. Понятие класса, объекта, методов на примере Turtle(). (1 час)**

Знакомство с ООП, способ создания исполнителей turtle как экземпляров класса Turtle. Классы, объекты, методы, атрибуты на примере модуля turtle.

### **6.2 Введение в ООП. Практика. Проект “Гонки”. (1 час)**

Проект “Гонки” – гонки черепашек с использованием функций модуля random.

### **6.3 Объявление и использование классов в Python. (1 час)**

Синтаксис и создание классов в Python. Понятие конструктора класса, атрибутов, методов.

### **6.4 Объявление и использование классов в Python. Практика. (1 час)**

Создание простой текстовой игры с применением Классов.

## **Раздел 7. Графический модуль tkinter. Создание приложения с графическим интерфейсом (4 часа)**

### **7.1 Графический модуль tkinter. Понятие графического интерфейса. Виджеты. (1 час)**

Что такое графический интерфейс. Виджеты label, button, entry, создание окна приложения.

### **7.2 Графический модуль tkinter. Виды позиционирования виджетов. (1 час)**

Абсолютное и относительное позиционирование виджетов, методы place(), pack(), grid().

### **7.3 Графический модуль tkinter. Практика. Интерфейс для проекта на модуле turtle. (1 час)**

Управление исполнителем turtle на основе виджетов tkinter.

### **7.4 Графический модуль tkinter. Практика. Приложение с графическим интерфейсом. (1 час)**

Разработка оконного приложения с графическим интерфейсом, основанном на модуле tkinter.

## **8. Итоговая аттестация (1 ак.ч.)**

Прохождение тестирования по всем темам курса .

### **Методическое обеспечение образовательной программы**

В ходе реализации образовательной программы используются следующие образовательные технологии:

- интерактивные лекции в видеоформате
- тренинги по решению заданий – выполнение тренировочных заданий, позволяющее приобрести опыт решения задач.

<b>Форма организации и форма проведения занятия</b>	<b>Методы и приемы организации учебно-воспитательного процесса</b>
<b>Форма организации детей на занятии:</b> фронтальная, индивидуально-фронтальная  <b>Формы проведения занятий:</b> Комбинированное занятие	<b>Словесные:</b> объяснение, беседа, дискуссия <b>Наглядные:</b> демонстрационные материалы, видеофрагменты, показ педагогом образца выполнения задания, и т.п. <b>Информационно-коммуникационные:</b> электронные и информационные ресурсы с аудио- и видеоинформацией. <b>Практические:</b> практические задания, упражнения <b>Методы проблемного обучения:</b>

	Поиск (самостоятельный поиск ответа на поставленные вопросы) <b>Методы стимулирования и мотивации деятельности и поведения:</b> одобрение, похвала, игровые эмоциональные ситуации, использование примера
--	---

## **Требования к условиям организации образовательного процесса**

Онлайн-платформа. Программное обеспечение, представляющее собой набор взаимосвязанных веб-сервисов и модулей, составляющих единое пространство предоставления услуг потребителям в сети Интернет. Включает в себя следующие модули, обеспечивающие учебный процесс по программе:

- модуль трансляции занятий с интерактивными возможностями;
- модуль практических заданий различного типа;
- модуль контроля и результативности обучения (тесты).

Электронные образовательные ресурсы:

- модуль мультимедийных материалов в формате видео разборов тем, рассматриваемых в рамках программы.

## **Оценка реализации программы и образовательные результаты программы**

По итогам прохождения программы обучающиеся проходят тестирование.

## **Требования к кадровому обеспечению**

Высшее образование по профилю предметной области “математика и информатика”.

### **Литература**

1. Марк Лутц. Программирование на Python, том I, 4-е издание. – Пер. с англ. – СПб.: Символ-Плюс, 2011. – 992 с., ил.
2. Мэтиз Эрик. Изучаем Python: программирование игр, визуализация данных, веб-приложения. 3-е изд. — СПб.: Питер, 2020. — 512 с.: ил. — (Серия «Библиотека программиста»).

### **Электронные ресурсы:**

1. Питонтьютор. [Электронный ресурс]. – URL: <https://pythontutor.ru> (дата обращения 24.07.2023)