

МУНИЦИПАЛЬНОЕ ОБЩЕОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
«СРЕДНЯЯ АВТОНОМНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ ШКОЛА № 8  
Г. РТИЩЕВО САРАТОВСКОЙ ОБЛАСТИ»  
(МАОУ "СОШ № 8 г. Ртищево Саратовской области")  
Центр образования естественно-научной и технической  
направленностей "Точка роста"

Принята  
на заседании педагогического совета  
МАОУ "СОШ № 8 г. Ртищево  
Саратовской области"  
Протокол № 1 от «30» 08 2023 г.



Утверждаю:  
Директор МАОУ "СОШ № 8  
г. Ртищево Саратовской области"  
Е.Л. Мареева  
Приказ № 212-о от «30» 08 2023 г.



Дополнительная общеобразовательная общеразвивающая  
программа  
(технической направленности)  
«ОСНОВЫ ПРОГРАММИРОВАНИЯ Python»

Возраст обучающихся  
14 – 18 лет  
Срок реализации 136 часа  
Автор-составитель:  
Т.В.Тулынина

г. Ртищево, 2023

## Содержание

<b>1. «КОМПЛЕКС ОСНОВНЫХ ХАРАКТЕРИСТИК ДОПОЛНИТЕЛЬНОЙ ОБЩОБРАЗОВАТЕЛЬНОЙ ОБЩЕРАЗВИВАЮЩЕЙ ПРОГРАММЫ»:</b> .....	3
<b>1.1 ПОЯСНИТЕЛЬНАЯ ЗАПИСКА:</b> .....	3
<b>1.2 ЦЕЛИ И ЗАДАЧИ ПРОГРАММЫ</b> .....	7
<b>1.3 ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ</b> .....	7
<b>1.4 СОДЕРЖАНИЕ УЧЕБНОГО ПЛАНА</b> .....	9
<b>1.4.1 УЧЕБНЫЙ ПЛАН</b> .....	9
<b>1.4.2 СОДЕРЖАНИЕ УЧЕБНОГО ПЛАНА</b> .....	12
<b>1.5 ФОРМЫ АТТЕСТАЦИИ И ИХ ПЕРИОДИЧНОСТЬ</b> .....	14
<b>2. «КОМПЛЕКС ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИХ УСЛОВИЙ»:</b> .....	16
<b>2.1 МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ</b> .....	16
<b>2.2 УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ</b> .....	17
<b>2.3 КАЛЕНДАРНЫЙ УЧЕБНЫЙ ГРАФИК</b> .....	18
<b>2.4 ОЦЕНОЧНЫЕ МАТЕРИАЛЫ</b> .....	26
<b>2.5 СПИСОК ЛИТЕРАТУРЫ</b> .....	26
<b>МЕТОДИЧЕСКИЙ КЕЙС</b> .....	27
<b>ПРИЛОЖЕНИЕ № 1</b> .....	27
<b>ПРИЛОЖЕНИЕ № 2</b> .....	29

# **1. «КОМПЛЕКС ОСНОВНЫХ ХАРАКТЕРИСТИК ДОПОЛНИТЕЛЬНОЙ ОБЩЕОБРАЗОВАТЕЛЬНОЙ ОБЩЕРАЗВИВАЮЩЕЙ ПРОГРАММЫ»:**

## **1.1 ПОЯСНИТЕЛЬНАЯ ЗАПИСКА:**

Дополнительная общеразвивающая программа «Основы программирования на языке Python» разработана в соответствии с Положением о порядке и разработке и утверждения ДОП в МАОУ «СОШ №8 г. Ртищево Саратовской области».

В соответствии с Постановлением администрации Ртищевского муниципального района Саратовской области «Об утверждении Положения о персонифицированном дополнительном образовании в Ртищевском муниципальном районе в новой редакции» от 29.06.2023 года № 599, п. 3.7., п/п. 3.7.1. «образовательная программа специально разработана в целях сопровождения отдельных категорий обучающихся»; п/п. 3.7.4. «образовательная программа реализуется в целях обеспечения развития детей по обозначенным на уровне Ртищевского муниципального района и (или) Саратовской области приоритетным видам деятельности» данная программа может быть зачислена в реестр значимых дополнительных общеобразовательных общеразвивающих программ ПФДО Саратовской области.

### **Направленность программы**

Программа технической направленности, которая направлена на

- формирование и развитие творческих способностей обучающихся
- развитие и поддержку детей, проявивших интерес и определенные способности к техническому творчеству, техническому моделированию.

### **Новизна программы**

Задача построения в стране новой инновационной экономики и достижения технологического уровня, запланированного Концепцией долгосрочного социально-экономического развития Российской Федерации до 2020 года и долгосрочным прогнозом научно-технологического развития Российской Федерации до 2025 года, не может быть решена без существенных изменений системы дополнительного образования детей, создания новых общеразвивающих программ технической направленности.

Изменение взглядов на программирование как науку, его место в системе научного знания требует существенных изменений в содержании образовательного процесса. В связи с этим особую актуальность приобретают раскрытие личностных резервов учащихся и создание соответствующей образовательной среды.

**Общепедагогическая направленность** занятий – гармонизация индивидуальных и социальных аспектов обучения в отношении к информационным технологиям.

- Умение составлять алгоритмы решения и навыки программирования являются элементами информационной компетенции – одной из ключевых компетенций современной школы.
- Умение находить решение, составлять алгоритм решения и реализовать его с помощью языков программирования — необходимое условие подготовки современных учащихся.

Особая роль отводится широко представленной в курсе системе рефлексивных заданий. Освоение рефлексии направлено на осознание учащимися того важного обстоятельства, что наряду с разрабатываемыми ими продуктами в виде программ на компьютере рождается основополагающий образовательный продукт: освоенный инструментарий. Именно этот образовательный продукт станет базой для творческого самовыражения учащихся в форме различных программ.

Никакая система задач, какой бы хорошей она ни была, никакие тренинги памяти, внимания и т. п. не дают того эффекта, который возникает в случае, если учащиеся осознают необходимость решения тех или иных задач, если у них появляется острая необходимость к преодолению интеллектуальных трудностей.

Содержание обучения, представленное в программе «Программирование на языке «Python», позволяет вести обучение в режиме актуального познания. Практическая направленность курса на создание внешних образовательных продуктов – блок-схем, алгоритмов, программ – способствует выявлению фактов, которые невозможно объяснить на основе имеющихся у учащихся знаний.

### **Актуальность программы**

С развитием современных информационных технологий сегодня любой учащийся под руководством опытного педагога может с лёгкостью научиться программировать. Компьютеры и компьютерные системы – неотъемлемая часть жизни нашего общества. Научившись программировать, мы можем быть не только пользователями информационных технологий, но и активными их создателями.

Языки программирования можно сравнить с иностранными языками, овладеть ими может каждый. Учиться программировать очень интересно. Результат программирования очень часто виден сразу. Кроме того, создание компьютерных игр и обучающих программ способствует развитию логики и креативного мышления. Ещё одной значимой стороной обучения программированию является спрос на рынке труда на специалистов данного направления деятельности.

Потребностью общества в специалистах, владеющих профессиональными навыками и умением. Определение и выбор учащихся (ещё на стадии школьного обучения) дальнейшего профессионального развития, обучения и освоения конкретных специальностей, и более лёгкой адаптацией «во взрослой» жизни.

Актуальность программы проявляется в том, что на современном этапе развития общества она отвечает запросам детей и родителей: формирует социально значимые знания, умения и навыки оказывает комплексное обучающее, развивающее, воспитательное и здоровьесберегающее воздействие, способствует формированию эстетических и нравственных качеств личности, приобщает детей к творчеству.

### **Адресат программы**

Дополнительная общеразвивающая программа «Основы программирования на языке Python» рассчитана на детей 14 - 18 лет, проявляющих интерес к информационно-коммуникационным технологиям.

Подростковый возраст от 14 до 18 лет. Переход от детства к взрослости составляет главный смысл и специфическое различие этого этапа. Подростковый период считается «кризисным», такая оценка обусловлена многими качественными сдвигами в развитии подростка. Именно в этом возрасте происходят интенсивные и кардинальные изменения в организации ребенка на пути к биологической зрелости и полового созревания. Характерными новообразованиями подросткового возраста есть стремление к самообразованию и самовоспитанию, полная определенность склонностей и профессиональных интересов.

Юношеский возраст от 14 до 18 лет. В юношеском возрасте происходит интенсивное физиологическое и психическое развития. Особое значение в юношеском возрасте приобретает моральное воспитание, основные виды деятельности — учение и посильный труд, увеличивается диапазон социальных ролей и обязательств.

Психическое развитие личности в юношеском возрасте тесно связано с обучением, трудовой деятельностью и усложнением общения со взрослыми. В связи с началом трудовой деятельности отношения между личностью и обществом значительно углубляются, что приводит к наиболее четкому пониманию своего места в жизни.

Программа составлена с учётом индивидуальных и возрастных особенностей учащихся. Психолого-педагогические особенности учащихся определяют и методы индивидуальной работы педагога с каждым из них, темпы прохождения образовательного маршрута.

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

Общее количество учебных часов, запланированных на весь период обучения и необходимых для освоения программы 136 часов. Продолжительность реализации программы: 1 учебный год.

**Формы обучения** - очное, с применением электронного обучения, дистанционных образовательных технологий. Занятия ведутся на русском языке.

### **Особенности организации образовательного процесса**

Группы являются основным составом объединения, состав группы постоянный от 10 до 20 человек. Возраст детей, участвующих в освоении программы – 14 - 18 лет

### **Режим занятий, периодичность и продолжительность занятий**

Режим занятий в день 2 часа. Периодичность занятий 2 раза в неделю. Продолжительность занятий -40 минут. Между занятиями предусматривается 10 минутный перерыв на отдых.

### **Педагогическая целесообразность**

Состоит в том, что активизация познавательного процесса позволяет учащимся более полно выражать свой творческий потенциал и реализовывать собственные идеи в изучаемой области знаний, создаёт предпосылки по применению информационных компетенций в других учебных курсах, а также способствует возникновению дальнейшей мотивации, направленной на освоение профессий, связанных с разработкой программного обеспечения.

Курс служит средством внутри профильной специализации в области новых информационных технологий, что способствует созданию дополнительных условий для проявления индивидуальных образовательных интересов учащихся, их дальнейшей профессиональной ориентации.

Ключевой особенностью курса является его направленность на формирование у учащихся навыков поиска собственного решения поставленной задачи, составления алгоритма решения и его реализации с помощью средств программирования.

**Целесообразность** программы выражена в подборе интерактивных и практико-ориентированных форм занятий, способствующих формированию основных компетенций (информационных, коммуникативных, компетенций личного развития и др.).

В рамках изучения курса программы «Основы программирования на языке Python» – это не столько средство подготовки к будущей профессиональной деятельности, сколько формирование новых общеинтеллектуальных умений и навыков.

Особую роль в программировании отводится формированию мыслительных и психических процессов учащихся (внимание, память, логика), освоение приёмов умственных действий, самостоятельного нахождения и составления алгоритмов решения задач, умения строить модели, чётко и лаконично реализовывать этапы решения задач.

Использование этих возможностей для формирования предметных и метапредметных результатов учащихся особенно важно, т.к. именно они активизируют процесс индивидуально-личностного становления учащихся.

### **Отличительные особенности программы**

Программа «Основы программирования на языке Python» является модифицированной, составлена на основе использования элементов авторских программ и курсов:

- А.А. Чернов «Программирование на языках высокого уровня». Волгоград: «Учитель», 2012;
- «Основы программирования». С.М. Окулов и др., М.: Бином. Лаборатория Знаний, 2012, М.
- Лутц «Изучаем Питон», Санкт-Петербург: Символ, 2013г.;

- «Программирование на языке Python» программа дополнительного образования, автор: Ефимов Кирилл Николаевич, педагог дополнительного образования, г. Калуга;
- «Программирование аркадных игр на языке Python» в дополнительном образовании, автор-составитель Прядкина Елена Владимировна, учитель информатики, г. Москва.

По уровню освоения программа относится к общекультурной. По уровню сложности – к стартовой.

Основное количество часов отводится практическому написанию программ. Каждый обучающийся реализует индивидуальный проект. Продукт, полученный в результате освоения программы, имеет прикладной характер и может быть использован по необходимости. Можно без преувеличения сказать, что программа имеет потенциальные предпосылки для развития коммуникативных и речевых действий в силу их действительно универсального, т. е. максимально обобщенного, характера.

Заложенные в Федеральном государственном образовательном стандарте второго поколения основы формирования универсальных учебных действий подчёркивают ценность современного образования – школа должна побуждать молодёжь принимать активную гражданскую позицию, усиливать личностное развитие и безопасную социальную включённость в жизнь общества.

## 1.2 ЦЕЛИ И ЗАДАЧИ ПРОГРАММЫ

**Цель:** способствовать **развитию инженерного мышления, воспитание конкурентно способной личности**, обладающей информационными компетенциями, владеющей базовыми понятиями теории алгоритмов, умеющей разрабатывать эффективные алгоритмы и реализовывать их в виде программы, написанной на языке программирования Python.

### **Задачи**

*образовательные:*

- освоение основных алгоритмических конструкций;
- обучение основам алгоритмизации и программирования;
- освоение первоначальных навыков программирования на языке программирования высокого уровня Python;
- приобщение к проектно-творческой деятельности;

*воспитательные:*

- воспитание интереса к информационной и коммуникационной сфере человеческой деятельности,
- воспитание потребности соблюдать этические и правовые нормы работы с информацией;
- воспитание бережного отношения к техническим устройствам;

*развивающие:*

- развитие творческого воображения, алгоритмического мышления учащихся;
- развитие навыков планирования проекта, умения работать в группе;
- развитие навыков ориентации в информационных потоках окружающего мира и применения точной и понятной инструкции для решения учебных задач и в повседневной жизни.

## 1.3 ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ

В рамках курса «Программирование на языке Python» обучающиеся

### **На предметном уровне**

*должны знать/ понимать:*

- понятие программы; структура программы на Python; режимы работы с Python.
- общую структуру программы; типы данных; целые, вещественные типы данных и операции над ними; оператор присваивания;
- назначение условного оператора; способ записи условного оператора; логический тип данных; логические операторы `or`, `and`, `not`;
- циклы с условием и их виды; правила записи циклов с условием; назначение и особенности использования цикла с параметром; формат записи цикла с параметром; примеры использования циклов различных типов.
- назначение строкового типа данных; операторы для работы со строками; процедуры и функции для работы со строками; операции со строками.
- сложные типы данных; способ описания списка; способ доступа к элементам списка; способ описания кортежа; способ описания словаря; операции, выполняемые со списками, кортежами и словарями;

- понятие множества; способы описания множества; операторы работы с множествами.
- что такое стиль программирования; правила именования объектов; основные рекомендации при написании программ.

*Должны уметь:*

- выполнить установку программы; выполнить простейшую программу в интерактивной среде; написать комментарии в программе.
- использовать условный оператор; создавать сложные условия с помощью логических операторов.
- определять вид цикла, наиболее удобный для решения поставленной задачи;
- использовать цикл с условием; определять целесообразность применения и использовать цикл с параметром для решения поставленной задачи.
- описывать строки; соединять строки; находить длину строки; вырезать часть строки; находить подстроку в строке; находить количество слов в строке.
- описывать списки; вводить элементы списка; выводить элементы списка; выполнять поиск элемента в списке, поиск минимума и максимума, нахождение суммы элементов списка; использовать вложенные списки; приводить примеры использования вложенных списков (матриц); описывать множества; определять принадлежность элемента множеству;
- вводить элементы множества; выводить элементы множества.
- определять вид ошибок и находить ошибки в программе; выполнять тестирование и отладку программ.

### **Личностные результаты**

- критическое отношение к информации и избирательность её восприятия; осмысление мотивов своих действий при выполнении заданий;
- развитие любознательности, сообразительности при выполнении разнообразных заданий проблемного и эвристического характера;
- развитие внимательности, настойчивости, целеустремленности, умения преодолевать трудности – качеств весьма важных в практической деятельности любого человека;
- развитие самостоятельности суждений, независимости и нестандартности мышления;
- воспитание чувства справедливости, ответственности;
- начало профессионального самоопределения, ознакомление с миром профессий, связанных с робототехникой.

### **Метапредметные результаты**

***Регулятивные универсальные учебные действия:***

- принимать и сохранять учебную задачу; планировать последовательность шагов алгоритма для достижения цели; формировать умения ставить цель – создание творческой работы, планировать достижение этой цели;
- осуществлять итоговый и пошаговый контроль по результату;
- адекватно воспринимать оценку учителя;
- различать способ и результат действия; вносить коррективы в действия в случае расхождения результата решения задачи на основе её оценки и учета характера сделанных ошибок;
- в сотрудничестве с учителем ставить новые учебные задачи; проявлять познавательную инициативу в учебном сотрудничестве;
- осваивать способы решения проблем творческого характера в жизненных ситуациях;
- оценивать получающийся творческий продукт и соотносить его с изначальным замыслом, выполнять по необходимости коррекции либо продукта, либо замысла.

***Познавательные универсальные учебные действия:***

- осуществлять поиск информации в индивидуальных информационных архивах учащегося, информационной среде образовательного учреждения, в федеральных хранилищах информационных образовательных ресурсов;
- использовать средства информационных и коммуникационных технологий для решения коммуникативных, познавательных и творческих задач; ориентироваться на разнообразие способов решения задач;
- осуществлять анализ объектов с выделением существенных и несущественных признаков; проводить сравнение, классификацию по заданным критериям;
- строить логические рассуждения в форме связи простых суждений об объекте; устанавливать аналогии, причинно-следственные связи;
- моделировать, преобразовывать объект из чувственной формы в модель, где выделены существенные характеристики объекта (пространственно-графическая или знаково-символическая);
- синтезировать, составлять целое из частей, в том числе самостоятельное достраивание с восполнением недостающих компонентов;
- выбирать основания и критерии для сравнения и классификации объектов;

#### **Коммуникативные универсальные учебные действия:**

- аргументировать свою точку зрения на выбор оснований и критериев при выделении признаков, сравнении и классификации объектов;
- выслушивать собеседника и вести диалог;
- признавать возможность существования различных точек зрения и права каждого иметь свою;
- планировать учебное сотрудничество с учителем и сверстниками — определять цели, функций участников, способов взаимодействия;
- осуществлять постановку вопросов — инициативное сотрудничество в поиске и сборе информации;
- разрешать конфликты – выявление, идентификация проблемы, поиск и оценка альтернативных способов разрешения конфликта, принятие решения и его реализация;
- управлять поведением партнера — контроль, коррекция, оценка его действий;
- уметь с достаточной полнотой и точностью выражать свои мысли в соответствии с задачами и условиями коммуникации;
- владеть монологической и диалогической формами речи.

## **1.4 СОДЕРЖАНИЕ УЧЕБНОГО ПЛАНА**

### **1.4.1 УЧЕБНЫЙ ПЛАН**

№	Наименование разделов и тем	Количество часов			Формы аттестации/ контроля
		Всего	Теория	Практика	
	Инструктаж по технике безопасности в компьютерном классе	2	2		Опрос
1.	<b>Раздел 1. Знакомство с языком Python</b>	6	2	4	Тестирование, решение практических задач
1.1	Занятие 1. Общие сведения о языке. Практическая работа: Установка программы Python	2	1	1	
1.2	Занятие 2. Режимы работы Практическая работа: Режимы работы с Python	2	1	1	
1.3	Тест №1. Знакомство с языком Python	2		2	
2.	<b>Раздел 2. Переменные и выражения</b>	11	4	7	Тестирование, решение
2.1	Занятие 3. Переменные Практическая работа: Работа со справочной системой	2	1	1	

№	Наименование разделов и тем	Количество часов			Формы аттестации/ контроля практически х задач
		Всего	Теория	Практика	
2.2	Практическая работа: Переменные	2		2	Формы аттестации/ контроля практически х задач
2.3	Занятие 4. Выражения Практическая работа: Выражения	2	1	1	
2.4	Занятие 5. Ввод и вывод	1	1		
2.5	Занятие 6. Задачи на элементарные действия с числами Практическая работа 2.5. Задачи на элементарные действия с числами	2	1	1	
2.6	Тест №2. Выражения и операции	2		2	
3.	<b>Раздел 3. Условные предложения</b>	20	6	14	
3.1	Занятие7. Логические выражения и операторы. Практическая работа: Логические выражения	2	1	1	
3.2	Занятие8. Условный оператор Практическая работа: «Условный оператор»	2	1	1	
3.3	Занятие9. Множественное ветвление Практическая работа: Множественное ветвление	4	1	3	
3.4	Занятие10. Реализация ветвления в языке Python. Практическая работа: «Условные операторы»	2	1	1	
3.5	Самостоятельная работа №1 по теме «Условные операторы».	4	2	2	
3.6	Занятие11. Зачетная работа №1. "Составление программ с ветвлением".	4		4	
3.7	Тест №3."Условные операторы".	2		2	
4.	<b>Раздел 4. Циклы</b>	46	9	37	Тестирование, решение практически х задач, творческая работа
4.1	Занятие12. Оператор цикла с условием Практическая работа "Числа Фибоначчи"	4	1	3	
4.2	Занятие13. Оператор цикла for Практическая работа Решение задачи с циклом for.	4	1	3	
4.3	Занятие14. Вложенные циклы Практическая работа: Реализация циклических алгоритмов	4	1	3	
4.4	Занятие15. Случайные числа Практическая работа: Случайные числа	4	1	3	
4.5	Занятие16. Примеры решения задач с циклом Практическая работа: Решение задач с циклом.	6	1	5	
4.6	Самостоятельная работа №2 "Составление программ с циклом"	10	2	8	
4.7	Тест №4. Циклы	2		2	
4.8	Занятие17. Творческая работа№1. "Циклы"	12	2	10	

№	Наименование разделов и тем	Количество часов			Формы аттестации/ контроля
		Всего	Теория	Практика	
5.	<b>Раздел 5. Функции</b>	14	6	8	Тестирование, решение практических задач
5.1	Занятие 18. Создание функций Практическая работа Создание функций	2	1	1	
5.2	Занятие 19. Локальные переменные Практическая работа Локальные переменные	2	1	1	
5.3	Занятие 20. Примеры решения задач с использованием функций Практическая работа Решение задач с использованием функций	2	1	1	
5.4	Самостоятельная работа №3 по теме "Функции"	4	2	2	
5.5	Занятие 21. Рекурсивные функции Практическая работа 5.4. Рекурсивные функции	2	1	1	
5.6	Тест № 5. Функции	2		2	
6.	<b>Раздел 6. Строки - последовательности символов</b>	10	3	7	Решение практических задач
6.1	Занятие 22. Строки Практическая работа: Строки	2	1	1	
6.2	Занятие 23. Срезы строк	2	1	1	
6.3	Занятие 24. Примеры решения задач со строками Практическая работа: Решение задач со строками.	6	1	5	
7.	<b>Раздел 7. Сложные типы данных</b>	17	7	10	Тестирование, Решение практических задач
7.1	Занятие 25. Списки Практическая работа: Списки	2	1	1	
7.2	Занятие 26. Срезы списков	1	1		
7.3	Занятие 27. Списки: примеры решения задач Практическая работа 7.2. Решение задач со списками	6	1	5	
7.4	Занятие 28. Матрицы	2	1	1	
7.5	Тест №7. Списки	2		2	
7.6	Занятие 29. Кортежи	1	1		
7.7	Занятие 30. Введение в словари	1	1		
7.8	Занятие 31. Множества в языке Python	2	1	1	
8.	<b>Раздел 8. Стиль программирования и отладка программ</b>	10	6	4	Решение практических задач, презентация проекта, рефлексия.
8.1	Занятие 32. Стиль программирования	2	1	1	
8.2	Занятие 33. Отладка программ	2	1	1	
8.3	Занятие 34. Зачет по курсу «Программирование на языке Python»	4	2	2	
8.4	Занятие 35. Что дальше?	2	2		
	Всего:	136	45	91	

## 1.4.2 СОДЕРЖАНИЕ УЧЕБНОГО ПЛАНА

### Вводное занятие (2 часа).

Теория: Знакомство с планом работы объединения, инструктаж по ТБ.

Практика: Опрос.

### Раздел 1. Знакомство с языком Python (6 часов)

Теория: Общие сведения о языке Python. Установка Python на компьютер. Режимы работы Python. Что такое программа. Первая программа. Структура программы на языке Python. Комментарии.

Практическая работа 1.1. Установка программы Python

Практическая работа 1.2. Режимы работы с Python

Тест №1. Знакомство с языком Python

*Учащиеся должны знать/ понимать:*

– понятие программы; структура программы на Python; режимы работы с Python.

*Учащиеся должны уметь:*

– выполнить установку программы; выполнить простейшую программу в интерактивной среде; написать комментарии в программе.

### Раздел 2. Переменные и выражения (11 часов)

Теория: Типы данных. Преобразование типов. Переменные. Оператор присваивания. Имена переменных и ключевые слова. Выражения. Операции. Порядок выполнения операций. Композиция. Ввод и вывод. Ввод данных с клавиатуры. Вывод данных на экран. Пример скрипта, использующего ввод и вывод данных. Задача на элементарные действия с числами. Решение задач на элементарные действия с числами.

Практическая работа 2.1. Работа со справочной системой

Практическая работа 2.2. Переменные

Практическая работа 2.3. Выражения

Практическая работа 2.5. Задачи на элементарные действия с числами

Тест №2. Выражения и операции.

*Учащиеся должны знать/ понимать:*

– общую структуру программы; типы данных; целые, вещественные типы данных и операции над ними; оператор присваивания;

### Раздел 3. Условные предложения (20 часов)

Теория:

Логический тип данных. Логические выражения и операторы. Сложные условные выражения (логические операции and, or, not). Условный оператор. Альтернативное выполнение. Примеры решения задач с условным оператором. Множественное ветвление. Реализация ветвления в языке Python.

Практическая работа 3.1. Логические выражения

Практическая работа 3.2. "Условный оператор"

Практическая работа 3.3. Множественное ветвление

Практическая работа 3.4. "Условные операторы"

Самостоятельная работа №1. Решение задач по теме "Условные операторы".

Зачетная работа №1. "Составление программ с ветвлением".

Тест №3. "Условные операторы".

*Учащиеся должны знать/ понимать:*

– назначение условного оператора; способ записи условного оператора; логический тип данных; логические операторы or, and, not;

*Учащиеся должны уметь:*

– использовать условный оператор; создавать сложные условия с помощью логических операторов.

### Раздел 4. Циклы (46 часов)

Теория:

Понятие цикла. Тело цикла. Условия выполнения тела цикла. Оператор цикла с условием. Оператор цикла while. Бесконечные циклы. Альтернативная ветка цикла while. Обновление переменной. Краткая форма записи обновления. Примеры использования циклов. Оператор цикла с параметром for. Операторы управления циклом. Пример задачи с использованием цикла for. Вложенные циклы. Циклы в циклах. Случайные числа. Функция randrange. Функция random. Примеры решения задач с циклом.

Практическая работа 4.1. "Числа Фибоначчи"

Практическая работа 4.2. Решение задачи с циклом for.

Практическая работа 4.3. Реализация циклических алгоритмов

Практическая работа 4.4. Случайные числа

Практическая работа 4.5. Решение задач с циклом.

Самостоятельная работа №2 "Составление программ с циклом"

Тест №4. Циклы

Творческая работа № 1. "Циклы"

*Учащиеся должны знать/ понимать:*

– циклы с условием и их виды; правила записи циклов с условием; назначение и особенности использования цикла с параметром; формат записи цикла с параметром; примеры использования циклов различных типов.

*Учащиеся должны уметь:*

– определять вид цикла, наиболее удобный для решения поставленной задачи; использовать цикл с условием; определять целесообразность применения и использовать цикл с параметром для решения поставленной задачи.

## **Раздел 5. Функции (14 часов)**

Теория: Создание функций. Параметры и аргументы. Локальные и глобальные переменные. Поток выполнения. Функции, возвращающие результат. Анонимные функции, инструкция lambda. Примеры решения задач с использованием функций. Рекурсивные функции. Вычисление факториала. Числа Фибоначчи.

Практическая работа 5.1. Создание функций

Практическая работа 5.2. Локальные переменные

Практическая работа 5.3. Решение задач с использованием функций

Практическая работа 5.4. Рекурсивные функции

Самостоятельная работа №3 по теме "Функции"

Тест №5. Функции

*Учащиеся должны знать/ понимать:*

– понятие функции; способы описания функции; принципы структурного программирования; понятие локальных переменных подпрограмм;

– понятие формальных и фактических параметров подпрограмм; способ передачи параметров.

*Учащиеся должны уметь:*

– создавать и использовать функции; использовать механизм параметров для передачи значений.

## **Раздел 6. Строки – последовательности символов (10 часов)**

Теория: Составной тип данных - строка. Доступ по индексу. Длина строки и отрицательные индексы. Преобразование типов. Применение цикла для обхода строки. Срезы строк. Строки нельзя изменить. Сравнение строк. Оператор in. Модуль string. Операторы для всех типов последовательностей (строки, списки, кортежи). Примеры решения задач со строками.

Практическая работа 6.1. Строки

Практическая работа 6.2. Решение задач со строками.

*Учащиеся должны знать/ понимать:*

– назначение строкового типа данных; операторы для работы со строками; процедуры и функции для работы со строками; операции со строками.

*Учащиеся должны уметь:*

– описывать строки; соединять строки; находить длину строки; вырезать часть строки; находить подстроку в строке; находить количество слов в строке.

### **Раздел 7. Сложные типы данных (17 часов)**

Теория: Списки. Тип список (list). Индексы. Обход списка. Проверка вхождения в список. Добавление в список. Суммирование или изменение списка. Операторы для списков. Срезы списков. Удаление списка. Клонирование списков. Списочные параметры. Функция range. Списки: примеры решения задач. Матрицы. Вложенные списки. Матрицы. Строки и списки. Генераторы списков в Python. Кортежи. Присваивание кортежей. Кортеж и как возвращаемые значения. Введение в словари. Тип словарь(dict). Словарные операции. Словарные методы. Множества в языке Python. Множества. Множественный тип данных. Описание множеств. Операции, допустимые над множествами: объединение, пересечение, разность, включение. Оператор определения принадлежности элемента множеству.

Практическая работа 7.1. Списки.

Практическая работа 7.2. Решение задач со списками.

Тест №7. Списки

*Учащиеся должны знать/ понимать:*

– сложные типы данных; способ описания списка; способ доступа к элементам списка; способ описания кортежа; способ описания словаря;  
– операции, выполняемые со списками, кортежами и словарями; понятие множества; способы описания множества; операторы работы с множествами.

*Учащиеся должны уметь:*

– описывать списки; вводить элементы списка; выводить элементы списка; выполнять поиск элемента в списке, поиск минимума и максимума, нахождение суммы элементов списка; использовать вложенные списки; приводить примеры использования вложенных списков (матриц); описывать множества; определять принадлежность элемента множеству; вводить элементы множества; выводить элементы множества.

### **Раздел 8. Стил программирования и отладка программ (10 часов)**

Стил программирования. Отладка программ.

Зачет по курсу «Программирование на языке Python»

*Учащиеся должны знать/ понимать:*

– что такое стил программирования;  
– правила именования объектов;  
– основные рекомендации при написании программ.

*Учащиеся должны уметь:*

– определять вид ошибок и находить ошибки в программе: выполнять тестирование и отладку программ.

## **1.5 ФОРМЫ АТТЕСТАЦИИ И ИХ ПЕРИОДИЧНОСТЬ**

**Входящий контроль** осуществляется при комплектовании группы в начале учебного года. *Цель* – определить исходный уровень знаний учащихся, определить формы и методы работы с учащимися.

*Форма контроля:* тестирование.

**Текущий контроль** осуществляется после изучения отдельных тем, раздела программы. В практической деятельности результативность оценивается качеством выполнения практических работ, поиску и отбору необходимого материала, умению работать с различными источниками информации. Анализируются положительные и отрицательные стороны работы, корректируются

недостатки. Контроль знаний осуществляется с помощью заданий педагога (решение практических задач средствами языка программирования); взаимоконтроля, самоконтроля и др. Они активизируют, стимулируют работу учащихся, позволяют более полно проявлять полученные знания, умения, навыки.

**Промежуточный контроль** осуществляется в конце I полугодия учебного года.

Форма контроля: тест, решение практических задач средствами языка программирования.

**Итоговый контроль** осуществляется в конце учебного года.

Форма контроля: защита творческого проекта. Отслеживание личностного развития учащихся осуществляется методом наблюдения, анкетирования. По итогам первого полугодия и по итогам года заполняется «Диагностическая карта», в которой проставляется уровень усвоения программы каждым учащимся объединения.

### **Формы отслеживания и фиксации образовательных результатов**

Формы определения результативности обучения. Результаты освоения программы отслеживаются по итогам опросов, выполнения практических заданий. Формы аттестации и оценочные материалы. В процессе реализации программы предусмотрены следующие виды контроля:

- входной контроль проводится с целью определения уровня знаний учащихся (Приложение № 1);

- промежуточный контроль проводится регулярно на занятиях с целью определения степени усвоения материала в форме опроса, решения задач и практических заданий;

- итоговый контроль – защита проекта.

Формы подведения итогов реализации программы. Результаты обучения по программе выявляются по итогам проведения олимпиад, соревнований по программированию, защиты проекта

### **Способы оценивания уровня достижений учащихся**

Предметом диагностики и контроля в курсе «Основы программирования на языке Python» являются внешние образовательные продукты учащихся (созданные блок-схемы, программы), а также их внутренние личностные качества (освоенные способы деятельности, знания, умения), которые относятся к целям и задачам курса.

Качество внешней образовательной продукции желательно оценивать по следующим параметрам:

– алгоритм должен быть оптимальным по скорости выполнения и максимально простым в реализации на языке программирования;

– программа должна выполнять поставленные задачи;

– по степени «читаемости кода» (должны быть соблюдены отступы, обязательное наличие комментариев к коду программы и т. д.).

Проверка достигаемых учащимися результатов производится в следующих формах:

– текущий рефлексивный самоанализ, контроль и самооценка учащимися выполняемых заданий;

– текущая диагностика и оценка педагогом деятельности учащихся;

– итоговая оценка деятельности и образовательной продукции ученика в соответствии с его индивидуальными и возрастными особенностями;

– Итоговый контроль проводится в конце всего курса. Он состоит из тестирования и решения практической задачи, защиты творческого проекта.

## **2. «КОМПЛЕКС ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИХ УСЛОВИЙ»:**

### **2.1 МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ**

**Особенности организации образовательного процесса** – очная форма.

**Методы обучения:**

1. Словесный: объяснение нового материала; рассказ обзорный для раскрытия новой темы; беседы с учащимися в процессе изучения темы.
2. Наглядный: применение демонстрационного материала, наглядных пособий, презентаций по теме.
3. Практический: индивидуальная и совместная продуктивная деятельность, выполнение учащимися определенных заданий, решение задач.
4. Интерактивный: создание специальных заданий, моделирующих реальную жизненную ситуацию, из которой учащимся предлагается найти выход.

**Формы организации образовательного процесса** - Индивидуально-групповая и групповая

**Формы организации учебного занятия**

Отбор методов обучения обусловлен необходимостью формировать информационную и коммуникативную компетентности учащихся, реализовывать лично-ориентированное обучение, направлять их на самостоятельное решение разнообразных проблем, развивать исследовательские и творческие способности.

Решение данных задач кроется в организации 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 учащихся навыкам самостоятельной творческой деятельности.

Теория преподносится в форме беседы, эвристической беседы, презентации, обзора и т.п.

Практические занятия проходят в форме тестирования, практикума, совместной продуктивной деятельности, моделирования и защиты проектов, проблемном изложении материала, с помощью которого дети сами решают познавательные задачи.

**Педагогические технологии**

1. Технология проблемного диалога. Учащимся не только сообщаются готовые знания, но и организуется такая их деятельность, в процессе которой они сами делают «открытия», узнают что-то новое и используют полученные знания и умения для решения жизненных задач.
2. Технология коллективного взаимообучения («организованный диалог», «сочетательный диалог», «коллективный способ обучения (КСО), «работа учащихся в парах сменного состава») позволяет плодотворно развивать у обучаемых самостоятельность и коммуникативные умения.
3. Игровая технология. Игровая форма в образовательном процессе создается при помощи игровых приемов и ситуаций, выступающих как средство побуждения к деятельности. Способствует развитию творческих способностей, продуктивному сотрудничеству с другими учащимися. Приучает к коллективным действиям, принятию решений, учит руководить и подчиняться, стимулирует практические навыки, развивает воображение.
4. Элементы здоровьесберегающих технологий являются необходимым условием снижения утомляемости и перегрузки учащихся.

5. Проектная технология предлагает практические творческие задания, требующие от учащихся их применение для решения проблемных заданий, знания материала на данный исторический этап. Овладевая культурой проектирования, школьник приучается творчески мыслить, прогнозировать возможные варианты решения стоящих перед ним задач.

6. Информационно-коммуникационные технологии активизируют творческий потенциал учащихся; способствует развитию логики, внимания, речи, повышению качества знаний; формированию умения пользоваться информацией, выбирать из нее необходимое для принятия решения, работать со всеми видами информации, программным обеспечением, специальными программами и т.д.

## 2.2 УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ

### *Материально-техническое обеспечение:*

- столы для компьютера;
- компьютерные стулья;
- шкафы для дидактических материалов, пособий;
- канцтовары;

### *Информационное обеспечение:*

- ноутбук (на каждого участника);
- мультимедийный проектор;
- оргтехника;
- выход в сеть Internet;

### *Программное обеспечение:*

- Интерпретатор Python версии 3.7 и выше;
- IDE Jupyter Lab;
- FoxitReader или другой просмотрщик PDF файлов;
- Архиватор;
- Пакет офисных программ;
- Растровый графический редактор;
- Браузер.

### **Характеристика помещений для занятий по программе**

Теоретические занятия проходят в компьютерном классе, который соответствует нормам СанПиН, оборудованном современными ноутбуками, мультимедийным проектором, стендами с литературой.

### **Информационное обеспечение**

- видеоматериалы разной тематики по программе;
- цифровые разработки педагога к занятиям (презентации, сайты, тесты и т.д.)

## 2.3 КАЛЕНДАРНЫЙ УЧЕБНЫЙ ГРАФИК

Дата занятия		Кол-во часов	Содержание занятия	Раздел, тема	Время на изучение	
план	факт				теория	практика
		2	<b>Теоретические занятия.</b> Знакомство с планом работы объединения, инструктаж по ТБ. <b>Практические занятия.</b> Общие сведения о языке Python. Установка Python на компьютер.	Общие сведения о языке Установка программы Python	1	1
		2	<b>Теоретические занятия.</b> Режимы работы Python. Что такое программа. Первая программа. Структура программы на языке Python. <b>Практические занятия</b> Программы Python. Режимы работы с Python	Режимы работы Режимы работы с Python	1	1
		2	<b>Теоретические занятия</b> Знакомство с языком Python	Знакомство с языком Python	2	0
		2	<b>Теоретические занятия</b> Типы данных. Преобразование типов. Переменные. Оператор присваивания. Имена переменных и ключевые слова. <b>Практические занятия</b> Выражения. Операции. Порядок выполнения операций. Композиция.	Переменные	1	1
		2	<b>Практические занятия</b> Задачи на элементарные действия с числами. Решение задач на элементарные действия с числами.	Практическая работа: Переменные	0	2
		2	<b>Теоретические занятия</b> Выражения. Операции. <b>Практические занятия</b> Задачи на элементарные действия с числами. Решение задач на элементарные действия с числами.	Выражения Практическая работа: Выражения	1	1

Дата занятия		Кол-во часов	Содержание занятия	Раздел, тема	Время на изучение	
план	факт				теория	практика
		2	<b>Практические занятия</b> Ввод и вывод. Ввод данных с клавиатуры. Вывод данных на экран. Пример скрипта, использующего ввод и вывод данных	Ввод и вывод	0	2
		2	<b>Теоретические занятия.</b> Решение задач на элементарные действия с числами. <b>Практические занятия</b> Выполнение практической работы	Задачи на элементарные действия с числами Практическая работа 2.5. Задачи на элементарные действия с числами	1	1
		2	<b>Теоретические занятия</b> Работа со справочной системой	Выражения и операции.	2	0
		2	<b>Теоретические занятия</b> Логический тип данных. Логические выражения и операторы. Сложные условные выражения (логические операции and, or, not). <b>Практические занятия</b> Примеры решения задач с условным оператором. Множественное ветвление.	Логические выражения и операторы. Практическая работа: Логические выражения	1	1
		2	<b>Теоретические занятия</b> Логический тип данных. Логические выражения и операторы. Сложные условные выражения (логические операции and, or, not). <b>Практические занятия</b> Работа с программными блоками.	Условный оператор Практическая работа: «Условный оператор»	1	1

Дата занятия		Кол-во часов	Содержание занятия	Раздел, тема	Время на изучение	
план	факт				теория	практика
		2	<b>Теоретические занятия</b> Сложные условные выражения (логические операции and, or, not). Условный оператор. Альтернативное выполнение. Примеры решения задач с условным оператором. Множественное ветвление. Реализация ветвления в языке Python. <b>Практические занятия</b> Примеры решения задач с условным оператором. Множественное ветвление.	Множественное ветвление	1	1
		2	<b>Теоретические занятия</b> Альтернативное выполнение. Примеры решения задач с условным оператором. Множественное ветвление. Реализация ветвления в языке Python	Множественное ветвление	2	0
		2	<b>Практические занятия</b> Работа с программными блоками.	Реализация ветвления в языке Python. Практическая работа: «Условные операторы»	0	2
		2	<b>Практические занятия</b> Логические выражения Практическая работа 3.2. "Условный оператор"	Самостоятельная работа № 1 по теме «Условные операторы».	0	2
		2	<b>Практические занятия</b> Логические выражения Практическая работа 3.3. Множественное ветвление	Самостоятельная работа № 1 по теме «Условные операторы».	0	2
		2	<b>Практические занятия</b> Логические выражения Практическая работа 3.4. "Условные операторы"	Зачетная работа № 1. "Составление программ с ветвлением".	0	2
		2	<b>Практические занятия</b> Логические выражения	Зачетная работа № 1. "Составление программ с ветвлением".	0	2

Дата занятия		Кол-во часов	Содержание занятия	Раздел, тема	Время на изучение	
план	факт				теория	практика
		2	<b>Практические занятия</b> Решение задач по теме "Условные операторы".	Тест № 3. "Условные операторы".	0	2
		2	<b>Теоретические занятия</b> Понятие цикла. Тело цикла. Условия выполнения тела цикла. Оператор цикла с условием. Оператор цикла while. Бесконечные циклы. Альтернативная ветка цикла while. Обновление переменной. Краткая форма записи обновления. Примеры использования циклов.	Оператор цикла с условием	2	0
		2	<b>Практические занятия</b> "Числа Фибоначчи" Практическая работа	Практическая работа "Числа Фибоначчи"	0	2
		2	<b>Практические занятия</b> Программирование робота на движение змейкой или по спирали.	Оператор цикла for	0	2
		2	<b>Теоретические занятия</b> Поворот на месте и знакомство с блоком EV3, генерирующим случайные числа.	Практическая работа Решение задачи с циклом for.	1	1
		2	<b>Теоретические занятия</b> Оператор цикла с параметром for. Операторы управления циклом. Пример задачи с использованием цикла for. Вложенные циклы. Циклы в циклах. Примеры решения задач с циклом. <b>Практические занятия</b> Реализация циклических алгоритмов Практическая работа	Вложенные циклы Реализация циклических алгоритмов	1	1
		2	<b>Практические занятия</b> Реализация циклических алгоритмов Практическая работа	Реализация циклических алгоритмов	0	2
		2	<b>Теоретические занятия</b> Случайные числа. Функция randrange. Функция random.	Случайные числа	2	0
		2	<b>Практические занятия</b> Решения задач с случайными числами	Случайные числа	0	2

Дата занятия		Кол-во часов	Содержание занятия	Раздел, тема	Время на изучение	
план	факт				теория	практика
		2	<i>Практические занятия</i> Примеры решения задач с циклом.	Примеры решения задач с циклом	0	2
		2	<i>Практические занятия</i> Решение задач с циклом	Решение задач с циклом.	0	2
		2	<i>Практические занятия</i> Решение задач с циклом	Решение задач с циклом.	0	2
		2	<i>Практические занятия</i> Решение задач с циклом	Решение задач с циклом.	0	2
		2	<i>Практические занятия</i> Выполнение самостоятельной работы	Самостоятельная работа № 2 "Составление программ с циклом"	0	2
		2	<i>Практические занятия</i> Выполнение самостоятельной работы	Самостоятельная работа № 2 "Составление программ с циклом"	0	2
		2	<i>Практические занятия</i> Выполнение самостоятельной работы	Самостоятельная работа № 2 "Составление программ с циклом"	0	2
		2	<i>Практические занятия</i> Выполнение самостоятельной работы	Самостоятельная работа № 2 "Составление программ с циклом"	0	2
		2	<i>Практические занятия</i> Выполнение самостоятельной работы	Самостоятельная работа № 2 "Составление программ с циклом"	0	2
		2	<i>Практические занятия</i> Решение теста	Тест № 4. Циклы	0	2
		2	<i>Практические занятия</i> Выполнение творческой работы.	Творческая работа № 1. "Циклы"	0	2
		2	<i>Практические занятия</i> Выполнение творческой работы.	Творческая работа № 1. "Циклы"	0	2
		2	<i>Практические занятия</i> Выполнение творческой работы.	Творческая работа № 1. "Циклы"	0	2
		2	<i>Практические занятия</i> Выполнение творческой работы.	Творческая работа № 1. "Циклы"	0	2
		2	<i>Практические занятия</i> Выполнение творческой работы.	Творческая работа № 1. "Циклы"	0	2
		2	<i>Практические занятия</i> Выполнение творческой работы.	Творческая работа № 1. "Циклы"	0	2

Дата занятия		Кол-во часов	Содержание занятия	Раздел, тема	Время на изучение	
план	факт				теория	практика
		2	<i>Теоретические занятия</i> Создание функций. Параметры и аргументы. <i>Практические занятия</i> Рекурсивные функции. Вычисление факториала.	Создание функций	1	1
		2	<i>Теоретические занятия</i> Локальные и глобальные переменные. Поток выполнения. Функции, возвращающие результат. <i>Практические занятия</i> Рекурсивные функции. Вычисление факториала.	Локальные переменные	1	1
		2	<i>Практические занятия</i> Примеры решения задач с использованием функций.	Решения задач с использованием функций	0	2
		2	<i>Практические занятия</i> Выполнение самостоятельной работы.	Самостоятельная работа № 3 по теме "Функции"	0	2
		2	<i>Практические занятия</i> Выполнение самостоятельной работы.	Самостоятельная работа № 3 по теме "Функции"	0	2
		2	<i>Теоретические занятия</i> Локальные и глобальные переменные. Поток выполнения. Функции, возвращающие результат. <i>Практические занятия</i> Решение практических задач с рекурсивными функциями	Рекурсивные функции	1	1
		2	<i>Практические занятия</i> Выполнение теста.	Тест № 5. Функции	0	2
		2	<i>Теоретические занятия</i> Составной тип данных - строка. Доступ по индексу. Длина строки и отрицательные индексы. Преобразование типов. Применение цикла для обхода строки. <i>Практические занятия</i> Решение практических задач, создание строк	Строки	1	1

Дата занятия		Кол-во часов	Содержание занятия	Раздел, тема	Время на изучение	
план	факт				теория	практика
		2	<b>Теоретические занятия</b> Срезы строк. Строки нельзя изменить. Сравнение строк. Оператор in. Модуль string. Операторы для всех типов <b>Практические занятия</b> Решение практических задач, создание строк	Срезы строк	1	1
		2	<b>Теоретические занятия</b> Разбор примеров решения задач со строками <b>Практические занятия</b> Написание программы для обнаружения роботом черной линии.	Примеры решения задач со строками	1	1
		2	<b>Практические занятия</b> Решение задач.	Решения задач со строками	0	2
		2	<b>Практические занятия</b> Решение задач.	Решения задач со строками	0	2
		2	<b>Теоретические занятия</b> Списки. Тип список (list). Индексы. Обход списка. Проверка вхождения в список. Добавление в список. Суммирование или изменение списка. Операторы для списков. Срезы списков. Удаление списка. Клонирование списков. Списочные параметры. Функция range. <b>Практические занятия</b> Решение практических задач, создание списков	Списки	1	1
		2	<b>Практические занятия</b> Сборка и программирование робота для движения вдоль черной линии.	Срезы списков	0	2
		2	<b>Практические занятия</b> Сборка и программирование робота для движения вдоль черной линии.	Списки: примеры решения задач	0	2
		2	<b>Теоретические занятия</b> Примеры задач со списками <b>Практические занятия</b> Решение задач.	Решение задач со списками	1	1
		2	<b>Практические занятия</b> Решение задач.	Решение задач со списками	0	2

Дата занятия		Кол-во часов	Содержание занятия	Раздел, тема	Время на изучение	
план	факт				теория	практика
		2	<b>Теоретические занятия</b> Матрицы. Вложенные списки. Матрицы. Строки и списки. Генераторы списков в Python. Кортежи. Присваивание кортежей. Кортежи как возвращаемые значения. Введение в словари. Тип словарь (dict). Словарные операции. Словарные методы.	Матрицы.	2	0
		2	<b>Практические занятия</b> Словарные операции.	Списки, Кортежи	0	2
		2	<b>Практические занятия</b> Словарные операции.	Введение в словари.	0	2
		2	<b>Теоретические занятия</b> Множества в языке Python. Множества. Множественный тип данных. Описание множеств. <b>Практические занятия</b> Операции, допустимые над множествами: объединение, пересечение, разность, включение. Оператор определения принадлежности элемента множеству	Множества в языке Python	1	1
		2	<b>Теоретические занятия</b> Стиль программирования. <b>Практические занятия</b> Практическая работа по программированию	Стиль программирования	1	1
		2	<b>Теоретические занятия</b> Отладка программ <b>Практические занятия</b> Практическая работа по программированию	Отладка программ	1	1
		2	<b>Теоретические занятия</b> Зачетная работа	Зачет по курсу «Программирование на языке Python»	2	0
		2	<b>Теоретические занятия</b> Перспективы и планы дальнейшего изучения языка	Перспективы и планы дальнейшего изучения языка	0	2
		136			35	101

## 2.4 ОЦЕНОЧНЫЕ МАТЕРИАЛЫ

### Формы предъявления и демонстрации образовательных результатов

Аналитический материал по итогам проведения диагностики, защита творческих работ, конкурс, научно-практическая конференция, открытое занятие.

### Перечень (пакет) диагностических методик

**Входной контроль:** *Тестирование (Приложение 1)*

**Промежуточный контроль:** *Тестирование, решение задач (Приложение 2)*

**Итоговый контроль:** *учащимся предлагается самостоятельно выбрать тему творческого проекта и на основании темы разработать программу, пояснительную записку, презентацию.*

## 2.5 СПИСОК ЛИТЕРАТУРЫ

### Дидактические материалы

-специальная и научно-популярная литература для педагога и учащихся;  
-раздаточные материалы, инструкционные, технологические карты, задания, упражнения, образцы изделий и т.п.

### Список литературы для педагога

1. Бизли, Дэвид М. Python. Подробный справочник. – М.–СПб.: Символ-Плюс, 2010.
2. Лутц, Марк Python. Справочник. – М.: Вильямс, 2015.
3. Официальный сайт программы [Электронный ресурс] – Режим доступа: <https://docs.python.org/>, свободный.
4. Сайт, среда разработки для языка Python. [Электронный ресурс] – Режим доступа: <https://www.jetbrains.com/pycharm/?fromMenu>, свободный.

### Список литературы для родителей и учащихся

1. Сайт / справочные материалы [Электронный ресурс] – Режим доступа: <https://metanit.com/python/>, свободный.
2. Сайт / интерактивный сборник задач для практики программирования [Электронный ресурс] – Режим доступа: <http://pythontutor.ru/>, свободный.
3. Сайт / Адаптивный тренажер Python[Электронный ресурс] – Режим доступа: <https://stepik.org/course/431> , свободный.
4. Сайт / среда разработки для языка Python[Электронный ресурс] – Режим доступа: <https://www.jetbrains.com/pycharm/?fromMenu> , свободный.

# МЕТОДИЧЕСКИЙ КЕЙС

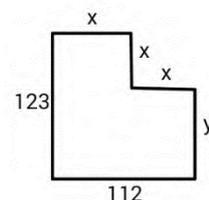
## ПРИЛОЖЕНИЕ № 1

### Входной контроль

Низкий уровень: 0-70% выполненных заданий;

Средний уровень: 70-85% правильно выполненных заданий;

Высокий уровень: 85-100% правильно выполненных заданий.



Фамилия, Имя	
--------------	--

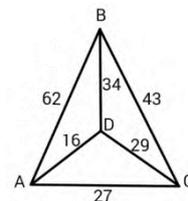
1. Из прямоугольника вырезали квадрат со стороной  $x$ , получилась фигура как на рисунке. Чему равна сумма цифр  $y$ ?

2. В алфавите племени мумба-юмба 32 буквы. Любое слово в языке этого племени состоит из пяти букв и должно одинаково читаться справа налево и слева направо, при этом первые две буквы слова обязательно различаются, а третья совпадает с пятой. Каково максимальное количество слов в этом языке?

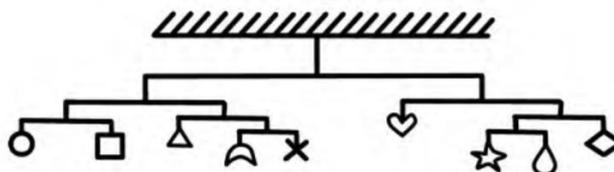
3. Катя наклеила на рулет тонкие поперечные кольца трёх разных цветов. Если разрезать по серым кольцам, получится 25 кусков рулета, если по малиновым — 47 кусков, а если по зеленым — 31 кусок. Сколько кусков рулета получится, если разрезать по кольцам всех трёх цветов?

**Примечания.** Учтите, что наклеить кольцо одного цвета на кольцо другого нельзя.

4. На рисунке показано расположение городов А, В, С и D расстояния между ними. Турист выходит из города В и собирается посетить остальные города, побывав в каждом по разу. Какова наименьшая возможная длина маршрута, если он хочет закончить свой путь в том же городе?



5. Фигурки, общей массой 432 грамма, при помощи невесомых нитей и планок собрали в конструкцию, изображённую на рисунке. Оказалось, что все её части находятся в равновесии. Сколько весит ромб?



6. Люди переезжают в города, за год численность людей удваивается. Если люди заселят весь город за 12 лет, то сколько лет понадобится, чтобы занять лишь половину города?

7. Сравните пары слов. Сколько среди них полностью идентичных?

O/Sanmarco  
Ф.Wagonerrte  
A.S.Schmetterling  
N.V.Murfreesboroque  
P.S.Splendoursec

O/Samnarco  
Ф.Wagonertre  
A.S.Schnetterling  
N.V.Munfreesboroque  
P.S.Sqlendoursec

8. Семь человек выясняли, какой сегодня день недели.

Первый сказал: «Послезавтра - воскресенье».

Второй: «Вчера был понедельник».

Третий: «Завтра будет суббота».

Четвертый: «Завтра будет среда».

Пятый: «Вчера был четверг».

Шестой: «Позавчера было воскресенье».

Седьмой: «Позавчера была среда».

Какой сегодня день недели, если трое ошибаются?

9. Вам предложены несколько высказываний и следствие из них (выделено жирным).

Согласны ли Вы с этим следствием?

1. Все клёны — растения.

2. Некоторые растения быстро желтеют.

**Значит, некоторые клёны быстро желтеют.**

Да

Нет

10. Гусеница прогрызает яблоко диаметром 6 сантиметров насквозь за 16 секунд, вылезая снаружи полностью. Известно, что середину яблока она начинает грызть уже через 6 секунд после начала пути. Какова длина гусеницы в сантиметрах?

<p><b>Вопросы для промежуточного контроля по усвоению материала</b> (<i>Промежуточный контроль проводится в последнюю неделю декабря</i>)</p>
<p>Составьте выражение для вычисления в интерпретаторе Python3 и вставьте в поле ответа результат вычисления:</p> <p style="text-align: center;">11111*1111111</p> <p>— произведение чисел 11111 (5 единиц) и 1111111 (7 единиц)</p>
<p>Запишите число 1.2345e3 в виде десятичной дроби.</p>
<p>Составьте и запишите выражение для вычисления: 2014.0^14(Возвестив 14 степень) Обратите внимание на запись числа: это вещественное число.</p>
<p>Приведите к целому типу число 2.99</p>
<p>Расставьте скобки в выражении a and b or not a and not b в соответствии с порядком вычисления выражения (приоритетом операций). Всего потребуется 5 пар скобок (внешние скобки входят в их число).</p>
<p>Найдите результат выражения для заданных значений a и b Учтите регистр символов при ответе. a = True b = False a and b or not a and not b</p>
<p>Отметьте выражения, значения которых равны True:</p> <ul style="list-style-type: none"> <li>• "239" &lt; "30" and 239 &lt; 30</li> <li>• "239" &lt; "30" and 239 &gt; 30</li> <li>• "239" &gt; "30" and 239 &lt; 30</li> <li>• "239" &gt; "30" and 239 &gt; 30</li> </ul>
<p>Укажите результат выражения: "123"+"42"</p>
<p>Какое значение будет у переменной i после выполнения фрагмента программы? i = 0 while i &lt;= 10: i = i + 1 if i &gt; 7: i = i + 2</p>
<p>Сколько итераций цикла будет выполнено в этом фрагменте программы? i = 0 while i &lt;= 10: i = i + 1 if i &gt; 7: i = i + 2</p>
<p>Сколько всего знаков * будет выведено после исполнения фрагмента программы: i = 0 while i &lt; 5: print('*') if i % 2 == 0: print('**') if i &gt; 2: print('***') i = i + 1</p>

Определите, какое значение будет иметь переменная  $i$  после выполнения следующего фрагмента программы:

```
i = 0
s = 0
while i < 10:
    i = i + 1
    s = s + i
    if s > 15:
        break
    i = i + 1
```

Определите, какое значение будет иметь переменная  $i$  после выполнения следующего фрагмента программы:

```
i = 0
s = 0
while i < 10:
    i = i + 1
    s = s + i
    if s > 15:
        continue
    i = i + 1
```

#### **Задачи для промежуточного контроля по усвоению материала**

*Напишите простой калькулятор, который считывает с пользовательского ввода три строки: первое число, второе число и операцию, после чего применяет операцию к введённым числам ("первое число" "операция" "второе число") и выводит результат на экран. Поддерживаемые операции: +, -, /, \*, mod, pow, div, где mod— это взятие остатка от деления, pow— возведение в степень, div— целочисленное деление. Если выполняется деление и второе число равно 0, необходимо выводить строку "Деление на 0!". Обратите внимание, что на вход программе приходят вещественные числа.*

Жители страны Малевии часто экспериментируют с планировкой комнат. Комнаты бывают треугольные, прямоугольные и круглые. Чтобы быстро вычислять жилплощадь, требуется написать программу, на вход которой подаётся тип фигуры комнаты и соответствующие параметры, которая бы выводила площадь получившейся комнаты.

Для числа  $\pi$  в стране Малевии используют значение 3.14.

Формат ввода, который используют Малевийцы:

Треугольник  $abc$ , где  $a$ ,  $b$  и  $c$ — длины сторон треугольника

Прямоугольник  $ab$ , где  $a$  и  $b$ — длины сторон прямоугольника

Круг  $r$ , где  $r$ — радиус окружности

*Напишите программу, которая получает на вход три целых числа, по одному числу в строке, и выводит на консоль в три строки сначала максимальное, потом минимальное, после чего оставшееся число. На ввод могут подаваться и повторяющиеся числа.*

Паша очень любит кататься на общественном транспорте, а получая билет, сразу проверяет, счастливый ли ему попался. Билет считается счастливым, если сумма первых трех цифр совпадает с суммой последних трех цифр номера билета. Однако Паша очень плохо считает в уме, поэтому попросил вас написать программу, которая проверит равенство сумм и выведет "Счастливый", если суммы совпадают, и "Обычный", если суммы различны.

На вход программе подаётся строка из шести цифр. (Пример: 123321)

Выводить нужно только слово "Счастливый" или "Обычный", с большой буквы.

Напишите программу, которая считывает с консоли числа (по одному в строке) до тех пор, пока сумма введённых чисел не будет равна 0 и **сразу после этого** выводит сумму квадратов всех считанных чисел.

Гарантируется, что в какой-то момент сумма введённых чисел окажется равной 0, **после этого считывание продолжать не нужно**.

В примере мы считываем числа 1, -3, 5, -6, -10, 13; в этот момент замечаем, что сумма этих чисел равна нулю и выводим сумму их квадратов, не обращая внимания на то, что остались ещё не прочитанные значения.

Напишите программу, которая выводит часть последовательности 1 2 2 3 3 4 4 4 5 5 5 5 ... (число повторяется столько раз, чему равно). На вход программе передаётся неотрицательное целое число  $n$ — столько элементов последовательности должна отобразить программа. На выходе ожидается последовательность чисел, записанных через пробел в одну строку.

Например, если  $n=7$ , то программа должна вывести 1 2 2 3 3 3 4.

Выведите таблицу размером  $n*n$ , заполненную числами от 1 до  $n^2$  по спирали, выходящей из левого верхнего угла и закрученной по часовой стрелке, как показано в примере (здесь  $n=5$ )

### Итоговый контроль

1. Назовите два числа, у которых количество цифр равно количеству букв, составляющих название каждого из этих чисел.
2. Собака была привязана к десятиметровой веревке, а прошла двести метров. Как ей это удалось?
3. Есть дорога, по которой может проехать только одна машина. По дороге едут две машины: одна с горы, другая под гору. Как им разъехаться?
4. Как с помощью только одной палочки образовать на столе треугольник?
5. В каком месяце болтливая девочка говорит меньше всего?
6. Что становится на треть больше, если его поставить вверх ногами?
7. Представьте себе, что вы кондуктор. Поезд везет сто вагонов, в каждом вагоне 10 купе, в каждом купе 4 пассажира. Сколько лет кондуктору?
8. Юра разрезал огромную пиццу на 10 кусков. Затем он взял один из кусков и разрезал его еще на 10. После этого из имеющихся кусков он выбрал два и разрезал каждый из них на 10. **Вопрос:** Сколько кусков пиццы получилось у Юры.
9. В компьютерной игре нужно победить монстра. Изначально у Юры было только 9 выстрелов. Но за каждое попадание он получал дополнительно еще 3 выстрела. **Вопрос:** Сколько раз Юра попал в цель, если всего он выстрелил 30 раз, израсходовав все выстрелы?

### Диагностическая карта

№	Имя	Входная диагностика	Промежуточная диагностика	Итоговая диагностика
1				
	Итого в %	Н С В		Н С В

Результативность отслеживается с помощью карты наблюдений, анализа участия детей в совместной продуктивной деятельности, разработки и защите творческих проектов. Усвоение программы возможно по 3-м уровням: низкий (Н), средний (С), высокий (В).

#### Низкий уровень

*Учащиеся должны знать/ понимать:*

- технику безопасного поведения во время занятий; правила поведения в общественных местах,
- понятие программы Python;

- общую структуру программы;
- основные типы данных;
- оператор присваивания;
- назначение условного оператора;
- способ записи условного оператора;
- логический тип данных;
- логические операторы or, and, not;
- основные циклы с условием;
- основные правила записи циклов условием; формат записи цикла с параметром;
- понятие функции;
- основные принципы структурного программирования;
- понятие локальных переменных подпрограмм;
- способ передачи параметров.
- назначение строкового типа данных;
- операторы для работы со строками;
- операции со строками;
- способ описания списка;
- способ описания кортежа;
- способ описания словаря;
- основные операции, выполняемые со списками, кортежами и словарями;
- понятие множества;
- способы описания множества;
- операторы работы с множествами.
- что такое стиль программирования;
- правила именования объектов;
- основные рекомендации при написании программ.

*Учащиеся должны уметь:*

- понимать учебную задачу, сохранять ее содержание в процессе ее выполнения под руководством педагога;
- работать в паре, малой группе; выполнить установку программы под руководством педагога;
- выполнить простейшую программу в интерактивной среде;
- написать комментарии в программе под руководством педагога;
- решать задачи на элементарные действия с числами;
- использовать условный оператор;
- определять вид цикла, наиболее удобный для решения поставленной задачи;
- использовать цикл с условием под руководством педагога;
- определять целесообразность применения и использования цикла с параметром для решения поставленной задачи по наводящим вопросам педагога;
- создавать и использовать основные функции;
- описывать и соединять строки;
- находить подстроку в строке с помощью педагога;
- находить количество слов в строке;
- вводить и выводить элементы списка под руководством педагога;
- приводить примеры использования вложенных списков (матриц) по наводящим вопросам педагога;
- описывать множества под руководством педагога; определять принадлежность элемента множеству по наводящим вопросам;
- определять вид ошибок и находить ошибки в программе под руководством педагога;
- составлять элементарные алгоритмы для решения задач;

- реализовывать алгоритмы на компьютере в виде программ, написанных на языке Python под руководством педагога.

### **Средний уровень**

*Учащиеся должны знать/ понимать:*

- основные положения техники безопасности на занятиях, правила поведения в общественных местах, правила дорожной безопасности;
- основные приемы взаимодействия в группе сверстников;
- понятие программы;
- структуру программы на Python;
- режимы работы Python.
- типы данных;
- целые, вещественные типы данных и операции над ними;
- оператор присваивания;
- назначение условного оператора;
- способ записи условного оператора;
- логический тип данных;
- логические операторы or, and, not;
- циклы с условием и их виды;
- назначение и особенности использования цикла с параметром;
- формат записи цикла с параметром; примеры использования циклов различных типов.
- понятие функции;
- основные способы описания функции;
- принципы структурного программирования;
- понятие локальных переменных подпрограмм;
- понятие формальных и фактических параметров подпрограмм;
- способ передачи параметров.
- назначение строкового типа данных;
- операторы для работы со строками;
- процедуры и функции для работы со строками;
- операции со строками;
- сложные типы данных;
- способ описания списка;
- способ доступа к элементам списка;
- способ описания кортежа;
- способ описания словаря;
- операции, выполняемые со списками, кортежами и словарями;
- понятие множества;
- способы описания множества;
- операторы работы с множествами.
- что такое стиль программирования;
- правила именования объектов;
- основные рекомендации при написании программ; основные шаги работы над проектом, его презентации.

*Учащиеся должны уметь:*

- уважительно относиться к преподавателям и сверстникам;
- применять некоторые приемы логического (абстрактное) мышления;
- концентрировать внимание на одном или двух объектах;
- понимать причины успеха/неуспеха с помощью анализа педагога;
- выполнить установку программы;
- выполнить простейшую программу в интерактивной среде;

- написать комментарии в программе;
- решать задачи на элементарные действия с числами;
- использовать условный оператор;
- создавать сложные условия с помощью логических операторов;
- определять вид цикла, наиболее удобный для решения поставленной задачи;
- использовать цикл с условием;
- определять целесообразность применения и использования цикла с параметром для решения поставленной задачи;
- создавать и использовать функции;
- использовать механизм параметров для передачи значений;
- описывать строки;
- соединять строки;
- находить длину строки;
- вырезать часть строки;
- находить подстроку в строке;
- находить количество слов в строке;
- описывать списки;
- вводить элементы списка;
- выводить элементы списка;
- выполнять поиск элемента в списке, поиск минимума и максимума, нахождение суммы элементов списка;
- использовать вложенные списки;
- приводить примеры использования вложенных списков (матриц);
- описывать множества;
- определять принадлежность элемента множеству;
- вводить элементы множества;
- выводить элементы множества.
- определять вид ошибок и находить ошибки в программе.
- составлять алгоритмы для решения задач;
- реализовывать алгоритмы на компьютере в виде программ, написанных на языке Python;
- отлаживать и тестировать программы, написанные на языке Python;
- понимать и сохранять цели и задачи учебной деятельности в ходе занятия;
- планировать свою деятельность с помощью взрослого;
- сотрудничать с взрослыми и сверстниками в процессе выполнения учебной задачи;
- понимать причины успеха/неуспеха учебной деятельности под руководством взрослого;
- делать выводы в ходе полученных заданий; выстраивать логические цепи рассуждений под руководством педагога;
- выражать творческие идеи, разработать творческий проект на основе образца;
- конструктивно взаимодействовать в составе группы в ходе работы над проектом.

### **Высокий уровень**

*Учащиеся должны знать/ понимать:*

- основные положения техники безопасности на занятиях, правила поведения в общественных местах, правила дорожной безопасности, правила поведения во время чрезвычайных происшествий;
- приемы конструктивного взаимодействия в группе сверстников;
- понятие программы;
- структуру программы на Python;
- режимы работы в Python.
- Общую структуру программы;
- типы данных;

- целые, вещественные типы данных и операции над ними;
- оператор присваивания;
- назначение условного оператора;
- способ записи условного оператора;
- логический тип данных;
- логические операторы or, and, not;
- циклы с условием и их виды;
- правила записи циклов с условием;
- назначение и особенности использования цикла с параметром;
- формат записи цикла с параметром;
- примеры использования циклов различных типов.
- понятие функции;
- способы описания функции;
- принципы структурного программирования;
- понятие локальных переменных подпрограмм;
- понятие формальных и фактических параметров подпрограмм;
- способ передачи параметров.
- назначение строкового типа данных;
- операторы для работы со строками;
- процедуры и функции для работы со строками;
- операции со строками;
- сложные типы данных;
- способ описания списка;
- способ доступа к элементам списка;
- способ описания кортежа;
- способ описания словаря;
- операции, выполняемые со списками, кортежами и словарями;
- понятие множества;
- способы описания множества;
- операторы работы с множествами.
- что такое стиль программирования;
- правила именования объектов;
- основные рекомендации при написании программ;
- правила и этапы работы над проектом;
- приемы успешной презентации проекта.

*Учащиеся должны уметь:*

- выполнить установку программы;
- выполнить простейшую программу в интерактивной среде;
- написать комментарии в программе;
- решать задачи на элементарные действия с числами;
- использовать условный оператор;
- создавать сложные условия с помощью логических операторов;
- определять вид цикла, наиболее удобный для решения поставленной задачи;
- использовать цикл с условием;
- определять целесообразность применения и использования цикла с параметром для решения поставленной задачи;
- создавать и использовать функции;
- использовать механизм параметров для передачи значений;
- описывать строки;

- соединять строки;
- находить длину строки;
- вырезать часть строки;
- находить подстроку в строке;
- находить количество слов в строке;
- описывать списки;
- вводить элементы списка;
- выводить элементы списка;
- выполнять поиск элемента в списке, поиск минимума и максимума, нахождение суммы элементов списка;
- использовать вложенные списки;
- приводить примеры использования вложенных списков (матриц);
- описывать множества;
- определять принадлежность элемента множеству;
- вводить элементы множества;
- выводить элементы множества.
- определять вид ошибок и находить ошибки в программе.
- составлять алгоритмы для решения задач;
- реализовывать алгоритмы на компьютере в виде программ, написанных на языке Python;
- отлаживать и тестировать программы, написанные на языке Python;
- понимать учебную задачу, анализировать достижение результата;
- делать выводы в ходе полученных заданий; самостоятельно выстраивать логические цепи рассуждений;
- понимать причины успеха/неуспеха, конструктивно принимать причины неуспеха, исправлять ошибки с минимальным участием педагога;
- планировать свою деятельность, находить оригинальные способы выполнения поставленной творческой задачи;
- создать проективную команду и организовать ее деятельность;
- разработать творческий проект по предложенной или самостоятельно выбранной теме в составе творческой группы;
- самостоятельно подготовить презентацию и защитить проект