

УПРАВЛЕНИЕ ОБРАЗОВАНИЯ АДМИНИСТРАЦИИ  
ГУБКИНСКОГО ГОРОДСКОГО ОКРУГА  
МУНИЦИПАЛЬНОЕ БЮДЖЕТНОЕ ОБЩЕОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
«ОБРАЗОВАТЕЛЬНЫЙ КОМПЛЕКС «СТАРТУМ»  
ГОРОДА ГУБКИНА БЕЛГОРОДСКОЙ ОБЛАСТИ  
СТРУКТУРНОЕ ПОДРАЗДЕЛЕНИЕ ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ  
«СТАНЦИЯ ЮНЫХ ТЕХНИКОВ»

Принята на заседании  
педагогического совета  
от «30» 08 2024 г.  
Протокол № 1

УТВЕРЖДАЮ  
Директор МБОУ «ОК «СтартУМ»  
Т.В. Солдатова  
Приказ от «02» сентября 2024 г. № 674



ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ  
ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА

«Язык программирования Python»

Направленность: техническая  
Возраст обучающихся: 14-16 лет  
Срок реализации: 1 год  
Уровень программы: базовый

Автор-составитель:  
Рыбалкина Алина Юрьевна,  
педагог дополнительного образования

Губкин, 2024 г.

## **Раздел 1. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

### **Направленность программы**

Дополнительная общеразвивающая программа «Язык программирования Python» имеет техническую направленность и предназначена для организации внеурочной деятельности по нескольким взаимосвязанным направлениям развития личности, таким как общеинтеллектуальное, общекультурное, социальное. Программа предлагает ее реализацию в факультативной или кружковой форме.

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

### **Уровень программы**

Данная дополнительная общеразвивающая программа ориентирована на один год обучения.

### **Актуальность программы**

Python — это язык программирования общего назначения, распространяемый с открытыми исходными текстами. Он оптимизирован для создания качественного программного обеспечения. Язык Python используется сотнями тысяч разработчиков по всему миру в таких областях, как создание веб-сценариев, системное программирование, создание пользовательских интерфейсов, настройка программных продуктов под пользователя, численное программирование и в других. Как считают многие, один из самых используемых языков программирования в мире.

### **Педагогическая целесообразность**

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

### **Цель программы**

Развить личность обучающегося, способного к творческому самовыражению через овладение основами программирования на одном из самых востребованных языков программирования Python.

### **Задачи программы**

#### ***Образовательные:***

- познакомить с возможностями и особенностями современного языка программирования Python;
- сформировать навыки выполнения технологической цепочки разработки программ средствами языка программирования Python;
- изучить основные конструкции языка программирования Python, позволяющие работать с простыми и составными типами данных (строками, списками, кортежами, словарями, множествами);
- научить применять функции при написании программ на языке

программирования Python;

***Развивающие:***

- развивать творческие способности, алгоритмическое и логическое мышление;
- развивать представления о возможностях применения современных компьютерных технологий в профессиональной деятельности;
- развивать внимательность и наблюдательность, прививать навыки аккуратности и точности в работе;
- повысить интеллектуальный уровень и расширить интеллектуальные навыки.

***Воспитательные:***

- воспитывать навыки взаимодействия при командной работе над проектом;
- воспитывать чувство ответственности за результаты деятельности;
- способствовать формированию культуры программирования;
- способствовать формированию у обучающихся культуры создания и этики представления проектов на конференциях и в Интернете.

**Отличительные особенности данной программы**

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

Занятия начинаются с практического знакомства со средой программирования Python, далее идет непосредственное изучение синтаксических конструкций языка и отработка навыков применения элементов программирования при решении задач и создании игр. Каждая новая тема завершается практическими задачами, способствующими овладению методики программирования и изучению языка Python.

Затем полученные знания и навыки программирования закрепляются, осваиваются новые методы, способы решения задач, рассматриваются более сложные задачи, новые технологии программирования.

**Возраст обучающихся. Условия набора**

Данная дополнительная общеобразовательная программа предназначена для детей 14-16 лет. Набирается 1 группа. В коллектив принимаются все желающие. Набор производится, начиная с 1 сентября текущего года. Возможен дополнительный набор отдельных обучающихся в течение года.

**Срок реализации программы**

Срок реализации программы «Язык программирования Python» составляет 1 год (36 недель, 144 часа). Период обучения: сентябрь - май.

**Формы и режим занятия**

**Форма занятий** – групповая. Наполняемость группы – от 5 до 15 человек. Набор воспитанников осуществляется на добровольной основе, по интересам. Ограничения могут быть только по медицинским показаниям. Состав групп разновозрастной, в зависимости от базовой подготовки

подростка.

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

### **Методы обучения**

Отбор методов обучения обусловлен необходимостью формировать информационную и коммуникативную компетентности обучающихся, реализовывать личностно-ориентированное обучение, направлять их на самостоятельное решение разнообразных проблем, развивать исследовательские и творческие способности. Решение данных задач кроется в организации системно-деятельностного подхода к обучению, в проблемном изложении материала педагогом, в переходе от репродуктивного вида работ к самостоятельным, поисково-исследовательским видам деятельности. Поэтому основная методическая установка в данном курсе — обучение навыкам самостоятельной творческой деятельности.

#### ***Методы проведения занятий***

1. Словесные методы обучения:

- устное изложение материала с презентационным сопровождением;
- беседа.

2. Наглядные методы:

- показ программного кода и результатов его выполнения на тестовых данных;
- показ приёмов отладки программного кода педагогом;
- показ приёмов работы со средой разработки;
- демонстрации различных вариантов оптимизации программного кода.

3. Практические методы обучения: написание и отладка программ.

Методы, в основе которых лежит уровень деятельности детей:

- объяснительно-иллюстративные методы обучения;
- репродуктивные методы обучения;
- частично-поисковые методы обучения;
- проектный метод обучения.

Программа не создает учебных перегрузок для школьников: содержание построено таким образом, что изучение всех последующих тем обеспечивается предыдущими, материал дробится на простые легко воспринимаемые части и полностью усваивается на занятии, домашнее задание носит творческий характер и выполняется по желанию обучающихся.

Санитарно-гигиенические требования при работе в компьютерном классе обеспечены.

Программа состоит из отдельных тем, которые должны изучаться последовательно. В зависимости от предварительной подготовки обучающихся в группе количество часов на изучение темы может варьироваться.

### **Ожидаемые результаты**

В результате обучения по программе обучающиеся овладеют основными навыками программирования на языке Python и будут готовы к дальнейшему

самостоятельному изучению языка программирования и совершенствованию своих навыков в области программирования.

Обучающиеся будут иметь представление о профессии программист и о своих возможностях преуспеть в данной профессии.

По окончании курса обучения обучающиеся:

- познакомятся с технологией создания компьютерных программ, и современными концепциями программирования;
- узнают особенности и возможности языка программирования Python и области его применения;
- научатся составлять алгоритмы для решения задач и реализовывать эти алгоритмы на компьютере в виде программ, написанных на языке Python;
- научатся работать в интерактивной среде программирования и использовать её основные возможности для отладки и тестирования программ;
- получают общие представления о характере труда программиста и специфике его работы;
- овладеют основами функционального программирования на языке Python;
- приобретут навыки объектного программирования;
- смогут самостоятельно создавать и отлаживать несложные приложения на языке программирования Python;
- освоят нестандартные приёмы программирования;
- результатом обучения будет - портфолио электронных учебных работ обучающегося; - самостоятельно выполненный проект.

#### **Виды и формы контроля:**

Текущий контроль знаний проводится на каждом занятии как текущий рефлексивный самоанализ;

- контроль и самооценка обучающимися выполняемых заданий;
- в форме взаимопроверки;
- в форме фронтального опроса;
- в форме письменного опроса;
- в форме выполнения практического задания.

Тематический контроль осуществляется по завершении крупного блока (темы). Он позволяет оценить знания и умения обучающихся, в ходе достаточно продолжительного периода работы.

Тематический контроль проводится в форме собеседования;

- в форме предъявления программного продукта;
- в форме практической работы, ориентированной на применение полученных знаний в нестандартной ситуации.

Итоговый контроль проводится по завершении учебного курса в форме защиты портфолио. Формы подведения итогов реализации дополнительной образовательной программы Предметом диагностики и контроля в курсе «Основы программирования на языке Python» являются внешние образовательные продукты обучающихся (созданные программы), а также их внутренние личностные качества (освоенные способы деятельности, знания, умения), которые относятся к целям и задачам курса. Практическим

результатом обучения воспитанников является также самостоятельно выполненный проект, включающий в себя не только компьютерную программу, но и описание алгоритма её работы, контрольные тесты для её проверки, описание используемых языковых конструкций и оригинальных приёмов программирования, область практического применения и другую документацию, отражающую работу над созданием проекта. Темы проектов выбираются обучающимися как самостоятельно, так и из числа предложенных преподавателем, но всегда обсуждается в группе. Необходимо, чтобы программа имела практическое применение и была интересна не только авторам проекта. Воспитанник должен почувствовать важность выбранной им темы в глазах общества, желание показать своё мастерство и уникальность технологического решения созданной им программы. Обеспечивается возможность презентации проекта на школьной научно-практической конференции, конкурсе или фестивале проектов.

### ***Примерные требования при защите проекта***

Обучающиеся должны (по желанию) продемонстрировать самостоятельно разработанную программу, показать владение техническими, программными средствами разработки проекта, знание основ программирования, уметь ответить на вопросы по содержанию.

По техническим средствам:

- какие технические средства были задействованы при создании проекта;
- объяснить минимальные требования к программному и аппаратному обеспечению для работы программы.

По технологии разработки проект:

- охарактеризовать программные средства, используемые при выполнении проекта;
- объяснить технологические приемы разработки;
- наличие документации;
- рекомендации по работе с программой.

По разработке программы:

- алгоритм должен быть оптимальным по скорости выполнения и максимально простым в реализации на языке программирования;
- код должен быть «читаемым» (должны быть соблюдены отступы, обязательное наличие комментариев к коду программы и т. д.);
- программа должна выполнять поставленные задачи, и правильно работать;
- наличие собственных конструктивных решений;
- знание основ программирования и знание технологий программирования;
- наличие дружественного интерфейса.

По содержанию проекта:

- актуальность работы,
- глубина и логика раскрытия темы,
- форма подачи информации,

- заинтересованность в проведении исследования,
- области применения разработки,
- источники информации.

### Оценка образовательных результатов освоения программы

При подведении итогов учитывается не только результаты диагностики и качество выполнения учебных заданий, но и личные достижения обучающихся.

Рекомендуется в начале и конце курса обучения заполнять информационную карту:

#### «Информационная карта освоения образовательной программы»

№ п/п	ФИО обучающегося	Месяц						Общий суммарный балл	Уровень освоения
		Критерии наблюдения							
		А	Б	В	Г	Д	Е		

Критерии:

- «А» Социальная активность обучающегося;
- «Б» Коммуникабельность, умение работать в коллективе;
- «В» Владение навыками алгоритмизации и знание основных конструкций языка программирования;
- «Г» Навыки анализа, синтеза и культуры представления информации;
- «Д» Умение работать на персональном компьютере в среде программирования;
- «Е» Владение технологией программирования;
- «Ж» Уровень самостоятельности;

По каждому критерию выставляется: от 0 до 5 баллов.

Общий уровень определяется суммированием баллов:

- низкий уровень: от 7 до 20 баллов;
- средний уровень: от 21 до 27 баллов;
- высокий уровень: от 28 до 35 баллов

#### Планируемые (ожидаемые) результаты

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

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

- умение самостоятельно планировать пути достижения цели, в том числе альтернативные, осознанно выбирать наиболее эффективные способы

решения задач;

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

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

- формирование ответственного отношения к обучению, готовности и способности обучающихся к саморазвитию и самообразованию на основе мотивации к обучению и познанию;
- формирование коммуникативной компетентности в общении и сотрудничестве со сверстниками, взрослыми в процессе образовательной, общественно-полезной, учебно-исследовательской, творческой и других видах деятельности.

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

- формирование знаний, умений и навыков при решении задач информатики и программирования разных видов;
- приобретение опыта использования информационных технологий в индивидуальной и коллективной учебной и познавательной, в том числе проектной деятельности.
- формирование информационной и алгоритмической культуры; формирование представления о компьютере как универсальном устройстве обработки информации;
- развитие основных навыков и умений использования компьютерных устройств;
- формирование представления об основных изучаемых понятиях: информация, алгоритм, модель – и их свойства;
- развитие алгоритмического мышления, необходимого для профессиональной деятельности в современном обществе;
- развитие умений составить и записать алгоритм для конкретного исполнителя;
- формирование знаний об алгоритмических конструкциях, логических значениях и операциях;
- знакомство с одним из языков программирования и основными

алгоритмическими структурами – линейной, условной и циклической;

- владение системой базовых знаний, отражающих вклад информатики в формирование современной научной картины мира;

- овладение понятием сложности алгоритма, знание основных алгоритмов обработки числовой и текстовой информации, алгоритмов поиска и сортировки;

- владение универсальным языком программирования высокого уровня Python, представлениями о базовых типах данных и структурах данных; умением использовать основные управляющие конструкции;

- владение навыками и опытом разработки программ в среде программирования Python, включая тестирование и отладку программ; владение элементарными навыками формализации прикладной задачи и документирования программ;

- формирование умения работать с библиотеками программ; получение опыта использования компьютерных средств представления и анализа данных.

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

- основные типы алгоритмов;

- иметь представление о структуре программы, основы программирования на языках высокого уровня;

- базовые алгоритмические конструкции;

- содержание этапов разработки программы: алгоритмизация-кодирование-отладка -тестирование;

- дополнительные возможности языка Python для выражения различных алгоритмических ситуаций;

- алгоритмы и программы на языке Python решения простых, сложных и нестандартных задач в математической области;

- основы разработки простых игр в системе программирования Python.

Обучающиеся должны **уметь:**

- записывать алгоритмические структуры на языке программирования Python;

- использовать Python для решения задач из области математики, физики;

- строить алгоритмы методом последовательного уточнения (сверху вниз), изображать эти алгоритмы в виде блок-схем;

- использовать основные алгоритмические приемы при решении математических задач;

- решать простые, сложные и нестандартные задачи;

- создавать простые игры;

- анализировать текст чужих программ, находить в них неточности, оптимизировать алгоритм, создавать собственные варианты решения.

### **Формы подведения итогов реализации программы**

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

**Раздел 2. Содержание программы.**  
**УЧЕБНЫЙ (ТЕМАТИЧЕСКИЙ) ПЛАН**

№ п/п	Название раздела, темы	Кол-во часов			Формы аттестации/ контроля
		Всего	Теория	Практика	
<b>1</b>	<b>Основы программирования на языке Python.</b>	<b>30</b>	<b>8</b>	<b>22</b>	
1.1.	История языков программирования. Введение в язык программирования Python. Среда программирования Python. Установка программы.	1	1		
1.2.	Типы данных и функции вывода. Определение переменной. Переменные и арифметические выражения.	2	1	1	
1.3.	Чтение данных. Операции над строками. Примеры решения задач.	2	1	1	
1.4.	Отработка навыков решения простейших задач.	4		4	Практические задания
1.5.	Логический тип данных и операции. Примеры использования логических выражений.	1	1		
1.6.	Условный оператор. Вложенный условный оператор. Примеры решения задач.	2	1	1	
1.7.	Отработка навыков решения простейших задач.	4		4	Практические задания
1.8.	Цикл WHILE. Примеры решения задач.	2	1	1	
1.9.	Подсчет суммы и оператор CONTINUE. Примеры решения задач.	2	1	1	
1.10.	Отработка навыков решения простейших задач.	4		4	Практические задания
1.11.	Вещественные числа. Основы работы с вещественными числами. Округление вещественных чисел. Примеры решения задач.	2	1	1	
1.12.	Отработка навыков решения простейших задач.	4		4	Практические задания
<b>2.</b>	<b>Методы программирования на языке Python. Введение в олимпиадное программирование.</b>	<b>42</b>	<b>7</b>	<b>35</b>	
2.1.	Срезы строк. Использование срезов. Метод FIND. Примеры решения задач.	2	1	1	
2.2.	Отработка навыков решения простейших задач.	4		4	Практические задания
2.3.	Методы RFINN, REPLACE и COUNT. Примеры решения задач.	2	1	1	
2.4.	Отработка навыков решения простейших задач.	2		2	Практические задания
2.5.	Решение несложных олимпиадных задач.	4		4	Практические задания
2.6.	Функции. Использование функций. Примеры решения задач.	2	1	1	

2.7.	Возврат значений. Локальные и глобальные переменные. Примеры решения задач.	2	1	1	
2.8.	Отработка навыков решения задач.	4		4	Практические задания
2.9.	Рекурсия. Использование рекурсии. Примеры решения задач.	2	1	1	
2.10.	Отработка навыков решения задач.	4		4	Практические задания
2.11.	Кортежи. Функция RANGE, цикл FOR. Примеры решения задач.	2	1	1	
2.12.	Отработка навыков решения задач.	4		4	Практические задания
2.13.	Списки. Метод SPLIT и JOIN. Примеры решения задач.	2	1	1	
2.14.	Отработка навыков решения задач.	2		2	Практические задания
2.15.	Решение олимпиадных задач.	4		4	Практические задания
<b>3.</b>	<b>Методы программирования на языке Python. Решение олимпиадных задач.</b>	<b>48</b>	<b>11</b>	<b>37</b>	
3.1.	Повторение. Основные конструкции языка программирования Python. Типы данных. Арифметические выражения. Условный оператор. Циклы. Решение задач.	4	2	2	
3.2.	Срезы строк. Кортежи. Методы. Функции и рекурсия в Python. Решение задач.	4	2	2	
3.3.	Списки. Методы работы со списками. Обработка списка. Примеры решения задач.	2	1	1	
3.4.	Отработка навыков решения задач.	4		4	Практические задания
3.5.	Сортировка. Сравнение списков и кортежей. Именованный параметр KEY. Примеры решения задач.	2	1	1	
3.6.	Отработка навыков решения задач.	4		4	Практические задания
3.7.	Структуры в Python. Лямбда- функции. Именованные параметры. Чтение из файла.	2	1	1	
3.8.	Сортировка подсчетом. Примеры решения задач.	2	1	1	
3.9.	Отработка навыков решения задач.	4		4	Практические задания
3.10.	Множества и хеш-функции. Создание и работа с множествами.	2	1	1	
3.11.	Отработка навыков решения задач.	4		4	Практические задания
3.12.	Решение олимпиадных задач.	4		4	Практические задания
3.13.	Словари. Методы строк. Примеры решения сложных задач.	4	2	2	
3.14.	Отработка навыков решения сложных задач.	6		6	Практические задания

<b>4.</b>	<b>Объектно-ориентированное программирование. Решение задач повышенной сложности.</b>	<b>24</b>	<b>4</b>	<b>20</b>	
4.1.	Парадигмы программирования и функциональное программирование. Встроенные функции для работы с последовательностями. Примеры решения задач в функциональном стиле.	2	1	1	
4.2.	Итераторы и генераторы. Генерация комбинаторных объектов ITERTOOLS. Функции PARTIAL, REDUCE, ACCUMULATE. Примеры решения задач.	2	1	1	
4.3.	Отработка навыков решения сложных задач.	4		4	Практические задания
4.4.	Объектно-ориентированное программирование. Комплексные числа. Инкапсуляция и конструкторы. Примеры решения задач.	2	1	1	
4.5.	Классы объектов. Обработка ошибок. Наследование и полиморфизм. Примеры решения задач.	2	1	1	
4.6.	Решение задач повышенной сложности.	4		4	Практические задания
4.7.	Решение сложных олимпиадных задач.	6		6	Практические задания
4.8.	Заключительное занятие	2		2	Практические задания
<b>Всего: 144 часа.</b>					

## СОДЕРЖАНИЕ УЧЕБНОГО (ТЕМАТИЧЕСКОГО) ПЛАНА

### Раздел 1. Основы программирования на языке Python.

Тема 1.1. История языков программирования. Введение в язык программирования Python. Среда программирования Python. Установка программы. Теория (1 ч.)

Тема 1.2. Типы данных и функции вывода. Определение переменной. Переменные и арифметические выражения. Теория (1 ч.). Практика (1 ч.) Тренировочные задания.

Тема 1.3. Чтение данных. Операции над строками. Примеры решения задач. Теория (1 ч.). Практика (1 ч.) Тренировочные задания.

Тема 1.4. Отработка навыков решения простейших задач. Практика (4 ч.) Форма подведения итогов: практические задания.

Тема 1.5. Логический тип данных и операции. Примеры использования логических выражений. Теория (1 ч.).

Тема 1.6. Условный оператор. Вложенный условный оператор. Примеры решения задач. Теория (1 ч.). Практика (1 ч.) Тренировочные задания.

Тема 1.7. Отработка навыков решения простейших задач. Практика (4 ч.) Форма подведения итогов: практические задания.

Тема 1.8. Цикл WHILE. Примеры решения задач. Теория (1 ч.). Практика (1 ч.) Тренировочные задания.

Тема 1.9. Подсчет суммы и оператор CONTINUE. Примеры решения задач. Теория (1 ч.). Практика (1 ч.) Тренировочные задания.

Тема 1.10. Отработка навыков решения простейших задач. Практика (4 ч.) Форма подведения итогов: практические задания.

Тема 1.11. Вещественные числа. Основы работы с вещественными числами. Округление вещественных чисел. Примеры решения задач. Теория (1 ч.). Практика (1 ч.) Тренировочные задания.

Тема 1.12. Отработка навыков решения простейших задач. Практика (4 ч.) Форма подведения итогов: практические задания.

## **Раздел 2. Методы программирования на языке Python. Введение в олимпиадное программирование.**

Тема 2.1. Срезы строк. Использование срезов. Метод FIND. Примеры решения задач. Теория (1 ч.). Практика (1 ч.) Тренировочные задания.

Тема 2.2. Отработка навыков решения простейших задач.

Тема 2.3. Методы RFIN, REPLACE и COUNT. Примеры решения задач.

Тема 2.4. Отработка навыков решения простейших задач. Практика (2 ч.) Форма подведения итогов: практические задания.

Тема 2.5. Решение несложных олимпиадных задач. Практика (4 ч.) Форма подведения итогов: практические задания.

Тема 2.6. Функции. Использование функций. Примеры решения задач. Теория (1 ч.). Практика (1 ч.) Тренировочные задания.

Тема 2.7. Возврат значений. Локальные и глобальные переменные. Примеры решения задач. Теория (1 ч.). Практика (1 ч.) Тренировочные задания.

Тема 2.8. Отработка навыков решения задач. Практика (4 ч.) Форма подведения итогов: практические задания.

Тема 2.9. Рекурсия. Использование рекурсии. Примеры решения задач. Теория (1 ч.). Практика (1 ч.) Тренировочные задания.

Тема 2.10. Отработка навыков решения задач. Практика (4 ч.) Форма подведения итогов: практические задания.

Тема 2.11. Кортежи. Функция RANGE, цикл FOR. Примеры решения задач. Теория (1 ч.). Практика (1 ч.) Тренировочные задания.

Тема 2.12. Отработка навыков решения задач. Практика (4 ч.) Форма подведения итогов: практические задания.

Тема 2.13. Списки. Метод SPLIT и JOIN. Примеры решения задач. Теория (1 ч.). Практика (1 ч.) Тренировочные задания.

Тема 2.14. Отработка навыков решения задач. Практика (2 ч.) Форма подведения итогов: практические задания.

Тема 2.15. Решение олимпиадных задач. Практика (4 ч.) Форма подведения итогов: практические задания.

## **Раздел 3. Методы программирования на языке Python. Решение олимпиадных задач.**

Тема 3.1. Повторение. Основные конструкции языка программирования Python. Типы данных. Арифметические выражения. Условный оператор. Циклы. Решение задач. Теория (2 ч.). Практика (2 ч.) Тренировочные задания.

Тема 3.2. Срезы строк. Кортежи. Методы. Функции и рекурсия в Python. Решение задач. Теория (2 ч.). Практика (2 ч.) Тренировочные задания.

Тема 3.3. Списки. Методы работы со списками. Обработка списка. Примеры решения задач. Теория (1 ч.). Практика (1 ч.) Тренировочные задания.

Тема 3.4. Отработка навыков решения задач. Практика (4 ч.) Форма подведения итогов: практические задания.

Тема 3.5. Сортировка. Сравнение списков и кортежей. Именованный параметр KEY. Примеры решения задач. Теория (1 ч.). Практика (1 ч.) Тренировочные задания.

Тема 3.6. Отработка навыков решения задач. Практика (4 ч.) Форма подведения итогов: практические задания.

Тема 3.7. Структуры в Python. Лямбда-функции. Именованные параметры. Чтение из файла. Теория (1 ч.). Практика (1 ч.) Тренировочные задания.

Тема 3.8. Сортировка подсчетом. Примеры решения задач. Теория (1 ч.). Практика (1 ч.) Тренировочные задания.

Тема 3.9. Отработка навыков решения задач. Практика (4 ч.) Форма подведения итогов: практические задания.

Тема 3.10. Множества и хеш-функции. Создание и работа с множествами. Теория (1 ч.). Практика (1 ч.) Тренировочные задания.

Тема 3.11. Отработка навыков решения задач. Практика (4 ч.) Форма подведения итогов: практические задания.

Тема 3.12. Решение олимпиадных задач. Практика (4 ч.) Форма подведения итогов: практические задания.

Тема 3.13. Словари. Методы строк. Примеры решения сложных задач. Теория (2 ч.). Практика (2 ч.) Тренировочные задания.

Тема 3.14. Отработка навыков решения сложных задач. Практика (6 ч.) Форма подведения итогов: практические задания.

#### **Раздел 4. Объектно-ориентированное программирование. Решение задач повышенной сложности.**

Тема 4.1. Парадигмы программирования и функциональное программирование. Встроенные функции для работы с последовательностями. Примеры решения задач в функциональном стиле. Теория (1 ч.). Практика (1 ч.) Тренировочные задания.

Тема 4.2. Итераторы и генераторы. Генерация комбинаторных объектов ITERTOOLS. Функции PARTIAL, REDUCE, ACCUMULATE. Примеры решения задач. Теория (1 ч.). Практика (1 ч.) Тренировочные задания.

Тема 4.3. Отработка навыков решения сложных задач. Практика (4 ч.) Форма подведения итогов: практические задания.

Тема 4.4. Объектно-ориентированное программирование. Комплексные числа. Инкапсуляция и конструкторы. Примеры решения задач. Теория (1 ч.). Практика (1 ч.) Тренировочные задания.

Тема 4.5. Классы объектов. Обработка ошибок. Наследование и

полиморфизм. Примеры решения задач. Теория (1 ч.). Практика (1 ч.)  
Тренировочные задания.

Тема 4.6. Решение задач повышенной сложности. Практика (4 ч.) Форма подведения итогов: практические задания.

Тема 4.7. Решение сложных олимпиадных задач. Практика (6 ч.) Форма подведения итогов: практические задания.

Тема 4.8. Заключительное занятие Практика (2 ч.) Форма подведения итогов: зачетное задание.

### Календарный учебный график

	Дата начала занятий	Дата окончания занятий	Количество о учебных недель	Количество о учебных дней	Количество учебных часов	Режим занятий
1 год обучения	сентябрь	май	36	36	144	2 раза в неделю по 2 часа

### Методическое обеспечение образовательной программы

1. Методическое пособие для учителя к завершённой предметной линии учебников «Информатика» для 10–11 классов общеобразовательных учреждений (углублённый уровень) Авторы: К.Ю. Поляков, Е. А. Еремин. БИНОМ. Лаборатория знаний / Автор-составитель: М. Н. Бородин. — Эл. изд. — М.: БИНОМ. Лаборатория знаний, 2013 ISBN 978-5-9963-1344-

2. «Здоровье и безопасность детей в мире компьютерных технологий и Интернета» Электронное методическое пособие. Партнёрство в образовании. Проект «Академия учителей».

### Материально-технические и информационно-технические средства обучения:

#### *Технические средства:*

1. Персональные компьютеры – 12 рабочих станций;
2. Рабочее место учителя – персональный компьютер, принтер, сканер, звуковые колонки.
3. Оборудование для презентаций – мультимедийный проектор, экран;
4. Маркерная доска.

#### *Программное обеспечение:*

1. Операционная система Windows или Linux;
2. Интегрированные среды программирования Python 3.3 - Python 3.5 и выше.

#### *Цифровые образовательные ресурсы:*

1. Л.Н. Горбунов и др. УМК Здоровье и безопасность детей в мире компьютерных технологий и Интернет. Электронное приложение. Москва. САЛОН-ПРЕСС. 2010;
2. Федеральный портал Единая коллекция образовательных ресурсов (school-collection.edu.ru/)
3. Домашняя страница Python [www.python.org](http://www.python.org) . Справочные материалы,

официальная документация;

4. Сайт проекта Интуит: Национальный открытый университет, курс «Введение в программирование на Python», <http://www.intuit.ru/studies/courses/12179/1172/info>;

5. Сайт проекта Интуит: Национальный открытый университет. Курс «Язык программирования Python» <http://www.intuit.ru/studies/courses/49/49/info>;

6. Сайт проекта Open Book Project [openbookproject.net](http://openbookproject.net) содержит серию практических примеров на Python Криса Мейерса;

7. Интерактивный Python. Trinket <https://trinket.io/python>;

8. Интерактивный учебник языка Питон <http://pythontutor.ru/> ;

9. Python 3 для начинающих <https://pythonworld.ru/> ;

10. Python GUI Programming (Tkinter)

[https://www.tutorialspoint.com/python/python\\_gui\\_programming.htm](https://www.tutorialspoint.com/python/python_gui_programming.htm)

<http://www.tkdocs.com/tutorial/> и <http://effbot.org/tkinterbook/>;

11. Python IDE for beginners <http://thonny.org/> ;

12. Бесплатный, простой и профессиональный инструмент для изучения программирования с помощью Python <https://www.jetbrains.com/pycharm-edu/> ;

13. 5. Язык Python (Электронный ресурс)

<http://kpolyakov.spb.ru/school/probook/python.htm>.

#### *Дидактические материалы:*

1. Раздаточный материал контролирующего и обучающего характера по каждой теме.

2. Задания на практические работы.

3. Карточки с индивидуальными заданиями.

4. Раздаточный материал справочного характера.

5. Раздаточный материал теоретического характера.

6. Демонстрационные материалы в электронном виде.

### **СПИСОК ЛИТЕРАТУРЫ**

#### *Литература для педагога*

1. Поляков К.Ю., Еремин Е.А. Информатика. 10-й класс. Углубленный уровень. В двух частях. М.: Бином, 2013.

2. Поляков К.Ю., Еремин Е.А. Информатика. 11-й класс. Углубленный уровень. В двух частях. М.: Бином, 2013.

3. Python. Подробный справочник Дэвида М. Бизли — книга со справочной информацией о языке Python и модулях стандартной библиотеки.

4. Python. Справочник Марка Лутца. Справочник по наиболее часто используемым функциям и модулям.

#### *Литература для обучающихся*

1. К.Ю. Поляков, Е. А. Еремин. «Информатика», углубленный уровень, М.: Бином. Лаборатория Знаний, 2013.

2. М. Лутц «Изучаем Питон», Санкт-Петербург: Символ, 2011г.

3. Самоучитель Python. Дмитрий Мусин. 2016 [pythonworld.ru](http://pythonworld.ru)