

Муниципальное бюджетное учреждение дополнительного образования  
«Центр дополнительного образования детей»

ПРИНЯТА

педагогическим советом,  
протокол №1 от 30.05.2022 г.



УТВЕРЖДАЮ  
И.о. директора МБУ ДО ЦДОД  
Е. А. Верхотурова  
Приказ № 089 от 23.08.2022 г.

ДОПОЛНИТЕЛЬНАЯ ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА  
«ПРОГРАММИРОВАНИЕ»  
технической направленности  
базового уровня  
возраст обучающихся 13-17 лет  
срок реализации 1 год

**Бондаренко Ольга Владимировна,**  
педагог дополнительного образования

## СОДЕРЖАНИЕ

1. Пояснительная записка .....	3
2. Учебный план .....	5
3. Содержание программы .....	7
4. Планируемые результаты освоения программы.....	11
5. Оценочные материалы образовательной программы....	11
6. Методическое обеспечение.....	12
7. Материально-техническое обеспечение.....	12
8. Список литературы.....	12

## **I. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

Дополнительная общеразвивающая программа «Программирование» относится к технической направленности, является модернизированной. Данная программа составлена на основе образовательной программы «Мастерская программирования на языке PYTHON», составитель – Митрофанов Иван Юрьевич, педагог дополнительного образования МОУ «Левобережная средняя школа города Тутаева».

### **Новизна, актуальность, педагогическая целесообразность программы**

Программа погружает учащихся в мир алгоритмов, где все строится на логических рассуждениях, выстраивании продуманных действий, которые обязательно приводят к результату, в мир четких команд языков программирования.

Запросы нашего времени таковы, что все научные отрасли, отрасли жизни основываются на использовании компьютерной техники для вычислений, создания баз данных, написания научных работ, построения графиков и схем. Требуется большое количество программных продуктов, отвечающих тем или иным требованиям. Данная программа открывает детям возможность попробовать себя в создании программных продуктов, решать задачи с помощью языка программирования Python.

Выбор языка обусловлен тем, что синтаксис языка достаточно прост и интуитивно понятен, а это понижает порог вхождения и позволяет сосредоточиться на логических и алгоритмических аспектах программирования, ознакомиться с фундаментальными понятиями алгоритмизации и программирования на доступном уровне. При этом Python является очень востребованным языком; он отлично подходит для знакомства с различными современными парадигмами программирования и активно применяется в самых разных областях от разработки веб-приложений до машинного обучения. Научившись программировать на языке Python, обучающиеся получают мощный и удобный инструмент для решения как учебных, так и прикладных задач. Вместе с тем чистота и ясность его конструкций позволит обучающимся с лёгкостью выучить любой другой язык программирования.

Отличительная особенность программы в том, что она является практико-ориентированной. Изучение основных принципов программирования невозможно без регулярной практики написания программ на каком-либо языке. Освоение подростками IT-навыков происходит в процессе практической и самостоятельной работы. Это позволяет обучающимся получать не только теоретические знания в области программирования, но и овладевать IT-технологиями, что поможет им самоопределиться и выстроить траекторию личностного роста в современном информационном обществе.

Педагогическая целесообразность Программы заключается в том, что занятие программированием, даёт необычайно сильный толчок для развития интеллекта обучающихся, формирует их логическое мышление, вырабатывает привычку аккуратной и систематической работы.

В большей степени обучение построено на использовании информационно-коммуникативных технологий. Данная технология применяется как средство обучения: занятия проводятся в компьютерном классе с использованием современных электронных учебных материалов.

**Цель программы** - формирование алгоритмического стиля мышления и

умения решать задачи при помощи языка программирования.

### **Задачи программы:**

#### **Обучающие:**

- сформировать навыки разработки эффективных алгоритмов и программ на языке программирования Python;
- сформировать у обучающихся умения анализировать содержание и структуру исходных данных;
- научить строить логическую последовательность действий для решения задач.

#### **Развивающие:**

- развить способность определять и формулировать цель своей деятельности, выстраивать план ее достижения;
- развить способность самостоятельно оценивать свою деятельность и вносить коррективы с учетом сделанных ошибок;
- развить умение выстраивать словесный алгоритм, высказывать и отстаивать свою точку зрения.

#### **Воспитательные:**

- создание условия для воспитания этики группового сотрудничества, взаимоуважения;
- создание условия для формирования мотивации, ответственного отношения к обучению и развитию.

#### **Формы обучения:**

В рамках программы применяются следующие формы обучения: лекции, беседы, семинары, занятия с применением информационных технологий, зачеты, практические занятия, где организуется работа учащихся индивидуально, парами или в малых группах.

Во время практической работы, когда каждый учащийся работает самостоятельно, он придерживается индивидуального темпа, так как все дети имеют различные способности и отличаются скоростью мыслительных процессов. Учащиеся, выполняющие работу быстрее, получают больший объем заданий, либо задания повышенного уровня сложности.

#### **Характеристика контингента обучающихся:**

На программу приглашаются дети в возрасте 13-17 лет. Занятия проводятся один раз в неделю по два академических часа. Образовательная программа «Программирование» рассчитана на 1 год обучения, 72 академических часа.

#### **Условия приема обучающихся в программу:**

Дети принимаются в программу по их желанию, стремлению попробовать себя в этой сфере деятельности.

**Уровень программы** – базовый предполагает использование и реализацию таких форм организации материала, которые допускают освоение специализированных знаний и языка, гарантированно обеспечивает трансляцию общей и целостной картины в рамках содержательно-тематического направления общеразвивающей программы.

#### **Количество обучающихся**

Состав группы не более 10 человек (по количеству компьютеров в кабинете), чтобы каждый учащийся мог самостоятельно работать.

## II. УЧЕБНЫЙ ПЛАН

№	Названия раздела/темы	Количество часов			Формы аттестации и контроля
		Всего	Теория	Практика	
<b>1.</b>	<b>Введение в язык программирования Python</b>	<b>5</b>	<b>2</b>	<b>4</b>	
1.1.	Вводное занятие. Понятия «алгоритм» и «программа». Начальное знакомство с языком. Инструктаж	2	1	1	Тест.
1.2.	Элементы языка. Структура программы. Операции и переменные. Типы данных	2	1	1	
1.3.	Ввод и вывод данных на языке Python. Ввод и отладка программ в среде	1	-	1	Текущий контроль. Практикум
<b>2.</b>	<b>Линейные алгоритмы и их реализация на языке Python</b>	<b>9</b>	<b>2</b>	<b>7</b>	
2.1.	Линейные алгоритмы целочисленных данных и их реализация на Python	2	1	1	
2.2.	Решение задач по теме «Обработка целочисленных данных»	1	-	1	Текущий контроль. Практикум
2.3.	Линейные алгоритмы вещественных чисел и их реализация на Python	2	1	1	
2.4.	Решение задач по теме «Обработка вещественных чисел»	2	-	2	Текущий контроль. Практикум
2.5.	Самостоятельная работа по темам раздела «Линейные алгоритмы и их реализация в среде Python»	2	-	2	Текущий контроль. Самостоятельная работа
<b>3.</b>	<b>Разветвляющиеся алгоритмы и их реализация на языке Python</b>	<b>14</b>	<b>4</b>	<b>10</b>	
3.1.	Алгоритм «выбор», графическое изображение, полное и неполное ветвление	2	1	1	
3.2.	Реализация алгоритма «выбор» на Python. Примеры решения задач	2	1	1	
3.3.	Решение задач по теме «Полное и неполное ветвление»	2	-	2	Текущий контроль. Практикум
3.4.	Разветвляющиеся алгоритмы. Сложные условия. Каскадные ветвления	2	1	1	
3.5.	Множественный выбор и его реализация с помощью вложенных ветвлений	2	1	1	

3.6.	Решение задач по теме «Сложные условия. Каскадные ветвления»	2	-	2	Текущий контроль. Практикум
3.7.	Контрольная работа по темам раздела: «Разветвляющиеся алгоритмы и их реализация на языке Python»	2	-	2	Промежуточный контроль. Зачетная работа
<b>4.</b>	<b>Циклические алгоритмы и их реализация на Python</b>	<b>26</b>	<b>4</b>	<b>22</b>	
4.1.	Циклические алгоритмы. Цикл с предусловием. Инструкции управления циклом	2	1	1	
4.2.	Решение задач по теме «Циклические алгоритмы. Цикл с предусловием»	6	-	6	Текущий контроль. Практикум
4.3.	Циклические алгоритмы. Цикл с постусловием. Инструкции управления	2	1	1	
4.4.	Решение задач по теме «Циклические алгоритмы. Цикл с постусловием»	2	-	2	Текущий контроль. Практикум
4.5.	Самостоятельная работа по теме «Циклические алгоритмы. Цикл с предусловием»	2	-	2	Текущий контроль. Самостоятельная работа
4.6.	Циклические алгоритмы. Цикл с параметром. Инструкции управления циклом	2	1	1	
4.7.	Решение задач по теме «Циклические алгоритмы. Цикл с параметром»	4	-	4	Текущий контроль. Практикум
4.8.	Самостоятельная работа по теме «Циклические алгоритмы. Циклы со счетчиком»	2	-	2	Текущий контроль. Самостоятельная работа
4.9.	Сложные циклические алгоритмы. Вложенные циклы	2	1	1	
4.10.	Самостоятельная работа по разделу: «Циклические алгоритмы и их реализация на Python». Анализ самостоятельной работы	2	-	2	Текущий контроль. Самостоятельная работа
<b>5.</b>	<b>Этапы решения задач на языке Python</b>	<b>6</b>	<b>1</b>	<b>5</b>	
5.1.	Последовательное конструирование алгоритма	2	1	1	
5.2.	Этапы решения задачи на компьютере. Последовательное конструирование алгоритма	4	-	4	Текущий контроль. Практикум
<b>6.</b>	<b>Вспомогательные алгоритмы и их реализация на языке Python</b>	<b>10</b>	<b>2</b>	<b>8</b>	
6.1.	Вспомогательные алгоритмы и подпрограммы. Функции	2	1	1	

6.2.	Вспомогательные алгоритмы и подпрограммы. Решение задач с применением функций	4	-	4	Текущий контроль. Практикум
6.3.	Вспомогательные алгоритмы и подпрограммы. Рекурсия	2	1	1	
6.4.	Вспомогательные алгоритмы и подпрограммы. Решение задач с применением рекурсии	2	-	2	Текущий контроль. Практикум
<b>7.</b>	<b>Итоги обучения</b>	<b>4</b>	<b>1</b>	<b>3</b>	
7.1.	Итоговая самостоятельная работа по теме: «Реализация основных типов алгоритмов»	2	-	2	Итоговый контроль. Контрольная работа
7.2.	Подведение итогов обучения	2	1	1	
	<b>ИТОГО</b>	<b>72</b>	<b>16</b>	<b>56</b>	

### КАЛЕНДАРНЫЙ УЧЕБНЫЙ ГРАФИК

Раздел / месяц	сентябрь	октябрь	ноябрь	декабрь	январь	февраль	март	апрель	май
Введение в язык программирования Python	5ч								
Линейные алгоритмы и их реализация на языке Python	5ч	4ч							
Разветвляющиеся алгоритмы и их реализация на языке Python.		4ч	8ч						
Промежуточная аттестация				Зачетная работа – 2ч					
Циклические алгоритмы и их реализация на Python				6ч	6ч	8ч	6ч		
Этапы решения задач на языке Python							2ч	4ч	
Вспомогательные алгоритмы и их реализация на языке Python								4ч	6ч
Итоги обучения									2ч
Итоговая аттестация									Контрольная работа – 2 ч
Всего	10ч	8ч	8ч	6ч	6ч	8ч	8ч	8ч	10ч

### III. СОДЕРЖАНИЕ

**Раздел 1.** Введение в язык программирования Python.

**Тема 1.1.** Вводное занятие. Понятия «алгоритм» и «программа». Начальное знакомство с языком. Инструктаж.

**Теория.** Структура образовательной программы, её цель и задачи, содержание обучения. Основные правила и требования техники безопасности при работе за компьютером. Понятие алгоритма и программы.

**Практика.** Первичная диагностика. Тестирование. Установка, работа в среде программирования Python.

**Тема 1.2.** Элементы языка. Структура программы. Операции и переменные. Типы данных.

**Теория.** Знакомство с основными типами переменных, синтаксисом языка

программирования, основными процедурами ввода исходных данных и вывода результатов.

**Практика.** Отработка процедур ввода исходных данных и вывода результатов. Тестирование.

**Тема 1.3.** Ввод и вывод данных на языке Python. Ввод и отладка программ в среде.

**Практика.** Практикум по созданию элементарных программ ввода вывода данных, работа со средой, отладка программ. Анализ возможных синтаксических ошибок.

**Раздел 2.** Линейные алгоритмы и их реализация на языке Python.

**Тема 2.1.** Линейные алгоритмы целочисленных данных и их реализация на языке Python.

**Теория.** Знакомство со структурой линейного алгоритма, правилами записи арифметических выражений. Различные типы данных, допустимые операции над ними и ресурсы оперативной памяти. Выполнение операции присваивания в оперативной памяти (далее – ОП) компьютера.

**Практика.** Создание блок-схемы линейного алгоритма. Запись арифметических выражений на языке программирования.

**Тема 2.2.** Решение задач по теме «Обработка целочисленных данных».

**Практика.** Практикум по разработке линейных алгоритмов. Ввод и отладка программ, реализующих линейный алгоритм обработки целых чисел. Анализ готовых линейных программ.

**Тема 2.3.** Линейные алгоритмы вещественных чисел и их реализация на Python

**Теория.** Различные типы данных, допустимые операциях над вещественными числами и ресурсы оперативной памяти.

**Практика.** Разбор типичных задач с линейной структурой алгоритма.

**Тема 2.4.** Решение задач по теме «Обработка вещественных чисел».

**Практика.** Практикум по реализации линейных алгоритмов вещественных чисел. Ввод и отладка программ, реализующих линейный алгоритм обработки вещественных чисел.

**Тема 2.5.** Самостоятельная работа по темам раздела: «Линейные алгоритмы и их реализация в среде Python».

**Практика.** Самостоятельная работа по составлению линейного алгоритма, написанию программы, вводу и отладке программного кода, анализу результатов.

**Раздел 3.** Разветвляющиеся алгоритмы и их реализация на языке Python.

**Тема 3.1.** Алгоритм «выбор», графическое изображение, полное и неполное ветвление.

**Теория.** Понятие алгоритма «выбор», графическое изображение.

**Практика.** Составление алгоритма «полное ветвление».

**Тема 3.2.** Реализация алгоритма «выбор» на Python. Примеры решения задач.

**Теория.** Типовые задачи, использующие алгоритм «выбор», изображение алгоритма в виде блок-схемы.

**Практика.** Ввод и отладка программ в среде Python.

**Тема 3.3.** Решение задач по теме «Полное и неполное ветвление».

**Практика.** Практикум по реализации алгоритмов «полное ветвление» и «неполное ветвление». Ввод и отладка программ в среде Python.



**Тема 3.4.** Разветвляющиеся алгоритмы. Сложные условия. Каскадные ветвления.

**Теория.** Составление и анализ алгоритмов.

**Практика.** Ввод и отладка программ в среде Python.

**Тема 3.5.** Множественный выбор и его реализация с помощью вложенных ветвлений

**Теория.** Понятие множественного выбора, изображение на блок-схеме.

**Практика.** Ввод и отладка программ в Python.

**Тема 3.6.** Решение задач по теме «Сложные условия. Каскадные ветвления».

**Практика.** Практикум по составлению алгоритмов для решения задач со сложными условиями и каскадными ветвлениями. Ввод и отладка программ в среде Python.

**Тема 3.7.** Контрольная работа по темам раздела «Разветвляющиеся алгоритмы и их реализация на языке Python».

**Практика.** Контрольная работа по составлению алгоритмов, написанию программного кода, вводу и отладке программ в среде Python. Анализ работы.

**Раздел 4.** Циклические алгоритмы и их реализация на языке Python.

**Тема 4.1.** Циклические алгоритмы. Цикл с предусловием. Инструкции управления циклом.

**Теория.** Понятие цикла с предусловием, графическое изображение.

**Практика.** Составление алгоритмов, написание программного кода, ввод и отладка программ в среде Python.

**Тема 4.2.** Решение задач по теме «Циклические алгоритмы. Цикл с предусловием».

**Практика.** Практикум по составлению блок-схем, написанию программного кода, вводу и отладке программ, реализующих цикл с предусловием в среде Python.

**Тема 4.3.** Циклические алгоритмы. Цикл с постусловием. Инструкции управления циклом.

**Теория.** Понятие цикла с постусловием, графическое изображение.

**Практика.** Ввод и отладка простейших программ, реализующих цикл с постусловием в среде Python.

**Тема 4.4.** Решение задач по теме «Циклические алгоритмы. Цикл с постусловием»

**Практика.** Практикум по решению типовых задач с помощью цикла с постусловием. Составление блок-схем, написание программного кода, ввод и отладка программ, реализующих цикл с постусловием в Python.

**Тема 4.5.** Самостоятельная работа по теме «Циклические алгоритмы. Цикл с предусловием».

**Практика.** Самостоятельная работа по решению типовых задач с помощью цикла с предусловием. Ввод и отладка программ, реализующих цикл с предусловием в среде Python. Анализ составленного алгоритма.

**Тема 4.6.** Циклические алгоритмы. Цикл с параметром. Инструкции управления циклом.

**Теория.** Понятие цикла с параметром, графическое изображение.

**Практика.** Составление алгоритмов, написание программного кода, ввод и отладка программ в среде Python.

**Тема 4.7.** Решение задач по теме: «Циклические алгоритмы. Цикл с параметром»

**Практика.** Практикум по разработке алгоритмов цикла с параметром, анализу логических ошибок. Ввод и отладка программ, реализующих цикл с параметром в среде Python.

**Тема 4.8.** Самостоятельная работа по теме «Циклические алгоритмы. Циклы со счетчиком».

**Практика.** Самостоятельная работа по разработке блок-схем и отладке программ, реализующих циклы с параметром в среде Python. Анализ самостоятельной работы.

**Тема 4.9.** Сложные циклические алгоритмы. Вложенные циклы. Решение задач

**Теория.** Понятие вложенного цикла, графическое изображение.

**Практика.** Решение задач по разработке и отладке программ, реализующих вложенные циклы в среде Python. Анализ разработанного алгоритма. Оценка эффективности разработанного алгоритма.

**Тема 4.10.** Самостоятельная работа по темам раздела «Циклические алгоритмы и их реализация на языке Python». Анализ самостоятельной работы.

**Практика.** Самостоятельная работа по разработке блок-схем и отладке программ, реализующих циклические алгоритмы в среде Python.

**Раздел 5.** Этапы решения задач на языке Python.

**Тема 5.1.** Последовательное конструирование алгоритма.

**Теория.** Этапы и особенности решения задачи на компьютере. Этап создания алгоритма. Использование принципа последовательного конструирования алгоритма. Будет также рассмотрен принцип его работы.

**Практика.** Разработка алгоритмов. Написание программного кода, ввод и отладка программ в среде Python.

**Тема 5.2.** Этапы решения задачи на компьютере. Последовательное конструирование алгоритма.

**Практика.** Практикум по последовательному конструированию алгоритмов. Написание программного кода, ввод и отладка программ в среде Python.

**Раздел 6.** Вспомогательные алгоритмы и их реализация на языке Python.

**Тема 6.1.** Вспомогательные алгоритмы и подпрограммы. Функции.

**Теория.** Понятие вспомогательного алгоритма. Формат записи вспомогательного алгоритма в виде функции. Типовые задачи.

**Практика.** Ввод и отладка программ с использованием функции в среде Python.

**Тема 6.2.** Вспомогательные алгоритмы и подпрограммы. Решение задач с применением функций.

**Практика.** Практикум по программированию решения отдельных подзадач с помощью отдельных функций, которые потом при необходимости вызываются в различных местах программы. Польза функций при решении задач. Важные принципы в программировании – модульность и повторное использование кода. Их польза при разработке сложных программ.

**Тема 6.3.** Вспомогательные алгоритмы и подпрограммы. Рекурсия Теория. Понятие рекурсии, в чём её сложность. Рекурсивные алгоритмы и их особенности: когда нужно использовать рекурсию в программировании, а когда лучше обойтись без этого.

**Практика.** Ввод и отладка программ с использованием рекурсии в среде Python.

**Тема 6.4.** Вспомогательные алгоритмы и подпрограммы. Решение задач с применением рекурсии

**Практика.** Практикум по решению типовых задач с вводом и отладкой программ с использованием рекурсии в среде Python.

**Раздел 7.** Итоги обучения.

**Тема 7.1.** Итоговая самостоятельная работа по теме «Реализация основных типов алгоритмов».

**Практика.** Зачетная работа по составлению алгоритма, написанию программы, вводу и отладке программного кода.

**Тема 7.2.** Подведение итогов обучения.

**Теория.** Обзор пройденного материала.

**Практика.** Анализ результатов работы.

#### **IV. ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОСВОЕНИЯ ПРОГРАММЫ**

**Предметные результаты:**

**Учащийся**

- разрабатывает алгоритмы и программы на языке программирования Python;
- анализирует содержание и структуру исходных данных;
- выстраивает логическую последовательность действий для решения задач.

**Метапредметные:**

**Учащийся**

- формулирует цель своей деятельности, и выстраивает план действий;
- оценивает свою деятельность и вносит коррективы с учетом сделанных ошибок;
- выстраивает словесный алгоритм, высказывает и отстаивает свою точку зрения.

**Личностные:**

**Учащийся**

- проявляет этичное отношение к участникам группового взаимодействия;
- проявляет ответственное отношение к обучению и саморазвитию.

#### **V. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ**

**Способы проверки результатов:**

На различных этапах обучения используются различные виды контроля: предварительный, текущий, тематический и итоговый.

Контроль осуществляется в виде тестирования, диктантов, самостоятельных, практических работ, контрольных работ, зачетов.

Педагог создает сводную таблицу, где записаны все итоговые работы.

В течение учебного процесса нужно выполнить все работы, которые отмечаются в таблице.

За итоговую контрольную работу или проект выставляются оценки: отлично, хорошо, удовлетворительно. Предусматривается участие детей в олимпиадах по программированию.

**Критерии оценивания итоговой контрольной работы на языке программирования Python.**

При решении итоговой контрольной работы учащийся должен продемонстрировать знание:

1. основ программирования и культуры построения программы;
2. составления алгоритма и представление его в графическом виде;
3. определения типов данных для используемых в программах переменных;
4. построения ветвления и выбора;
5. построения цикла по счетчику;
6. построения циклов с постусловием и предусловием;
7. использования в общей программе подпрограмм: процедур и функций; массивов: одномерных и двумерных, и принципов работы с ними;
8. Принципов работы со строками;
9. Принципов работы с файлами.

Контрольная работа оценивается **отлично** (3 балла), если выполнены безошибочно 100% – 80 % работы.

Контрольная работа оценивается **хорошо** (2 балла), если выполнено 70% – 50 % работы.

Контрольная работа оценивается **удовлетворительно** (1 балл), если выполнено 50% - 30% работы.

## VI. МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ

При реализации программы используются следующие методы обучения

- Объяснительно-иллюстративный: педагог, проводя беседы, рассказывает и показывает на схемах, чертежах новый материал;
- Репродуктивный: при использовании этого метода обучающиеся выполняют задания по образцу, чтобы закрепить изученный материал;
- Диалогический: применение этого метода предусматривает диалог между педагогом и обучающимся, обеспечивает более прочное усвоение знаний, путем обсуждения возникающих проблем при выполнении заданий.
- Частично - поисковый: обучающиеся самостоятельно, или совместно с педагогом проводят поиск новых решений нестандартных заданий
- Наглядный (наблюдение учащимися естественных объектов, явлений, процессов или их изображений-таблиц, моделей демонстрируемых учителем), при котором источником познания служат наглядные средства;
- Практический (наблюдения предметов и явлений в процессе труда или экспериментов, упражнений, решение задач), при котором источником познания служит практическая работа учащихся.

## VII. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ

Для реализации программы необходимы:

- компьютерный класс на 10 рабочих мест,
- программное обеспечение: Python,
- проектор,
- принтер.

## VIII. СПИСОК ЛИТЕРАТУРЫ

1. Банкрашков А. Программирование для детей на языке Python. – Москва: АСТ, 2017. 2. Бриггс Дж. Python для детей. Самоучитель по программированию. – Москва: Манн, 2018.
2. Бэрри П. Изучаем программирование на Python. – Москва: Эксмо, 2017.
3. ВандерПлас Дж. Python для сложных задач. Наука о данных и машинное

- обучение. – Санкт-Петербург: Питер, 2018.
4. Васильев А. Python на примерах. Практический курс. Москва: Наука и техника, 2018.
  5. Вордэрман К. Программирование на Python. Иллюстрированное руководство для детей. – Москва: Манн, Иванов и Фербер, 2018.
  6. Гэддис Т. Начинаем программировать на Python. – Санкт-Петербург: БВХ-Петербург, 2019.
  7. Доусон М. Програмируем на Python. – Санкт-Петербург: Прогресс книга, 2019.
  8. Жуков Р. Язык программирования Python: практикум. Учебное пособие. – Москва: Инфра-М, 2020.
  9. Златопольский Д.М. Основы программирования на языке Python. – Москва: ДМК Пресс, 2018.
  10. Лутц М. Изучаем Python. В 2-х т. – Москва: Вильямс, 2019.
  11. Любанович Б. Простой Python. Современный стиль программирования. – Санкт-Петербург: Питер, 2019.
  12. МакГрат М. Программирование на Python для начинающих. Москва: Эксмо, 2015.
  13. Маккини У. Python и анализ данных. – Москва: ДМК Пресс, 2018.
  14. Мартелли А., Рейвенскрофт А., Холден С. Python. Справочник. Полное описание языка. – Москва: Диалектика, 2018.
  15. Мэтиз Э. Изучаем Python. Программирование игр, визуализация данных, веб-приложения. – Санкт-Петербург: Питер, 2020.
  16. Мюллер Дж.П. Python для чайников. – Москва: Вильямс, 2019.
  17. Персиваль Г. Python. Разработка на основе тестирования. – Москва: ДМК Пресс, 2018.
  19. Пратик Дж. Искусственный интеллект с примерами на Python. Создание приложений искусственного интеллекта с помощью Python для взаимодействия с окружающим миром. – Москва: Вильямс, 2019.
  18. Пэйн Б. Python для детей и родителей. Играй и программируй. – Москва, Эксмо, 2018.
  19. Рамальо Л. Python к вершинам мастерства. – Москва: ДМК Пресс, 2016.
  22. Свейгарт Э. Автоматизация рутинных задач с помощью Python. Практическое руководство для начинающих. – Москва: Вильямс, 2018.
  20. Свейгарт Э. Учим Python, делая крутые игры. – Москва: Бомбора, 2018.
  21. Солем Я.Э. Программирование компьютерного зрения на языке Python. – Москва: ДМК Пресс, 2016.
  22. Томашевский П. Привет, Python. Моя первая книга по программированию. – Москва: Наука и техника, 2018.
  23. Харрисон М. Как устроен Python. – Санкт-Петербург: Питер, 2019.