

Министерство культуры Иркутской области



Государственное бюджетное учреждение культуры  
Иркутская областная государственная универсальная научная библиотека  
им. И. И. Молчанова-Сибирского

Принята на заседании методического  
совета ГБУК ИОГУНБ  
от «28» сентября 2023 г.  
Протокол №7/1\_\_\_\_\_

Утверждена приказом ГБУК ИОГУНБ  
от «29» сентября 2023 г.  
№ 169-д

**ДОПОЛНИТЕЛЬНАЯ  
ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА**

«Основы программирования на языке Python  
(наименование программы)  
(углублённый уровень)»

Направленность: техническая  
Возраст обучающихся: 14 – 19 лет  
Срок реализации программы: 1 год  
Форма обучения: очная  
ФИО, должность автора:

М.Г. Рейнгольд, ведущий методист СУЦ НМО  
А.В. Осипова, ведущий методист СУЦ НМО  
А.А. Васильева, ведущий методист СУЦ НМО

Иркутск  
2023 г.

# 1. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

## 1.1. Общая характеристика дополнительной общеразвивающей программы

В современном мире все большую ценность приобретает навык не только грамотного использования готовых программных решений, но и умение создавать новые программы. Недостаточное количество часов, которое отводится на изучение программирования на уроках информатики в общеобразовательных школах, нередко представляет программирование лишь на элементарном уровне, как однообразный и зачастую непонятный процесс.

Python – один из самых популярных скриптовых языков программирования, выбираемых для обучения программированию. Это объясняется простотой синтаксиса языка, интуитивной понятностью написанного кода, а также его практической востребованностью: обучающийся не просто изучает программирование, а знакомится с различными современными парадигмами программирования, которые он в будущем сможет применять в самых разных областях от разработки веб-приложений до машинного обучения.

Содержание программы включает 3 модуля:

1. Модуль 1 «Введение в программирование»
2. Модуль 2 «Базовые конструкции в Python»
3. Модуль 3 «Решение прикладных задач в Python»

В основу курса «Основы программирования на языке Python (углублённый уровень)» заложены принципы модульности и практической направленности, что обеспечит вариативность обучения. Содержание учебных модулей направлено на:

- детальное изучение алгоритмизации;
- реализацию межпредметных связей;
- организацию проектной и исследовательской деятельности обучающихся.

Важным аспектом программы является самостоятельная работа над заданиями: школьники учатся решать задачи без помощи преподавателя. Для этого в содержании курса фигурируют задания, в которых:

- для решения задачи необходимо найти какую-то информацию в сети Интернет;
- может потребоваться устранение ошибки, которую не так просто быстро обнаружить;
- условие сформулировано недостаточно прозрачно и ученику необходимо самостоятельно формализовать его (или задать правильные вопросы преподавателю).

Общеразвивающая программа имеет техническую направленность, составлена с учетом возрастных особенностей обучающихся.

## 1.2. Нормативные документы, регламентирующие разработку дополнительной общеразвивающей программы

Нормативно–правовую базу разработки ДОП составляют:

- Федеральный закон от 29 декабря 2012 г. № 273–ФЗ «Об образовании в Российской Федерации»;
- Приказ Министерства просвещения Российской Федерации от 27 июля 2022 г. № 629 «Об утверждении порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам»;
- Письмо Минобрнауки России от 18.11.2015 N 09–3242 «О направлении информации» (вместе с «Методическими рекомендациями по проектированию дополнительных общеразвивающих программ (включая разноуровневые

- программы)»;
- Локальные нормативные акты, регламентирующие образовательную деятельность по дополнительным образовательным программам.

### **1.3. Актуальность программы**

В настоящее время мы переживаем большие изменения в развитии общества. В современную жизнь человека всё больше внедряются компьютеры и информационные технологии. Всё большее значение приобретает умение человека грамотно обращаться с компьютером, причём зачастую не на пользовательском уровне, а на уровне начинающего программиста.

В обязательном школьном курсе информатики программирование нередко представлено лишь на элементарном уровне, на это выделяется недостаточное количество часов. Лишь немногие школы могут себе позволить преподавать программирование на достойном уровне. Следствием этого является формальное восприятие учащимися основ современного программирования и неумение применять полученные знания на практике.

Данная программа разработана на основе дополнительной общеобразовательной (общеразвивающей) программы АНО ДПО «ШАД» и опирается на уникальный опыт программирования на факультете компьютерных наук НИУ ВШЭ. В ней большое внимание уделяется практической работе на компьютере, самостоятельному написанию кода.

Изучение основных принципов программирования невозможно без регулярной практики написания программ на каком-либо языке. Для обучения был выбран язык Python. Данный выбор обусловлен тем, что синтаксис языка достаточно прост и интуитивно понятен, а это понижает порог вхождения и позволяет сосредоточиться на логических и алгоритмических аспектах программирования, а не на выучивании тонкостей синтаксиса. При этом Python является очень востребованным языком; он отлично подходит для знакомства с различными современными парадигмами программирования и активно применяется в самых разных областях от разработки веб-приложений до машинного обучения и его знание позволит учащимся с лёгкостью выучить любой другой язык программирования.

Знания и умения, приобретённые 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.

### **1.4. Направленность программы: техническая.**

### **1.5. Объемы и сроки освоения программы**

**Форма обучения:** очная

**Срок реализации программы:** 1 год

**Объем программы:** 144 академических часа

**Адресат программы:** учащиеся 14-19 лет общеобразовательных образовательных организаций и организаций среднего профессионального образования.

**Максимальная недельная нагрузка:** 4 академических часа в неделю.

**Режим занятий:** два раза в неделю по 2 академических часа.

Продолжительность академического часа – 45 минут. Между сдвоенными занятиями предусмотрен перерыв 10 минут.

## 1.6. Отличительные особенности программы от уже существующих

Программа курса:

- обеспечивает знакомство с фундаментальными понятиями алгоритмизации и программирования на доступном уровне;
- имеет практическую направленность с ориентацией на реальные потребности, соответствующие возрасту ученика;
- охватывает как алгоритмическое направление, так и вопросы практического использования полученных знаний при решении задач из различных областей знаний;
- ориентирована на существующий парк вычислительной техники и дополнительные ограничения;
- допускает возможность варьирования в зависимости от уровня подготовки и интеллектуального уровня учащихся (как группового, так и индивидуального);
- предусматривает возможность индивидуальной работы с учащимися.

Практическая значимость курса заключается в том, что он способствует более успешному овладению знаниями и умениями по направлению «Программирование» через развитие самостоятельности обучающихся и оптимизацию средств и методов обучения.

Элементы программы курса могут быть рекомендованы для использования учителями информатики при проведении лабораторно-практических и практических занятий.

## 1.7. Цель 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:**

**ОБУЧАЮЩИЕ ЗАДАЧИ:**

- определять результат выполнения алгоритма при заданных исходных данных, узнавать изученные алгоритмы обработки чисел и числовых последовательностей, создавать на их основе несложные программы анализа данных, читать и понимать несложные программы, написанные на выбранном для изучения универсальном алгоритмическом языке высокого уровня;
- формировать представления об основных предметных понятиях («информация», «алгоритм», «исполнитель», «модель») и их свойствах;
- развивать логических способностей и алгоритмического мышления, умения составить и записать алгоритм для конкретного исполнителя, знакомство с основными алгоритмическими структурами — линейной, условной и циклической;
- уметь выполнять пошагово (с использованием компьютера или вручную) несложные алгоритмы управления исполнителями и анализа числовых и текстовых данных;
- приобретать навыки и опыт разработки программ в выбранной среде программирования, включая тестирование и отладку программ; умение использовать основные управляющие конструкции объектно-ориентированного программирования и библиотеки прикладных программ, выполнять созданные программы;
- уметь 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;

- формировать умения соблюдать нормы информационной этики и права.

#### РАЗВИВАЮЩИЕ ЗАДАЧИ:

- формировать ответственное отношение к учению, способности довести до конца начатое дело аналогично завершённым творческим учебным проектам;
- формировать способности к саморазвитию и самообразованию средствами информационных технологий на основе приобретённой благодаря иллюстрированной среде программирования мотивации к обучению и познанию;
- приобретать опыт участия в социально значимых проектах, повышение уровня самооценки благодаря реализованным проектам;
- формировать коммуникативную компетентность в общении и сотрудничестве со сверстниками в процессе образовательной, учебно-исследовательской и проектной деятельности, участия в конкурсах и конференциях различного уровня;
- формировать целостное мировоззрение, соответствующее современному уровню развития информационных технологий;
- формировать осознанное позитивное отношение к другому человеку, его мнению, результату его деятельности;
- формировать ценности здорового и безопасного образа жизни; усвоение правил индивидуального и коллективного безопасного поведения при работе с компьютерной техникой.

#### ВОСПИТАТЕЛЬНЫЕ ЗАДАЧИ:

- уметь самостоятельно ставить и формулировать для себя новые задачи, развивать мотивы своей познавательной деятельности;
- уметь самостоятельно планировать пути решения поставленной проблемы для получения эффективного результата; понимание, что в программировании длинная программа не всегда лучшая;
- уметь критически оценивать правильность решения учебно-исследовательской задачи;
- уметь корректировать свои действия, вносить изменения в программу и отлаживать её в соответствии с изменяющимися условиями;
- владеть основами самоконтроля, способность к принятию решений;
- уметь создавать, применять и преобразовывать знаки и символы, модели и схемы для решения учебно-исследовательских и проектных работ;
- формировать и развивать компетентности в области использования информационно-коммуникационных технологий (ИКТ-компетенция);
- уметь организовывать учебное сотрудничество и совместную деятельность с учителем и сверстниками в процессе проектной и учебно-исследовательской деятельности.

#### **1.8. Формы подведения итогов реализации программы**

При организации занятий по курсу «Основы программирования на языке Python (углублённый уровень)» для достижения поставленных целей и решения поставленных задач используются формы проведения занятий с активными методами обучения:

- занятие в форме проблемно-поисковой деятельности;
- занятие с использованием межпредметных связей;
- занятие в форме мозгового штурма;
- занятие в форме частично-поисковой деятельности.

***Формы и методы контроля:***

- тестирование;
- устный опрос;
- самостоятельные и контрольные работы;
- участие в проектной деятельности.

Контроль качества освоения Программы включает в себя промежуточную и итоговую аттестацию обучающихся.

Итоговая аттестация слушателей – оценивание промежуточных и окончательных результатов обучения по учебным модулям. По окончании обучения при условии успешного освоения Программы выдается сертификат о прохождении обучения.

Лицам, не прошедшим итоговую аттестацию, а также лицам, освоившим часть Программы и отчисленным из ГБУК ИОГУНБ, выдается справка об обучении или периоде обучения по установленному образцу.

## 2. СОДЕРЖАНИЕ ПРОГРАММЫ

### 2.1. Учебный план

№ п/п	Наименование дисциплин	Общая трудоёмкость, час	Аудиторные занятия, час		Дистанционные занятия, час		СР	Формы промежуточной аттестации		
			всего	из них		всего			из них	
				теория	практика				теория	практика
<b>1.</b>	<b>Модуль 1. Введение в программирование. Базовые конструкции в Python</b>	<b>26</b>	<b>18</b>		<b>18</b>			<b>8</b>	<b>Контрольная работа</b>	
1.1.	Знакомство со средой. Понятия кода, интерпретатора, программы. Интегрированные среды, исполнение кода и отладка. Переменные, основные операторы.	2	1		1			1		
1.2.	Условный оператор	4	3		3			1		
1.3.	Простые встроенные функции	2	1		1			1		
1.4.	Знакомство с циклом while	4	3		3			1		
1.5.	Отладчик	2	1		1			1		
1.6.	Знакомство с циклом for	4	3		3			1		
1.7.	True, False, break, continue	2	1		1			1		
1.8.	Вложенные циклы	4	3		3			1		
1.9.	Контрольная работа	2	2		2					
<b>2.</b>	<b>Модуль 2. Коллекции в Python</b>	<b>42</b>	<b>32</b>		<b>32</b>			<b>10</b>	<b>Контрольная работа</b>	

2.1.	Множества	4	2		2				2	
2.2.	Строки. Индексация. Срезы	6	4		4				2	
2.3.	Знакомство со списками. Кортежи. Преобразование коллекций	6	4		4				2	
2.4.	Методы split и join. Списочные выражения. Методы списков и строк	6	4		4				2	
2.5.	Повторение материала.	4	4		4					
2.6.	Вложенные списки. Знакомство со словарями	4	4		4					
2.7.	Повторение материала.	4	4		4					
2.8.	Практика по работе с графикой и tkinter	6	4		4				2	
2.9.	Контрольная работа	2	2		2					
3.	<b>Модуль 3. Решение прикладных задач в Python</b>	<b>74</b>	<b>74</b>		<b>52</b>				<b>22</b>	<b>Контрольная работа</b>
3.1	Функции. Возвращение значений из функций	6	6		4				2	
3.2	Области видимости переменных	4	4		3				1	
3.3	Функции: передача параметров. Рекурсия	6	6		4				2	
3.4	Функции с переменным числом аргументов. Лямбда-функции	6	6		4				2	
3.5	Обработка коллекций. Потоковый ввод sys.stdin	2	2		1				1	
3.6	Контрольная работа	2	2		2					
3.7	Библиотеки Python. Встроенные модули.	6	6		4				2	

3.8	Библиотеки Python. Работа с графическими файлами и звуком.	6	6		4				2	
3.9	Библиотеки Python. Морфология	6	6		4				2	
3.10	Библиотеки Python Документы	6	6		4				2	
3.11	Контрольная работа	2	2		2					
3.12	Введение в ООП. Полиморфизм. Определение операторов	6	6		4				2	
3.13	Введение в ООП. Наследование.	6	6		4				2	
3.14	Проектирование и разработка классов.	6	4		4				2	
3.15.	Повторение материала.	6	6		6					
<b>4.</b>	<b>Итоговая аттестация</b>				<b>2</b>					<b>Контрольная работа</b>

\*Промежуточная аттестация проводится за счет времени, отведенного на изучение дисциплины

## 2.2.Календарный учебный график

№ п/ п	Наименование модуля номер недели	Общая трудоемко сть, час	октябрь				ноябрь				декабрь				январь				февраль				март				апрель			
			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
1	Введение в программирование. Базовые конструкции в Python	26	6	6	6	6	2																							
2	Коллекции в Python	42					4	6	6	6	6	6	6	2																
3	Решение прикладных задач в Python	76											4	6	6	6	6	6	6	4	4	4	4	4	4	4	4	4	4	2
	<b>Всего общая трудоемкость, час</b>	144																												

## 2.3.Содержание учебных дисциплин/модулей

### 2.3.1.Цель изучения:

Умения	Знания
<ul style="list-style-type: none"><li>- работать в среде разработки, использовать отладку для исправления ошибок;</li><li>- создавать программы, которые умеют выводить на экран и считывать информацию с клавиатуры;</li><li>- использовать переменные при написании программы, работать с базовыми типами данных и основными операторами;</li><li>- использовать простые встроенные функции;</li><li>- организовывать ветвление в программе;</li><li>- организовывать циклы в программе, прерывать их и продолжать;</li><li>- гибко применять данные знания на практике для решения конкретных задач</li></ul>	<ul style="list-style-type: none"><li>- что такое код, интерпретатор, программа;</li><li>- что такое интегрированные среды разработки, исполнение кода и отладка;</li><li>- что такое переменные и какие существуют базовые типы данных. Изучат принципы работы с ними;</li><li>- какие существуют основных операторах</li><li>- что из себя представляют ветвления и циклы</li></ul>

### 2.3.2. Объем учебной дисциплины/модуля и виды учебной работы

Вид учебной работы	Объем часов
<b>Объем учебного модуля</b>	26
в том числе:	
теоретическое обучение	-
практические занятия	18
самостоятельная работа	8
итоговая аттестация	-

### 2.3.3. Тематический план и содержание учебного модуля

Наименование разделов и тем	Содержание учебного материала, практические работы, самостоятельная работа слушателей	Кол-во часов
<b>Модуль 1. Введение в программирование. Базовые конструкции в Python</b>		
<b>Тема 1.1. Знакомство со средой. Понятия кода, интерпретатора, программы. Интегрированные среды, исполнение кода и отладка. Переменные, основные операторы.</b>	<b>Содержание учебного материала</b>	
	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. Работа с переменными. Ввод-вывод в программе.	1
	<b>Самостоятельная работа</b>	
	Решение задач разного уровня на платформе Яндекс Лицей модуля «Знакомство со средой»	1
<b>Тема 1.2. Условный оператор</b>	<b>Содержание учебного материала</b>	
	Знакомство с условным оператором if. Разбор на примерах, как с помощью данного условного оператора построить алгоритм ветвления в программе. Разбор сложных ветвлений и логические операции. Вложенные условия	3
	<b>Самостоятельная работа</b>	
	Решение задач разного уровня на платформе Яндекс Лицей модуля «Условный оператор».	1
<b>Тема 1.3. Простые встроенные функции</b>	<b>Содержание учебного материала</b>	
	Повторение материала прошлых тем. Знакомство с типами данных. Операции над числами. Приоритет операций. Простейшие встроенные функции.	1
	<b>Самостоятельная работа</b>	
	Решение задач разного уровня на платформе Яндекс Лицей модуля «Простые встроенные функции».	1
	<b>Содержание учебного материала</b>	

<b>Тема 1.4. Знакомство с циклом while</b>	Знакомство с понятием цикла и для чего они нужны в программе. Знакомство с оператором цикла while. Разрабатываем алгоритмы с использованием циклов.	3
	<b>Самостоятельная работа</b>	
	Решение задач разного уровня на платформе Яндекс Лицей модуля «Знакомство с циклом while».	1
<b>Тема 1.5. Отладчик</b>	<b>Содержание учебного материала</b>	
	Рассмотрение типов ошибок в программах и способов их исправления. Знакомство с отладчиком и точками остановки. В уроке рассматривается работа с отладчиком на примере программы, которую нужно исправить.	1
	<b>Самостоятельная работа</b>	
	Решение задач разного уровня на платформе Яндекс Лицей модуля «Отладчик».	1
<b>Тема 1.6. Знакомство с циклом for</b>	<b>Содержание учебного материала</b>	
	Понятие цикла и для чего он нужен в программе. Знакомство с оператором цикла for. Сравнение операторов цикла while и for. Ситуации, когда цикл for удобно использовать.	3
	<b>Самостоятельная работа</b>	
	Решение задач разного уровня на платформе Яндекс Лицей модуля «Знакомство с циклом for».	1
<b>Тема 1.7. True, False, break, continue</b>	<b>Содержание учебного материала</b>	
	Условия выхода из циклов. Рассматривается булев тип, даются задачи на использование флагов. Затем рассматриваются операторы break и continue, позволяющие в некоторых случаях избавиться от флагов.	1
	<b>Самостоятельная работа</b>	
	Решение задач разного уровня на платформе Яндекс Лицей модуля «True и False, break и continue».	1
<b>Тема 1.8. Вложенные циклы</b>	<b>Содержание учебного материала</b>	
	Вложенные циклы, позволяющие запустить цикл внутри циклического оператора. Приведем несколько примеров вложенности разных циклов, а также применение операторов break и continue со вложенными циклами.	3
	<b>Самостоятельная работа</b>	
	Решение задач разного уровня на платформе Яндекс Лицей модуля «Вложенные циклы».	1
<b>Тема 1.9. Контрольная работа</b>	<b>Содержание учебного материала</b>	
	Контрольная работа по модулю 1	2

#### 2.3.4. Цель изучения:

Умения	Знания
<ul style="list-style-type: none"><li>- решать прикладные задачи используя коллекции;</li><li>- с учетом особенностей задач подбирать нужный тип коллекции для решения</li></ul>	<ul style="list-style-type: none"><li>- что такое множества, списки, кортежи, словари и вложенные списки;</li><li>- какие способы преобразования коллекций существуют;</li><li>- какие основные методы для каждого типа;</li><li>- для чего нужны коллекции.</li></ul>

#### 2.3.5. Объем учебной дисциплины/модуля и виды учебной работы

Вид учебной работы	Объем часов
<b>Объем учебного модуля</b>	42
в том числе:	
теоретическое обучение	-
практические занятия	32
самостоятельная работа	10
итоговая аттестация	-

### 2.3.6. Тематический план и содержание учебного модуля

Наименование разделов и тем	Содержание учебного материала, практические работы, самостоятельная работа слушателей	Кол-во часов
<b>Модуль 2. Коллекции в Python</b>		
<b>Тема 2.1. Множества</b>	<b>Содержание учебного материала</b>	
	Множества в Python. Этот тип данных аналогичен математическим множествам, он поддерживает быстрые операции проверки наличия элемента в множестве, добавления и удаления элементов, операции объединения, пересечения и вычитания множеств.	2
	<b>Самостоятельная работа</b>	
	Решение задач разного уровня на платформе Яндекс Лицей модуля «Множества».	2
<b>Тема 2.2. Строки. Индексация. Срезы</b>	<b>Содержание учебного материала</b>	
	На этом занятии углубим свои знания о строках. Теперь мы сможем не только считывать строку, но и работать с ней, в том числе делать посимвольный перебор. Познакомимся с новым методом извлечения подстроки — срезами.	4
	<b>Самостоятельная работа</b>	
	Решение задач разного уровня на платформе Яндекс Лицей модуля «Строки. Индексация» и модуля «Строки. Срезы».	2
<b>Тема 2.3. Знакомство со списками. Кортежи. Преобразование коллекций</b>	<b>Содержание учебного материала</b>	
	Новый тип данных — списки (list), обращение к элементам списка по индексу (аналогично строкам, но с возможностью изменения элементов списка), метод append и вопросы перебора элементов списка и срезов списка. Вводится еще один контейнер — кортеж (tuple). Более подробно рассматривается операция присваивания кортежей, знакомая нам по конструкции $a, b = b, a$ , и применение этой операции в реализации классического алгоритма — сортировки пузырьком. Рассматриваются и вопросы преобразования одной коллекции в другую.	4
	<b>Самостоятельная работа</b>	
	Решение задач разного уровня на платформе Яндекс Лицей модуля «Знакомство со списками» и модуля «Кортежи. Преобразование коллекций».	2
	<b>Содержание учебного материала</b>	

<b>Тема 2.4. Методы split и join. Списочные выражения. Методы списков и строк</b>	Рассматриваются два очень популярных метода строк — split и join. Во второй части рассматриваются списочные выражения (python list comprehensions [... for ... in ...]), которые в сочетании с методом split позволяют, например, считывать из стандартного потока ввода несколько значений одной строкой. Приводятся таблицы с почти полным перечнем методов списков и строк, которые можно использовать как справочный материал. Рассматривается неявное приведение объектов к булеву типу. Приводятся примеры цепочек вызова методов. Появление метода pop позволяет познакомиться с понятием стека. Урок отличается от прочих: большую часть его материала не нужно запоминать, но можно использовать как справочный материал. С этой же целью вводятся функции dir и help.	4
	<b>Самостоятельная работа</b>	
	Решение задач разного уровня на платформе Яндекс Лицей модуля «Методы split и join. Списочные выражения» и модуля «Методы списков и строк».	2
<b>Тема 2.5. Повторение материала</b>	<b>Содержание учебного материала</b>	
	Повторение пройденного материала	4
<b>Тема 2.6. Вложенные списки. Знакомство со словарями</b>	<b>Содержание учебного материала</b>	
	Рассмотрим подробнее списки, элементами которых являются другие (вложенные) списки. Рассказывается о словарях — встроенной в Python мощной структуре данных. В других языках аналогичная структура называется map, HashMap, Dictionary. Базовые функции работы со словарями показаны на простых примерах хранения библиотеки знаний о фильмах и актерах.	4
<b>Тема 2.7. Повторение материала</b>	<b>Содержание учебного материала</b>	
	Повторение пройденного материала	4
<b>Тема 2.8. Практика по работе с графикой и tkinter</b>	<b>Содержание учебного материала</b>	
	Построения графика функции по вычисляемым точкам с помощью Tkinter. <b>Графика.</b>	4
	<b>Самостоятельная работа</b>	
<b>Тема 2.9. Контрольная работа</b>	Решение задач разного уровня на платформе Яндекс Лицей	2
	<b>Содержание учебного материала</b>	
	Контрольная работа по модулю 2	2

### 2.3.7. Цель изучения:

Умения	Знания
<ul style="list-style-type: none"><li>- приобретение навыков поиска информации в сети Интернет, анализ выбранной информации на соответствие запросу, использование информации при решении задач;</li><li>- приобретение навыков работы с библиотеками;</li><li>- приобретение навыков решения прикладных задач;</li><li>- приобретение навыков разработки классов.</li></ul>	<ul style="list-style-type: none"><li>- подключать и использовать модули стандартной библиотеки;</li><li>- модульный принцип компоновки программы;</li><li>- работа с документацией в стандартной библиотеке;</li><li>- понятие репозитория различных пакетов Python;</li><li>- работа с внешними библиотеками Python и утилитой pip;</li><li>- основы ООП.</li></ul>

### 2.3.8. Объем учебной дисциплины/модуля и виды учебной работы

Вид учебной работы	Объем часов
<b>Объем учебного модуля</b>	74
в том числе:	
теоретическое обучение	52
практические занятия	-
самостоятельная работа	22
итоговая аттестация	-

### 2.3.9. Тематический план и содержание учебного модуля

Наименование разделов и тем	Содержание учебного материала, практические работы, самостоятельная работа слушателей	Кол-во часов
<b>Модуль 3. Решение прикладных задач в Python</b>		
<b>Тема 3.1. Функции. Возвращение значения из функции</b>	<b>Содержание учебного материала</b>	
	Группировка команды в функции — участки кода, которые можно использовать многократно. Обсудим, как можно сделать так, чтобы код функции работал по-разному в зависимости от параметров. Локальные переменные.	4
	<b>Самостоятельная работа</b>	
	Решение задач разного уровня на платформе Яндекс Лицей модуля «Функции» и модуля «Возвращение значений из функций».	2
<b>Тема 3.2. Область видимости переменных</b>	<b>Содержание учебного материала</b>	
	Области видимости переменных. Занятие будет посвящено в первую очередь переменным и лишь во вторую — функциям. Эти знания нечасто нужны для написания кода, но совершенно необходимы для понимания того, как программа работает. Это поможет ученикам не гадать, как ведет себя программа, и значительно сократит время, которое они тратят на отладку.	3
	<b>Самостоятельная работа</b>	
	Решение задач разного уровня на платформе Яндекс Лицей модуля «Области видимости переменных»	1
<b>Тема 3.3. Функции. Передача параметров. Рекурсия</b>	<b>Содержание учебного материала</b>	
	Понятия «переменная» и «значение», ведь при вызове функций одно постоянно превращается в другое. Мы обсудим, почему аргумент, переданный в функцию, нельзя перезаписать, но иногда можно изменить его содержимое. Обсудим, какие объекты изменить можно, а какие не получится. Разберемся, почему равенство объектов неэквивалентно их идентичности. Также мы познакомимся с понятием рекурсии, покажем ее связь с уже известными нам конструкциями (циклами и функциями). Разберем наиболее часто встречающиеся ошибки и классические примеры.	4

	<b>Самостоятельная работа</b>	
	Решение задач разного уровня на платформе Яндекс Лицей модуля «Функции. Передача параметров» и модуля «Рекурсия»	2
<b>Тема 3.4. Функции с переменным числом аргументов» и модуля. Функции как объект. Лямбда-функции.</b>	<b>Содержание учебного материала</b>	
	Сложные функции, принимающие неопределенное число аргументов, и передавать в функцию именованные параметры. Попутно мы немного поговорим о том, как работает множественное присваивание. Как получить и использовать соответствующий объект. Кроме того, мы научимся создавать и использовать крошечные, буквально в полстроки функции и передавать их в качестве аргумента другой функции. Мы узнаем о двух важных функциях: <code>filter</code> и <code>map</code> , которые позволяют проводить множество преобразований над коллекциями и другими наборами объектов. Заодно мы вспомним списочные выражения и расширим знания о них.	4
	<b>Самостоятельная работа</b>	
	Решение задач разного уровня на платформе Яндекс Лицей модуля «Функции с переменным числом аргументов» и модуля “Лямбда-функции”	2
<b>Тема 3.5. Обработка коллекций. Поточный ввод sys.stdin</b>	<b>Содержание учебного материала</b>	
	Сведение сложного вычисления к одной строке, если правильно подобрать порядок преобразования данных. В этом уроке мы будем изучать арсенал имеющихся инструментов и учиться их использовать.	1
	<b>Самостоятельная работа</b>	
	Решение задач разного уровня на платформе Яндекс Лицей модуля «Обработка коллекций. Поточный ввод sys.stdin”	1
<b>Тема 3.6. Контрольная работа</b>	<b>Содержание учебного материала</b>	
	Написание контрольной работы	2
<b>Тема 3.7. Библиотеки Python. Встроенные модули</b>	<b>Содержание учебного материала</b>	
	Стандартные библиотеки языка Python	4
	<b>Самостоятельная работа</b>	
	Решение задач разного уровня на платформе Яндекс Лицей модуля “Библиотеки Python. Встроенные модули”	2

<b>Тема 3.8. Библиотеки Python. Работа с графическими файлами и звуком</b>	<b>Содержание учебного материала</b>	
	Использование библиотек для обработки изображений и аудиофайлов, создание собственных изображений с использованием библиотеки Pillow и модуля wave.	4
	<b>Самостоятельная работа</b>	
	Решение задач разного уровня на платформе Яндекс Лицей модуля «Библиотеки Python. Работа с графическими файлами и звуком»	2
<b>Тема 3.9. Библиотеки Python. Морфология</b>	<b>Содержание учебного материала</b>	4
	Библиотека, позволяющая работать с текстами. rpython2, морфологический анализ, работа с тегами. Постановка слов в начальную форму, согласование с числительными.	4
	<b>Самостоятельная работа</b>	
	Решение задач разного уровня на платформе Яндекс Лицей модуля «Библиотеки Python. Морфология»	2
<b>Тема 3.10. Библиотеки Python. Документы</b>	<b>Содержание учебного материала</b>	
	Библиотеки, с помощью которых можно работать с документами (текстами, презентациями, таблицами). Эти библиотеки можно использовать для автоматизации документооборота.	4
	<b>Самостоятельная работа</b>	
	Решение задач разного уровня на платформе Яндекс Лицей модуля «Библиотеки Python. Документы»	2
<b>Тема 3.11. Контрольная работа</b>	<b>Содержание учебного материала</b>	
	Проведение контрольной работы	2
<b>Тема 3.12. Введение в ООП. Полиморфизм. Определение операторов</b>	<b>Содержание учебного материала</b>	
	Объектно-ориентированное программирование (ООП). Создание классов, методы классов, инициализация экземпляров класса, соглашение об именовании, вызов методов атрибутов. Возможности предоставления одинаковых средств взаимодействия с объектами разной природы. Понятие «полиморфизм».	4
	<b>Самостоятельная работа</b>	
	Решение задач разного уровня на платформе Яндекс Лицей модуля «Введение в ООП. Полиморфизм. Определение операторов»	2

<b>Тема 3.13. ООП. Наследование</b>	<b>Содержание учебного материала</b>	
	Понятие наследования, его связь с уже известными понятиями объектно-ориентированного программирования (инкапсуляцией и полиморфизмом), описывается синтаксис и семантика наследования в Python. Урок посвящен приемам применения наследования: расширению и переопределению методов. Обсуждается и множественное наследование.	4
	<b>Самостоятельная работа</b>	
	Решение задач разного уровня на платформе Яндекс Лицей модуля «ООП. Наследование»	2
<b>Тема 3.14. Проектирование и разработка классов</b>	<b>Содержание учебного материала</b>	
	Проектирование программ. Чем больше и сложнее программа, тем важнее еще до написания кода понять, что она должна делать и какова будет ее внутренняя структура.	2
	<b>Самостоятельная работа</b>	
	Решение задач разного уровня на платформе Яндекс Лицей «Проектирование и разработка программ»	2
<b>Тема 3.15. Повторение материала</b>	<b>Содержание учебного материала</b>	
	Повторение изученного материала	6

### 2.3.10. Информационное обеспечение реализации программы

1. Любанович Б. Простой Python. Современный стиль программирования / Билл Любанович ; [пер. с англ. Е. Зазноба]. – Санкт-Петербург [и др.] : Питер, 2019. – 476 с. : ил., табл. – (Бестселлеры O'Reilly).

2. Ночка Е. И. Основы алгоритмизации и программирования (на языке Питон) : учебник для студентов среднего профессионального образования по направлениям подготовки 09.02.01 "Компьютерные системы и комплексы", 09.02.04 "Информационные системы (по отраслям)" / Е. И. Ночка. – Москва : КУРС, 2019. – 203 с. : ил.

3. Седер Н. Python : экспресс-курс / Наоми Седер ; перевел с английского Е. Матвеев. – 3-е издание. – Санкт-Петербург [и др.] : Питер, 2019. – 480 с.

4. Сэнд, Уоррен. Hello World! Занимательное программирование / Уоррен Сэнд, Картер Сэнд ; [пер. И. Рuzмайкина]. – Санкт-Петербург [и др.] : Питер, 2017. – 400 с.

5. <https://lms.yandex.ru/>

### 2.3.11. Контроль и оценка результатов освоения учебной дисциплины/модуля

Результаты обучения	Критерии оценки	Методы оценки
В результате освоения дисциплины обучающихся должен знать: <ul style="list-style-type: none"><li>- что такое код, интерпретатор, программа;</li><li>- что такое интегрированные среды разработки, исполнение кода и отладка;</li><li>- что такое переменные и какие существуют базовые типы данных. Изучат принципы работы с ними;</li><li>- какие существуют основных операторах</li><li>- что из себя представляют ветвления и циклы;</li><li>- что такое множества, списки, кортежи, словари и вложенные списки;</li><li>- какие способы преобразования коллекций существуют;</li><li>- какие основные методы для каждого типа;</li><li>- для чего нужны коллекции;</li><li>- подключать и использовать модули стандартной библиотеки;</li><li>- модульный принцип компоновки программы;</li><li>- работа с документацией в стандартной библиотеке;</li><li>- понятие репозитория различных пакетов Python;</li><li>- работа с внешними библиотеками</li></ul>	Правильность, полнота выполнения заданий, точность формулировок, точность расчетов, соответствие требованиям Адекватность, оптимальность выбора способов действий, методов, техник, последовательностей действий и т.д. Точность оценки, самооценки выполнения Соответствие требованиям инструкций, регламентов Рациональность действий и т.д.	Итоговая аттестация в форме <b>контрольной работы (решение задач)</b>

Python и утилитой pip; - основы ООП.		
В результате освоения дисциплины обучающийся должен уметь: <ul style="list-style-type: none"> <li>- работать в среде разработки, использовать отладку для исправления ошибок;</li> <li>- создавать программы, которые умеют выводить на экран и считывать информацию с клавиатуры;</li> <li>- использовать переменные при написании программы, работать с базовыми типами данных и основными операторами;</li> <li>- использовать простые встроенные функции;</li> <li>- организовывать ветвление в программе;</li> <li>- организовывать циклы в программе, прерывать их и продолжать;</li> <li>- гибко применять данные знания на практике для решения конкретных задач;</li> <li>- решать прикладные задачи используя коллекции;</li> <li>- с учетом особенностей задач подбирать нужный тип коллекции для решения;</li> <li>- искать информацию в сети Интернет, анализ выбранной информации на соответствие запросу, использование информации при решении задач;</li> <li>- работать с библиотеками;</li> <li>- решать прикладные задачи;</li> <li>- уметь разрабатывать классы.</li> </ul>		

### **3. МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ПРОГРАММЫ**

#### **3.1. Формы занятий, планируемых по каждому разделу**

Основной тип занятий — комбинированный, сочетающий в себе элементы теории и практики. Большинство заданий курса выполняется самостоятельно с помощью персонального компьютера и необходимых программных средств.

Единицей учебного процесса является блок уроков (модуль). Каждый такой блок охватывает отдельную информационную технологию или её часть. Внутри блоков разбивка по времени изучения производится учителем самостоятельно, но с учётом рекомендованного календарно-тематического плана. С учётом регулярного повторения

ранее изученных тем темп изучения отдельных разделов блока определяется субъективными и объективными факторами.

Каждая тема курса начинается с постановки задачи — характеристики предметной области или конкретной программы на языке Python, которую предстоит изучить. С этой целью учитель проводит демонстрацию презентации или показывает саму программу, а также готовые работы, выполненные в ней. Закрепление знаний проводится с помощью практики отработки умений самостоятельно решать поставленные задачи, соответствующих минимальному уровню планируемых результатов обучения.

Основные задания являются обязательными для выполнения всеми обучающимися в классе. Задания выполняются на компьютере с использованием интегрированной среды разработки. При этом ученики не только формируют новые теоретические и практические знания, но и приобретают новые технологические навыки.

Методика обучения ориентирована на индивидуальный подход. Для того чтобы каждый ученик получил наилучший результат обучения, программой предусмотрены индивидуальные задания для самостоятельной работы на домашнем компьютере. Такая форма организации обучения стимулирует интерес ученика к предмету, активность и самостоятельность учащихся, способствует объективному контролю глубины и широты знаний, повышению качества усвоения материала обучающимися, позволяет педагогу получить объективную оценку выбранной им тактики и стратегии работы, методики индивидуального обучения и обучения в группе, выбора предметного содержания.

Для самостоятельной работы используются разные по уровню сложности задания, которые носят репродуктивный и творческий характер. Количество таких заданий в работе может варьироваться.

В ходе обучения проводится промежуточное тестирование по темам для определения уровня знаний учеников. Выполнение контрольных заданий способствует активизации учебно-познавательной деятельности и ведёт к закреплению знаний, а также служит индикатором успешности образовательного процесса.

### **3.2. Приемы и методы организации учебно-воспитательного процесса**

При организации занятий по курсу «Основы программирования на языке Python (углублённый уровень)» для достижения поставленных целей и решения поставленных задач используются формы проведения занятий с активными методами обучения:

- занятие в форме проблемно-поисковой деятельности;
- занятие с использованием межпредметных связей;
- занятие в форме мозгового штурма;
- занятие в форме частично-поисковой деятельности.

#### ***Общая характеристика учебного процесса:***

- при изучении курса используются практические и самостоятельные работы;
- курс обучения заканчивается написанием программы для решения одной из задач;
- курс обучения заканчивается написанием программы для решения одной из задач.

## 4. ОРГАНИЗАЦИОННО–ПЕДАГОГИЧЕСКИЕ УСЛОВИЯ РЕАЛИЗАЦИИ ДОПОЛНИТЕЛЬНОЙ ОБЩЕРАЗВИВАЮЩЕЙ ПРОГРАММЫ

### 4.1. Материально–техническое обеспечение

№	Наименование учебной дисциплины в соответствии с учебным планом	Наименование специализированных аудиторий, кабинетов, лабораторий и пр. с перечнем основного оборудования	Форма владения, пользования (собственность, оперативное управление, аренда и т. п.)
1	Модуль 1. Введение в программирование. Базовые конструкции в Python	<b>Кабинет № 503</b> персональный компьютер – 18 шт., мультимедийный проектор – 1 шт., доска-флипчарт – 1 шт., модуль беспроводного доступа – 1 шт., стол компьютерный – 4 шт., стол письменный (преподавательский) – 1 шт., стол читательский – 17 шт., стул рабочий – 20 шт.	Оперативное управление по договору
2	Модуль 2. Коллекции в Python		Собственность
3	Модуль 3. Решение прикладных задач в Python		

### 4.2. Обеспеченность учебными материалами, техническими и электронными средствами обучения и контроля знаний

№ п/п	Наименование	Вид	Форма доступа	Количество
1.	Интернет-ресурсы. Статьи	Электронный	Кабинеты	503
2.	Мультимедийные презентации лекций	Электронный	Кабинеты	503
3.	Методические материалы	Электронный	Кабинеты	503

### 4.3. Кадровое обеспечение

#### 4.3.1. Авторы – составители программы

М.Г. Рейнгольд, ведущий методист СУЦ НМО

А.В. Осипова, ведущий методист СУЦ НМО

А.А. Васильева, ведущий методист СУЦ НМО

#### 5.3.2. Преподавательский состав

Ведущими преподавателями программы являются специалисты с высшим техническим образованием, имеющие опыт работы в преподавании и IT-индустрии.

Васильев А.В., ИРГУПС, 2011, инженер путей сообщения, ПензГТУ, 2018, прикладная информатика.

Рейнгольд М.Г., ИГУ, 2009, прикладная информатика.

## 6. ФОРМЫ И МЕТОДЫ КОНТРОЛЯ. КРИТЕРИИ

### 5.1. Виды контроля:

Оценка качества рабочей программы включает в себя:

- текущий контроль успеваемости;
- промежуточную аттестацию;
- итоговую аттестацию.

Для текущего контроля успеваемости предусмотрены следующие формы:

- проверка домашней (самостоятельной) работы;
- проведение устных опросов;
- тестирование.

Текущий контроль успеваемости учащихся проводится в счет аудиторного времени, предусмотренного учебным процессом.

Цель текущего контроля – проверка и коррекция выполнения текущих заданий, упражнений.

При проверке контрольных работ учитываются качество выполнения заданий, установок и пожеланий преподавателя, творческая инициативность и самостоятельность при выполнении домашних заданий, темпы освоения теоретического материала и наработки умений.

Промежуточная аттестация проводится в виде контрольных заданий.

Контрольные задания могут проходить в форме тестирования, практической работы, защиты проекта и др. Контрольные задания в рамках промежуточной аттестации проводятся на завершающих полугодие учебных занятиях в счет аудиторного времени, предусмотренного на изучение учебной дисциплины.

Виды и содержание контроля: устный опрос, проверка основных теоретических и практических знаний, защита проекта, просмотр творческих работ

## 7. СПИСОК ЛИТЕРАТУРЫ

1. Любанович Б. Простой Python. Современный стиль программирования / Билл Любанович ; [пер. с англ. Е. Зазноба]. – Санкт-Петербург [и др.] : Питер, 2019. – 476 с. : ил., табл. – (Бестселлеры O'Reilly).

2. Ночка Е. И. Основы алгоритмизации и программирования (на языке Питон) : учебник для студентов среднего профессионального образования по направлениям подготовки 09.02.01 "Компьютерные системы и комплексы", 09.02.04 "Информационные системы (по отраслям)" / Е. И. Ночка. – Москва : КУРС, 2019. – 203 с. : ил.

3. Седер Н. Python : экспресс-курс / Наоми Седер ; перевел с английского Е. Матвеев. – 3-е издание. – Санкт-Петербург [и др.] : Питер, 2019. – 480 с.

4. Сэнд, Уоррен. Hello World! Занимательное программирование / Уоррен Сэнд, Картер Сэнд ; [пер. И. Рузмайкина]. – Санкт-Петербург [и др.] : Питер, 2017. – 400 с.