

Муниципальное бюджетное общеобразовательное учреждение  
Инзенская средняя школа № 2 имени П.И.Бодина

Рассмотрена и принята на  
заседании педагогического  
совета от «21» 06 2023  
г. протокол № 6



Директор МБОУ ИСШ № 2  
Шкунова И.К.  
«22» 06 2023 г.

**ДОПОЛНИТЕЛЬНАЯ ОБЩЕРАЗВИВАЮЩАЯ  
ОБЩЕОБРАЗОВАТЕЛЬНАЯ ПРОГРАММА  
«Основы программирования на языке Python»  
технической направленности**

Срок обучения: 1 год  
(70 часов, 1 раз в неделю по 2 часа)  
Уровень программы: стартовый

Возраст учащихся: 15 - 16 лет.

Учитель: Тюрина А.А.  
Учитель информатики,  
педагог допол. образования

Инза, 2023 г.

## Содержание

1. Комплекс основных характеристик программы.....	3
1.1 Пояснительная записка.....	3
1.2 Цель и задачи программы.....	6
1.3 Планируемые результаты освоения программы.....	7
1.4 Учебный план.....	9
1.5 Содержание программы.....	11
2. Комплекс организационно-педагогических условий.....	13
2.1 Календарно-учебный график.....	13
2.2 Условия реализации программы.....	16
2.3 Формы аттестации.....	16
2.4 Методические материалы.....	17
Список литературы.....	18
Приложения.....	19

# РАЗДЕЛ 1. КОМПЛЕКС ОСНОВНЫХ ХАРАКТЕРИСТИК ПРОГРАММЫ

## 1.1. Пояснительная записка

Данная дополнительная общеобразовательная общеразвивающая программа разработана в рамках реализации образовательного проекта «Точка роста» и направлена на оптимизацию личностно-ориентированного обучения и становление проектной деятельности обучающихся в области программирования на языке Python.

### **Нормативно-правовое обеспечение программы.**

В настоящее время содержание, роль, назначение и условия реализации программ дополнительного образования закреплены в следующих нормативных документах:

- Федеральным законом от 29 декабря 2012 года № 273-ФЗ «Об образовании в Российской Федерации» (ст. 2, ст. 15, ст.16, ст.17, ст.75, ст. 79);

- Проект Концепции развития дополнительного образования детей до 2030 года;

- приказом Минпросвещения РФ от 09.11.2018 года № 196 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам»;

- приказом от 30 сентября 2020 г. N 533 «О внесении изменений в порядок организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам, утвержденный приказом Министерства просвещения Российской Федерации от 9 ноября 2018 г. № 196»;

- методическими рекомендациями по проектированию дополнительных общеразвивающих программ № 09-3242 от 18.11.2015 года;

- СП 2.4.3648-20 Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи;

- приказ Министерства образования и науки РФ от 23.08.2017 года № 816 «Порядок применения организациями, осуществляющими образовательную деятельность электронного обучения, дистанционных образовательных технологий при реализации образовательных программ»

- методические рекомендации от 20 марта 2020 г. по реализации образовательных программ начального общего, основного общего, среднего общего образования, образовательных программ среднего профессионального образования и дополнительных общеобразовательных программ с применением электронного обучения и дистанционных образовательных технологий.

- Устав МБОУ Инзенская СШ № 2;

- Положение о разработке, структуре и порядке утверждения дополнительной общеразвивающей программы (локальный акт МБОУ Инзенская СШ № 2);

- Положение о проведении промежуточной и итоговой аттестации обучающихся (локальный акт МБОУ Инзенская СШ № 2).

- Письма Минобрнауки РФ от 11.12.2006 N 06-1844 "О примерных требованиях к программам дополнительного образования детей"

**Уровень освоения программы:** стартовый

**Направленность (профиль) программы** – техническая

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.

**Новизна программы.** 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** 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 внутрипрофильной специализации в области новых информационных технологий, что способствует созданию дополнительных условий для проявления индивидуальных образовательных интересов учащихся, их дальнейшей профессиональной ориентации.

### **Отличительные особенности программы:**

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 практикоориентированных форм занятий, способствующих формированию основных компетенций (информационных, коммуникативных, компетенций личного развития и др.).

### **Практическая значимость:**

В рамках предлагаемого курса «Программирование на языке Python» изучение основ программирования на языке Python – это не столько средство подготовки к будущей профессиональной деятельности, сколько формирование новых общеинтеллектуальных умений и навыков: разделение задачи на этапы решения, построение алгоритма и др. Особую роль программирование служит для формирования мыслительных и психических процессов учащихся (внимание, память, логика), освоения 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 активизирует процесс индивидуально-личностного становления учащихся.

### **Адресат программы**

Дополнительная общеразвивающая программа «Программирование на языке Python» рассчитана на детей 15–16 лет, проявляющих интерес к информационно-коммуникационным технологиям. Учащиеся объединения являются разными по возрасту и социальному статусу. Объединение комплектуется на основании заявлений самих учащихся. Группы формируются из школьников разного возраста на добровольной внеконкурсной основе.

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.

**Состав группы** – постоянный.

**Количество учащихся** в группе 12 - 15 человек.

### **Принципы комплектования группы**

Из обучающихся 15 – 16 лет формируется объединение дополнительного образования, занятия в котором проводятся по данной дополнительной общеразвивающей программе. В объединении может быть как одна группа, так и несколько групп обучающихся в зависимости от возраста. Деятельность осуществляется в разновозрастных и/или разновозрастных группах. С учетом потребностей и возрастных возможностей обучающихся программа может осваиваться в различных формах занятий: лекции, семинары, практикумы, конференции, организационно-деловые игры, тренинги, мастер-классы и другое, что

определяется рабочей общеразвивающей программой дополнительного образования. Для наиболее полного удовлетворения запросов обучающихся и их родителей (законных представителей) наряду с очной формой программа может частично реализовываться в режиме дистанционного обучения

### **Основные методы обучения**

В образовательной программе используются методы обучения, которые обеспечивают продуктивное научно-техническое образование. Обучение опирается на такие виды образовательной деятельности, которые позволяют обучающимся:

- познавать окружающий мир (когнитивные);
- создавать при этом образовательную продукцию (креативные);
- организовывать образовательный процесс (оргдеятельностные).

Использование совокупности методов, представленных в данной классификации, позволяет наиболее точно охарактеризовать (проанализировать) образовательный процесс и, при необходимости, корректировать его в соответствии с поставленной в программе целью.

Когнитивные методы, или методы учебного познания окружающего мира - это, прежде всего, методы исследований в различных науках – методы сравнения, анализа, синтеза, классификации.

Применение когнитивных методов приводит к созданию образовательной продукции, т.е. к креативному результату, хотя первичной целью использования данных методов является познание объекта.

**Метод эвристических вопросов** предполагает для отыскания сведений о каком-либо событии или объекте задавать следующие семь ключевых вопросов: Кто? Что? Зачем? Чем? Где? Когда? Как?

**Метод сравнения** применяется для сравнения разных версий моделей обучающихся с созданными аналогами.

**Метод эвристического наблюдения** ставит целью научить детей добывать и конструировать знания с помощью наблюдений. Одновременно с получением заданной педагогом информации многие обучающиеся видят и другие особенности объекта, т.е. добывают новую информацию и конструируют новые знания.

**Метод фактов** учит отличать то, что видят, слышат, чувствуют обучающиеся, от того, что они думают. Таким образом, происходит поиск фактов, отличие их от не фактов, что важно для инженера-робототехника.

**Метод конструирования** понятий начинается с актуализации уже имеющихся представлений обучающихся. Сопоставляя и обсуждая детские представления о понятии, педагог помогает достроить их до некоторых культурных форм. Результатом выступает коллективный творческий продукт – совместно сформулированное определение понятия.

**Метод прогнозирования** применяется к реальному или планируемому процессу. Спустя заданное время прогноз сравнивается с реальностью. Проводится обсуждение результатов, делаются выводы.

**Метод ошибок** предполагает изменение устоявшегося негативного отношения к ошибкам, замену его на конструктивное использование ошибок. Ошибка рассматривается как источник противоречий, феноменов, исключений из правил, новых знаний, которые рождаются на противопоставлении общепринятым.

**Креативные методы** обучения ориентированы на создание обучающимися личного образовательного продукта – совершенного робота, путем проб, ошибок, накопленных знаний и поиском оптимального решения проблемы.

**Метод «Если бы...»** предполагает составить описание того, что произойдет, если в автоматизированной системечто-либо изменится.

**«Мозговой штурм»** ставит основной задачей сбор как можно большего числа идей в результате освобождения участников обсуждения от инерции мышления и стереотипов.

**Метод планирования** предполагают планирование образовательной деятельности на определенный период - занятие, неделю, тему, творческую работу.

**Метод контроля** в научно-техническом обучении образовательный продукт юного конструктора и программиста оценивается по степени отличия от заданного, т.е. чем больше оптимальных конструкторских идей выдумывают обучающиеся, тем выше оценка продуктивности его образования.

**Метод рефлексии** помогают обучающимся формулировать способы своей деятельности, возникающие проблемы, пути их решения и полученные результаты, что приводит к осознанному образовательному процессу.

**Метод самооценки** вытекают из методов рефлексии, носят количественный и качественный характер, отражают полноту достижения обучающимся цели.

Для контроля:

**Метод предварительный** (анкетирование, диагностика, наблюдение, опрос).

**Метод текущий** (наблюдение, ведение таблицы результатов);

**Метод тематический** (билеты, тесты);

**Метод итоговый** (соревнования).

**Объем программы:** 70 часов, по 2 часа в неделю.

**Формы обучения и виды занятий:** лекции, практические занятия, мастер-классы, сценические постановки.

**Срок освоения программы:** Программа составлена на один учебный год

## 1.2.Цели и задачи

**Цель:** Способствовать формированию творческой личности, обладающей информационными компетенциями, владеющей базовыми понятиями теории алгоритмов, умеющей разрабатывать эффективные алгоритмы и реализовывать их в виде программы, написанной на языке программирования Python.

**Задачи:**

**Образовательные:**

познакомить с понятиями алгоритма, вычислимой функции, языка программирования;

научить составлять и читать блок-схемы;

сформировать навыки выполнения технологической цепочки разработки программ средствами языка программирования Python;

объяснить основные конструкции языка программирования Python, позволяющие работать с простыми и составными типами данных (строками, списками, кортежами, словарями, множествами);

научить применять функции при написании программ на языке программирования Python;

научить отлаживать и тестировать программы, делать выводы о работе этих программ.

**Воспитательные:**

воспитать уважительное отношение к преподавателям и сверстникам, культуру поведения во время занятий и совместной продуктивной деятельности;

сформировать культуру занятий, направленную на воспитание личностных и социальных качеств;

**Развивающие:**

развить познавательные процессы (внимание, восприятие, логическое мышление, память),

развить креативность,

развить способности к самореализации.

### 1.3. Планируемые результаты

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

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

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

умение соотносить свои действия с планируемыми результатами, осуществлять контроль всей деятельности в процессе достижения результата, корректировать свои действия в соответствии с изменяющейся ситуацией;

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

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

умение организовывать сотрудничество и совместную деятельность с учителем и сверстниками; работать индивидуально в группе: находить общее решение и разрешать конфликты на основе согласования позиций и учета интересов; формулировать, аргументировать и отстаивать свое мнение;

формирование и развитие компетентности в области использования информационно-коммуникационных технологий.

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

формирование ответственного отношения к обучению, готовности и способности обучающихся к саморазвитию и самообразованию на основе мотивации к обучению и познанию;

формирование коммуникативной компетентности в общении и сотрудничестве со сверстниками, взрослыми в процессе образовательной, общественно-полезной, учебно-исследовательской, творческой и других видах деятельности.

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

формирование знаний, умений и навыков при решении задач информатики и программирования разных видов;

приобретение опыта использования информационных технологий в индивидуальной и коллективной учебной и познавательной, в том числе проектной деятельности.

формирование информационной и алгоритмической культуры; формирование представления о компьютере как универсальном устройстве обработки информации; развитие основных навыков и умений использования компьютерных устройств;

формирование представления об основных изучаемых понятиях: информация, алгоритм, модель - и их свойства;

развитие алгоритмического мышления, необходимого для профессиональной деятельности в современном обществе; развитие умений составить и записать алгоритм для конкретного исполнителя; формирование знаний об алгоритмических конструкциях, логических значениях и операциях; знакомство с одним из языков программирования и основными алгоритмическими структурами - линейной, условной и циклической;

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

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

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

владение навыками и опытом разработки программ в среде программирования Python, включая тестирование и отладку программ; владение элементарными навыками формализации прикладной задачи и документирования программ; формирование умения работать с библиотеками программ; получение опыта использования компьютерных средств представления и анализа данных.

В результате изучения дисциплины учащиеся должны знать:

основные типы алгоритмов;

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

базовые алгоритмические конструкции;

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

дополнительные возможности языка Python для выражения различных алгоритмических ситуаций;

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

основы разработки простых игр в системе программирования Python.

Учащиеся должны уметь:

записывать алгоритмические структуры на языке программирования Python;

использовать Python для решения задач из области математики, физики;

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

использовать основные алгоритмические приемы при решении математических задач;

решать простые, сложные и нестандартные задачи;

создавать простые игры;

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

**Формы подведения итогов реализации программы**

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

#### **Формы демонстрации результатов обучения**

Представление результатов образовательной деятельности пройдет в форме публичной презентации решений кейсов командами и последующих ответов выступающих на вопросы наставника и других команд.

#### **Формы диагностики результатов обучения**

Беседа, тестирование, опрос.

## 1.4. Учебный план

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

2.	<b>Модуль 2 Методы программирования на языке Python. Введение в олимпиадное программирование.</b>	<b>42</b>	<b>7</b>	<b>35</b>	
2.1.	Срезы строк. Использование срезов. Метод FIND. Примеры решения задач.	2	1	1	Тестирование
2.2.	Отработка навыков решения простейших задач.	4		4	Практические задания
2.3.	Методы RFIND, REPLACE и COUNT. Примеры решения задач.	2	1	1	Тестирование
2.4.	Отработка навыков решения простейших задач.	2		2	Практические задания
2.5.	Решение несложных олимпиадных задач.	4		4	Практические задания
2.6.	Функции. Использование функций. Примеры решения задач.	2	1	1	Тестирование
2.7.	Возврат значений. Локальные и глобальные переменные. Примеры решения задач.	2	1	1	Тестирование
2.8.	Отработка навыков решения задач.	4		4	Практические задания
2.9.	Рекурсия. Использование рекурсии. Примеры решения задач.	2	1	1	Тестирование
2.10.	Отработка навыков решения задач.	2		2	Практические задания
2.11.	Кортежи. Функция RANGE, цикл FOR. Примеры решения задач.	2	1	1	Тестирование
2.12.	Отработка навыков решения задач.	4		4	Практические задания
2.13.	Списки. Метод SPLIT и JOIN. Примеры решения задач.	2	1	1	Тестирование
2.14.	Отработка навыков решения задач.	4		4	Практические задания
2.15.	Решение олимпиадных задач.	4		4	Практические задания
	<b>ИТОГО:</b>	<b>70</b>	<b>15</b>	<b>55</b>	

## 1.5. Содержание учебного плана

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

Тема 1.1. История языков программирования. Введение в язык программирования Python.

Среда программирования Python. Установка программы. Теория (1 ч.)

Тема 1.2. Типы данных и функции вывода. Определение переменной. Переменные и арифметические выражения.

Теория (1 ч.).

Практика (1 ч.) Тренировочные задания.

Форма контроля. Тестирование.

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

Практика (1 ч.) Тренировочные задания.

Форма контроля. Тестирование.

Тема 1.4. Отработка навыков решения простейших задач.

Практика (4 ч.)

Форма контроля: практические задания.

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

Форма контроля. Тестирование.

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

Практика (1 ч.) Тренировочные задания.

Форма контроля. Тестирование.

Тема 1.7. Отработка навыков решения простейших задач.

Практика (2 ч.)

Форма контроля: практические задания.

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

Практика (1 ч.) Тренировочные задания.

Форма контроля. Анализ практических заданий

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

Практика (1 ч.) Тренировочные задания.

Форма контроля. Анализ практических заданий

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

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

Практика (1ч.) Тренировочные задания.

Форма контроля. Анализ практических заданий

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

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

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

Практика (1 ч.) Тренировочные задания.

Форма контроля. Анализ практических заданий

Тема 2.2. Отработка навыков решения простейших задач.

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

Форма контроля. Анализ практических заданий

Тема 2.4. Отработка навыков решения простейших задач.

Практика (2 ч.)

Форма подведения итогов: практические задания.

Тема 2.5. Решение несложных олимпиадных задач. Практика (4 ч.)

Форма подведения итогов: практические задания.

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

Практика (1 ч.) Тренировочные задания.

Форма контроля. Анализ практических заданий

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

Практика (1 ч.) Тренировочные задания.

Форма контроля. Анализ практических заданий

Тема 2.8. Отработка навыков решения задач. Практика (4 ч.)

Форма подведения итогов: практические задания.

Тема 2.9. Рекурсия. Использование рекурсии.

Примеры решения задач. Теория (1 ч.).

Практика (1 ч.) Тренировочные задания.

Форма контроля. Анализ практических заданий

Тема 2.10. Отработка навыков решения задач. Практика (4 ч.)

Форма подведения итогов: практические задания.

Тема 2.11. Кортежи. Функция RANGE, цикл FOR.

Примеры решения задач. Теория (1 ч.).

Практика (1 ч.) Тренировочные задания.

Форма контроля. Анализ практических заданий

Тема 2.12. Отработка навыков решения задач. Практика (4 ч.)

Форма подведения итогов: практические задания.

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

Практика (1 ч.) Тренировочные задания.

Форма контроля. Анализ практических заданий

Тема 2.14. Отработка навыков решения задач. Практика (2 ч.)

Форма подведения итогов: практические задания.

Тема 2.15. Решение олимпиадных задач. Практика (4 ч.)

Форма подведения итогов: практические задания.

## 2. КОМПЛЕКС ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИХ УСЛОВИЙ

### 2.1. Календарно-учебный график

**Примерный календарный учебный график на 2023/2024 учебный год**

**Период обучения** — сентябрь -май.

**Количество учебных недель** — 35.

**Количество часов** — 70.

**Режим проведения занятий:** пятница с 14.15 до 15.00, с 15.15 до 16.00

№ п/п	дата по плану	дата по факту	Кол-во часов	Тема занятия	Форма контроля	форма занятия	примечание
<b>1 МОДУЛЬ</b>							
1.	01.09		2	История языков программирования. Введение в язык программирования Python. Среда программирования Python. Установка программы.	Тестирование	теория	
2.	08.09		2	Типы данных и функции вывода. Определение переменной. Переменные и арифметические выражения.	Тестирование	теория	
3.	15.09		2	Чтение данных. Операции над строками. Примеры решения задач.	Тестирование	комплексное	
4.	22.09		2	Отработка навыков решения простейших задач.	Практические задания	комплексное	
5	29.09		2	Отработка навыков решения простейших задач.	Практические задания	комплексное	
6.	06.10		2	Логический тип данных и операции. Примеры использования логических выражений.	Тестирование	комплексное	
7	20.10		2	Условный оператор. Вложенный условный оператор. Примеры решения задач.	Тестирование	комплексное	

8.	27.10		2	Отработка навыков решения простейших задач.	Практические задания	комплексное	
9	03.11		2	Цикл WHILE. Примеры решения задач.	Тестирование	комплексное	
10.	10.11		2	Подсчет суммы и оператор CONTINUE. Примеры решения задач.	Тестирование	комплексное	
11.	17.11 01.12		4	Отработка навыков решения простейших задач.	Практические задания	комплексное	
12.	08.12		2	Вещественные числа. Основы работы с вещественными числами. Округление вещественных чисел. Примеры решения задач.	Тестирование	комплексное	
13.	15.12		2	Отработка навыков решения простейших задач.	Практические задания	комплексное	
14	22.12		2	Итоговое занятие по модулю	зачет	итоговое	
<b>2 МОДУЛЬ</b>							
15.	29.12		2	Срезы строк. Использование срезов. Метод FIND. Примеры решения задач.	Тестирование	комплексное	
16.	12.01		2	Отработка навыков решения простейших задач.	Практические задания	комплексное	
17	19.01		2	Отработка навыков решения простейших задач.	Практические задания	комплексное	
18.	26.01		2	Методы RFIN, REPLACE и COUNT. Примеры решения задач.	Тестирование	комплексное	
19	02.02		2	Отработка навыков решения простейших задач.	Практические задания	комплексное	
20	09.02		2	Решение несложных олимпиадных задач.	Практические задания	комплексное	
21	16.02		2	Решение несложных олимпиадных задач.	Практические задания	комплексное	
22	02.03		2	Функции. Использование функций. Примеры решения задач.	Тестирование	комплексное	
23.	09.03		2	Возврат значений. Локальные и глобальные переменные. Примеры	Тестирование	комплексное	

				решения задач.			
24	16.03		2	Отработка навыков решения задач.	Практические задания	комплексное	
25	23.03		2	Отработка навыков решения задач.	Практические задания	комплексное	
26.	30.03		2	Рекурсия. Использование рекурсии. Примеры решения задач.	Тестирование	комплексное	
27	06.04		2	Отработка навыков решения задач.	Практические задания	комплексное	
28.	20.04		2	Кортежи. Функция RANGE, цикл FOR. Примеры решения задач.	Тестирование	комплексное	
29.	27.04		2	Отработка навыков решения задач.	Практические задания	комплексное	
30	04.05		2	Отработка навыков решения задач.	Практические задания	комплексное	
31	11.05		2	Списки. Метод SPLIT и JOIN. Примеры решения задач.	Тестирование	комплексное	
32	18.05		2	Отработка навыков решения задач.	Практические задания	комплексное	
33	22.05		2	Отработка навыков решения задач.	Практические задания	комплексное	
34	25.05		2	Решение олимпиадных задач.	Практические задания	комплексное	
35	31.05		2	Итоговый мониторинг	итоговое	зачет	
		<b>итого</b>	<b>70</b> <b>ч.</b>				

## 2.2 Условия реализации программы

### ***Материально-техническое обеспечение:***

- столы для компьютера;
- компьютерные стулья;
- шкафы для дидактических материалов, пособий;
- специальная и научно-популярная литература для педагога и учащихся;
- канцтовары;

### ***Информационное обеспечение:***

- персональный компьютер (на каждого участника);
- мультимедийный проектор;
- видеоматериалы разной тематики по программе;
- оргтехника;
- выход в сеть Internet;

### ***Аппаратное обеспечение:***

- Процессор не ниже Core2 Duo;
- Объем оперативной памяти не ниже 4 ГбDDR3;
- Дисковое пространство на менее 128 Гб;
- Монитор диагональю на мене 19”;

### ***Программное обеспечение:***

- Операционная система Windows 10 Профессиональная или выше;
- Интерпретатор Python версии 3.7 и выше;
- IDE JupyterLab PyCharm;
- Foxit Reader или другой просмотрщик PDF файлов;
- WinRAR;
- Пакет офисных программ;
- Adobe Photoshop или другой растровый графический редактор;
- Любой браузер для интернет серфинга.

## 2.3 Формы аттестации

**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.

### **Оценочные материалы**

**Входящий 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.

## **2.4 Методические материалы**

Программа разработана на основе концептуальных положений Стратегии инновационного развития Российской Федерации на период до 2030 года.

### **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.

### **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 создаётся при помощи игровых приёмов и ситуаций, выступающих как средство побуждения к деятельности. Способствует развитию творческих способностей, продуктивному сотрудничеству с другими учащимися. Приучает к коллективным действиям, при-

нятию 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

## Список литературы и методического материала

### *Литература 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.

### *Литература 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.
5. Сайт проекта Open Book Project. Практические примеры на Python Криса Мейерса [Электронный ресурс] – Режим доступа: [openbookproject.net](http://openbookproject.net) , свободный.

**Входящий контроль**

Низкий уровень: 0-70% выполненных заданий;

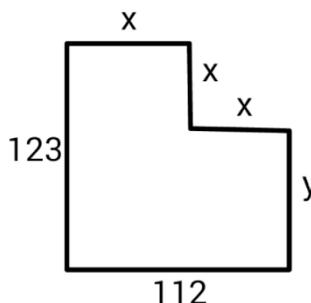
Средний уровень: 70-85% правильно выполненных заданий;

Высокий уровень: 85-100% правильно выполненных заданий.

Для учащихся – 14-17 лет

<b>Фамилия, Имя</b>	
---------------------	--

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.

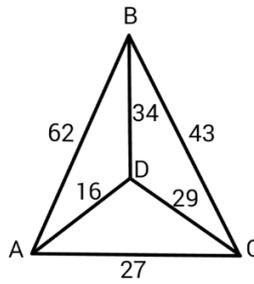
Сколько кусков рулета получится, если разрезать по кольцам всех трёх цветов?

**Примечания**

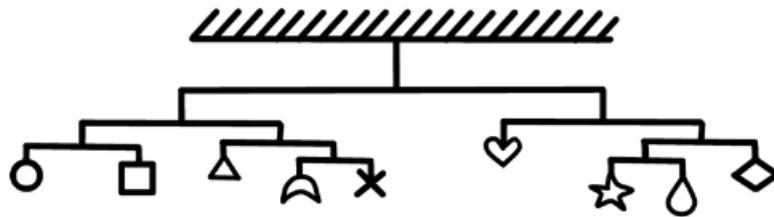
Учтите, что наклеить кольцо одного цвета на кольцо другого нельзя.

4. На рисунке показано расположение городов А, В, С и D и 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

O/Samnarco

Ф.Wagonerrte

Ф.Wagonertre

A.S.Schmetterling

A.S.Schnetterling

N.V.Murfreesboroque

N.V.Munfreesboroque

P.S.Splendoursec

P.S.Sqlendoursec

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?

8. 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.**

<input type="checkbox"/>	Да
<input type="checkbox"/>	Нет

9. 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?

### Для учащихся 11-13 лет

<b>Фамилия, Имя.</b>	
----------------------	--

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?

**Итоговый контроль**  
**Диагностическая карта**

№ /п	Имя	Входящая Диагностика	Промежуточная диагностика	Итоговая диагностика
1				
2				
3+				
	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 с 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 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 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 выпол-

нения поставленной творческой задачи;

- создать проектировочную команду и организовать ее деятельность;
- разработать творческий проект по предложенной или самостоятельно выбранной теме в составе творческой группы;
- самостоятельно подготовить презентацию и защитить проект.

<p style="text-align: center;"><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 Python 3 и вставьте в поле ответа результат вычисления: <math>11111 \cdot 1111111</math> — произведение чисел <b>11111 (5 единиц)</b> и <b>1111111 (7 единиц)</b></p>
<p>Запишите число <b>1.2345e3</b> в виде десятичной дроби.</p>
<p>Составьте и запишите выражение для вычисления: <b>2014.0<sup>14</sup>(Возвестив 14 степень)</b> Обратите внимание на запись числа: это вещественное число.</p>
<p><b>Приведите к целому типу число 2.99</b></p>
<p>Расставьте скобки в выражении <b>a and b or not a and not b</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 <i>a</i> и <i>b</i> Учитывайте регистр символов при ответе. <b>a = True</b> <b>b = False</b> <b>a and b or not a and not b</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>i</i> после выполнения фрагмента программы? <b>i = 0</b> <b>while i &lt;= 10:</b>     <b>i = i + 1</b>     <b>if i &gt; 7:</b>         <b>i = i + 2</b></p>

Сколько итераций цикла будет выполнено в этом фрагменте программы?

```
i = 0  
while i <= 10:  
    i = i + 1  
    if i > 7:  
        i = i + 2
```

Сколько всего знаков \* будет выведено после исполнения фрагмента программы:

```
i = 0  
while i < 5:  
print('*')  
    if i % 2 == 0:  
        print('**')  
    if i > 2:  
        print('***')  
    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:  
        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

a

b

c

*где a, b и c — длины сторон треугольника*

прямоугольник

a

b

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