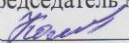


Муниципальное бюджетное общеобразовательное учреждение  
средняя общеобразовательная школа №4  
Невьянского городского округа

г. Невьянск, ул. Долгих, 69

Согласовано

Председатель методического совета  
 (Колташова С.Н.)

протокол № 1  
от «30» августа 2024 г.



тел. 2-17-36, факс 4-20-09

Утверждаю:

Директор ОУ

(Колногоров С.Г.)

приказ № 221 од  
от «30» августа 2024 г.

ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ  
ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА

«Программирования на языке Python»

Составила: Сивкова О.Л.

Невьянск 2024

## Содержание

<b>1. Пояснительная записка .....</b>	<b>3</b>
<b>2. Структура программы.....</b>	<b>9</b>
2.1. Объем программы.....	9
2.2. Учебный план .....	10
2.3. Содержание учебного плана .....	12
<b>3. Условия реализации программы.....</b>	<b>14</b>
3.1. Материально-техническое оснащение.....	14
3.2. Методическое обеспечение реализации программы.....	15
<b>4. Список литературы.....</b>	<b>17</b>
4.1. Список литературы, используемой педагогом.....	17
4.2. Список рекомендуемой литературы для обучающихся.....	18

## 1. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

**Актуальность и направленность программы.** В настоящее время прослеживается острый кризис квалифицированных кадров в различных сферах рынка труда. Работодатели, хоть и обращают внимание на область предметных знаний, но все же отдают предпочтение универсальным навыкам. Подобная позиция связана с постоянно изменяющимися условиями труда, технологическим прогрессом. Подобные универсальные навыки (способность работать в команде, многозадачность, творческий подход, критическое мышление, продуктивное мышление, нацеленность на результат) вырабатываются в процессе профессионального становления.

Направленность дополнительной образовательной 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 навы

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 языке 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.

Наш выбор Python для преподавания призван помочь обучающимся сделать первые шаги по одному из современных и перспективных путей развития IT-индустрии.

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.

Данная программа относится к углубленному уровню, так как направлена на овладение знаниями и развитие навыков, она дает возможность ребенку погрузиться в атмосферу дополнительного образования, попробовать себя в новом лично значимом виде деятельности. Родителям дает возможность разобраться с логикой дополнительного образования и наметить линию индивидуального развития своего ребенка.

Программа базируется на решении кейсов и проектной деятельности. Для повышения качества образования данные кейсы основываются на реальных ситуациях и проблемах потенциальных работодателей. Это дает возможность ранней профессиональной ориентации обучающихся.

По форме организации содержания и процесса педагогической деятельности программа «Программирование на языке Python» интегрированная, так как объединяет в целое отдельные образовательные области на основе единства математики и информатики. Главной целью является приобретение обучающимся теоретических знаний и практических навыков и компетенций, достаточных для дальнейшего успешного обучения в системе непрерывного образования: школа – дообразование – 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

языке Python» направлена на устранение данного пробела.

**Адресат программы.** Программа ориентирована на дополнительное образование обучающихся 14-17 лет, интересующихся программированием.

**Срок и этапы реализации программы.** Программа рассчитана на 1 год обучения - 68 академических часа.

Основной формой обучения являются групповые занятия. В основе образовательного процесса лежит проектный подход.

**Режим занятий:** 1 раз по 2 часа в неделю.

Продолжительность 1 занятия: 2 академических часа.

Структура двухчасового занятия:

- 40 минут – рабочая часть;
- 10 минут – перерыв (отдых);
- 40 минут – рабочая часть.

Основная форма работы теоретической части – лекционные занятия в группах до 15 человек. Практические задания планируется выполнять индивидуально, в парах и в малых группах. Занятия проводятся в виде бесед, семинаров, лекций: для наглядности изучаемого материала используется различный мультимедийный материал – презентации, видеоролики.

### **Цель программы**

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

### **Задачи программы**

Достижение поставленной цели складывается из выполнения следующих задач:

Обучающие:

1. Познакомить с базовыми понятиями основ программирования.
2. Познакомить с особенностями синтаксиса языка программирования Python.
3. Познакомить с принципами объектно-ориентированного программирования.
4. Сформировать навык правильного оформления кода.
5. Сформировать навык использования итеративного подхода при решении различных задач.
6. Формировать навык решения задач и разработки проектов с помощью Python.

7. Формировать навык разработки игр с помощью языка программирования Python.

8. Формировать навык тестирования и оптимизации созданных проектов.

Развивающие:

1. Формировать и развивать логическое, алгоритмическое, критическое, креативное и пространственное мышление.

2. Формировать навык публичного выступления и презентации.

3. Формировать навык планирования деятельности и основ тайм-менеджмента.

4. Формировать и расширять словарный запас на английском языке.

5. Формировать функциональную грамотность.

6. Расширять кругозор, развивать память, внимание, творческое воображение, абстрактно-логические и наглядно-образные виды мышления и типы памяти, основные мыслительные операции и свойства внимания.

7. Совершенствовать диалогическую речь учащихся: уметь слушать собеседника, понимать вопросы, смысл знаний, уметь задавать вопросы, отвечать на них.

Воспитательные:

1. Воспитывать у учащихся потребность в сотрудничестве, взаимодействии со сверстниками, умение подчинять свои интересы определенным правилам, развивать эмоциональный интеллект.

2. Формировать умение давать качественную обратную связь и реагировать на нее.

3. Формировать информационную культуру.

В результате освоения дисциплины обучающийся должен:

**уметь:**

- работать в среде программирования;
- реализовывать построенные алгоритмы в виде программ на конкретном языке программирования;
- настраивать рабочую среду Python;

**знать:**

- этапы решения задачи на компьютере;
- типы данных;
- базовые конструкции изучаемых языков программирования;
- принципы структурного и модульного программирования;
- принципы объектно-ориентированного программирования;
- основы создания приложений в Python;

- инструментальные средства Python.

**Методы:** кейс-метод, проектная деятельность.

**Формы работы:**

- практическое занятие;
- занятие – соревнование;
- консультация;
- самостоятельная работа.

**Виды учебной деятельности:**

- анализ проблемных учебных ситуаций;
- построение гипотезы на основе анализа имеющихся данных;
- поиск необходимой информации;
- выполнение практических работ;
- подготовка выступлений и докладов с использованием разнообразных источников информации;
- публичное выступление.

**Требования к результатам освоения программы:**

Результат (освоенные компетенции)	Основные показатели оценки результата	Формы и методы контроля и оценки
Личностные компетенции	умение работать в команде: работа в общем ритме, эффективное распределение задач и др.	проектная деятельность в команде, презентации и защиты проектов
	формирование высокого познавательного интереса учащихся	проектная деятельность
	формирование критического мышления	проектная деятельность
	проявление технического мышления, познавательной деятельности, творческой инициативы, самостоятельности	проектная деятельность, выполнение кейсов
Метапредметные компетенции	умение ориентироваться в информационном пространстве, продуктивно использовать техническую литературу для поиска сложных решений	проектная деятельность, презентации и защиты проектов, выполнение кейсов
	способность творчески решать технические задачи	выполнение кейсов
	готовность и способность применения теоретических знаний по физике, информатике для решения задач в реальном мире	проектная деятельность, выполнение кейсов

	способность правильно организовывать рабочее место и время для достижения поставленных целей	выполнение практических заданий
	Знание основ ТРИЗ, навыки публичного выступления и презентации результатов, навык генерации идей	выполнение практических заданий
Предметные компетенции	знание основ и принципов программирования	- проектная деятельность, выполнение кейсов; - участие в конференциях, выставках, конкурсах, соревнованиях и т.п.; - выполнение практических заданий
	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	

### Формы подведения итогов реализации программы

Основной формой подведения итогов дополнительной общеразвивающей программы «Язык программирования языке Python» является решение задач, проектная деятельность (разработка и отладка программ на языке программирования Python).

Критерии оценки защиты проекта:

Критерии	Ф.И.О. обучающегося, тема проекта		
Постановка цели, проблематизация: 1. Проектная работа соответствует цели и отвечает на проблемные вопросы – 3 балла 2. Проектная работа соответствует цели и отвечает на некоторые проблемные вопросы – 2 балла 3. Проектная работа не совсем точно отражает цель проекта и его проблемные вопросы – 1 балл			
Формулировка задач проекта: 1. Поставленные задачи ведут к достижению цели проекта – 3 балла 2. Не все задачи ведут к достижению цели проекта – 2 балла 3. Представленные задачи не ведут к достижению цели проекта – 1 балл			
Результаты работы:			



<p>1. Результаты работы, представленные при помощи компьютерных средств, оформлены в соответствие с правилами – 3 балла</p> <p>2. Результаты работы, представленные при помощи компьютерных средств, содержат незначительные ошибки оформления – 2 балла</p> <p>3. Результаты работы, представленные при помощи компьютерных средств, содержат значительные ошибки в оформлении – 1 балл</p>		
<p>Выступление:</p> <p>1. Устное выступление участника логично, отсутствуют грамматические и лексические ошибки – 3 балла</p> <p>2. Устное выступление участника логично, присутствуют незначительные грамматические и лексические ошибки, не мешающие пониманию материала – 2 балла</p> <p>3. Устное выступление участника не всегда логично, присутствуют грамматические и лексические ошибки, которые затрудняют понимание – 1 балл</p>		
<p>Соответствие выступления и презентации:</p> <p>1. Выступление не повторяет текст презентации или публикации – 3 балла</p> <p>2. Выступление частично повторяет текст презентации или публикации – 2 балла</p> <p>3. Выступление полностью повторяет текст презентации или публикации – 1 балл</p>		
<p>Ответы на вопросы:</p> <p>1. В ходе устного выступления даны ответы на все вопросы – 3 балла</p> <p>2. В ходе устного выступления даны ответы на некоторые вопросы – 2 балла</p> <p>3. Обучающийся затруднялся давать правильные ответы на вопросы – 1 балл</p>		

## 2. СТРУКТУРА ПРОГРАММЫ

### 2.1. Объем программы

Год обучения	Уровень	Кол-во часов
1 год	вводный уровень	68

### 2.2. Учебный план

№ п/п	Наименование раздела, темы	Количество часов			Формы аттестации/контроля
		Всего	Теория	Практика	
1.	<b>Раздел 1. Введение в Python</b>	<b>4</b>	<b>3</b>	<b>1</b>	<b>Тест по модулю</b>
	1.1. Инструкции и структура программы.	2	2	0	Устный опрос
	1.2. Установка Python. Ввод и вывод данных.	2	1	1	Решение задач
2.	<b>Раздел 2. Типы данных и операции</b>	<b>12</b>	<b>6</b>	<b>6</b>	<b>Тест по модулю</b>
	2.1. Переменные. Операторы.	4	2	2	Тест по теме
	2.2. Типы данных. Преобразование типов. Различия типов данных. Базовые функции (abs, round, int, math)	8	4	4	Решение задач
3.	<b>Раздел 3. Инструкции и синтаксис</b>	<b>14</b>	<b>7</b>	<b>7</b>	<b>Тест по модулю</b>
	3.1. Условный оператор if, Альтернативный условный оператор elif, else.	4	2	2	Решение задач
	3.2. Циклы while. Цикл for. Обработка включений. Функция random. Случайные числа.	10	5	5	Решение задач с использованием циклов
4.	<b>Раздел 4. Функции и модули в программировании</b>	<b>18</b>	<b>9</b>	<b>9</b>	<b>Тест по модулю</b>
	4.1. Встроенные и пользовательские функции. Создание функций. Функции, возвращающие результат.	6	3	3	Тест по теме
	4.2. Строки, последовательность символов. Доступ по индексу. Длина строки и	12	6	6	Решение задач с использованием строк

	отрицательные индексы. Преобразование типов. Применение цикла для обхода строки. Срезы строк. Сравнение строк.				
5.	<b>Раздел 5. Сложные типы данных</b>	<b>12</b>	<b>7</b>	<b>9</b>	<b>Тест по модулю</b>
	5.1. Списки. Срезы списков. Решение задач со списками	6	1	5	Решение задач
	5.2. Матрицы	2	1	1	Решение задач
	5.3. Словари	2	1	1	Решение задач
	5.4. Множества в языке Python	2	1	1	Решение задач
	<b>Раздел 6. Написание и отладка программ. Самостоятельная работа</b>	<b>10</b>	<b>2</b>	<b>8</b>	<b>Защита проекта</b>
	<b>Итого</b>	<b>68</b>	<b>15</b>	<b>53</b>	

### 2.3. Содержание учебного плана

#### 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.

## **в. 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.

Практическая работа: Практическое закрепление знаний по условным операторам. Создание программ.

### **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».

### **3. УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ**

#### **3.1. Материально-техническое оснащение**

##### **Компьютерное оборудование:**

- персональный компьютер – 15 шт.
- моноблок – 1 шт.
- маршрутизатор – 1 шт.
- коммутатор – 1 шт.

##### **Программное обеспечение:**

- ОС Windows
- Python
- Pyzo
- Pygame
- Текстовый редактор Блокнот
- KompoZer
- Microsoft Power point

##### **Презентационное оборудование:**

- проектор – 1 шт.
- ноутбук – 1 шт.

##### **Дополнительное оборудование:**

- учительский стол – 1 шт.
- учительский стул – 1 шт.
- парты двухместные – 8 шт.
- стулья ученические – 16 шт.

#### **3.2. Методическое обеспечение реализации программы**

Методическое обеспечение программы включает приёмы и методы организации образовательного процесса, дидактические материалы, техническое оснащение занятий.

Методы и приёмы организации образовательного процесса при реализации программы:

Словесные методы: объяснение, беседа, комментированное чтение, рассказ.

Практические методы: работа с текстом, составление планов, работа над

проектами, выполнение творческих заданий: составление кроссвордов, сочинение загадок, рассказов, выпуск бюллетеней, сборников или альбомов с творческими работами и проектами.

Игровые методы: фантазирование, театральная импровизация, живая наглядность.

Наглядные методы: показ видеоматериалов, посещение выставок, проведение экскурсий.

Виды дидактических материалов, используемые при реализации программы:

Для обеспечения наглядности и доступности изучаемого материала педагог использует наглядные пособия следующих видов:

- схематические или символические (оформленные стенды и планшеты, таблицы, схемы, рисунки, графики, плакаты, диаграммы, чертежи, шаблоны и т.п.);
- картинные (иллюстрации, слайды, фотоматериалы и др.);
- звуковые (аудиозаписи);
- смешанные (видеозаписи, учебные кинофильмы и т.д.);
- дидактические пособия (карточки, рабочие тетради, раздаточный материал, вопросы и задания для опроса, тесты, практические задания, упражнения и др.);
- компьютерные программы в электронном виде (компьютеры с программами, CD, флеш-носители);
- учебные пособия, журналы, книги, Интернет-ресурсы.

При реализации программы с целью повышения качества и эффективности процесса обучения применяются современные эффективные технологии обучения, ориентированные не на накопление знаний, а на организацию активной деятельности обучающихся:

- технологии проектной деятельности;
- компьютерные (информационные) технологии;
- технологии учебно-игровой деятельности (моделирование);
- технологии коммуникативно-диалоговой деятельности;
- модульные технологии;
- квест-технологии;
- технологии личностно-ориентированного обучения;
- кейс-технологии.

Информационные технологии используются в различных видах деятельности:

- при подготовке и проведении занятий;
- для создания авторских мультимедийных презентаций;
- в рамках индивидуальной и групповой проектной деятельности;



- для самостоятельной работы;
- для накопления демонстрационных материалов к занятиям (видеоматериалы, таблицы, презентации, карты);

Одним из основных методов является метод проектного обучения, так как он является неотъемлемой частью учебного процесса. Исходный лозунг основателей системы проектного обучения – «Все из жизни, все для жизни». Обучение строится на активной основе, через практическую деятельность ученика, ориентируясь на его личный интерес и практическую востребованность полученных знаний в дальнейшей жизни, обучающийся имеет возможность через проектную деятельность освоить получаемые знания. Проекты представляются в виде готовых программ, презентаций проектов, научных докладов, моделей, демонстрации видеофильма. Достоинствами проектной деятельности являются:

- Уметь работать в коллективе;
- Брать ответственность за выбор решения на себя;
- Разделять ответственность с другими;
- Предоставлять ребенку свободу выбора темы, методов работы;
- Понимание каждым обучающимся важности работы и др.

## 4. СПИСОК ЛИТЕРАТУРЫ

### 4.1. Список литературы, используемой педагогом

#### Основная:

1. Клейнберг Дж. Алгоритмы: разработка и применение. СПб: Питер, 2016. - 800 с.
2. Златопольский Д. М. Основы программирования на языке Python. – М.: ДМК Пресс, 2017. – 284 с.: ил.
3. Бхаргава А. Грокаем алгоритмы. Иллюстрированное пособие для программистов и любопытствующих. - СПб.: Питер, 2017. - 288 с.

#### Дополнительная:

1. Луц М. Изучаем Python. СПб: Симво-плюс, 2011. - 1280 с.
2. Паронджанов В.Д. Учись писать, читать и понимать алгоритмы. М: Ямб, 2012. – 520 с.
4. Златопольский Д.М. Сборник задач по программированию. СПб: БХВ-Петербург, 2011. - 295 с.
5. ООП на Python: концепции, принципы и примеры реализации [Электронный ресурс]. – Режим доступа: <https://proglib.io/p/python-oop/>

### 4.2. Список рекомендуемой литературы для обучающихся

#### Основная:

1. Свейгар.Эл. Учим python, делая крутые игры М: Эксмо, 2018. – 416 с.
2. Семакин И.Г. Основы алгоритмизации и программирования. М.: Академия, 2016. - 304 с.

#### Дополнительная:

6. Луц М. Изучаем Python. СПб: Симво-плюс, 2011. - 1280 с.
7. Уроки по Python для начинающих [Электронный ресурс]. – Режим доступа: <https://pythonru.com/uroki/vvedenie-uroki-po-python-dlja-nachinajushhih>
8. Алгоритмизация. Программирования Python 3 [Электронный ресурс]. – Режим доступа: <https://stepik.org/course/7215/promo>