

ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ  
«ЦЕНТР ДЛЯ ОДАРЕННЫХ ДЕТЕЙ «ПОИСК»

РЕКОМЕНДОВАНА

педагогическим советом  
Протокол № 6 от «2» апреля 2024 г.



Дополнительная общеобразовательная общеразвивающая  
программа технической направленности

**«ОСНОВЫ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ PYTHON»**

Направление:	техническое
Возраст обучающихся:	14-17 лет
Объем программы:	136 часов
Срок освоения:	1 год
Форма обучения:	очная
Авторы программы:	Журавлёв Алексей Борисович, педагог дополнительного образования ЦЦО «IT-куб» Савельева Ольга Александровна, методист ЦЦО «IT-куб»

Михайловск, 2024

## ОГЛАВЛЕНИЕ

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА .....	3
1. ОСНОВНЫЕ ХАРАКТЕРИСТИКИ ПРОГРАММЫ .....	5
2. ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИЕ УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ.....	11
УЧЕБНЫЙ ПЛАН.....	13
КАЛЕНДАРНЫЙ УЧЕБНЫЙ ГРАФИК .....	14
РАБОЧАЯ ПРОГРАММА УЧЕБНОГО КУРСА «Основы программирования на языке Python».....	15
СОДЕРЖАНИЕ КУРСА «Основы программирования на языке Python» ....	23
ОЦЕНОЧНЫЕ МАТЕРИАЛЫ .....	34
МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ .....	35
КАДРОВОЕ ОБЕСПЕЧЕНИЕ .....	39
ОПИСАНИЕ МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЙ БАЗЫ, НЕОБХОДИМОЙ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО КУРСУ .....	39
УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ПРОГРАММЫ.....	39

## ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Программа «Основы программирования на языке Python» разработана в соответствии с требованиями нормативных документов:

Федерального закона РФ от 29.12.2012 г. № 273-ФЗ «Об образовании в Российской Федерации».

Концепции развития дополнительного образования детей, утвержденной распоряжением Правительства РФ от 4 сентября 2014 г. № 1726-р.

Приказа Минпросвещения РФ от 09.11.2018 г. N 196 «Об утверждении

Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам».

Постановления Главного государственного санитарного врача РФ от 28.09.2020 г. № 28 СП 2.4.3648-20 «Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи».

Методических рекомендаций по проектированию дополнительных общеразвивающих программ (письмо Минобрнауки РФ от 18.11.2015 г. N 09-3242).

Методических рекомендаций по созданию и функционированию центров цифрового образования «IT-куб» (утв. распоряжением Министерства просвещения Российской Федерации от 12.01.2021 № Р-5). Паспорт национального проекта «Образование» (утв. президиумом Совета при Президенте РФ по стратегическому развитию и национальным проектам, протокол от 24.12.2018 № 16).

Государственной программы Российской Федерации «Развитие образования» (утв. постановлением Правительства РФ от 26.12.2017 № 1642 (ред. от 15.03.2021) «Об утверждении государственной программы Российской Федерации “Развитие образования”»).

Стратегии развития воспитания в Российской Федерации на период до 2025 года (утв. распоряжением Правительства РФ от 29.05.2015 № 996-р «Об

утверждении Стратегии развития воспитания в Российской Федерации на период до 2025 года»).

# **1. ОСНОВНЫЕ ХАРАКТЕРИСТИКИ ПРОГРАММЫ**

## **1.1. Направленность программы**

Дополнительная общеобразовательная общеразвивающая программа «Основы программирования на языке Python» имеет техническую направленность.

## **1.2. Адресат программы**

Программа адресована обучающимся от 14 до 17 лет.

Программа предназначена для одаренных школьников 7-10 классов, проявляющих повышенный интерес к информатике, математике, анализу данных.

Возрастная категория обучающихся – разновозрастная.

Необходимы базовые знания по следующим школьным предметам: информатика, математика.

Наличие определенной физической и практической подготовки для изучения учебной программы не требуется. Опыт программирования будет способствовать более глубокому освоению материала программы.

## **1.3. Актуальность программы**

Актуальность данной программы состоит в том, что она составлена с учётом современных потребностей рынка в специалистах в области информационных технологий, особенно в области программирования.

Научившись программировать на языке Python, учащиеся получат мощный и удобный инструмент для решения как учебных, так и прикладных задач. Вместе с тем чистота и ясность его конструкций позволит учащимся потом с легкостью выучить любой другой язык программирования.

Знания и умения, приобретенные в результате освоения курса, могут быть использованы обучающимися при сдаче ЕГЭ, при участии в олимпиадах по программированию, при решении задач по физике, химии, биологии,

лингвистике и другим наукам, а также они являются фундаментом для дальнейшего совершенствования мастерства программирования.

Данная программа дает возможность детям творчески мыслить, находить самостоятельные индивидуальные решения, а полученные умения и навыки применять в жизни. Развитие творческих способностей помогает также в профессиональной ориентации подростков.

#### **1.4. Новизна программы**

Новизна программы состоит в том, что она учитывает новые технологические уклады, которые требуют новый способ мышления и тесного взаимодействия при постоянном повышении уровня междисциплинарности проектов, а также использует новые формы диагностики и подведения итогов реализации программы, выполняемые в формате защиты проектов.

В основу программы «Основы программирования на языке Python» заложены принципы модульности практической направленности, что обеспечит вариативность обучения. Содержание учебных модулей направлено на детальное изучение алгоритмизации, реализацию межпредметных связей и организацию проектной и исследовательской деятельности.

**Уровень освоения программы – базовый.**

#### **1.5 Объем и срок освоения программы**

Объем программы – 136 часов.

Срок реализации программы – 1 год.

#### **1.6 Цели и задачи программы**

**Цель** - создание условий для изучения методов программирования на языке Python, рассмотрение различных парадигм программирования, предлагаемых этим языком (процедурная, функциональная, объектно-ориентированная), подготовка к использованию как языка программирования,

так и методов программирования на Python в учебной и последующей профессиональной деятельности в различных предметных областях.

## **Задачи программы**

### **1. Обучающие:**

Основные обучающие цели программы:

- познакомить с принципами и методами фундаментального программирования;
- познакомить с принципами и методами объектно-ориентированного программирования;
- привить навыки работы в интегрированной среде разработки на языке Python;
- изучить конструкции языка программирования Python;
- привить навыки разработки эффективных алгоритмов и программ на основе изучения языка программирования Python;
- дать представление о значении информационных технологий в развитии общества и в изменении характера труда человека.

### **2. Развивающие:**

Обучающиеся в процессе изучения образовательной программы получают возможность:

- развить навыки алгоритмического и логического мышления, грамотной разработки программ;
- сформировать и развить навыки поиска информации в сети Интернет, анализа выбранной информации на соответствие запросу, использования информации при решении задач;
- развивать познавательные способности, память, внимание, пространственное мышление, аккуратность и изобретательность;

- формировать творческий подход к поставленной задаче;
- развивать навыки инженерного мышления, умения работать как по предложенным инструкциям, так и находить свои собственные пути решения поставленных задач;
- развивать навыки эффективной деятельности в проекте;
- развивать стрессоустойчивость;
- формировать и развивать навыки публичного выступления;
- развивать способности к самоанализу, самопознанию;
- формировать навыки рефлексивной деятельности.

### 3. Воспитательные:

Воспитательными задачами, реализуемыми в процессе освоения программы учениками, являются следующие:

- воспитать мотивацию учащихся к изобретательству, созданию собственных программных реализаций;
- привить стремление к получению качественного законченного результата в проектной деятельности;
- привить информационную культуру: ответственное отношение к информации с учетом правовых и этических аспектов её распространения, избирательного отношения к полученной информации;
- формировать правильное восприятие системы ценностей, принципов, правил информационного общества;
- формировать потребность в самостоятельном приобретении и применении знаний, потребность к постоянному саморазвитию;
- воспитывать социально-значимые качества личности человека: ответственность, коммуникабельность, добросовестность, взаимопомощь, доброжелательность.



## 1.7. Планируемые результаты освоения программы

### 1. Предметные результаты:

- знание правил работы с персональным компьютером и технику безопасности;
- понимание назначения и функции используемых информационных технологий;
- знание особенностей работы с интегрированной средой разработки;
- оперирование базовыми и сложными конструкциями, знание способов организации процедур и функций в языке программирования Python;
- умение эффективно использовать интегрированную среду разработки;
- определять результат выполнения алгоритма при заданных исходных данных, узнавать изученные алгоритмы обработки чисел и числовых последовательностей, создавать на их основе программы анализа данных, читать и понимать программы, написанные на языке программирования высокого уровня Python;
- выполнять пошагово алгоритмы управления исполнителями и анализа числовых и текстовых данных;
- разрабатывать и использовать компьютерно-математические модели, оценивать числовые параметры моделируемых объектов и процессов, интерпретировать результаты, получаемые в ходе моделирования реальных процессов, анализировать готовые модели на предмет соответствия реальному объекту или процессу;
- умение писать код программы на языке Python, используя специальные средства и библиотеки языка Python.

### 2. Метапредметные результаты:

- умение искать информацию с применением правил поиска в компьютерных сетях при выполнении заданий и проектов по различным темам;
- умение следовать требованиям техники безопасности, гигиены, эргономики и ресурсосбережения при работе со средствами информационных и

коммуникационных технологий;

- навыки проектирования, разработки, документирования и представления собственных проектов в составе команды;

- умение самостоятельно определять цели своего обучения, ставить и формулировать для себя новые задачи в учебе и познавательной деятельности, развивать мотивы и интересы своей познавательной деятельности;

- умение использовать принципы тайм-менеджмента;

- умение соотносить свои действия с планируемыми результатами, осуществлять контроль своей деятельности в процессе достижения результата, определять способы действий в рамках предложенных условий и требований, корректировать свои действия в соответствии с изменяющейся ситуацией;

- умение оценивать правильность выполнения учебной задачи, собственные возможности ее решения;

- владение основами самооценки, принятия решений и осуществления осознанного выбора в учебной и познавательной деятельности;

- умение определять понятия, создавать обобщения, устанавливать аналогии, классифицировать, самостоятельно выбирать основания и критерии для классификации, устанавливать причинно-следственные связи, строить логическое рассуждение, умозаключение (индуктивное, дедуктивное и по аналогии) и делать выводы;

- умение организовывать учебное сотрудничество и совместную деятельность с учителем и сверстниками; работать индивидуально и в команде; формулировать, аргументировать и отстаивать свое мнение.

### 3. Личностные результаты:

- готовность и способность обучающихся к самообразованию, саморазвитию, личностному и профессиональному самоопределению;

- сформированность мотивации к обучению и целенаправленной познавательной деятельности;

- систему значимых социальных и межличностных отношений,

ценностно-смысловых установок, отражающих личностные и гражданские позиции в деятельности;

- умение ставить цели и строить жизненные планы;
- способность к осознанию российской идентичности в поликультурном социуме.

## **2. ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИЕ УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ**

### **2.1 Язык реализации программы**

Реализация дополнительной общеобразовательной общеразвивающей программы «Основы программирования на языке Python» осуществляется на государственном языке Российской Федерации.

### **2.2. Форма обучения:**

- очная.

### **2.3. Особенности реализации программы**

Программа реализуется по модульному принципу.

### **2.4. Условия набора и формирования групп**

#### **Условия набора обучающихся.**

На обучение зачисляются ученики 7-10 классов общеобразовательных организаций Ставропольского края.

Зачисление на обучение осуществляется по результатам конкурсного отбора в соответствии Положением об образовательном проекте «Яндекс Лицей», реализуемого в рамках дополнительного образования детей.

Условия конкурсного отбора гарантируют соблюдение прав обучающихся в области дополнительного образования и обеспечивают зачисление наиболее способных и подготовленных обучающихся к освоению программы.

Условия формирования групп: разновозрастная.

## **2.5. Формы организации и проведение занятий**

Формы организации занятий:

- аудиторные (под непосредственным руководством преподавателя).

Формы проведения занятий:

- теоретические;
- практические;
- лабораторные;
- контрольные.

Формы организации деятельности обучающихся:

*Интерактивные проблемные лекции* - предполагает наиболее полное вовлечение всех участников лекционного занятия в процесс изучаемого материала, демонстрация слайд-презентации или фрагментов учебных фильмов.

*Мозговой штурм* - предполагает генерацию идей, которую применяют для выявления проблем или поиска решений

*Практикум* – предполагает выполнение практических заданий.

Режим занятий.

Очная форма обучения: 7-10 классы – 2 урока 2 раза в неделю. Программа реализуется в г. Михайловске.

## УЧЕБНЫЙ ПЛАН

Наименование модуля, учебного курса	Количество часов			Форма контроля/ аттестации
	Теория	Практика	Всего	
<b>Модуль 1.</b> Введение в программирование на Python.	9	16	25	Самостоятельная работа № 1
<b>Модуль 2.</b> Множества, строки, списки и словари в Python.	11	15	26	Самостоятельная работа № 2
<b>Модуль 3.</b> Функции в Python.	7	14	21	Контрольная работа №1
<b>Модуль 4.</b> Библиотеки Python.	5	13	18	Самостоятельная работа № 3
<b>Модуль 5.</b> Объектно-ориентированное программирование.	8	13	21	Контрольная работа №2
<b>Модуль 6.</b> Практическое применение языка Python.	11	14	25	Проектная работа
<b>Итого:</b>	<b>51</b>	<b>85</b>	<b>136</b>	

## КАЛЕНДАРНЫЙ УЧЕБНЫЙ ГРАФИК

Наименование модуля, учебного курса	Год обучения	Дата начала обучения	Дата окончания обучения	Количество учебных недель	Количество учебных дней	Количество учебных часов	Режим занятий
<b>Модуль 1.</b> Введение в программирование на Python.	1 год обучения	04.10.2024	08.11.2024	3	12	25	2 урока 2 раза в неделю
<b>Модуль 2.</b> Множества, строки, списки и словари в Python.		11.11.2024	23.12.2024	4	14	26	
<b>Модуль 3.</b> Функции в Python.		27.12.2024	24.01.2025	2	8	21	
<b>Модуль 4.</b> Библиотеки Python.		27.01.2025	03.03.2025	3	12	18	
<b>Модуль 5.</b> Объектно-ориентированное программирование.		07.03.2025	31.03.2025	3	11	21	
<b>Модуль 6.</b> Практическое применение языка Python.		04.04.2025	19.05.2025	4	13	25	

## РАБОЧАЯ ПРОГРАММА УЧЕБНОГО КУРСА «Основы программирования на языке Python»

Курс «Основы программирования на языке Python» предназначен для обучающихся 7-10 классов.

Курс знакомит обучающихся с принципами и методами фундаментального и объектно-ориентированного программирования, развивает навыки работы в интегрированной среде разработки на языке Python, даёт практические навыки применения конструкций языка программирования Python, навыки разработки эффективных алгоритмов и программ, понимание значения информационных технологий в развитии общества и в изменении характера труда человека.

### **Модуль 1.** Введение в программирование на Python.

Изучая данный модуль, обучающиеся разрабатывают первые алгоритмы и программы, а также анализируют, на какие функциональные блоки может быть разбита программа, и определяют работоспособность разработанной программы.

В течение модуля ученики начинают решать задачи: от самых простых до продвинутых.

*Учащиеся должны знать:*

- базовые конструкции языка программирования Python (операции присваивания, ветвления, цикла, ввод\вывод, запись констант и выражений);
- основные методы ветвления программы - условные операторы;
- логические типы данных, прерывания, переходы (сброс циклов на определённой итерации, пропуск итерации при выполнении условий).

*Учащиеся должны уметь:*

- устанавливать необходимый инструментарий для разработки приложений;
- составлять программный код с использованием разных типов данных

и операций над ними;

- составлять, отлаживать и исполнять на персональном компьютере программы решения учебных задач по программированию;
- применять условные конструкции;
- решать задачи с использованием разных видов циклов;
- искать информацию в свободных источниках и структурировать ее;
- объективно оценивать результаты своей работы.

## **Модуль 2. Множества, строки, списки и словари в Python.**

Изучая данный модуль, обучающиеся получают знания о языке высокого уровня Python, структуре программ, строковых переменных и константах, работе с массивами, конкатенацией, основными управляющими конструкциями алгоритмов с ветвлением, методов split и join и примеров работы с ними.

Так же рассматривается работа со списками, строками, множествами и кортежами в Python, понятие словаря и особенности работы со словарями в Python. Для закрепления материала ученики решают задачи.

*Учащиеся должны знать:*

- базовые конструкции языка программирования Python (операции присваивания, ветвления, цикла, ввод\вывод, запись констант и выражений);
- основные методы сортировки списков;
- алгоритмы обработки списков (поиск элемента с заданными свойствами, удаление элементов, вставка элементов, замена элементов);
- способы организации процедур и функций.

*Учащиеся должны уметь:*

- записывать конструкции языка программирования Python;
- тестировать и осуществлять отладку программ;
- объективно оценивать результаты своей работы.

## **Модуль 3. Функции в Python.**



В данном модуле рассматриваются: понятие функции, возвращаемые значения, области видимости переменных, определение объекта, принимающего аргументы и возвращающего значения.

Для закрепления и проверки усвоения материала проводится самостоятельная работа: работа с online-методичкой, в которой описываются способы решения задач, разработка решений по заданиям.

*Учащиеся должны знать:*

- основные понятия объектно-ориентированного проектирования;
- понятия ассоциативного массива, репозитория;
- модульный принцип компоновки программ;
- алгоритмы решения основных видов прикладных задач.

*Учащиеся должны уметь:*

- генерировать идеи;
- создавать и использовать стандартные функции языка Python;
- работать с документацией в интернете;
- объективно оценивать результаты своей работы.

#### **Модуль 4. Библиотеки Python.**

Материал данного модуля предполагает изучение таких вопросов, как использование модулей в Python, подключение и использование модулей стандартной библиотеки, модульный принцип компоновки программы, работа с документацией в стандартной библиотеке, понятие репозитория различных пакетов Python, работа с внешними библиотеками Python и утилитой pip. Решение задач.

*Учащиеся должны знать:*

- стандартные подключаемые библиотеки и модули Python;
- модульный принцип компоновки программ;
- способы поиска и добавления библиотек python, не входящих в стандартный набор.

*Учащиеся должны уметь:*

- подключать и использовать модули стандартной библиотеки;
- работать с документацией в стандартной библиотеке;
- работать с внешними библиотеками и утилитой pip;
- объективно оценивать результаты своей работы.

### **Модуль 5. Объектно-ориентированное программирование.**

Материал данного модуля предполагает изучение таких вопросов, как структурирование данных в виде объектов определённого типа, формализация основных аспектов задачи для создания классов, инкапсуляция логики работы программы внутри объекта вместе с относящейся к нему информацией и наследование логики и информации от одного или нескольких объектов.

Форма закрепления материал - решение задач.

*Учащиеся должны знать:*

- основные понятия объектно-ориентированного проектирования;
- динамические, статические и классовые методы;
- способы переопределения класса его дочерними классами.

*Учащиеся должны уметь:*

- создавать экземпляры классов;
- определять атрибуты и методы внутри класса;
- вызывать определённый метод у объекта;
- учитывать предыдущие ошибки при работе.

### **Модуль 6. Практическое применение языка Python.**

Изучение данного модуля предполагает рассмотрение таких вопросов, как решение повседневных задач средствами Python, автоматизация рутинных процессов, разбор примеров олимпиадных задач и решений к ним. В завершении изучения материала ученики выполняют проектную работу: работу с конспектом, который описывает способы решения задач, поиск информации в интернете, самостоятельная работа над решением задач.

Тематика проектной работы:

- задачи, в которых по заданному алгоритму необходимо написать программу;

- задачи, в которых необходимо составить алгоритм решения и написать по нему программу;

- задачи, для решения которых необходимо найти некоторую информацию в Интернете.

Выполненная проектная работа презентуется перед одноклассниками.

*Учащиеся должны знать:*

- способы создания графического интерфейса программы;
- основные математические методы анализа данных;
- алгоритмы решения прикладных задач программными методами.

*Учащиеся должны уметь:*

- создавать программы с поточной обработкой данных;
- работать с функциями, вызывающими сами себя;
- искать необходимые для решения задач библиотеки и модули в интернете;
- адекватно воспринимать критику со стороны более опытных разработчиков.

## Тематический план курса

№	Наименование кейса, темы	Количество часов			Формы контроля
		Теория	Практика	Всего	
<b>Модуль 1. Введение в программирование на Python.</b>		<b>9</b>	<b>16</b>	<b>25</b>	
1	Тема 1.1. Знакомство со средой.	1	2	3	Опрос
2	Тема 1.2. Условный оператор.	1	2	3	Опрос
3	Тема 1.3. Простые встроенные функции.	1	2	3	Решение задач
4	Тема 1.4. Знакомство с циклом while.	1	2	3	Решение задач
5	Тема 1.5. Отладчик.	1	2	3	Решение задач
6	Тема 1.6. Знакомство с циклом for.	1	2	3	Решение задач
7	Тема 1.7. Погружение в условия. True, False, Break, Continue.	1	2	3	Решение задач
8	Тема 1.8. Вложенные циклы.	2	2	4	Самостоятельная работа № 1
<b>Модуль 2. Множества, строки, списки и словари в Python.</b>		<b>11</b>	<b>15</b>	<b>26</b>	
9	Тема 2.1. Множества	1	2	3	Решение задач
10	Тема 2.2. Строки. Индексация	1	2	3	Решение задач
11	Тема 2.3. Строки. Срезы.	1	2	3	Решение задач
12	Тема 2.4. Знакомство со списками.	1	2	3	Решение задач
13	Тема 2.5. Кортежи. Преобразование коллекций.	2	1	3	Решение задач
14	Тема 2.6. Методы Split и join. Списочные	2	1	3	Решение задач

	выражения.				
15	Тема 2.7. Методы списков и строк.	1	2	3	Решение задач
16	Тема 2.8. Вложенные списки.	1	2	3	Решение задач
17	Тема 2.9. Знакомство со словарями.	1	1	2	Самостоятельная работа № 2
<b>Модуль 3. Функции в Python.</b>		<b>7</b>	<b>14</b>	<b>21</b>	
18	Тема 3.1. Функции.	1	2	3	Решение задач
19	Тема 3.2. Функции. Возвращение значений из функций.	1	2	3	Решение задач
20	Тема 3.3. Области видимости переменных.	1	2	3	Решение задач
21	Тема 3.4. Функции: передача параметров.	1	2	3	Решение задач
22	Тема 3.5. Функции с переменным числом аргументов.	1	2	3	Решение задач
23	Тема 3.6. Функции как объект. Лямбда-функции.	1	2	3	Решение задач
24	Тема 3.7. Обработка коллекций. Поточковый ввод sys.stdin.	1	2	3	Контрольная работа №1
<b>Модуль 4. Библиотеки Python.</b>		<b>5</b>	<b>13</b>	<b>18</b>	
25	Тема 4.1. Random.	1	2	3	Решение задач
26	Тема 4.2. Графика.	1	2	3	Решение задач
27	Тема 4.3. Графика + звук.	1	3	4	Решение задач
28	Тема 4.4. Морфология.	1	3	4	Решение задач
29	Тема 4.5. Документы.	1	3	4	Самостоятельная работа № 3
<b>Модуль 5. Объектно-ориентированное программирование.</b>		<b>8</b>	<b>13</b>	<b>21</b>	

30	Тема 5.1. Введение в ООП.	2	4	6	Решение задач
31	Тема 5.2. Полиморфизм.	2	3	5	Решение задач
32	Тема 5.3. Определение операторов.	2	3	5	Решение задач
33	Тема 5.4. Наследование.	2	3	5	Контрольная работа №2
<b>Модуль 6. Практическое применение языка Python.</b>		<b>11</b>	<b>14</b>	<b>25</b>	
34	Тема 6.1. Основы ведения проектной деятельности и «Тайм-менеджмент».	1	1	2	Решение задач
35	Тема 6.2. Постановка проблемной ситуации и поиск путей решения. Деление на команды.	2	3	5	Решение задач
36	Тема 6.3. Работа команд над реализацией своего проекта.	2	3	5	Решение задач
37	Тема 6.4. Работа над возможными ошибками и недочётами в проекте.	1	2	3	Решение задач
38	Тема 6.5. Подытожим и суммируем полученные навыки и знания по результатам работы над проектом.	1	2	3	Проектная работа
39	Тема 6.6. Готовим презентацию для итоговой защиты.	2	2	4	Проектная работа
40	Тема 6.7. Итоговая защита.	2	1	3	Проектная работа
<b>Итого</b>		<b>51</b>	<b>85</b>	<b>136</b>	

## СОДЕРЖАНИЕ КУРСА «Основы программирования на языке Python»

### **Модуль 1. Введение в программирование на Python.**

Знакомство с виртуальной средой взаимодействия: регистрация, организация личного кабинета, поиск и выкладывание материалов. Знакомство с системой автоматизированной проверки задач.

Основные понятия программирования: исполнитель, система команд, алгоритм, программа, среда разработки, интерпретатор, код программы и редактор кода.

Ввод-вывод в программе, условный оператор, оператор цикла с предусловием. Простейшие программы с использованием условного оператора if, оператора циклов while и операторов ввода-вывода. Технология разработки программы.

На этом этапе обучающиеся разрабатывают первые алгоритмы и программы, а также анализируют, на какие функциональные блоки может быть разбита программа, и определяют работоспособность разработанной программы.

В течение модуля ученики начинают решать задачи: от самых простых до продвинутых.

*Учащиеся должны знать:*

- базовые конструкции языка программирования Python (операции присваивания, ветвления, цикла, ввод\вывод, запись констант и выражений);
- основные методы ветвления программы - условные операторы;
- логические типы данных, прерывания, переходы (сброс циклов на определённой итерации, пропуск итерации при выполнении условий).

*Учащиеся должны уметь:*

- устанавливать необходимый инструментарий для разработки

приложений;

- составлять программный код с использованием разных типов данных и операций над ними;
- составлять, отлаживать и исполнять на персональном компьютере программы решения учебных задач по программированию;
- применять условные конструкции;
- решать задачи с использованием разных видов циклов;
- искать информацию в свободных источниках и структурировать ее;
- объективно оценивать результаты своей работы.

*Формы занятий*, используемые при изучении данного модуля:

- лекционная,
- групповая работа,
- групповые консультации,
- самостоятельная работа,
- контрольная работа.

**Тема 1.1.** Знакомство со средой.

*Теория.* Знакомство с теми техническими средствами, которые понадобятся детям для обучения.

*Практика.* Ознакомительная работа с системой.

**Тема 1.2.** Условный оператор.

*Теория.* Изучаем оператор ветвления (if), реализующий выполнение определённых команд при условии, что некоторое логическое выражение истинно.

*Практика.* Разбор примеров программ с использованием ветвления. Вычисление логических выражений с помощью операции ветвления средствами языка python. Решение задач по теме.

**Тема 1.3.** Простые встроенные функции.

*Теория.* Знакомство с типами данных и функциями.

*Практика.* Разбор примеров создания и использования функций.

**Тема 1.4.** Знакомство с циклом while.



*Теория.* Оператор цикла с постусловием. Оператор с неизвестным числом повторов.

*Практика.* Решение задач с использованием цикла while.

**Тема 1.5.** Отладчик.

*Теория.* Проверка программы на ошибки. Способы поиска ошибок и их исправлений.

*Практика.* Примеры задач, перестающие работать при неверных входных данных. Самостоятельный поиск возможностей ошибки.

**Тема 1.6.** Знакомство с циклом for.

*Теория.* Изучаем цикл с известным количеством повторений.

*Практика.* Решение задач с использованием цикла for.

**Тема 1.7.** Погружение в условия. True, False, Break, Continue.

*Теория.* Принцип работы операторов True, False, Break, Continue в циклах.

*Практика.* Разбор примеров и решений усложнённых задач с использованием операторов True, False, Break, Continue.

**Тема 1.8.** Вложенные циклы.

*Теория.* Использование циклов внутри тела других циклов. Внутренние и внешние циклы.

*Практика.* Решение задач по теме «Вложенные циклы»

*Форма подведения итогов:* самостоятельная работа по теме модуля.

## **Модуль 2. Множества, строки, списки и словари в Python.**

Понятие о языке высокого уровня Python. Структура программы, строковые переменные и константы, работа с массивами, конкатенация в Python. Основные управляющие конструкции алгоритмов с ветвлением в Python. Принципы методов split и join. Примеры работы с ними.

Работа со списками, строками, множествами и кортежами в Python. Понятие словаря. Особенности работы со словарями в python. Решение задач.

*Учащиеся должны знать:*

- базовые конструкции языка программирования Python (операции присваивания, ветвления, цикла, ввод\вывод, запись констант и выражений);
- основные методы сортировки списков;
- алгоритмы обработки списков (поиск элемента с заданными свойствами, удаление элементов, вставка элементов, замена элементов);
- способы организации процедур и функций.

*Учащиеся должны уметь:*

- записывать конструкции языка программирования Python;
- тестировать и осуществлять отладку программ;
- объективно оценивать результаты своей работы.

*Формы занятий, используемые при изучении данного модуля:*

- лекционная,
- групповая работа,
- групповые консультации,
- самостоятельная работа,
- контрольная работа.

### **Тема 2.1. Множества.**

*Теория.* Структура множества. Неповторяющиеся элементы в случайном порядке.

*Практика.* Решение задач с использованием множеств.

### **Тема 2.2. Строки. Индексация.**

*Теория.* Определение понятия строки. Посимвольная нумерация.

*Практика.* Решение задач по теме «Строки. Индексация.»

### **Тема 2.3. Строки. Срезы.**

*Теория.* Понятие срезов строки. обращение к элементам по индексу.

*Практика.* Решение задач по теме «Строки. Срезы.»

### **Тема 2.4. Знакомство со списками.**

*Теория.* Элементы списка. Создание и обращение к спискам.

*Практика.* Решение задач с использованием списков.

### **Тема 2.5. Кортежи. Преобразование коллекций.**

*Теория.* Определение кортежа. Отличия от списков. Хранение элементов различных типов

*Практика.* Решение задач на тему «Кортежи. Преобразование коллекций».

**Тема 2.6.** Методы Split и join. Списочные выражения.

*Теория.* Работа со списками. Применение методов split и join.

*Практика.* Решение задач на тему «Методы split и join».

**Тема 2.7.** Методы списков и строк.

*Теория.* Компьютерная графика и её виды. Кроссплатформенная библиотека для разработки графического интерфейса TKInter.

*Теория.* Генераторы списков. Методы find и rfind. Метод replace и count.

*Практика.* Решение задач на методы списков и строк.

**Тема 2.8.** Вложенные списки.

*Теория.* Обработка и вывод вложенных списков.

*Практика.* Решение задач на тему «Вложенные списки».

**Тема 2.9.** Знакомство со словарями.

*Теория.* Словари (dict) и работа с ними. Методы словарей.

*Практика.* Решение задач на тему «Знакомство со словарями».

*Форма подведения итогов:* самостоятельная работа по теме модуля.

### **Модуль 3. Функции в Python.**

Понятие функции. Возвращаемые значения. Области видимости переменных. Определение объекта, принимающего аргументы и возвращающего значения.

Самостоятельная работа: работа с online-методичкой, в которой описываются способы решения задач, разработка решений по заданиям.

*Учащиеся должны знать:*

- основные понятия объектно-ориентированного проектирования;
- понятия ассоциативного массива, репозитория;

- модульный принцип компоновки программ;
- алгоритмы решения основных видов прикладных задач.

*Учащиеся должны уметь:*

- генерировать идеи;
- создавать и использовать стандартные функции языка Python;
- работать с документацией в интернете;
- объективно оценивать результаты своей работы.

*Формы занятий, используемые при изучении данного модуля:*

- лекционная,
- групповая работа,
- групповые консультации,
- самостоятельные работы,
- контрольная работа.

### **Тема 3.1. Функции.**

*Теория.* Понятие функции. Способы определения функций.

*Практика.* Решение задач по теме урока.

### **Тема 3.2. Функции. Возвращение значений из функций.**

*Теория.* Оператор Return. Ограничения на возвращаемые значения.

*Практика.* Решение задач по теме урока.

### **Тема 3.3. Области видимости переменных.**

*Теория.* Локальная, глобальная и нелокальная области видимости

*Практика.* Решение задач по теме урока.

### **Тема 3.4. Функции: передача параметров.**

*Теория.* Ключевые аргументы, область видимости аргументов.

*Практика.* Решение задач по теме урока.

### **Тема 3.5. Функции с переменным числом аргументов.**

*Теория.* Изменяемое количество передаваемых аргументов при вызове функции.

*Практика.* Решение задач по теме урока.

### **Тема 3.6. Функции как объект. Лямбда-функции.**

*Теория.* Создание и вызов функции в одной строке кода. Анонимные функции.

*Практика.* Решение задач по теме урока.

**Тема 3.7.** Обработка коллекций. Поточковый ввод `sys.stdin`.

*Теория.* Взаимодействие с интерпретатором python напрямую. Хранение набора значений одного или различных типов.

*Практика.* Решение задач по теме урока.

*Форма подведения итогов:* самостоятельная работа по теме модуля.

#### **Модуль 4. Библиотеки Python.**

Модули в Python. Подключение и использование модулей стандартной библиотеки. Модульный принцип компоновки программы. Работа с документацией в стандартной библиотеке. Понятие репозитория различных пакетов Python. Работа с внешними библиотеками Python и утилитой `pip`. Решение задач.

*Учащиеся должны знать:*

- стандартные подключаемые библиотеки и модули Python;
- модульный принцип компоновки программ;
- способы поиска и добавления библиотек python, не входящих в стандартный набор.

*Учащиеся должны уметь:*

- подключать и использовать модули стандартной библиотеки;
- работать с документацией в стандартной библиотеке;
- работать с внешними библиотеками и утилитой `pip`;
- объективно оценивать результаты своей работы.

*Формы занятий,* используемые при изучении данного модуля:

- лекционная,
- групповая работа,
- групповые консультации,
- самостоятельная работа.

#### **Тема 4.1. Random.**

*Теория.* Знакомство с модулем для генерации случайного набора элементов. Управление случайной генерацией. Ограничение выборки и типов элементов.

*Практика.* Решение задач по теме урока.

#### **Тема 4.2. Графика.**

*Теория.* Визуализация математических вычислений. Создание анимированных изображений.

*Практика.* Решение задач по теме урока.

#### **Тема 4.3. Графика + звук.**

*Теория.* Построение спектрограммы. Визуализация звуковых волн.

*Практика.* Решение задач по теме урока.

#### **Тема 4.4. Морфология.**

*Теория.* Морфологический анализ языка. Определение характеристик слова на основе того, как это слово пишется.

*Практика.* Решение задач по теме урока.

#### **Тема 4.5. Документы.**

*Теория.* Открытие, закрытие, чтение и изменение документов.

*Практика.* Решение задач по теме урока.

*Форма подведения итогов:* самостоятельная работа по теме модуля.

### **Модуль 5. Объектно-ориентированное программирование.**

Структурирование данных в виде объектов определённого типа. Формализация основных аспектов задачи для создания классов. Инкапсуляция логики работы программы внутрь объекта вместе с относящейся к нему информацией. Наследование логики и информации от одного или нескольких объектов.

Решение задач.

*Учащиеся должны знать:*

– основные понятия объектно-ориентированного проектирования;

- динамические, статические и классовые методы;
- способы переопределения класса его дочерними классами.

*Учащиеся должны уметь:*

- создавать экземпляры классов;
- определять атрибуты и методы внутри класса;
- вызывать определённый метод у объекта;
- учитывать предыдущие ошибки при работе.

*Формы занятий, используемые при изучении данного модуля:*

- лекционная,
- групповая работа,
- групповые консультации,
- самостоятельная работа.

### **Тема 5.1. Введение в ООП.**

*Теория.* Понятия классов, объектов, методов. Преимущества использования ООП.

*Практика.* Решение задач по теме урока.

### **Тема 5.2. Полиморфизм.**

*Теория.* Способность использовать в ООП общий интерфейс для нескольких форм (типов данных).

*Практика.* Решение задач по теме урока.

### **Тема 5.3. Определение операторов.**

*Теория.* Реализации полиморфизма, когда можно задать отдельную реализацию какого-либо метода в своём классе.

*Практика.* Решение задач по теме урока.

### **Тема 5.4. Наследование.**

*Теория.* Использование методов и свойств уже существующего объекта при создании нового.

*Практика.* Решение задач по теме модуля.

## **Модуль 6. Практическое применение языка Python.**

Решение повседневных задач средствами Python. Автоматизация рутинных процессов. Разбор примеров олимпиадных задач и решений к ним.

Проектная работа: работа с конспектом, который описывает способы решения задач, поиск информации в интернете, самостоятельная работа над решением задач.

Тематика проектной работы:

- задачи, в которых по заданному алгоритму необходимо написать программу;
- задачи, в которых необходимо составить алгоритм решения и написать по нему программу;
- задачи, для решения которых необходимо найти некоторую информацию в Интернете.

Выполненная проектная работа презентуется перед одноклассниками.

*Учащиеся должны знать:*

- способы создания графического интерфейса программы;
- основные математические методы анализа данных;
- алгоритмы решения прикладных задач программными методами.

*Учащиеся должны уметь:*

- создавать программы с поточной обработкой данных;
- работать с функциями, вызывающими сами себя;
- искать необходимые для решения задач библиотеки и модули в интернете;
- адекватно воспринимать критику со стороны более опытных разработчиков.

*Формы занятий, используемые при изучении данного модуля:*

- лекционная,
- групповая работа,
- групповые консультации,
- итоговое выступление.



**Тема 6.1.** Основы ведения проектной деятельности и «Тайм-менеджмент».

*Теория.* Что такое проект. Виды проектов. Этапы проектной деятельности. Теория управления временем.

*Практика.* Решение тематических задач.

**Тема 6.2.** Постановка проблемной ситуации и поиск путей решения. Деление на команды.

*Теория.* Учимся искать пути решения, аргументировать свою точку зрения.

*Практика.* Мозговой штурм, решение командных задач.

**Тема 6.3.** Работа команд над реализацией своего проекта.

*Теория.* Консультация по теме задач проекта.

*Практика.* Решение задач по теме проекта.

**Тема 6.4.** Работа над возможными ошибками и недочётами в проекте.

*Теория.* Методы отладки и оптимизации кода.

*Практика.* Исправление неточностей и доработка проекта.

**Тема 6.5.** Подытожим и суммируем полученные навыки и знания по результатам работы над проектом.

*Теория.* Подводим итоги. Рефлексия.

*Практика.* Обсуждение полученных результатов. Внесение финальных изменений в проект.

**Тема 6.6.** Готовим презентацию для итоговой защиты.

*Теория.* Принципы хорошей презентации. Тренировка выступления.

*Практика.* Готовим проект и презентацию по полученным результатам.

**Тема 6.7.** Итоговая защита.

*Теория.* Формируем навыки публичного выступления. Умение усваивать информацию из выступления сотоварищей.

*Практика.* Представляем результат своей работы и полученных навыков. Анализируем полученный результат других выступающих.

## ОЦЕНОЧНЫЕ МАТЕРИАЛЫ

Во время проведения курса предполагается текущий, промежуточный и итоговый контроль. Промежуточная аттестация обучающихся по данной программе проводится в форме проведения самостоятельных и контрольных работ. Кроме того, проверка результатов освоения программы осуществляется постоянно: после изучения каждой темы модуля, учащиеся решают задачи и проходят тестирования в формате пятиминуток.

*Входной контроль* – тестирование и собеседование.

*Текущий контроль* осуществляется на занятиях в течение всего обучения для отслеживания уровня освоения учебного материала программы.

Формы:

- решение задач;
- пятиминутки в виде тестов или задач.

*Промежуточная аттестация* проводится с целью выявления уровня освоения программ обучающимися и уровня развития личностных качеств по завершению каждого курса программы.

Формы:

- самостоятельная работа;
- контрольная работа;
- решение задач;

*Итоговое оценивание* проводится в конце обучения по курсу.

Форма: выполнение проектной работы.

## МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ

Тема модуля	Форма занятий	Приёмы и методы организации образовательного процесса	Дидактический материал. Электронные источники	Техническое оснащение и расходный материал	Форма подведения итогов
<b>Раздел 1. «Основы программирования на языке Python»</b>					
<b>Модуль 1.</b> Введение в программирование на Python	Комбинированная	Объяснительно-иллюстративный. Метод упражнений. Метод практических работ. Метод мозгового штурма. Проблемно-поисковый.	1. Яндекс.Контест, сервис, позволяющий программистам со всего мира соревноваться на предложенных задачах. Сервис способен одновременно обрабатывать терабайты данных, поэтому легко выдержит нагрузку в более чем тысячу участников. ( <a href="https://contest.yandex.ru">https://contest.yandex.ru</a> ); 2. LMS AnyTask – система управления обучением, позволяющая автоматизировать большинство внутренних процессов Яндекс.Лицея ( <a href="http://anytask.org">http://anytask.org</a> ); 3. Сервис PythonTutor, позволяющий визуализировать исполнение кода на языке Python ( <a href="http://pythontutor.com">http://pythontutor.com</a> ); 4. Федеральный портал Единая коллекция образовательных ресурсов ( <a href="http://school-collection.edu.ru">http://school-collection.edu.ru</a> ); 6. Официальная документация языка Python ( <a href="http://docs.python.org">http://docs.python.org</a> ).	<ul style="list-style-type: none"> <li>• Ноутбуки с мышкой и доступом к сети Интернет.</li> <li>• Презентационное оборудование.</li> <li>• Принтер.</li> <li>• Маркерная доска.</li> </ul>	Самостоятельная работа № 1

<p><b>Модуль 2.</b> Множества, строки, списки и словари в Python.</p>	<p>Комбинируемая</p>	<p>Объяснительно-иллюстративный. Метод упражнений. Метод практических работ. Метод мозгового штурма. Проблемно-поисковый.</p>	<p>1. Яндекс.Контест, сервис, позволяющий программистам со всего мира соревноваться на предложенных задачах. Сервис способен одновременно обрабатывать терабайты данных, поэтому легко выдержит нагрузку в более чем тысячу участников. (<a href="https://contest.yandex.ru">https://contest.yandex.ru</a>); 2. LMS AnyTask – система управления обучением, позволяющая автоматизировать большинство внутренних процессов Яндекс.Лицея (<a href="http://anytask.org">http://anytask.org</a>); 3. Сервис PythonTutor, позволяющий визуализировать исполнение кода на языке Python (<a href="http://pythontutor.com">http://pythontutor.com</a>); 4. Федеральный портал Единая коллекция образовательных ресурсов (<a href="http://school-collection.edu.ru">http://school-collection.edu.ru</a>); 6. Официальная документация языка Python (<a href="http://docs.python.org">http://docs.python.org</a>).</p>	<ul style="list-style-type: none"> <li>• Ноутбуки с мышкой и доступом к сети Интернет.</li> <li>• Презентационное оборудование.</li> <li>• Принтер.</li> <li>• Маркерная доска.</li> </ul>	<p>Самостоятельная работа № 2</p>
---	----------------------	---	--	--	-----------------------------------

<p><b>Модуль 3.</b> Функции в Python.</p>	<p>Комбинируемая</p>	<p>Объяснительно-иллюстративный. Метод упражнений. Метод практических работ. Метод мозгового штурма. Проблемно-поисковый.</p>	<p>1. Яндекс.Контест, сервис, позволяющий программистам со всего мира соревноваться на предложенных задачах. Сервис способен одновременно обрабатывать терабайты данных, поэтому легко выдержит нагрузку в более чем тысячу участников. (<a href="https://contest.yandex.ru">https://contest.yandex.ru</a>); 2. LMS AnyTask – система управления обучением, позволяющая автоматизировать большинство внутренних процессов Яндекс.Лицея (<a href="http://anytask.org">http://anytask.org</a>); 3. Сервис PythonTutor, позволяющий визуализировать исполнение кода на языке Python (<a href="http://pythontutor.com">http://pythontutor.com</a>); 4. Федеральный портал Единая коллекция образовательных ресурсов (<a href="http://school-collection.edu.ru">http://school-collection.edu.ru</a>); 6. Официальная документация языка Python (<a href="http://docs.python.org">http://docs.python.org</a>).</p>	<ul style="list-style-type: none"> <li>• Ноутбуки с мышкой и доступом к сети Интернет.</li> <li>• Презентационное оборудование.</li> <li>• Принтер.</li> <li>• Маркерная доска.</li> </ul>	<p>Контрольная работа №1</p>
---	----------------------	---	--	--	------------------------------

<p><b>Модуль 4.</b> Библиотеки Python.</p>	<p>Комбинированная</p>	<p>Объяснительно-иллюстративный. Метод упражнений. Метод практических работ. Метод мозгового штурма. Проблемно-поисковый.</p>	<p>1. Яндекс.Контест, сервис, позволяющий программистам со всего мира соревноваться на предложенных задачах. Сервис способен одновременно обрабатывать терабайты данных, поэтому легко выдержит нагрузку в более чем тысячу участников. (<a href="https://contest.yandex.ru">https://contest.yandex.ru</a>); 2. LMS AnyTask – система управления обучением, позволяющая автоматизировать большинство внутренних процессов Яндекс.Лицея (<a href="http://anytask.org">http://anytask.org</a>); 3. Сервис PythonTutor, позволяющий визуализировать исполнение кода на языке Python (<a href="http://pythontutor.com">http://pythontutor.com</a>); 4. Федеральный портал Единая коллекция образовательных ресурсов (<a href="http://school-collection.edu.ru">http://school-collection.edu.ru</a>); 6. Официальная документация языка Python (<a href="http://docs.python.org">http://docs.python.org</a>).</p>	<ul style="list-style-type: none"> <li>• Ноутбуки с мышкой и доступом к сети Интернет.</li> <li>• Презентационное оборудование.</li> <li>• Принтер.</li> <li>• Маркерная доска.</li> </ul>	<p>Самостоятельная работа № 3</p>
<p><b>Модуль 5.</b> Объектно-ориентированное программирование.</p>	<p>Комбинированная</p>	<p>Объяснительно-иллюстративный. Метод упражнений. Метод практических работ. Метод мозгового штурма. Проблемно-поисковый.</p>	<p>2. LMS AnyTask – система управления обучением, позволяющая автоматизировать большинство внутренних процессов Яндекс.Лицея (<a href="http://anytask.org">http://anytask.org</a>);</p>	<ul style="list-style-type: none"> <li>• Ноутбуки с мышкой и доступом к сети Интернет.</li> <li>• Презентационное оборудование.</li> <li>• Принтер.</li> <li>• Маркерная доска.</li> </ul>	<p>Контрольная работа №2</p>
<p><b>Модуль 6.</b> Практическое применение языка Python.</p>	<p>Комбинированная</p>	<p>Объяснительно-иллюстративный. Метод мозгового штурма. Проблемно-поисковый.</p>	<p>3. Сервис PythonTutor, позволяющий визуализировать исполнение кода на языке Python (<a href="http://pythontutor.com">http://pythontutor.com</a>);</p>	<ul style="list-style-type: none"> <li>• Ноутбуки с мышкой и доступом к сети Интернет.</li> <li>• Презентационное оборудование.</li> <li>• Принтер.</li> <li>• Маркерная доска.</li> </ul>	<p>Проектная работа</p>

## **КАДРОВОЕ ОБЕСПЕЧЕНИЕ**

Преподавание данной программы могут осуществлять педагогические работники, владеющие набором профессиональных навыков в области программирования на языке Python, информационно-телекоммуникационных технологий и информатики, при наличии необходимых компетенций и уровня профильной подготовки.

### **ОПИСАНИЕ МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЙ БАЗЫ, НЕОБХОДИМОЙ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО КУРСУ**

Для реализации курса «Основы программирования на языке Python» помещение должно соответствовать следующим характеристикам:

– аудитории, оборудованы интерактивной доской, проектором, ноутбуком.

– каждый обучающийся выполняет практические работы за отдельным компьютером с сохранением результатов в облачном хранилище.

### **УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ПРОГРАММЫ**

#### **1. Перечень литературы, необходимой для освоения программы:**

##### **1.1. Перечень литературы, использованной при написании программы:**

1. К. Ю. Поляков, Е. А. Еремин. Информатика. Углублённый уровень. Учебник для 10 класса в 2 частях. М.: БИНОМ. Лаборатория знаний, 2014.

2. М. Лутц. Изучаем Python. СПб.: Символ-Плюс, 2011.

3. Задачи по программированию. Под ред. С. М. Окулова, М.: БИНОМ. Лаборатория знаний, 2006.

4. С. М. Окулов. Основы программирования. М.: Бином. Лаборатория знаний, 2012.

## **1.2. Перечень литературы, рекомендованной обучающимся:**

1. Информатика и ИКТ. Задачник-практикум в 2 частях. Под ред. И. Г. Семакина и Е. К. Хеннера. М.: БИНОМ. Лаборатория знаний, 2014.
2. Материалы и презентации к урокам в LMS Яндекс.Лицея.
3. Сайт [pythonworld.ru](http://pythonworld.ru) — «Python 3 для начинающих».
4. Сайт [pythontutor.ru](http://pythontutor.ru) — «Питонтьютор».

## **1.3. Перечень литературы, рекомендованной родителям:**

1. М. Лутц. Изучаем Python. СПб.: Символ-Плюс, 2011.

## **1.4 Перечень раздаточного материала:**

1. Тематические презентации.

## **2. Информационное обеспечение**

Программное обеспечение:

Операционная система (Windows, Linux, macOS), IDE PyCharm

## **2.1 Перечень ресурсов информационно-телекоммуникационной сети**

**«Интернет», необходимых для освоения программы:**

1. <https://www.youtube.com/playlist?list=PLJOzdkh8T5kpIBTG9mM2wVBjh-5OpdwBl> — Лекции А.В. Умнова, прочитанные в Школе Анализа Данных Яндекса.