



Директор ЧОУ ДО ЦИТО «Познание»

И.В. Вылегжанина

Киров, 23.04.2020

Частное образовательное учреждение дополнительного образования  
Центр информационных технологий в обучении «Познание»

Дополнительная общеобразовательная (общеразвивающая) программа  
по программированию  
«Программирование Python: от основ до собственных приложений»  
Возраст: 11-14 лет  
Срок реализации: 72 часа

Автор программы:  
Кибешева И.Р., преподаватель ЧОУ ДО ЦИТО «Познание»

Киров, 2020

## ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

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. Как считают многие, один из самых используемых языков программирования в мире.

Содержание курса программирования на языке Python объединяет в себе основы алгоритмизации и проектирования, математическое моделирование и математическую логику, программирование, информационно-коммуникационные технологии. Обучающиеся смогут анализировать информацию, выявлять в ней факты и проблемы, самостоятельно ставить задачи, структурировать и преобразовывать информацию, использовать ее для решения учебных и жизненных задач.

При организации обучения программированию учитываются возрастные и индивидуальные особенности детей, через использование педагогических технологий, ориентированных на включение обучающихся в активную деятельность, сотрудничество, на достижение не только предметных, но и метапредметных и личностных образовательных результатов, раннюю профессиональную ориентацию обучающихся.

Программирование позволяет успешно готовиться к участию в олимпиадах, конкурсах. Кроме того, данный курс поможет учащимся, выбравшим предмет «Информатика» для сдачи экзамена по выбору, а также облегчит изучение других языков программирования. Школьники, проявившие особые успехи, приглашаются для участия в региональных, российских и международных соревнованиях, олимпиадах, тем самым осуществляется выявление и поддержка детей, проявивших выдающиеся способности.

Содержание обучения, представленное в программе «Программирование на языке «Python», позволяет вести обучение в режиме актуального познания. Практическая направленность курса на создание внешних образовательных продуктов – блок-схем, алгоритмов, программ – способствует выявлению фактов, которые невозможно объяснить на основе имеющихся у учащихся знаний.

Актуальность программы состоит в том, что активизация познавательного процесса позволяет учащимся более полно выражать свой творческий потенциал и реализовывать собственные идеи в изучаемой области знаний, создаёт предпосылки по применению информационных компетенций в других учебных курсах, а также способствует возникновению дальней мотивации, направленной на освоение профессий, связанных с разработкой программного обеспечения.

Курс служит средством внутрипрофильной специализации в области новых информационных технологий, что способствует созданию дополнительных условий для проявления индивидуальных образовательных интересов учащихся, их дальнейшей профессиональной ориентации.

Программа имеет техническую направленность, является прикладной, носит практико-ориентировочный характер и направлена на овладение обучающимися основных приемов программирования. Обучение по данной программе создает благоприятные условия для интеллектуального и духовного воспитания личности ребенка, социально-культурного и профессионального самоопределения, развития познавательной активности и творческой самореализации учащихся.

При разработке программы учтены положения и требования следующих нормативных документов:

1. Федеральный Закон «Об образовании в Российской Федерации» от 29.12.2012 № 273-ФЗ;
2. Концепция развития дополнительного образования детей (утверждена распоряжением Правительства РФ от 04.09.2014 г. № 1726-р);
3. Конвенция о правах ребёнка;

4. Национальная доктрина образования в РФ на период до 2025 года;
5. Распоряжение Правительства Российской Федерации от 24.04.2015 г. № 729-р «О плане мероприятий на 2015-2020 годы по реализации Концепции развития дополнительного образования детей»;
6. СанПиН 2.4.4.3172-14 «Санитарно-эпидемиологические требования к устройству, содержанию и организации режима работы образовательных организаций дополнительного образования детей» (утверждено постановлением Главного государственного санитарного врача РФ от 04.07.2014 № 41);
7. Письмо Минобрнауки России от 11.12.2006 г. № 06-1844 «О примерных требованиях к программам дополнительного образования детей»;
8. Приказ Министерства образования и науки Российской Федерации (Минобрнауки России) от 09.11.2018 г. № 196 г. Москва «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам»;
9. Письмо Минобрнауки России от 18.11.2015г. № 09-3242 «О направлении информации» (вместе «с Методическими рекомендациями по проектированию ДООП);
10. Устав ЧОУ ДО ЦИТО «Познание».

В настоящее время Россия остро нуждается в инженерах, других специалистах, разрабатывающих высокие технологии, выпускающих наукоемкие изделия, умеющих осуществлять инновации. Учитывая общественный запрос, в области внедряются новые виды технического творчества: программирование, программы мультипликационной анимации, робототехника и др. Одной из важных задач развития дополнительного образования детей является обеспечение доступности дополнительных образовательных программ технической направленности, соответствующих приоритетным направлениям технологического развития Российской Федерации.

**Цель:** 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.

**Задачи:**

Обучающие:

- способствовать формированию учебно-интеллектуальных умений, приёмов мыслительной деятельности, освоению рациональных способов её осуществления на основе учета индивидуальных особенностей учащихся;
- способствовать формированию активного, самостоятельного, креативного мышления;
- познакомить учащихся с основными алгоритмическими конструкциями и правилами их записи, с основными способами организации данных;
- научить учащихся составлять и записывать алгоритмы с использованием соответствующих алгоритмических конструкций;
- научить распознавать необходимость применения той или иной алгоритмической конструкции при решении задач;

- научить организовывать данные для эффективной алгоритмической обработки;
- научить учащихся разрабатывать алгоритмы и реализовывать их на языке программирования Python;
- научить учащихся осуществлять отладку и тестирование программы.

**Развивающие:**

- формировать новый тип мышления – операционный, который направлен на выбор оптимальных решений;
- развивать психические познавательные процессы: мышление, восприятие, память, воображение у учащихся;
- развивать представление учащихся о практическом значении информатики.

**Воспитательные:**

- воспитание волевых качеств: настойчивости, целеустремленности;
- воспитывать культуру алгоритмического мышления;
- воспитывать у учащихся усидчивость, терпение, трудолюбие.
- ознакомление с миром технических профессий, связанных с программированием.

**Возраст участников, особенности изучения программы:**

Средние классы (11-14 лет).

**Сроки реализации образовательной программы:** 72 часа.

**Форма обучения:** комбинированные занятия (занятия состоят из теоретической и практической частей, причём большее количество времени занимает практическая часть).

**Виды учебной деятельности:** лекция, беседа, демонстрация, практическая работа в среде IDLE Python, проектная деятельность, соревнования.

**Формы контроля и оценки образовательных результатов:** текущий контроль осуществляется по результатам выполнения обучающихся

практических заданий, итоговый контроль – в форме проектов с открытым решением, соревнований по программированию, олимпиад.

**Средства обучения:** среда программирования «Pascal ABC.NET», цифровые образовательные ресурсы (презентации, видеоролики, программы), дистанционные образовательные технологии.

### **Планируемые образовательные результаты:**

#### **Метапредметные**

##### **Регулятивные универсальные учебные действия:**

- планировать последовательность шагов алгоритма для решения поставленной задачи;
- вносить коррективы в действия в случае расхождения результата решения задачи на основе ее оценки и учета характера сделанных ошибок;
- проявлять познавательную инициативу в учебном сотрудничестве;
- овладение основами самоконтроля, самооценки, принятия решений и осуществления осознанного выбора в учебной и познавательной деятельности.

##### **Познавательные универсальные учебные действия:**

- формирование информационно-логических умений: определять понятия, создавать обобщения, устанавливать аналогии, классифицировать, устанавливать причинно-следственные связи, строить логическое рассуждение, умозаключение и делать выводы;
- овладение основными универсальными умениями информационного характера: постановка и формулирование проблемы; поиск и выделение необходимой информации, применение методов информационного поиска; структурирование и визуализация информации; выбор наиболее эффективных способов решения задач в зависимости от конкретных условий; самостоятельное создание алгоритмов деятельности при решении проблем творческого и поискового характера.

##### **Коммуникативные универсальные учебные действия:**

- аргументировать свою точку зрения;
- выслушивать собеседника и вести диалог;
- признавать возможность существования различных точек зрения и права каждого иметь свою;
- уметь с достаточной полнотой и точностью выражать свои мысли в соответствии с задачами и условиями коммуникации.

### **Предметные результаты**

По окончании обучения обучающиеся должны

- **знать:** роль программного обеспечения и его виды; основные алгоритмические конструкции и правила их записи; основные способы организации данных; основные изучаемые понятия: информация, алгоритм, модель – и их свойства.
- **уметь:** составлять и записывать алгоритмы с использованием соответствующих алгоритмических конструкций; распознавать необходимость применения той или иной алгоритмической конструкции при решении задачи; организовывать данные для эффективной алгоритмической обработки; разрабатывать алгоритмы и реализовывать их на языке программирования Python; осуществлять отладку и тестирование программы в среде «IDLE (Python)»; использовать идею координат на плоскости для графической интерпретации объектов, использовать компьютерные программы для иллюстрации решений, для построения, проведения экспериментов.
- **владеть:** алгоритмическим мышлением, пониманием необходимости формального описания алгоритмов; умением понимать программы, написанные на выбранном для изучения универсальном алгоритмическом языке высокого уровня знание основных конструкций программирования (ветвление, цикл, подпрограмма); умение анализировать алгоритмы с использованием таблиц; стандартными приемами написания программы для



решения стандартной задачи с использованием основных конструкций программирования.

### **Личностные**

- формирование собственного жизненного опыта значимости подготовки в области программирования в условиях развития информационного общества;
- формирование коммуникативной компетентности в общении и сотрудничестве со сверстниками, детьми старшего и младшего возраста, взрослыми в процессе образовательной, творческой и других видов деятельности;
- формирование способности обучающихся к саморазвитию и личностному самоопределению, мотивации к целенаправленной познавательной деятельности с целью приобретения профессиональных навыков в ИТ-сфере.

**Учебно-тематический план**

№	Наименование	Количество часов
1.	Сентябрь – Линейные программы	8
2.	Октябрь – Ветвления	8
3.	Ноябрь – Циклы	8
4.	Декабрь – Строки и файлы	8
5.	Январь – Списки и кортежи	8
6.	Февраль – Подпрограммы	8
7.	Март – Tkinter: Графика	8
8.	Апрель – Tkinter: Оконные приложения	8
9.	Май – Tkinter: Создание игр	8
	Итого	72

## СОДЕРЖАНИЕ ПРОГРАММЫ

### СЕНТЯБРЬ – ЛИНЕЙНЫЕ ПРОГРАММЫ.

1. Знакомство с интерфейсом среды. Структура программы. Операторы ввода и вывода. Первая программа. Дружественный интерфейс. Этикет программиста.
2. Переменные. Основные типы данных. Оператор присваивания. Арифметические операции.
3. Целочисленное деление и нахождение остатка от деления. Задачи на округление. Задачи с цифрами числа. Кратность чисел.
4. Представление алгоритма в виде блок-схемы. Решение задач с помощью математического моделирования. Разработка и исполнение линейных программ.

### ОКТАБРЬ – УСЛОВНЫЙ ОПЕРАТОР.

5. Логический тип данных (Boolean). Операции отношения. Ввод-вывод булевых переменных. Базовые логические операции.
6. Условные алгоритмы. Условный оператор if...elif...else. Сокращенное ветвление. Полное ветвление. Блок-схемы, изображающие условные операторы.
7. Операторные скобки. Вложенные условные операторы. Оператор выбора.
8. Разработка и исполнение разветвляющихся программ с использованием сложных условных операторов в среде PascalABC.NET.

### НОЯБРЬ – ЦИКЛЫ.

9. Циклические алгоритмы и их назначение. Конструкция range. Оператор цикла с параметром for. Блок-схемы, изображающие циклы.
10. Оператор цикла с предусловием while . Досрочное прерывание цикла.
11. Оператор цикла с постусловием. Оператор безусловного перехода goto.
12. Вложенные циклы.

### ДЕКАБРЬ – СТРОКИ И ФАЙЛЫ

13. Символьный тип данных. Таблица символов ASCII. Процедуры для работы с символами: нахождение символа по номеру, поиск номера символа.
14. Строковый тип данных. Основные операции. Программирование алгоритмов обработки текста. Операции поиска и замены в символьных строках и массивах. Шифровка и дешифровка текста.
15. Файлы. Текстовые файлы. Файлы с фиксированной структурой записи. Процедуры и функции для работы с файлами. Программирование ввода-вывода.

16. Разработка и исполнение программ обработки текстового файла. Сохранение нового файла с ответом.

#### ЯНВАРЬ – СПИСКИ И КОРТЕЖИ

17. Одномерные массивы. Понятие одномерного массива. Константы. Объявление массива. Инициализация массива. Вывод элементов массива. Заполнение массива случайными числами. Заполнение массива с клавиатуры.
18. Вычисление суммы и количества элементов массива с заданными свойствами. Разработка и исполнение программ обработки массива с изменением элементов, нахождение среднего арифметического всех элементов
19. Разработка и исполнение программ обработки массива на нахождение минимального, максимального элементов и их номеров, нахождение количества нулевых, положительных, четных, нечетных элементов.
20. Сортировка элементов массива методом «пузырька».

#### ФЕВРАЛЬ – ПОДПРОГРАММЫ

21. Подпрограммы (функции и процедуры). Назначение и различия. Способы описания. Обмен информацией между основной программой и подпрограммой. Глобальные и локальные переменные.
22. Использование подпрограмм для работы с массивами.
23. Использование подпрограмм для работы с текстовыми файлами.
24. Рекурсия. Примеры рекурсивного программирования. Комбинаторика.

#### МАРТ – TKINTER: ГРАФИКА

25. Окна. Создание окна. Геометрия окна. Заголовок окна. Положение окна.
26. Виджет Canvas. Геометрия холста. Оконная система координат. Методы «Линия», «Текст» и их свойства.
27. Методы «Прямоугольник», «Овал», «Дуга», «Сектор», «Многоугольник» и их свойства. Создание изображения из фигур.
28. Идентификаторы. Теги. Анимация в Tkinter.

#### АПРЕЛЬ – TKINTER: ОКОННЫЕ ПРИЛОЖЕНИЯ

29. Виджеты Button, Label, Entry, Text и их свойства. Метод Pack(). Создание калькулятора.
30. Виджеты Radiobutton, Checkbutton и их свойства. Переменные в Tkinter. Метод grid(). Создание Графического редактора.

31. Виджет Listbox. Метод bind(). Категории событий. Привязка событий к виджету.

32. Диалоговые окна. Виджет Menu(). Метод Place().

### МАЙ – TKINTER: СОЗДАНИЕ ИГР

33. Игра «Вратарь»

34. Игра «Питомец»

35. Игра «Змейка»

36. Игра «Змейка»

## МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ

### Структура занятия

1. Актуализация. Ответ на вопросы обучающихся по прошлым темам, домашним задачам.
2. Теоретическая часть (презентации, примеры готового кода, устный разбор и компиляция отдельных алгоритмов).
3. Практическая часть – решение задач с преподавателем, самостоятельное решение задач, сверка этапов решения, обсуждение идей и различных способов решения.
4. Подведение итогов занятия.

### **ЛИТЕРАТУРА:**

1. Лутц М. Программирование на Python, том I, 4-е издание. — Пер. с англ. — СПб.: Символ-Плюс, 2011. — 992 с., ил.
2. Майк МакГрат «Программирование на Python для начинающих» Эксмо, 2015.
3. Федоров Д. Ю. Основы программирования на примере языка Python. //Учебное пособие. — Санкт-Петербург: 2016.
4. Сэнд У., Сенд К. «Hello World! Занимательное программирование на языке Python» - М.: – 2016.
5. Свейгарт, Эл. Автоматизация рутинных задач с помощью Python: практическое руководство для начинающих.: Пер. с англ.—М.: ООО “ИД. Вильямс”, 2016.—592с.: ил.—Парад. тит. англ.
6. Мэтиз Эрик Изучаем Python. Программирование игр, визуализация данных, веб-приложения. — СПб.: Питер, 2017. — 496 с.: ил. — (Серия «Библиотека программиста»).