



## **СОДЕРЖАНИЕ**

### **1. Пояснительная записка**

- 1.1. Нормативно-правовая основа программы
- 1.2. Актуальность (педагогическая целесообразность) и новизна программы
- 1.3. Направленность программы, ее цель и задачи
- 1.4. Требования к категории обучающихся
- 1.5. Общая характеристика программы
- 1.6. Принципы обучения по программе
- 1.7. Организационно-педагогические условия реализации программы
- 1.8. Планируемые результаты обучения
- 1.9. Формы подведения итогов и определение результативности программы

### **2. Учебный план**

### **3. Календарный учебный график**

### **4. Рабочая программа учебного модуля «ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ PYTHON»**

## 1. Пояснительная записка

### 1.1. Нормативно-правовая основа программы

Дополнительная образовательная общеразвивающая программа «ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ PYTHON» (далее – программа) составлена на основе следующих нормативных правовых документов:

- Федеральный закон от 29.12.2012 №273-ФЗ «Об образовании в Российской Федерации»;
- Приказ Министерства просвещения России от 09.11.2018 N 196 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам»;
- Концепция развития дополнительного образования детей от 04.09.2014 г. № 1726-р;
- Санитарно-эпидемиологические требования к устройству, содержанию и организации режима работы образовательных организаций дополнительного образования детей (СанПиН 2.4.4.3172- 14).

### 1.2. Актуальность (педагогическая целесообразность) и новизна программы

С развитием современных информационных технологий сегодня любой учащийся под руководством опытного педагога может с лёгкостью научиться программировать.

Компьютеры и компьютерные системы – неотъемлемая часть жизни нашего общества. Научившись программировать, мы можем быть не только пользователями информационных технологий, но и активными их создателями.

Учиться программировать очень интересно. Результат программирования очень часто виден сразу. Кроме того, создание компьютерных игр и обучающих программ способствует развитию логики и креативного мышления. Ещё одной значимой стороной обучения программированию является спрос на рынке труда на специалистов данного направления деятельности.

#### Новизна программы

Новизна Программы заключается в том, что Python дает более широкие возможности в области программирования, чем Pascal, который входит в школьный курс информатики. На языке Python можно легко и быстро создавать простые компьютерные игры, трёхмерные модели и программировать роботов. Этот язык быстрее и легче усваивается, чем Pascal. Многие мировые компании такие, как Intel, Cisco, Hewlett-Packard, используют этот язык при реализации своих проектов. Крупнейшие интернет-ресурсы такие, как Google, YouTube, также разработаны с помощью языка программирования Python.

#### Педагогическая целесообразность Программы

Педагогическая целесообразность программы заключается в привлечении учащихся к занятиям техническим творчеством, что способствует развитию логического мышления, творческих способностей и навыков решения задач программирования. Программирование мотивирует к занятиям в различных научных областях (физики, информатики, алгебры, геометрии и др.), развивает воображение и способствует ранней профориентации подростков. Для достижения поставленных задач занятия проводятся в формате «от простого к сложному». Учащиеся вспоминают свои знания по основам алгоритмизации и программирования и на их основе, углубляя их, учатся составлять простые и сложные программы.

### 1.3. Направленность программы, ее цель и задачи

**Цель Программы:** 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 Программы**

### ***Обучающие:***

- обучить языку программирования Python и созданию программ на его основе;
- научить создавать прикладное программное обеспечение;
- расширять кругозор обучающихся в области программирования;
- научить дизайнерскому оформлению созданного ПО.

### ***Развивающие:***

- развивать память и внимание, познавательную и творческую активность;
- развивать творческие способности, эстетическое и эргономическое восприятие объектов труда;
- развивать логическое мышление.

### ***Воспитательные:***

- прививать интерес к активному творческому самовыражению, культуре труда;
- воспитывать упорство в достижении желаемого результата;
- воспитывать эстетический вкус;
- воспитывать чувство взаимопомощи, доверия, коллективизма.

## **1.4. Требования к категории обучающихся**

Программа ориентирована на детей, склонных связать свою будущую жизнь с профессией программиста.

Набор производится по желанию учащихся и их родителей.

Возрастная категория обучающихся по программе от 12 до 17 лет.

## **1.5. Общая характеристика программы**

Программа построена с учетом возрастных и индивидуальных особенностей детей, режима и временных параметров организации образовательной деятельности (занятий), что делает ее доступной, обеспечивает достижение положительного результата в обучении, познавательном и творческо-личностном развитии.

Программа рассчитана на 72 общих часа нагрузки. Структура курса предусматривает лекции и практические занятия.

На лекциях раскрываются основные теоретические положения, практические занятия направлены на закрепление теоретических знаний. Реализация программы построена на использовании активных методов обучения, совместной творческой деятельности преподавателя и детей.

Структура образовательной программы включает комплекс модулей, направленных на изучение основ программирования:

- Основные логические принципы программирования.
- Основы работы с объектно-ориентированной средой Python.
- Основные программные модули и компоненты.
- Работа с программными функциями и процедурами.
- Работа с логическими операторами и циклами.
- Работа с графикой.

### **Отличительные особенности программы.**

Основное количество часов отводится практическому написанию программ. Каждый обучающийся реализует индивидуальный проект в результате освоения программы. Продукт, полученный в результате освоения программы, имеет прикладной характер и может быть использован по необходимости.

## **1.6. Принципы обучения по программе**

В качестве основных принципов обучения по данной программе определены:

- принцип взаимодействия и сотрудничества;

- принцип единства развития, обучения и воспитания;
- принцип учета возрастных и индивидуальных особенностей детей;
- принцип систематичности и последовательности;
- принцип доступности;
- принцип наглядности;
- принцип вариативности и вариантности;
- принцип комплексного подхода.

### **1.7. Организационно-педагогические условия реализации программы**

Образовательный процесс осуществляется на основе учебного плана, рабочей программы учебного модуля и регламентируется расписанием занятий.

*Сроки реализации программы.* Программа реализуется в течение учебного 1 года. Учебный год начинается с 1 ноября. Обучение детей рассчитано на 33 учебных недель, общее количество учебных часов – 72.

*Формы реализации программы и режим занятий.* Форма проведения занятий – групповая. Услуги оказываются в том числе с применением электронного обучения и дистанционных образовательных технологий.

Порядок организации занятий:

- режим занятий: 1 раз в неделю;
- продолжительность занятий – 2 аудиторных часа (занятия по 45 мин. с перерывом на отдых 10 минут).

*Организации образовательного процесса для учащихся с ограниченными возможностями здоровья, детей-инвалидов, инвалидов.*

Обучение учащихся с ограниченными возможностями здоровья, детей-инвалидов и инвалидов по дополнительным общеобразовательным программам осуществляется с учетом особенностей психофизического развития, индивидуальных возможностей и состояния здоровья таких учащихся.

Занятия с учащимися с ограниченными возможностями здоровья, детьми-инвалидами и инвалидами организуются совместно с другими учащимися.

Образовательная деятельность учащихся с ограниченными возможностями здоровья по дополнительным общеобразовательным программам осуществляется на основе дополнительных общеобразовательных программ, адаптированных при необходимости для обучения указанных учащихся, с привлечением специалистов в области коррекционной педагогики, а также педагогическими работниками, прошедшими соответствующую переподготовку.

Сроки обучения по дополнительным общеразвивающим программам для учащихся с ограниченными возможностями здоровья, детей-инвалидов и инвалидов с учетом особенностей их психофизического развития при необходимости могут быть увеличены.

*Педагогические кадры.* Образовательный процесс по образовательной программе обеспечивает Размологов Д.М. – программист УИНФ АГГПУ им. В.М. Шукшина.

*Учебно-методическое и информационное обеспечение программы.* Содержательная составляющая модулей образовательной программы отражена в рабочих программах, в которых отдельным блоком представлены учебно-методические (списки основной и дополнительной литературы) и информационные материалы (ссылки на электронные ресурсы).

### **1.8. Планируемые результаты обучения**

Ожидаемые результаты и способы определения их результативности

По итогам реализации программы, учащиеся будут знать:

- принципы программирования на языке Python;

- основы дизайнерского оформления созданных программ.
- По итогам реализации программы, учащиеся будут уметь:
- производить чтение и запись программ на языке Python;
  - запускать и отлаживать программу.

### **1.9. Формы подведения итогов и определение результативности программы**

Результативность усвоения учебного материала отслеживается по итогам освоения каждого учебного блока (модуля) посредством:

- тестов текущего контроля;
- проверочных (контрольных) работ;
- творческих (проектных) заданий.

МИНИСТЕРСТВО ПРОСВЕЩЕНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Алтайский государственный гуманитарно-педагогический университет  
имени В.М. Шукшина»  
(АГГПУ им. В.М. Шукшина)

Образовательный центр

Утверждено  
на Ученом совете вуза  
Протокол № 2/1  
от « 27 » 09 2022 г.

Утверждаю:  
Проректор по УР М.В. Довьцова  
от « 27 » 09 2022 г.



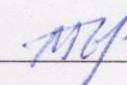
**УЧЕБНЫЙ ПЛАН  
ДОПОЛНИТЕЛЬНОЙ ОБРАЗОВАТЕЛЬНОЙ  
ОБЩЕРАЗВИВАЮЩЕЙ ПРОГРАММЫ  
«ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ PYTHON»**

**Категория обучающихся:** дети 12-17 лет  
**Количество учебных групп:** 1  
**Количество учебных часов:** 72

**Срок обучения:** 12.11.2022-27.05.2022  
**Режим занятий:** 14.00 -15.40  
**Место проведения:** АГГПУ им. В.М.  
Шукшина

**СОГЛАСОВАНО:**

И.о. начальника отдела непрерывного  
дополнительного образования

 /Л.А. Романова/

## УЧЕБНЫЙ ПЛАН

№ п/п	Наименование разделов учебного модуля	Распределение учебной нагрузки				
		всего	лекции	практические	самостоятель ная работа	форма контроля
1	Модуль 1. Введение в Python	6	2	4		контрольная работа
2	Модуль 2. Алгоритмические инструкции	6	2	4		контрольная работа
3	Модуль 3. Строки	6	2	4		контрольная работа
4	Модуль 4. Функции	6	2	4		контрольная работа
5	Модуль 5. Списки и кортежи	6	2	4		контрольная работа
6	Модуль 6. Словари и множества	6	2	4		контрольная работа
7	Модуль 7. Объектно-ориентированное программирование (ООП)	18	4	8	6	контрольная работа
8	Модуль 8. Подведение итогов. Защита индивидуального проекта	18	2	6	10	творческое задание
	<b>ИТОГО:</b>	<b>72</b>	<b>18</b>	<b>38</b>	<b>16</b>	



Проректор по УР

Утверждаю:  
М.В.Довыдова  
от « 27 » 09 2022 г.

### КАЛЕНДАРНЫЙ УЧЕБНЫЙ ГРАФИК НА 2022-2023 УЧЕБНЫЙ ГОД

№ п/п	Наименование учебных модулей	октябрь					ноябрь				декабрь			
		Модуль 3					Модуль 4				Модуль 5			
		01.10.2022	08.10.2022	15.10.2022	22.10.2022	29.10.2022	05.11.-2022	12.11.-2022	19.11.-2022	26.11.-2022	03.12.2022	10.12.2022	17.12.2022	24.12.2022
1	Введение в Python	-	-	-	-	-	2	2	2					
2	Алгоритмические инструкции									2	2	2		
3	Строки												2	2
№ п/п	Наименование учебных модулей	январь				февраль				март				
		Модуль 3				Модуль 4				Модуль 5				
		=	14.01-2023	21.01-2023	28.01-2023	04.02.2023	11.02.2023	18.02.2023	25.02.2023	04.03.2023	11.03.2023	18.03.2023	25.03.2023	
3	Строки		2											
4	Функции			2	2	2								
5	Списки и кортежи						2	2	2					
6	Словари и множества									2	2	2		
7	Объектно-ориентированное программирование (ООП)												2	
п/п	Наименование учебных модулей	апрель					май							
		Модуль 6					Модуль 7							
		01.04.2022	08.04.2022	15.04.2022	22.04.2022	29.04.2022	06.05.2022	13.05.2022	20.05.2022	27.05.2022				
7	Объектно-ориентированное программирование (ООП)	2	2	2	2	2								
8	Подведение итогов. Защита индивидуального проекта						2	2	2	2				

Примечание: 2 – кол-во часов в неделю; = – учебный процесс не осуществляется



## Распределение учебной нагрузки

Общий объем	Аудиторная			Форма контроля (зачет, экзамен, контр. работа, проект)
	Лекции	Практические работы	Самостоятельная работа	
72	18	38	18	Защита индивидуального проекта

Программа одобрена на заседании кафедры математики, физики, информатики  
Протокол № 1 от 02.09.2022

Заведующий кафедрой Т.А. Гусева

## **АННОТАЦИЯ ПРОГРАММЫ «ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ PYTHON»**

Программа Python дает более широкие возможности в области программирования, чем Pascal, который входит в школьный курс информатики. На языке Python можно легко и быстро создавать простые компьютерные игры, трёхмерные модели и программировать роботов. Этот язык быстрее и легче усваивается, чем Pascal. Многие мировые компании такие, как Intel, Cisco, Hewlett-Packard, используют этот язык при реализации своих проектов. Крупнейшие интернет-ресурсы такие, как Google, YouTube, также разработаны с помощью языка программирования Python. Учитывая особенности обучаемых групп, в частности разновозрастной состав учащихся, метод проектов обеспечивает как целенаправленную работу обучаемых в группе в целом на достижение результата (разработку конечного проекта), так и развитию навыков самообучения в ходе «исследования» предложенной среды программирования. Курс рассчитан на достижение практических навыков в проектировании задачи, определения составных частей проекта и разработки конечной версии разрабатываемого программного продукта.

Педагогическая целесообразность программы заключается в привлечении учащихся к занятиям техническим творчеством, что способствует развитию логического мышления, творческих способностей и навыков решения задач программирования. Программирование мотивирует к занятиям в различных научных областях (физики, информатики, алгебры, геометрии и др.), развивает воображение и способствует ранней профориентации подростков. Для достижения поставленных задач занятия проводятся в формате «от простого к сложному». Учащиеся вспоминают свои знания по основам алгоритмизации и программирования и на их основе, углубляя их, учатся составлять простые и сложные программы.

Цель программы: 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.

Программа ориентирована на детей, склонных связать свою будущую жизнь с профессией программиста. Набор производится по желанию учащихся и их родителей.

Возрастная категория обучающихся по программе от 12 до 17 лет.

## Цели и задачи занятий

**Цель Программы:** 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 Программы**

Обучающие:

- обучить языку программирования Python и созданию программ на его основе;
- научить создавать прикладное программное обеспечение;
- расширять кругозор обучающихся в области программирования;
- научить дизайнерскому оформлению созданного ПО.

Развивающие:

- развивать память и внимание, познавательную и творческую активность;
- развивать творческие способности, эстетическое и эргономическое восприятие

объектов труда;

- развивать логическое мышление.

Воспитательные:

- прививать интерес к активному творческому самовыражению, культуре труда;
- воспитывать упорство в достижении желаемого результата;
- воспитывать эстетический вкус;
- воспитывать чувство взаимопомощи, доверия, коллективизма.

### **Планируемые результаты обучения**

К ожидаемым результатам реализации программы относятся:

*знание:*

синтаксис изучаемого языка программирования;  
алгоритмические структуры изучаемого языка программирования;  
методы обработки информации;  
принципы построения и разработки алгоритмов создаваемых программ;  
основные элементы объектно-ориентированной среды и принципы работы с ними;  
принципы моделирования;

*умение:*

грамотно оптимизировать разрабатываемые алгоритмы создаваемых проектов;  
работать с составляющими объектно-ориентированной среды программирования

Lazarus;

грамотно и корректно ставить конечную задачу разрабатываемого проекта, а так же формулировать промежуточные цели;

*владение:*

проектировать модели разрабатываемых приложений;  
грамотно разрабатывать интерфейс создаваемых проектов.

*Метапредметным результатом* реализации программы является формирование у школьников стойкой познавательной мотивации к осуществлению дальнейшей деятельности по освоению программирования.

### **Сроки реализации программы**

Программа реализуется в течение 1 учебного года. Учебный год начинается с ноября месяца. Обучение детей рассчитано на 23 учебных недель, общее количество учебных часов – 72.

### **Формы реализации программы и режим занятий**

Образовательный процесс осуществляется на основе учебного плана, рабочей программы учебного модуля и регламентируется расписанием занятий.

Формы реализации программы и режим занятий. Форма проведения занятий – групповая. Услуги оказываются в том числе с применением электронного обучения и дистанционных образовательных технологий.

Порядок организации занятий: режим занятий: 1 раз в неделю; продолжительность занятий – 2 аудиторных часа (занятия по 45 мин. с перерывом на отдых 10 минут).

## Учебно-тематический план занятий

### Содержание разделов модулей

#### **Модуль 1. Введение в Python.**

*Основные понятия:* трансляция, интерпретация, компиляция, синтаксис, семантика, прагматика, переменная, динамическая типизация, служебные слова, идентификаторы, простые типы данных, приоритеты операций, литералы чисел, операция присваивания, PEP 8.

##### **1.1. Язык Python. Среда программирования. Особенности ввода-вывода**

*Теория.* Язык программирования Python. Достоинства и недостатки.

Области применения. Интерактивный режим работы программы.

*Практика.* Установка языка программирования Python3.5 и среды программирования WingIDE 100. Регистрация на Интернет-ресурсах.

**1.2. Типы данных, операции. Оператор присваивания** *Теория.* Ввод и вывод числовой информации.

*Практика.* Тренировочное задание на ввод и вывод числовой информации.

##### **1.3. Числа. Стандартные операции**

*Теория.* Стандартные операции с целыми и действительными числами.

Стиль программирования Python (PEP 8).

*Практика.* Решение простых задач в интерактивном режиме.

#### **Модуль 2. Алгоритмические инструкции**

*Основные понятия:* логический тип данных, логические операции (and, or, not, A), условный оператор, условное и альтернативное исполнение алгоритма, операторы сравнения, вложенность операторов, оператор цикла, переменная-флаг, генерация псевдослучайной последовательности, инструкции break, continue и pass.

##### **2.1. Условный оператор**

*Теория.* Условная и циклическая инструкции. Каскадность и вложенность алгоритмических инструкций.

##### **2.2. Цикл while**

*Теория.* Теория чисел. Фильтрация потока чисел. НОД иНОК. Проверка числа на простоту. Алгоритм Евклида. Нахождение максимума и минимума.

*Практика.* Решение задач на анализ чисел потока и целочисленной арифметики.

### **2.3. Цикл for**

*Теория.* Теория чисел. Фильтрация потока чисел. НОД иНОК. Проверка числа на простоту. Алгоритм Евклида. Нахождение максимума и минимума.

*Практика.* Решение задач на анализ чисел потока и целочисленной арифметики повышенной трудности.

## **Модуль 3. Строки**

*Основные понятия:* символ, строка, литерал, таблицы кодов ASCII, UTF-8, отладка кода, неизменяемый объект, формат вывода строки, экранированные escape-последовательности, положительная и отрицательная нумерация символов в строке, срез, конкатенация, длина строки.

### **3.1. Литералы строк**

*Теория.* Понятие «литералы строк».

*Практика.* Ввод-вывод строки. Решение задач на ввод строки, поиск подстроки.

### **3.2. Срезы строк**

*Теория.* Форматирование строки.

*Практика.* Преобразование строки. Решение упражнений.

### **3.3. Методы строк**

*Теория.* Методы работы со строкой.

*Практика.* Применение методов строки. Решение задач.

## **Модуль 4. Функции**

*Основные понятия:* подпрограмма, функция, процедура, рекурсия, глубина рекурсии, объявление, определение и вызов функции, возврат значений, глобальные и локальные переменные, передача параметров, работа с памятью, граф вызовов, стек вызовов, полиморфизм функций, утиная типизация, lambda-функции.

### **4.1. Парадигма и преимущества структурного программирования**

*Теория.* Обзор парадигм программирования. Особенности применения языков программирования.

### **4.2. Граф и стек вызовов функций. Области видимости переменных**

*Теория.* Применение стек и граф вызовов функций.

*Практика.* Выполнение тренировочных упражнений.

### **4.3. Прямая рекурсия**

*Теория.* Понятие прямой рекурсии и ее применение.

**Практика.** Выполнение тренировочных упражнений.

#### **4.4. Косвенная рекурсия**

**Теория.** Понятие косвенной рекурсии и ее применение.

**Практика.** Выполнение тренировочных упражнений.

*Практические занятия к темам 4.2., 4.3., 4.4.:* Нахождение суммы чисел. Вычисление степени. Числа Фибоначчи. Ханойские башни

Использование библиотеки математических функций. Решение задач повышенной трудности.

### **Модуль 5. Списки и кортежи**

*Основные понятия:* список, кортеж, элемент списка и кортежа, индекс, срез списка, матрица, многомерный список, сортировка, сложность алгоритма, устойчивость сортировки, квадратичная, быстрая, синхронная, поразрядная сортировки списка, случайное перемешивание.

#### **5.1. Списки и кортежи в Python. Сходства и различия**

**Теория.** Представление списка и кортежа в памяти компьютера, сходства и различия.

#### **5.2. Операции со списками**

**Теория.** Способы заполнения списка (с клавиатуры, из файла, случайным образом, по формуле).

**Практика.** Решение задач на ввод-вывод элементов одномерного и многомерного списка.

#### **5.3. Срезы списков**

**Теория.** Методы работы со списком и кортежем. Методы сортировки списка.

**Практика.** Решение задач на ввод-вывод элементов одномерного и многомерного списка и кортежа, преобразование, поиск, замену, подсчет.

#### **5.4. Матрицы. Операции над матрицами**

**Теория.** Вычисление сложности алгоритма. Многомерные списки.

**Практика.** Решение задач повышенной трудности.

### **Модуль 6. Словари и множества**

*Основные понятия:* словарь, множество, ключ, кодирование.

#### **6.1. Словари**

**Теория.** Понятие словаря. Способы создания. Словарь, преимущества и недостатки, методы работы со словарем. Словари со смешанными значениями. Кодирование и декодирование текста.

**Практика.** Решение задач на заполнение, преобразование, поиск, замену, подсчет, вывод элементов словаря.

## **6.2. Множества**

**Теория.** Понятие множества. Создание множеств. Множество, преимущества и недостатки, методы работы с множеством.

**Практика.** Решение задач повышенной трудности.

## **Модуль 7. Объектно-ориентированное программирование (ООП)**

**Основные понятия:** ООП, класс, метод INIT, экземпляр, наследование, полиморфизм, исключения, виджет, интерфейс, событие, техническое задание, проект, проектная деятельность, виды проектов.

### **7.1. Классы в Python**

**Теория.** Понятия «класс», «метод INIT», «экземпляр», наследование», «полиморфизм», «исключения», «виджет», «интерфейс», «интерфейс», «событие».

### **7.2. Разработка собственного класса**

**Теория.** Принципы разработки собственного класса. Обработка и генерация исключений. Виджет, методы виджета. Графическая библиотека tkinter, класс Tk. Системные методы. События.

**Практика.** Создание собственного класса.

### **7.3. Разработка и программирование собственного проекта**

**Практика.** Выбор вида и темы проекта. Составление технического задания. Программирование. Разработка технической документации и презентации проекта.

## **Модуль 8. Подведение итогов.**

### **8.1. Защита индивидуального проекта.**

## УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ПРОГРАММЫ

### ОСНОВНАЯ ЛИТЕРАТУРА

1. Информатика. Общий курс [Текст]: учебник для вузов / Под ред. А. Н. Гуда. – М.: Дашков и К°; Ростов на Дону: Наука-Пресс, 2019. – 400 с.
2. Комлева, Н. В. Информатика и программирование [Текст]: учебно-методический комплекс для студентов вузов, обучающихся по экономическим специальностям / Н. В. Комлева, А. А. Смирнов, Д. В. Хрипков. - Электрон. текстовые дан. - М. : Евразийский открытый институт, 2018. - 88 с.
3. Могилев, А.В. Информатика [Текст]: учебное пособие для студентов вузов, обучающихся по педагогическим специальностям / А.В. Могилев, Н.И. Пак, Е.К. Хеннер; ред. Е.К. Хеннер. – М.: Академия, 2017. – 848 с.
4. Степанов, А.Н. Информатика [Текст]: учебное пособие для вузов / А.Н. Степанов. – СПб.: Питер, 2016. – 765 с.

### ДОПОЛНИТЕЛЬНАЯ ЛИТЕРАТУРА

1. Бородин, А.Н. Элементарный курс теории вероятностей и математической статистики [Текст]: учебное пособие для вузов / А. Н. Бородин. – СПб.: Лань, 2008. – 256 с.
2. Гришин, М.П. Математика и информатика [Текст]: учебное пособие / М.П. Гришин. – М.: МГИУ, 2008. – 116 с.
3. Жолков, С.Ю. Математика и информатика для гуманитариев [Текст]: учебник для вузов / С.Ю. Жолков. – М.: Альфа – М: ИНФРА-М, 2008. – 528 с.
4. Информатика. Базовый курс [Текст]: учебное пособие для вузов / Под ред. С.В. Симоновича. – СПб.: Питер, 2010. – 640 с.
5. Левин, А.Ю. Самоучитель полезных программ [Текст] / А.Ю. Левин. – СПб.: Питер, 2008. – 704 с.
6. Могилев, А.В. Практикум по информатике [Текст]: учебное пособие для вузов / А.В. Могилев, Н.И. Пак, Е.К. Хеннер; под ред. Е.К. Хеннера. – М.: Академия, 2009. – 608 с.
7. Уткин, В.Б. Математика и информатика: учебное пособие / В.Б. Уткин, К.В. Балдин, А.В. Рукосуев. – М.: Дашков и К°, 2008. – 472 с.
8. Программирование на Паскаль [Текст]: учебное пособие для дисциплин "Программирование", "Технология программирования и работа на ЭВМ" / сост.: С. Ю. Кречетова, М. Ю. Беликова, Н. В. Фотиев. - Горно-Алтайск: Горно-Алтайский гос. университет, 2013. - 80 с.

### ПРОГРАММНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ

1. Бизли, Дэвид М. Python. Подробный справочник. – М.–СПб.: Символ-Плюс, 2018.
2. Лутц, Марк Python. Справочник. – М.: Вильямс, 2017.
3. Официальный сайт программы [Электронный ресурс] – Режим доступа: <https://docs.python.org/>, свободный.
4. Сайт, среда разработки для языка Python. [Электронный ресурс] – Режим доступа: <https://www.jetbrains.com/pycharm/?fromMenu>, свободный.
5. Сайт / справочные материалы [Электронный ресурс] – Режим доступа: <https://metanit.com/python/>, свободный.
6. Сайт / интерактивный сборник задач для практики программирования [Электронный ресурс] – Режим доступа: <http://pythontutor.ru/>, свободный.
7. Сайт / Адаптивный тренажер Python [Электронный ресурс] – Режим доступа: <https://stepik.org/course/431>, свободный.
8. Сайт / среда разработки для языка Python [Электронный ресурс] – Режим доступа: <https://www.jetbrains.com/pycharm/?fromMenu>, свободный.
9. Сайт проекта Open Book Project. Практические примеры на Python Криса Мейерса [Электронный ресурс] – Режим доступа: [openbookproject.net](http://openbookproject.net), свободный.