

**Государственное бюджетное общеобразовательное учреждение  
«Гимназия №1 г.Малгобек им. С. Чахкиева»**

<b>РАССМОТРЕНО</b> на заседании педагогического совета  Протокол № 1 от « 28 » августа 2024 г.	<b>УТВЕРЖДАЮ</b> Директор ГБОУ «Гимназия №1 г.Малгобек им. С. Чахкиева» _____/З.У. Булгучева./ Приказ № 217 от « 29 » сентября 2024 г.
--	---

**ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ  
(ОБЩЕРАЗВИВАЮЩАЯ) ПРОГРАММА**

**«Программирование «Python»**

Срок реализации: 1 год

Возраст обучающихся: 12-17 лет

Составил:

учитель информатики Яндиев  
Юсуп Микаилович



## ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Дополнительная общеобразовательная общеразвивающая программа технической направленности «Программирование на Python» составлена для организации внеурочной деятельности учащихся среднего и старшего звена основной школы и ориентирована на обучающихся, проявляющих интересы и склонности в области информатики, математики, физики, моделирования, компьютерной графики.

Данная дополнительная общеразвивающая программа направлена на развитие логического и пространственного мышления слушателя, способствует раскрытию творческого потенциала личности, формированию усидчивости и трудолюбия, приобретению практических умений и навыков в области компьютерных технологий, способствует интеллектуальному развитию ребенка.

Программа ориентирована на обобщение и углубление знаний и умений по курсу информатики основной школы.

Данная дополнительная общеразвивающая программа ориентирована на **один год** обучения. Программа имеет **базовый уровень**.

### **Актуальность программы.**

Поддержка и развитие детского технического творчества соответствуют актуальным и перспективным потребностям личности и стратегическим национальным приоритетам Российской Федерации.

Базовой основой для проектирования региональной стратегии развития научно-технического творчества, учебно-исследовательской деятельности обучающихся и молодежи являются нормативные и правовые акты:

- Федеральный закон «О науке и государственной научно-технической политике» от 23.08.1996 № 127-ФЗ (ред. от 02.07.2013).
- Федеральный закон «Об образовании в Российской Федерации» от 29 декабря 2012 г. № 273-ФЗ.
- Указ Президента РФ от 1 июня 2012 года № 761 «Национальная стратегия действий в интересах детей на 2012-2017 годы».
- Концепция развития дополнительного образования детей от 4 сентября 2014 года № 1726-р.
- Нормативно-правовая база образовательной программы системы внеурочной деятельности. ФГОС НОО.
- Порядок организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам (от 29.08.2013 г.).
- Постановление Главного государственного санитарного врача РФ от 4 июля 2014 г. № 41 «Об утверждении СанПиН 2.4.4.3172-14 «Санитарно-эпидемиологические требования к устройству, содержанию и организации режима работы образовательных организаций дополнительного образования детей».

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.

Основной **целью** данного учебного курса является ознакомление с объектно-ориентированным языком программирования Python, с возможностями, синтаксисом языка, технологией и методами программирования в среде Python, обучение практическим навыкам программирования на языке Python для решения типовых задач математики и информатики, а также при разработке простейших игр.

Реализация поставленной цели предусматривает решение следующих **задач**:

- развитие интереса учащихся к изучению программирования;
- знакомство учащихся с основами программирования в среде Python;
- овладение базовыми понятиями теории алгоритмов при решении математических задач;
- формирование навыков работы в системе программирования Python;
- формирование самостоятельности и творческого подхода к решению задач с помощью средств современной вычислительной техники;
- формирование навыков алгоритмического и логического мышления;
- формирование навыков грамотной разработки программ;
- формирование практических навыков решения прикладных задач;
- формирование практических навыков разработки игр.

#### **Отличительные особенности данной программы.**

На занятиях учащиеся познакомятся с теоретическими аспектами и синтаксисом языка, а также обучатся практическим навыкам программирования в среде Python.

Занятия начинаются с практического знакомства со средой программирования Python, далее идет непосредственное изучение синтаксических конструкций языка и отработка навыков применения элементов программирования при решении задач и создании игр. Каждая новая тема завершается практическими задачами, способствующими овладению методикой программирования и изучению языка Python. На втором году обучения закрепляются полученные знания и навыки программирования, осваиваются новые методы, способы решения задач, рассматриваются более сложные задачи, новые технологии программирования.

Данная дополнительная общеобразовательная программа предназначена для детей **14-16 лет**.

**Срок реализации** программы «Программирование на Python» составляет 1 год (35 недель, 140 часов). Период обучения: сентябрь - май.

**Форма занятий** - групповая. Наполняемость группы - от 5 до 15 человек (набор осуществляется без предварительного отбора, по желанию и интересу учащегося).

#### **Формы контроля:**

- практические работы;
- мини-проекты.

#### **Методы обучения:**

- Познавательный (восприятие, осмысление и запоминание учащимися нового материала с привлечением наблюдения готовых примеров, моделирования, изучения иллюстраций, восприятия, анализа и обобщения демонстрируемых материалов).
- Метод проектов (при усвоении и творческом применении навыков и умений в процессе разработки собственных моделей).
- Систематизирующий (беседа по теме, составление систематизирующих таблиц, графиков, схем и т.д.).
- Контрольный метод (при выявлении качества усвоения знаний, навыков и умений и их коррекция в процессе выполнения практических заданий).
- Групповая работа.

**Режим занятий:** Обучающиеся занимаются 2 раза в неделю по 2 часа.

### **ПЛАНИРУЕМЫЕ (ОЖИДАЕМЫЕ) РЕЗУЛЬТАТЫ**

В результате изучения дисциплины получают дальнейшее развитие личностные регулятивные, коммуникативные и познавательно-универсальные учебные действия, учебная (общая и предметная) и общепользовательская ИКТ-компетентность обучающихся, составляющая психолого-педагогическую, инструментальную основу формирования способности и готовности к освоению систематических знаний, к их самостоятельному пополнению, переносу и интеграции, способности к сотрудничеству и коммуникации, решению личностно и социально значимых проблем и воплощению решений в практику, способности к самоорганизации, саморегуляции и рефлексии.

В ходе изучения данного курса в основном формируются и получают развитие следующие **метапредметные результаты:**

- умение самостоятельно планировать пути достижения цели, в том числе альтернативные, осознанно выбирать наиболее эффективные способы решения задач;
- умение соотносить свои действия с планируемыми результатами, осуществлять контроль всей деятельности в процессе достижения результата, корректировать свои действия в соответствии с изменяющейся ситуацией;
- умение оценивать правильность выполнения учебной задачи, собственные возможности ее решения;
- владение основами самоконтроля, самооценки, принятия решений и осуществления осознанного выбора в учебной и познавательной деятельности;
- умение организовывать сотрудничество и совместную деятельность с учителем и сверстниками; работать индивидуально в группе: находить общее решение и разрешать конфликты на основе согласования позиций и учета интересов; формулировать, аргументировать и отстаивать свое мнение;
- формирование и развитие компетентности в области использования информационно-коммуникационных технологий.

Вместе с тем, вносятся существенный вклад в развитие **личностных результатов:**

- формирование ответственного отношения к обучению, готовности и спо-

способности, обучающихся к саморазвитию и самообразованию на основе мотивации к обучению и познанию;

- формирование коммуникативной компетентности в общении и сотрудничестве со сверстниками, взрослыми в процессе образовательной, общественно-полезной, учебно-исследовательской, творческой и других видах деятельности.

В части развития **предметных результатов** наибольшее влияние изучение курса оказывает на:

- формирование знаний, умений и навыков при решении задач информатики и программирования разных видов;
- приобретение опыта использования информационных технологий в индивидуальной и коллективной учебной и познавательной, в том числе проектной деятельности;
- формирование информационной и алгоритмической культуры;
- формирование представления о компьютере как универсальном устройстве обработки информации;
- развитие основных навыков и умений использования компьютерных устройств;
- формирование представления об основных изучаемых понятиях: информация, алгоритм, модель - и их свойства;
- развитие алгоритмического мышления, необходимого для профессиональной деятельности в современном обществе;
- развитие умений составить и записать алгоритм для конкретного исполнителя;
- формирование знаний об алгоритмических конструкциях, логических значениях и операциях;
- знакомство с одним из языков программирования и основными алгоритмическими структурами - линейной, условной и циклической;
- владение системой базовых знаний, отражающих вклад информатики в формирование современной научной картины мира;
- овладение понятием сложности алгоритма, знание основных алгоритмов обработки числовой и текстовой информации, алгоритмов поиска и сортировки;
- владение универсальным языком программирования высокого уровня Python, представлениями о базовых типах данных и структурах данных; умением использовать основные управляющие конструкции;
- владение навыками и опытом разработки программ в среде программирования Python, включая тестирование и отладку программ; владение элементарными навыками формализации прикладной задачи и документирования программ;
- формирование умения работать с библиотеками программ;
- получение опыта использования компьютерных средств представления и анализа данных.

В результате изучения дисциплины учащиеся должны **знать:**

- основные типы алгоритмов;

- иметь представление о структуре программы, основы программирования на языках высокого уровня;
- базовые алгоритмические конструкции;
- содержание этапов разработки программы: алгоритмизация-кодирование-отладка- тестирование;
- дополнительные возможности языка Python для выражения различных алгоритмических ситуаций;
- алгоритмы и программы на языке Python решения простых, сложных и нестандартных задач в математической области;
- основы разработки простых игр в системе программирования Python.

Учащиеся должны **уметь**:

- записывать алгоритмические структуры на языке программирования Python;
- использовать Python для решения задач из области математики, физики;
- строить алгоритмы методом последовательного уточнения (сверху вниз), изображать эти алгоритмы в виде блок-схем;
- использовать основные алгоритмические приемы при решении математических задач;
- решать простые, сложные и нестандартные задачи;
- создавать простые игры;
- анализировать текст чужих программ, находить в них неточности, оптимизировать алгоритм, создавать собственные варианты решения.

## ТЕМАТИЧЕСКОЕ ПЛАНИРОВАНИЕ

№ п/п	Тема	Количество часов
1.	Знакомство с Python.	2
2.	Основные типы данных. Операторы ввода/вывода.	13
3.	Арифметические операции и математический модуль.	8
4.	Циклы и условия.	29
5.	Массивы.	18
6.	Функции и процедуры.	7
7.	Строки.	3
8.	Текстовые файлы	4
9.	Контейнеры и структуры в языке Python.	18
10.	Создание оконных приложений на Python.	28
11.	Творческие проекты	10
	<b>ИТОГО:</b>	<b>140</b>

## СОДЕРЖАНИЕ ПРОГРАММЫ

### *Введение в 3D моделирование (2 часа)*

Инструктаж по технике безопасности.

3D технологии. Понятие 3D модели и виртуальной реальности. Область применения и назначение. Стереоскопия.

### *Знакомство с Python (2 часа)*

Инструктаж по технике безопасности. Знакомство с Python.

### *Основные типы данных. Операторы ввода/вывода (13 часов)*

Структура программы. Вывод на экран. Поток ввода/вывода. Создание дружественного интерфейса. Базовые типы данных. Переменные. Идентификаторы. Инициализация. Инструкция присваивания. Ключевые слова. Константы и литералы. Ввод данных в программу.

### *Арифметические операции и математический модуль (8 часов)*

Арифметические операции. Стандартная библиотека. Импорт библиотечных модулей. Математический модуль. Знакомство с другими модулями.

### *Циклы и условия (29 часов)*

Операции сравнения. Условная инструкция if-(elif)-else. Полное и неполное ветвление. Тернарная операция. Логические операции and (&&), or (||), not (!). Инструкция switch. Инструкция while (do-while). Вывод числовых рядов. Накопление сумм и произведений. Инструкция ввода и инструкция if внутри цикла. Арифметика остатков. Вычисление рядов с заданной точностью. Инструкции break и continue. Инструкция for. Функция range(). Вложенные циклы. Вывод прямоугольных таблиц. Разложение числа на простые множители. Поиск простых чисел.

### *Массивы (18 часов)*



Одномерные массивы. Ввод и вывод одномерных массивов. Перебор элементов массива. Поиск максимального и минимального элемента. Пузырьковая сортировка. Сортировка выбором. Сдвиг элементов массива. Ротация и реверс. Определение в заданном массиве серии элементов. Двумерные массивы. Ввод и вывод. Двумерные массивы. Матрицы. Динамические массивы.

### ***Функции и процедуры (7 часов)***

Функции. Параметры и аргументы. Операция return. Вызов функции. Ссылки. Передача аргументов по ссылке и указателю. Процедуры. Тип void. Рекурсивная функция (рекурсия).

### ***Строки (3 часа)***

Контейнер string. (Базовый тип str). Кодировка. Unicode. Ввод и вывод объекта класса string. Методы класса string. Поиск, удаление и вставка подстроки. Строковые потоки.

### ***Текстовые файлы (4 часа)***

Текстовые файлы. Файловые потоки. Файловый ввод/вывод. Указатель чтения. Построчное чтение файла. EOF.

### ***Контейнеры и структуры в языке Python (18 часов)***

Контейнер array. Итераторы. Цикл for по диапазону. Контейнер vector (динамический массив). Эффективность алгоритма. Передача контейнера в функцию. Контейнер deque. Контейнеры-адаптеры Структуры данных стек и дек. Контейнер list. Структура данных очередь. Структура (struct). Класс (class). Поля класса. Объект. Наследование. Полиморфизм. Инкапсуляция. Операция точка «.» и операция стрелка «->». Перегрузка операций. Методы. Конструкторы и деструкторы. Функтор. Предикат. Лямбда-функция. Классы контейнеров set и map, unordered set и unordered map.

### ***Создание оконных приложений на Python (28 часов)***

Основы работы в Tkinter. IDE Qt Creator. Установка и настройка среды Qt5. Класс QWidget. Иерархия классов виджетов. Управление автоматическим размещением элементов (компоновки). Элементы управления: надписи. Элементы управления: кнопки, флажки, элементы ввода, элементы выбора. События и взаимодействие с пользователем.

События клавиатуры и мыши. Таймер. Стили элементов. QStyle. Быстрая разработка приложений. Qt Designer. Язык сценариев QtScript.

### ***Творческие проекты (10 часов)***

Разработка итогового проекта. Защита итогового проекта.

## КАЛЕНДАРНО-ТЕМАТИЧЕСКОЕ ПЛАНИРОВАНИЕ (1-й год обучения)

№ п/п	Тема занятия	К-во часов	Дата по плану	Дата факт.
	<b>Знакомство с Python.</b>	<b>2</b>		
1.	Знакомство с Python.	1		
2.	Знакомство с Python.	1		
	<b>Основные типы данных. Операторы ввода/вывода.</b>	<b>13</b>		
3.	Структура программы. Вывод на экран.	1		
4.	Структура программы. Вывод на экран.	1		
5.	Структура программы. Вывод на экран.	1		
6.	Потоки ввода/вывода.	1		
7.	Потоки ввода/вывода.	1		
8.	Создание дружественного интерфейса.	1		
9.	Создание дружественного интерфейса.	1		
10.	Базовые типы данных. Переменные. Идентификаторы. Инициализация.	1		
11.	Инструкция присваивания. Ключевые слова.	1		
12.	Инструкция присваивания. Ключевые слова.	1		
13.	Константы и литералы. Ввод данных в программу.	1		
14.	Константы и литералы. Ввод данных в программу.	1		
15.	Константы и литералы. Ввод данных в программу.	1		
	<b>Арифметические операции и математический модуль.</b>	<b>8</b>		
16.	Арифметические операции.	1		
17.	Арифметические операции.	1		
18.	Арифметические операции.	1		
19.	Стандартная библиотека. Импорт библиотечных модулей.	1		
20.	Математический модуль.	1		
21.	Математический модуль.	1		
22.	Математический модуль.	1		
23.	Знакомство с другими модулями.	1		
	<b>Циклы и условия.</b>	<b>29</b>		
24.	Операции сравнения. Условная инструкция if-(elif)-else. Полное и неполное ветвление.	1		
25.	Операции сравнения. Условная инструкция if-(elif)-else. Полное и неполное ветвление.	1		

26.	Операции сравнения. Условная инструкция if-(elif)-else. Полное и неполное ветвление.	1		
27.	Тернарная операция.	1		
28.	Тернарная операция.	1		
29.	Логические операции and (&&), or (  ), not (!).	1		
30.	Логические операции and (&&), or (  ), not (!).	1		
31.	Логические операции and (&&), or (  ), not (!).	1		
32.	Инструкция switch.	1		
33.	Инструкция switch.	1		
34.	Инструкция while (do-while).	1		
35.	Вывод числовых рядов.	1		
36.	Накопление сумм и произведений.	1		
37.	Инструкция ввода и инструкция if внутри цикла.	1		
38.	Инструкция ввода и инструкция if внутри цикла.	1		
39.	Инструкция ввода и инструкция if внутри цикла.	1		
40.	Арифметика остатков.	1		
41.	Арифметика остатков.	1		
42.	Вычисление рядов с заданной точностью.	1		
43.	Вычисление рядов с заданной точностью.	1		
44.	Инструкции break и continue.	1		
45.	Инструкция for. Функция range().	1		
46.	Инструкция for. Функция range().	1		
47.	Инструкция for. Функция range().	1		
48.	Вложенные циклы. Вывод прямоугольных таблиц.	1		
49.	Вложенные циклы. Вывод данных по образцу.	1		
50.	Вложенные циклы. Вывод данных по образцу.	1		
51.	Разложение числа на простые множители.	1		
52.	Поиск простых чисел.	1		
	<b>Массивы.</b>	<b>18</b>		
53.	Одномерные массивы. Ввод и вывод одномерных массивов.	1		
54.	Одномерные массивы. Ввод и вывод одномерных массивов.	1		
55.	Перебор элементов массива. Поиск максимального и минимального элемента.	1		
56.	Перебор элементов массива. Поиск максимального и минимального элемента.	1		
57.	Пузырьковая сортировка.	1		
58.	Сортировка выбором.	1		

59.	Сдвиг элементов массива. Ротация и реверс.	1		
60.	Определение в заданном массиве серии элементов.	1		
61.	Определение в заданном массиве серии элементов.	1		
62.	Определение в заданном массиве серии элементов.	1		
63.	Двумерные массивы. Ввод и вывод.	1		
64.	Двумерные массивы. Матрицы.	1		
65.	Двумерные массивы. Матрицы.	1		
66.	Двумерные массивы. Матрицы.	1		
67.	Двумерные массивы. Матрицы.	1		
68.	Динамические массивы.	1		
69.	Динамические массивы.	1		
70.	Динамические массивы.	1		
	<b>Функции и процедуры.</b>	<b>7</b>		
71.	Функции. Параметры и аргументы. Операция return. Вызов функции.	1		
72.	Функции. Параметры и аргументы. Операция return. Вызов функции.	1		
73.	Ссылки. Передача аргументов по ссылке и указателю.	1		
74.	Процедуры. Тип void.	1		
75.	Рекурсивная функция (рекурсия).	1		
76.	Рекурсивная функция (рекурсия).	1		
77.	Рекурсивная функция (рекурсия).	1		
	<b>Строки.</b>	<b>3</b>		
78.	Контейнер string. (Базовый тип str). Кодировка. Unicode.	1		
79.	Ввод и вывод объекта класса string. Методы класса string.	1		
80.	Поиск, удаление и вставка подстроки. Строковые потоки.	1		
	<b>Текстовые файлы.</b>	<b>4</b>		
81.	Текстовые файлы. Файловые потоки. Файловый ввод/вывод. Указатель чтения.	1		
82.	Текстовые файлы. Файловые потоки. Файловый ввод/вывод. Указатель чтения.	1		
83.	Текстовые файлы. Файловые потоки. Файловый ввод/вывод. Указатель чтения.	1		
84.	Построчное чтение файла. EOF.	1		
	<b>Контейнеры и структуры в языке Python.</b>	<b>18</b>		
85.	Контейнер array. Итераторы. Цикл for по диапазону.	1		

86.	Контейнер array. Итераторы. Цикл for по диапазону.	1		
87.	Контейнер vector (динамический массив). Эффективность алгоритма.	1		
88.	Контейнер vector (динамический массив). Эффективность алгоритма.	1		
89.	Передача контейнера в функцию.	1		
90.	Контейнер deque. Контейнеры-адаптеры Структуры данных стек и дек.	1		
91.	Контейнер deque. Контейнеры-адаптеры Структуры данных стек и дек.	1		
92.	Контейнер list. Структура данных очередь.	1		
93.	Контейнер list. Структура данных очередь.	1		
94.	Структура (struct). Класс (class). Поля класса. Объект. Наследование. Полиморфизм. Инкапсуляция.	1		
95.	Структура (struct). Класс (class). Поля класса. Объект. Наследование. Полиморфизм. Инкапсуляция.	1		
96.	Операция точка «.» и операция стрелка «->». Перегрузка операций.	1		
97.	Операция точка «.» и операция стрелка «->». Перегрузка операций.	1		
98.	Методы. Конструкторы и деструкторы. Функтор. Предикат.	1		
99.	Методы. Конструкторы и деструкторы. Функтор. Предикат.	1		
100.	Лямбда-функция.	1		
101.	Лямбда-функция.	1		
102.	Классы контейнеров set и map, unordered set и unordered map.	1		
	<b>Создание оконных приложений на Python.</b>	<b>28</b>		
103.	Основы работы в Tkinter.	1		
104.	Основы работы в Tkinter.	1		
105.	IDE Qt Creator. Установка и настройка среды Qt5.	1		
106.	Класс QWidget. Иерархия классов виджетов.	1		
107.	Управление автоматическим размещением элементов (компоновки).	1		
108.	Управление автоматическим размещением элементов (компоновки).	1		
109.	Элементы управления: надписи.	1		
110.	Элементы управления: кнопки.	1		
111.	Элементы управления: флажки.	1		

112.	Элементы управления: элементы ввода.	1		
113.	Элементы управления: элементы выбора.	1		
114.	События и взаимодействие с пользователем.	1		
115.	События клавиатуры и мыши.	1		
116.	События клавиатуры и мыши.	1		
117.	События клавиатуры и мыши.	1		
118.	Таймер.	1		
119.	Таймер.	1		
120.	Стили элементов. QStyle.	1		
121.	Стили элементов. QStyle.	1		
122.	Быстрая разработка приложений. Qt Designer.	1		
123.	Быстрая разработка приложений. Qt Designer.	1		
124.	Быстрая разработка приложений. Qt Designer.	1		
125.	Быстрая разработка приложений. Qt Designer.	1		
126.	Быстрая разработка приложений. Qt Designer.	1		
127.	Язык сценариев QtScript.	1		
128.	Язык сценариев QtScript.	1		
129.	Язык сценариев QtScript.	1		
130.	Язык сценариев QtScript.	1		
	<b>Творческие проекты</b>	<b>10</b>		
131.	Разработка итогового проекта.	1		
132.	Разработка итогового проекта.	1		
133.	Разработка итогового проекта.	1		
134.	Разработка итогового проекта.	1		
135.	Разработка итогового проекта.	1		
136.	Разработка итогового проекта.	1		
137.	Разработка итогового проекта.	1		
138.	Разработка итогового проекта.	1		
139.	Защита итогового проекта.	1		
140.	Защита итогового проекта.	1		
	<b>ИТОГО:</b>	<b>140</b>		

## УЧЕБНО-МЕТОДИЧЕСКОЕ И МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА

### Учебно-методическое обеспечение

Предлагаемая программа составлена в соответствии с требованиями к курсу «Программирование на Python» в соответствии с ФГОС среднего (полного) общего образования. В состав УМК, кроме школьных учебников по информатике, также входят:

- компьютерный практикум в электронном виде с комплектом электронных учебных средств, размещенный на сайте: <http://kpolyakov.spb.ru/school/probook.htm>
- материалы для подготовки к итоговой аттестации по информатике в форме ОГЭ, размещенные на сайте <http://kpolyakov.spb.ru/school/oge.htm>;
- методическое пособие для учителя;
- комплект Федеральных цифровых информационно-образовательных ресурсов (далее ФЦИОР), помещенный в коллекцию ФЦИОР (<http://www.fcior.edu.ru>);
- сетевая методическая служба авторского коллектива для педагогов на сайте издательства <http://metodist.lbz.ru/authors/informatika/7/>.

Учитель может изменять предлагаемую авторскую учебную программу с учетом специфики региональных условий, образовательного учреждения и уровня подготовленности учеников:

- вносить изменения в порядок изучения материала;
- перераспределять учебное время;
- вносить изменения в содержание изучаемой темы;
- дополнять требования к уровню подготовки учащихся и т.д.

Практикум для учащихся, представляемый в электронном виде, позволяет расширить используемый теоретический, задачный и проектный материал.

Для подготовки к итоговой аттестации по информатике предлагается использовать материалы, размещенные на сайте <http://kpolyakov.spb.ru/school/oge.htm>.

Для реализации учебного курса «Программирование на Python» необходимо наличие компьютерного класса в соответствующей комплектации:

### Требования к комплектации компьютерного класса

Наиболее рациональным с точки зрения организации деятельности детей в школе является установка в компьютерном классе 13–15 компьютеров (рабочих мест) для школьников и одного компьютера (рабочего места) для педагога.

Предполагается объединение компьютеров в локальную сеть с возможностью выхода в Интернет, что позволяет использовать сетевые цифровые образовательные ресурсы.

Минимальные требования к техническим характеристикам компьютера следующие:

- процессор – не ниже Celeron с тактовой частотой 2 ГГц;

- оперативная память – не менее 1 Гб;
- жидкокристаллический монитор с диагональю не менее 15 дюймов;
- жесткий диск – не менее 40 Гб;
- клавиатура;
- мышь;
- аудио-карта и акустическая система (наушники или колонки).

Периферийное оборудование в кабинете информатики:

- принтер на рабочем месте учителя;
- сканер на рабочем месте учителя;
- интерактивный вычислительный комплекс на рабочем месте учителя.

### **Требования к программному обеспечению компьютеров**

На компьютерах, которые расположены в кабинете информатики, должна быть установлена операционная система Windows 7+ или Linux, а также необходимое программное обеспечение:

- текстовый редактор (Notepad++) и текстовый процессор (MS Word или OpenOffice Writer);
- табличный процессор (MS Excel или OpenOffice Calc);
- средства для работы с базами данных (MS Access или OpenOffice Base);
- среда программирования Python 3+, интегрированная среда разработки или среда разработки PyScripter.

### **СПИСОК ЛИТЕРАТУРЫ**

1. Python. Подробный справочник Дэвида М. Бизли – книга со справочной информацией о языке Python и модулях стандартной библиотеки.
2. Python. Справочник Марка Лутца. Справочник по наиболее часто используемым функциям и модулям.
3. Долинский М.С. Решение сложных и олимпиадных задач по программированию - Учебное пособие - М.: Просвещение, 2006.
4. Лутц М. «Изучаем Питон», Санкт-Петербург: Символ, 2011.
5. Майк МакГрат «Программирование на Python для начинающих». - Эксмо, 2015.
6. Поляков К.Ю., Еремин Е.А. Информатика. 10-й класс. Углубленный уровень. В двух частях. М.: Бином, 2013.
7. Поляков К.Ю., Еремин Е.А. Информатика. 11-й класс. Углубленный уровень. В двух частях. М.: Бином, 2013.
8. Поляков К.Ю., Еремин Е.А. «Информатика», углубленный уровень, М.: Бином. Лаборатория Знаний, 2013.
9. Россум Г., Дж. Дрейк Ф.Л., Откидач Д.С. Язык программирования Python. 2001.
10. Самоучитель Python. Дмитрий Мусин. 2019 (pythonworld.ru)
11. Сэнд У., Сенд К. «Занимательное программирование на языке Python». -М.: Просвещение, 2016.



12. Федоров Д. Ю. Основы программирования на примере языка Python. // Учебное пособие. - Санкт-Петербург: 2016.