

**Администрация города Искитима Новосибирской области
Муниципальное бюджетное образовательное учреждение
дополнительного образования «Центр дополнительного образования»
города Искитима Новосибирской области**

Рассмотрена и одобрена
на заседании педагогического совета
(протокол № 1 от «09» сентября 2026 года)



Утверждаю
Директор МБОУ ДО ЦДО
М.И. Курдюмова
(приказ № 43 - т от 11.09.26)



**Дополнительная общеобразовательная
общеразвивающая программа
технической направленности
«Программирование на Python». Базовый уровень
объединение «Программист»**

Срок реализации: 2 года

Возраст учащихся: 11-17 лет

Уровень: базовый

**Автор-составитель:
педагог дополнительного образования
Яголович Валентина Игнатьевна**

**г. Искитим
2026**

Оглавление

1.Комплекс основных характеристик программы	Ошибка! Закладка не определена.
1.1. Пояснительная записка	Ошибка! Закладка не определена.
1.1.1 Направленность программы	Ошибка! Закладка не определена.
1.1.2 Актуальность программы, целесообразность, новизна	Ошибка! Закладка не определена.
1.1.3 Отличительной особенностью программы «Программирование на Python»..	Ошибка! Закладка не определена.
Целевая аудитория (адресат программы)	Ошибка! Закладка не определена.
1.1.5 Объем, срок освоения программы.....	Ошибка! Закладка не определена.
1.1.6 формы обучения	Ошибка! Закладка не определена.
1.1.7 Уровень программы.....	Ошибка! Закладка не определена.
1.1.8 Организационные формы обучения	Ошибка! Закладка не определена.
1.1.9 Режим занятий	Ошибка! Закладка не определена.
1.2 Цель и задачи программы.....	Ошибка! Закладка не определена.
1.3 Содержание программы	Ошибка! Закладка не определена.
1.4.Содержание учебного плана	Ошибка! Закладка не определена.
1.4.Планируемые результаты	Ошибка! Закладка не определена.
2.Комплекс организационно-педагогических условий.....	Ошибка! Закладка не определена.
2.1.Календарный учебный график.....	Ошибка! Закладка не определена.
2.2.Условия реализации программы	Ошибка! Закладка не определена.
2.3 Формы аттестации	Ошибка! Закладка не определена.
2.4 Оценочные материалы.....	Ошибка! Закладка не определена.
2.5.Методические материалы.....	Ошибка! Закладка не определена.
2.7.Рабочая программа воспитания.....	Ошибка! Закладка не определена.
3.Список литературы	Ошибка! Закладка не определена.
3.1. Список литературы для педагогических работников	Ошибка! Закладка не определена.
3.2. Список литературы для обучающихся.....	Ошибка! Закладка не определена.
4.Приложения.....	Ошибка! Закладка не определена.
4.1 Календарно -тематическое планирование образовательной программы	Ошибка! Закладка не определена.
4.2 Оценочные материалы промежуточной аттестации 1год обучения	Ошибка! Закладка не определена.
4.3 Оценочные материалы промежуточной аттестации 2 год обучения	Ошибка! Закладка не определена.
4.4 Оценочные материалы аттестация по итогам освоения ДОП	Ошибка! Закладка не определена.
4.5 Итоговый уровень освоения общеобразовательной общеразвивающей программы	
технической направленности «Программирование на Python»	Ошибка! Закладка не определена.

4.6 Динамика уровня сформированности познавательного интереса. Ошибка! Закладка не определена.

4.7 «Лист наблюдения» по методике Щукиной Г.И. Ошибка! Закладка не определена.

1.Комплекс основных характеристик программы

1.1. Пояснительная записка

Современное общество переживает активную стадию цифровой трансформации. Все больше сфер жизни людей становятся зависимыми от информационных технологий и электроники. Одной из составляющих информационной компетентности является владение языком программирования, вследствие чего встаёт вопрос о выборе языка программирования, который удобен в освоении подростками.

Язык программирования Python является современным, востребованным и перспективным языком разработки, позволяющим не только заложить фундамент знаний будущих программистов и инженеров, но получить востребованные и актуальные навыки, необходимые в учебной, проектной и повседневной деятельности. Программы, и синтаксис языка не перегружен и интуитивно понятен, и это делает Python максимально эффективным инструментом при обучении детей программированию.

Программа разработана в соответствии со следующими нормативно-правовыми актами и государственными программными документами:

- Федеральный закон от 29.12.2012 № 273-ФЗ «Об образовании в Российской Федерации» (в редакции, вступающей в силу с 01.01.2026);
- Федеральный закон от 24.07.1998 № 124-ФЗ «Об основных гарантиях прав ребёнка в РФ»;
- Распоряжение Правительства РФ от 31.03.2022 №678-р «Об утверждении Концепции развития дополнительного образования детей до 2030 года» (с изменениями на 1 июля 2025);
- Указ Президента РФ от 07.05.2024 г. «О национальных целях развития РФ на период до 2030 года и перспективу до 2036 года»;
- «Методические рекомендации по разработке и реализации дополнительных общеобразовательных общеразвивающих программ (включая разноуровневые и модульные программы) разработанные ГАУ ДО НСО «ОЦРТДиЮ», РМЦ 2022 г.;
- СанПиНы СП 2.4.2.4283-26 «Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи»

Направленность программы

Дополнительная общеобразовательная программа «Программирование на Python» имеет техническую направленность и является программой базового уровня, позволяя обучающимся приобщиться к инженерно-техническим знаниям в области программирования и сформировать техническое мышление.

Актуальность программы, целесообразность, новизна

Актуальность

Актуальность программы обусловлена несколькими ключевыми факторами.

Во-первых, существует высокий социальный спрос на раннее изучение языков программирования, в частности Python. Этот язык является неотъемлемой частью школьного курса информатики, необходим для успешной сдачи ЕГЭ и решения олимпиадных задач, а также имеет широкий спектр прикладного применения в повседневной жизни.

Во-вторых, владение компьютерными технологиями сегодня рассматривается как фундаментальный компонент современного образования. Изучение программирования

способствует формированию системно-информационной картины мира, развитию алгоритмического мышления и учебных навыков.

В-третьих, программа создаёт условия для профессиональной ориентации школьников. Она позволяет обучающимся познакомиться с профессией программиста и смежными IT-специальностями, оценить свои способности и сформировать профессионально важные качества ещё на этапе школьного обучения.

Целесообразность использования Python

Выбор языка Python для изучения в среднем звене является целесообразным и методически обоснованным благодаря его характеристикам.

- Простота и низкий порог вхождения -минималистичный синтаксис делает язык лёгким для освоения.
- Кроссплатформенность - работает на всех распространённых операционных системах.
- Открытость - свободное программное обеспечение с открытым исходным кодом.
- Богатая экосистема - обширная стандартная библиотека, позволяющая реализовывать проекты различной сложности.

Новизна программы

Новизна программы заключается в смещении акцента с простого изучения синтаксиса языка на формирование системного инженерного мышления у обучающихся. Это позволяет им не только овладевать широкой областью теоретических знаний, но и эффективно применять их на практике.

Такой подход готовит их к реальным условиям работы в сфере IT, где умение мыслить системно и находить нестандартные решения ценится выше знания конкретных команд.

Отличительные особенности программы «Программирование на Python»

В отличие от ряда других программ дополнительного образования в сфере IT, например, курсов по программированию (автор А. С. Уваров) или «Основы алгоритмизации и программирования» (авторы И. И. Иванов, П. П. Петров), программа «Программирование на Python» выделяется ярко выраженным практико-ориентированным подходом.

Ключевое отличие заключается в структуре учебного процесса: каждая тема завершается большим набором различных упражнений. Это позволяет не просто закрепить теоретический материал, но и научиться применять его для решения реальных задач. Такой подход обеспечивает:

- глубокое погружение в предмет и расширение знаний за рамками школьной программы;
- целенаправленную подготовку к успешному участию в олимпиадах и конкурсах по информатике;
- формирование портфолио проектов, что является весомым преимуществом при поступлении в профильные учебные заведения.

Таким образом, программа «Программирование на Python» предлагает более прикладной и ориентированный на результат формат обучения по сравнению с аналогичными курсами.

Целевая аудитория

Программа ориентирована на учащихся 11–17 лет, проявляющих устойчивый интерес к сфере информационных технологий и программированию. Выбор этой возрастной группы обусловлен психологическими особенностями подростков.

В возрасте 11–13 лет происходит активное развитие абстрактного и логического мышления. В этом возрасте активно развиваются навыки коммуникации. Парное программирование или работа над общим проектом в мини-группах (2-3 человека). Дети начинают лучше понимать сложные концепции, такие как переменные, циклы

Внимание становится более устойчивым, чем у младших школьников. Они уже способны выстраивать логические цепочки и понимать причинно-следственные связи. Это идеальная база для изучения алгоритмов и структур данных

Учащиеся 14–17 лет - подростки уже обладают более развитым абстрактным мышлением. Обучение программированию для них — это не просто развлечение, а инструмент для достижения конкретных результатов. В этом возрасте у учащихся:

- формируется способность к *алгоритмическому и логическому мышлению*, необходимому для решения сложных технических задач;
- наблюдается стремление к *самостоятельности и самовыражению* через создание собственных цифровых продуктов, что является ключевым мотиватором в программировании;
- появляется потребность в познании *собственных возможностей* через преодоление вызовов, что соответствует процессу отладки кода и решению нестандартных задач.

Кроме того, в 14–17 лет подростки активно задумываются о будущей профессии. Программа помогает им сделать осознанный выбор в пользу технического или IT-направления профессионального становления.

Группы комплектуются на основании заявлений законных представителей или самих учащихся.

Объем, срок освоения программы

Объем 296 часов по 148 часов в год. Срок освоения программы 2 года.

Для каждой группы предусмотрено резервное учебное время, которое может быть использовано участниками образовательного процесса в целях формирования вариативной составляющей содержания программы. В резервные часы входят часы на повторение, презентации, конкурсы. Общий объем резервного учебного времени составляет 4 часа из общего количества часов.

Формы обучения

Очная. Обучение происходит на русском языке.

Уровень программы- базовый

Особенности организации образовательного процесса

Используется традиционная модель реализации программы в виде линейной последовательности «от простого к сложному» в течение всего периода. Обучение проводится в разновозрастных группах.

На первый год обучения зачисляются обучающиеся в возрасте от 11 до 13 лет. Данный возрастной этап характеризуется активным формированием познавательных интересов и необходимостью адаптации к новой образовательной среде, что учитывается при разработке методик преподавания.

На второй год переводятся учащиеся успешно освоившие первый год обучения. Также на второй год обучения могут зачисляться обучающиеся в возрасте от 14 до 17 лет, успешно прошедшие входное тестирование (письменно)

Организационные формы обучения

Занятия проводятся по группам, группы формируются из обучающихся разного возраста. Количественный состав группы 9 человек.

В процессе реализации программы используются следующие формы организации учебных занятий:

- лекции;
- выполнение домашних заданий;
- защита проектов;
- практические занятия;

- проектная деятельность;
- консультации.

Режим занятий

Занятия проводятся 2 раза в неделю по 2 академических часа. Общее количество часов в неделю–4 часа (2 занятия по 45 минут с 10-минутным перерывом). Режим организации занятий по данной дополнительной общеобразовательной программе определяется календарным учебным графиком.

1.2 Цель и задачи программы

Цель программы: формирование у обучающихся устойчивых навыков программирования на языке Python, развитие алгоритмического мышления, готовности к самостоятельной проектной деятельности.

Задачи:

1 год обучения

Предметные:

- обучить учащихся основам программирования в среде Python;
- сформировать базовые представления о теории алгоритмов;
- научить работать в системе программирования Python;
- развить логическое мышление через практические задания.

Метапредметные:

- учить применять нестандартные подходы при решении задач;
- развивать навыки самоанализа своей деятельности;
- формировать умение работать в команде, обсуждать идеи и совместно искать решения;
- обучать навыкам самообразования и самостоятельного поиска информации.

Личностные:

- формировать трудолюбие, усидчивость, ответственность и умение доводить начатое до конца;
- прививать учащимся этические и нравственные нормы поведения;
- воспитывать культуру общения, уважение к чужому мнению;
- развивать уверенность в своих силах и положительное отношение к результатам своего труда;
- поддерживать устойчивый интерес к программированию и информационным технологиям

2 год обучения

Предметные:

- Освоить составление базовых алгоритмов, включая алгоритмы поиска и сортировки.
- Научить использовать стандартные приёмы написания программ на Python для решения практических задач.
- Сформировать инженерное мышление: алгоритмическое и логическое мышление, модульность задачи, поиск и устранение ошибок (отладка), работа в команде;
- Развить навыки самостоятельной работы над проектами и задачами повышенной сложности.

Метапредметные:

- учить применять комплексные методы анализа для решения нестандартных задач повышенной сложности;
- формировать умение планировать и организовывать самостоятельную работу над проектом, распределяя задачи в команде;

- обучать навыкам проектной деятельности: от постановки задачи до презентации готового продукта

Личностные:

- формировать высокий уровень ответственности за результат индивидуальной и командной работы;
- прививать навыки самоорганизации и умение доводить до конца сложные проекты;
- формировать умение критически оценивать свои силы и результаты труда, стремясь к профессиональному росту в области ИТ.

1.3 Содержание программы

Учебный план

Первый год обучения 4 часа в неделю					
№п/п	Название раздела, темы	Количество часов			Формы аттестации/контроля
		Всего	теория	практика	
1	Техника безопасности в компьютерном классе	2	1	1	Опрос, викторина на знание правил техники безопасности
2	Алгоритмы	46	11	29	Текущий контроль осуществляется регулярно во время занятий с помощью педагогического наблюдения, опросов, решения задач, кейсов, разбора ситуаций, практических работ
3	Знакомство с языком Python	32	10	22	
4	Переменные	36	8	28	
5	Вычисления	32	12	20	
6	Резерв	2		2	
	Итого	148	41	107	
Второй год обучения 4 часа в неделю					
1	Введение. Техника безопасности в компьютерном классе	2	1	1	Опрос, тестирование
2	Основные алгоритмические конструкции	48	16	32	Тестирование. Устный опрос. Текущий контроль осуществляется регулярно во время занятий с помощью педагогического наблюдения, опросов, решения задач, кейсов, разбора ситуаций,

					практических работ.
3	Строки	30	9	21	Тестирование. Устный опрос. Текущий контроль осуществляется регулярно во время занятий с помощью педагогического наблюдения, опросов, решения задач, кейсов, разбора ситуаций, практических работ.
4	Списки	34	10	24	
4	Случайные числа	26	13	13	
5	Резерв. Повторение и углубление тем	2	1	1	тестирование
6	Итоговый проект	6	-	8	Защита проекта
	Итого	148	50	98	

Второй год обучения два часа в неделю					
1	Введение. Техника безопасности в компьютерном классе	2	1	1	Опрос, тестирование
2	Основные алгоритмические конструкции	18	6	12	Тестирование. Устный опрос. Текущий контроль осуществляется регулярно во время занятий с помощью педагогического наблюдения, опросов, решения задач, кейсов, разбора ситуаций, практических работ.
3	Строки	16	6	10	Тестирование. Устный опрос. Текущий контроль осуществляется регулярно во время занятий с помощью педагогического наблюдения, опросов, решения задач, кейсов, разбора ситуаций, практических работ.
4	Списки	14	6	8	
4	Случайные числа	16	6	10	
5	Резерв. Повторение и углубление тем	2	1	1	тестирование
6	Итоговый проект	6	-	6	Защита проекта
	Итого	74	27	47	

1.4.Содержание учебного плана Первый год обучения

Введение. Техника безопасности и правила работы на компьютере - знакомство с правилами поведения в кабинете информатики. Основные устройства компьютера, их функции и взаимосвязь(2часа)

Раздел 1.Алгоритмы(46часов)

1.1 Понятие алгоритма, свойства алгоритмов, виды алгоритмов.

Теория:

Знакомство с историей происхождения термина «алгоритм». Понятие основных свойства алгоритмов: определенность, результативность, массовость, формальность, дискретность и виды алгоритмов.

1.2 Способы описания алгоритмов, исполнители алгоритмов.

Теория:

Знакомство со способами описания алгоритмов: словесным, графическим в виде блок-схем, программным, понятие формального исполнителя.

Практика:

Создание и описание линейных алгоритмов

1.3 Виды и анализ линейных алгоритмов и алгоритмов с ветвлениями.

Теория:

Освоение и реализация линейных алгоритмов и алгоритмов с ветвлением, использование при анализе алгоритмов следующих аспектов :**последовательность выполнения, завершение, сложность выполнения алгоритмов.**

Практика:

Описание алгоритмов с ветвлением. Блок-схемы;

1.4 Простые и составные условия, вложенные ветвления.

Теория:

Знакомство и реализация простого ветвления и множественного ветвления, графическое представление алгоритма с ветвлением, понятие вложенного ветвления, реализация алгоритмов в виде блок-схем.

Практика:

Алгоритмы с вложенными ветвлениями, выполнение проверочных тестовых заданий.

2.3 Знакомство с языком Python(32 часа)

Теория:

Знакомство с историей развития языков программирования, способами трансляции программного кода, знакомство с Python и его особенностями, простейшая программа

Практика:

Составление простейших программ

2.1 Установка программы языка Python

Теория:

Практические рекомендации по установке.

Практика:

Установка программы python, запуск

2.2 Среда разработки IDLE

Теория:

Понятие стандартной среды разработки Python, правила работы в режимах: (ShellMode) и (EditMode), запуск режимов.

Практика:

запуск режимов, знакомство со стандартной средой разработки

2.3 Структура программы. Комментарии. Этапы решения задач на ПК

Теория:

Объясняется понятие модуля, инструкции, выражения, знакомство с вводом каждой инструкции(отступы), правила установки и обозначение комментариев. Разбираются этапы решения задач: постановка задачи, формализация задачи, создание алгоритма, программирование, тестирование программы, ошибки программы, отладка.

Практика:

Составление простейших программ, тестирование, ошибки в программах, создание комментариев.

2.4 Основы синтаксиса Python и арифметические выражения.

Теория:

Понятия операторов и операндов, операторы для математических вычислений, для сравнений и логических конструкций.

Практика:

Создание простейших программ с использованием арифметических выражений;
работа с числами в Python

2.5 Ввод вывод данных.

Теория:

Применение операторов Print, Input.

Практика:

Ввод и вывод данных, решение задач по карточкам.

3.Переменные и выражения(36 часов)

3.1Понятие переменной. Правила именования переменных. Допустимые и недопустимые символы в обозначении переменных

Теория:

Объяснение особенностей именования переменных: имя может содержать буквы (a-z, A-Z), цифры (0-9) и символ подчёркивания (_). Имя должно начинаться с буквы или символа подчёркивания. Нельзя использовать пробелы.

Практика:

Создание программ с использованием переменных различных типов.

3.2 Объявление переменной

Теория:

Разбираются правила объявления переменной, базовый синтаксис (имя_переменной = значение).

Практика:

Операции присваивания, создание программ с использованием переменных различных типов.

3.3 Инициализация значений (операция присваивания) множественное присваивание значений.

Теория:

Разбирается понятие динамической типизации, множественного присваивания.

Практика:

Множественное присваивание значений, составление программ.

3.4 Основные типы данных переменных в Python

Теория:

Знакомство с числовыми, строковыми, логическими типами данных .

Практика:

Выполнение практических заданий со строковыми и логическими типами данных, выполнение проверочных тестовых заданий.

4.Вычисления(32часа)

4.1 Понятие целых и вещественных чисел, преобразователи при вводе данных

Теория:

Объяснение понятия –целое и вещественное число, преобразователи int и float при вводе чисел.

Практика:

Практические работы с использованием простейших арифметических действий с целыми и вещественными числами.

4.2 Понятие целочисленного деления: нацело и с остатком.

Теория:

Разбираются понятие и практика использования операций целочисленного деления(//, %)

Практика:

Практические работы с использованием математических операторов `://`, `%` и `**`.

Комбинированная работа по карточкам.

4.3 Стандартные функции модуля `math`.

Теория:

Импортирование и использование модуля `math`, использование математических, тригонометрических и логарифмических функций

Практика:

Практические работы с использованием функций модуля `math`.

Итоговая контрольная работа

5.Резерв(2часа)

Практика:

Практическая отработка наиболее сложных тем

Второй год обучения

Требования ТБ поведения в компьютерном классе - правила поведения в кабинете информатики(2 часа)

Практика:

Работа по карточкам.

Раздел 1.Основные алгоритмические конструкции(48 часов)

1.1 Повторение:виды и анализ алгоритмов: линейных, условных и циклических, и их блок-схемы

Теория:

Реализация линейных алгоритмов и алгоритмов с ветвлением, использование при анализе алгоритмов следующих аспектов: **последовательность выполнения, завершение, сложность выполнения алгоритмов.**

Практика:

Решение задач, составление блок-схем всех видов алгоритмов.

1.2 Условные алгоритмы в Python:

1.2.1 Ветвление в Python. Оператор `if-else`.

Теория:

Разбирается базовый синтаксис оператора `if`, ошибки, возникающие при использовании оператора `if`.

Практика:

Практические задания с использованием оператора `if-else`.

1.2.2 Множественное ветвление. Оператор `if-elif-else`.

Теория:

Разбирается понятие множественного ветвления,(конструкция, которая позволяет организовать более двух веток выполнения программы в зависимости от условий.) синтаксис.

Практика:

Практические задание с использованием оператора `if-elif-else`, разработка игровых программ.

1.3 Циклы

1.3.1 Циклические алгоритмы `FOR`

Теория:

Разбирается понятие цикла `FOR`, синтаксис, встроенная функция `range`, синтаксис функции `range (start, stop, step)`.

Практика:

Решение задач с использованием операторов if-elif-else, for, функцией range.

1.3.2 Цикл с условием WHILE. Вложенные циклы.

Теория:

Разбирается понятие цикла WHILE, структура цикла, понятие вложенного цикла, условие организации внешнего и внутреннего циклов.

Практика:

Решение различных задач с использованием оператора while.

1.3.3 Операторы управления циклом: break, continue, else.

Теория:

Разбираем особенности работы с вложенными циклами: **Оператор break** (прерывает только тот цикл, в котором он вызван), **continue** (пропускает текущую итерацию и переходит к следующей), **оператор else** в циклах выполняется, если цикл завершился естественно (без использования break).

Практика:

Решение задач с использованием логики true, false, флагов;

решение задач с использованием операторов управления циклом: break, continue, else; программирование простейших игр типа «угадай код»

1.4 Логические выражения и операторы в Python.

1.4.1 Логические операторы в Python: and, or и not. Операции сравнения.

Теория:

Назначение операторов and, or и not. **Приоритет логических операторов**, операции сравнения в Python, ошибки при использовании операций сравнения.

Практика:

Выполнение практических заданий, выполнение проверочных тестовых заданий.

1.4.2 Простые и сложные логические выражения, результат вычисления логического выражения.

Теория:

Понятие логического выражения, составление простых и сложных (путём комбинирования нескольких простых выражений с помощью логических операторов.) логических выражений. Результат вычисления логического выражения (**булево значение True или False**).

Практика:

Выполнение практических заданий, выполнение проверочных тестовых заданий.

1.4.3 Вычисление суммы последовательностей.

Теория:

Знакомство с использованием встроенной функции sum().

Практика:

Выполнение практических заданий с использованием функции sum(), выполнение проверочных тестовых заданий.

2. Строки (30 часов)

2.1 Что такое строка. Создание строки. Специальные символы. Основные операции со строками.

Теория:

Понятие строки, создание строк с помощью кавычек, специальные символы. Основные операции со строками в Python: **Конкатенация (сложение), повторение строки, определение длины строки, проверка вхождения, методы строк, индексация и срезы.**

Практика:

Задания на создание и описание строк, соединение строк, повторение строки.

2.2 Действия над строками: обращение к элементу по индексу, проверка на вхождение.

Теория:

Объясняется понятие индексации строк, обращение к элементам по индексу и проверка на вхождение подстроки.

Практика:

Задачи на использование обращений к элементу по индексу, проверку на вхождение.

2.3 Длина строки и отрицательные индексы.

Теория:

Объясняется понятие длины строки(функция `len()`), вызов функции `len(s)`. Отсчитывание элементов с конца последовательности с использованием отрицательных индексов.

Практика:

Задания на вхождение, повтор и использование функции `len()`.

2.4. Срезы строк. Сравнение строк.

Теория:

Объясняется понятие среза, особенности работы со срезами, часто используемые шаблоны срезов, сравнение строк с помощью операторов сравнения.

Практика:

Решение задач с использованием шаблонов срезов, задачи на сравнение строк с помощью операторов сравнения.

2.5 Оператор `in`. Модуль `string`.

Теория:

Объяснение использования оператора `in` для проверки содержания одного элемента (например, подстрока или символ) в другом. Объяснение некоторых особенностей использования оператора `in`. Импорт модуля `string`.

Практика:

Задачи с использованием оператора `in`, импорт модуля `string`.

2.6 Операторы для всех типов последовательностей (строки, списки, кортежи).

Теория:

Операторы для всех типов последовательностей (строки, списки, кортежи).

Практика:

Выполнение практических заданий по карточкам для разных типов последовательностей.

2.7 Поиск и замена подстроки или символа в строке.

Теория:

Объяснение и отработка применения метода `find()`. Синтаксис: `string.find(substring, start, end)`.

Практика:

Примеры решения задач со строками.

Практика:

Выполнение практических заданий на :

поиск и замену символа в строке, находить подстроку в строке;

находить количество слов в строке.

3.Списки (34 часа)

3.1 Создание списков.

Теория:

Объясняется понятие и создание списков.

Практика:

Задачи на создание списков.

3.2 Операции со списками.

Теория:

Объяснение операций: **индексации, изменения элементов, объединения списков, разложения списков, перебор элементов списка** с помощью циклов for и while. Сравнение списков.

Практика:

Задачи на действия со списками: объединение, разложение, перебор списков.

3.3 Встроенные функции для списков `len()`, `sorted()`, `min()` и `max()`.

Теория:

Использование встроенных функций:определяющих длину списка, сортировку списка, минимальный и максимальный элементы списка.

Практика:

Задания на сортировку списков, сумму элементов списка, определение *min()* и *max()*.

3.4 Методы списков в Python.

Теория:

Методы списков в Python: `append()`, `extend()`, `insert()`, `remove()`,`clear()`,`index()`,`pop()`,`count()`,`sort()`, `reverse()`,`copy()`.

Практика:

Выполнение практических заданий на: перемешивание элементов списка и выбор случайного элемента, работа по карточкам.

3.5 Поиск элементов в списке, сортировка списка, преобразование списка в строку.

Теория:

Объяснение использования оператора `in`, методов `index()`, `sort()`, `join()`.

Практика:

Задачи на преобразование списка в строку, использование оператора `in` в списках, поиск элемента в списке, добавление и удаление элементов в списке.

3.6 Кортежи, понятие кортежей, создание кортежей.

Теория:

Объясняется понятие кортеж, индексация элементов кортежа, создание кортежей с помощью круглых скобок, без скобок, с помощью функции `tuple()`.

Практика:

Задания с использованием кортежей.

3.7 Перебор элементов кортежей.

Теория:

Объяснение перебора элементов кортежа с помощью цикл for или while.

Практика:

Задания с использованием кортежей с помощью циклов for или while.

4.Случайные числа.(26часов)

4.1 Для чего нужен модуль `random` в Python. Как работает генерация случайных чисел в Python.

Теория:

Разбираем понятия: **модуль `random` в Python**, использование модуля `random` для генерации случайных чисел, для перемешивания объектов, для работы с последовательностями.

Практика:

Решение задач с использованием модуля `random`.

4.2 Загрузка модуля `Random`, случайные целые числа (`int`):функция `randint (a, b)`, функция `randrange (start, stop[, step])`.

Теория:

Объясняется необходимость импортировать модуль (import random) , разбирается понятие случайного целого числа, объясняется **синтаксис и работа** функции random.randint(a, b) и функции randrange(start, stop[, step]).

Практика:

Решение задач с использованием функций randint(a, b), randrange (start, stop,[step]).

4.3 Случайные вещественные числа (float): функция random(), функция uniform(a, b).

Теория:

Разбирается понятие случайного вещественного числа, объясняется **синтаксис и работа** функции random() и функции uniform(a, b).

Практика:

Решение задач с использованием случайных вещественных чисел (float): random(), uniform(a, b).

4.4 Функции для работы с последовательностями: choice(seq), choices(seq), [shuffle(x[, random]).

Теория:

Объясняется работа функций с последовательностями: choice(seq), choices(seq), shuffle(x[, random]).

Практика:

Составление программ для работы с последовательностями: choice(seq), choices(seq), shuffle(x[, random]), выполнение проверочных тестовых заданий.

5. Резерв (2 часа)

Практика:

Повторение и углубление тем.

6. Итоговый проект (6 часов)

Теория:

Концепция проекта, понятия, постановка задачи, подготовка презентации проекта.

Практика:

Разработка программы проекта , работа по его созданию с применением всех изученных тем: условия, циклы, функции, работа с данными пользователя. Учащийся создаёт программу, которая взаимодействует с пользователем через текст. Программа предлагает выбор действий, и в зависимости от выбора пользователя сюжет развивается по-разному. Защита проекта.

1.5. Планируемые результаты

Итоговый образовательный результат:

Выпускник образовательной программы «Программирование на Python» владеет базовыми навыками составлять алгоритмы для решения задач различной сложности и реализовывать их на языке Python. Демонстрирует устойчивый интерес к программированию, трудолюбие, ответственность, самоорганизацию и готовность к созданию проектов повышенной сложности.

Предметные результаты

По окончании 1 года обучения учащиеся будут:

- знать основы синтаксиса языка программирования Python;
- иметь базовые представления о теории алгоритмов;
- уметь работать в системе программирования Python (написание, запуск и сохранение кода);
- владеть основами алгоритмического и логического мышления на уровне выполнения практических заданий.

Метапредметные результаты

По окончании 1 года обучения учащиеся будут уметь:

- применять нестандартные подходы при решении учебных задач;
- осуществлять самоанализ своей деятельности, оценивать правильность выполнения заданий и вносить коррективы;
- эффективно работать в команде: обсуждать идеи, аргументировать свою точку зрения и совместно искать решения;
- самостоятельно находить, отбирать и использовать необходимую информацию для решения учебных задач (навыки самообразования).

Личностные результаты:

По окончании 1 года обучения учащиеся будут:

- проявлять трудолюбие, усидчивость и ответственность при выполнении учебных заданий;
- соблюдать этические и нравственные нормы поведения в учебной деятельности и общении;
- демонстрировать культуру общения, уважение к чужому мнению и умение работать в команде;
- проявлять уверенность в своих силах и положительное отношение к результатам своего труда;
- иметь устойчивый интерес к программированию и информационным технологиям.

2 год обучения

Предметные результаты.

По окончании 2 года обучения учащиеся будут:

- знать и уметь составлять базовые алгоритмы, включая алгоритмы поиска и сортировки;
- владеть стандартными приёмами написания программ на Python для решения практических задач;
- применять инженерный подход к решению задач: декомпозировать задачу на модули, осуществлять отладку (поиск и устранение ошибок);
- уметь работать над проектами повышенной сложности как самостоятельно, так и в команде.

Метапредметные результаты

По окончании 2 года обучения учащиеся будут уметь:

- применять комплексные методы анализа для решения нестандартных задач повышенной сложности;
- планировать и организовывать самостоятельную работу над проектом, распределяя задачи в команде;
- владеть навыками проектной деятельности: от постановки задачи до презентации готового продукта.

Личностные результаты:

По окончании 2 года обучения учащиеся будут:

- проявлять высокий уровень ответственности за результат индивидуальной и командной работы;
- демонстрировать навыки самоорганизации и умение доводить до конца сложные проекты;
- критически оценивать свои силы и результаты труда, стремясь к профессиональному росту в области ИТ.

2.Комплекс организационно-педагогических условий

Календарно-тематическое планирование (Приложение №1)

2.1.Календарный учебный график

Год обучения	1 год обучения	2 год обучения	2 год обучения
Дата начала занятий	10 сентября текущего года	10 сентября текущего года	10 сентября текущего года
Дата окончания занятий	31 мая текущего года	31 мая текущего года	31 мая текущего года
Количество учебных недель	37	37	37
Количество учебных дней	74	74	37
Количество учебных часов	148	148	74
Продолжительность каникул	согласно графику образовательной организации	согласно графику образовательной организации	согласно графику образовательной организации
Сроки проведения аттестации	с 25.05 текущего года	с 25.05 текущего года	с 25.05 текущего года
Режим занятий	2 раза в неделю по 2 часа	2 раза в неделю по 2 часа	1 раз в неделю по 2 часа

Ссылка на положение о

КУГ: <https://cdoisk.edusite.ru/sveden/files/ec18be8536af4b6bf227b87a120a372e.pdf>

2.2.Условия реализации программы

Материально-техническое обеспечение

Компьютерный класс. Оборудование компьютерного класса:

- рабочие места по количеству обучающихся, оснащенные персональными компьютерам;
- рабочее место преподавателя, оснащенное персональным компьютером;
- комплект учебно-методической документации: программа, раздаточный материал, задания;
- цифровые компоненты учебно-методических комплексов (презентации, обучающие диски);
- наличие локальной сети и доступа к сети Интернет.

Требования к комплектации компьютерного класса

Для организации учебного процесса необходимы:

- учебный кабинет, удовлетворяющий санитарно-гигиеническим требованиям, для занятий группы 10-12 человек, компьютеры с выделенным каналом выхода в Интернет и программным обеспечением;
- мультимедийная проекционная установка, принтер, сканер, ксерокс.
- учебное помещение должно соответствовать требованиям санитарных норм и правил, установленных Санитарными правилами СанПиНы СП 2.4.2.4283-26 «Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи»
-

Информационное обеспечение

На компьютерах, которые расположены в кабинете информатики, должна быть установлена операционная система Windows или Linux, а также необходимое программное обеспечение:

- текстовый процессор (MS Word или OpenOfficeWriter);
- табличный процессор (MS Excel или OpenOfficeCalc);
- средства для работы с базами данных (MS Access или OpenOfficeBase);
- среда программирования Python 3+
- Windows 10/11, Ubuntu, Yandex Browser.

Кадровое обеспечение

Реализовывать программу могут педагоги дополнительного образования, обладающие достаточными знаниями в области педагогики, психологии и методологии, знающие особенности технологии обучения основам программирования на языке Python.

2.3 Формы аттестации

Аттестация позволяет определить достигнуты ли обучающимися планируемые результаты, освоена ли программа.

Формы аттестации:

- устный опрос;
- комбинированный опрос;
- тестирование;
- проверка самостоятельных работ;
- зачетное занятие;
- контрольная работа;
- защита проектов.

Система отслеживания результатов обучающихся выстроена следующим образом:

- текущий контроль;
- промежуточная аттестация;
- аттестация по итогам освоения ДОП.

Текущий контроль осуществляется регулярно во время занятий.

Цель: регулярная оценка усвоения материала, выявление пробелов, поддержание мотивации.

Формы:

- Краткие тесты (5–10 вопросов) по итогам темы.
- Проверка и оценка домашних заданий (написать функцию, исправить код, выполнить мини-задачу).
- Практические работы на занятии (написать фрагмент кода, решить задачу у доски или на компьютере).
- Активности на занятии (ответы на вопросы, участие в обсуждениях).

Критерии оценки:

Правильность выполнения задания (отсутствие синтаксических и логических ошибок).
Соблюдение стиля кода (<i>PEP 8</i>).
Полнота решения (все требования задания выполнены).
Своевременность сдачи.

Промежуточная аттестация

Проводится по окончании каждого года обучения

Промежуточная аттестация для первого года обучения проводится в форме компьютерного тестирования. (Приложение № 2).

Промежуточная аттестация для второго года обучения проводится в форме компьютерного тестирования. (Приложение № 3).

Аттестация по итогам освоения программы проводится в форме защиты проекта(приложение №4).

Сумма баллов результатов *промежуточной* аттестации каждого года и *аттестации по итогам освоения ДОП(защиты проекта)* переводится в один из уровней освоения программы :

Низкий: 30%-40% от общего количества баллов;

Средний: 40%-70% от общего количества баллов;

Высокий: 70%-100% от общего количества баллов.

2.4 Оценочные материалы

Результаты отслеживаются по следующим параметрам:

1.Уровнем освоения детьми содержания предмета:

- компьютерное тестирование первого года обучения (приложение №2);
- компьютерное тестирование второго года обучения (приложение №3);
- защита проекта.

2.Итоговым уровнем освоения программы (приложение №4)

Оценка успехов учащегося – основная составляющая образовательного процесса. В конечной оценке воспитанник должен точно увидеть:

- какими были его успехи в усвоении программного материала в целом;
- на каком уровне он его освоил, каковы его умения и навыки;
- какова оценка его творческой деятельности;
- в какой мере он способен проявить свое личностное отношение к изучаемой программе;
- как действовал в коллективе сверстников и взрослых;

Оценка эффективности программы

Эффективность работы программы оценивается следующими показателями:

- Знаниями учащихся, которые определяются, контрольными и зачетными занятиями (в конце полугодия).
- Успехами и достижениями учащихся - в конце каждого полугодия проводится мониторинг результатов обучения ребенка по освоению программы.

При решении развивающих и воспитательных задач применяются также элементы технологии развития критического мышления, которая способствует формированию качеств социально активной личности.

Результаты отслеживаются по следующим параметрам:

- уровень освоения детьми содержания предмета освоения образовательной программы «Программирование на Python» (Приложение №5)
- уровень сформированности познавательного интереса «Лист наблюдения» по методике Щукиной Г.И. (Приложение №6).

2.4. Методико-дидактический комплекс

Дидактические материалы

Дидактические материалы представлены:

- методическими пособиями, разработанными педагогом с учётом конкретных задач в виде раздаточного материала, (презентаций, рабочих карточек, карточек с контрольными заданиями)

- демонстрационные материалы по теме занятия.
- методические видео с подробным разбором материалов, рекомендуемых для использования на занятии.

Методические материалы

В образовательном процессе используются следующие *методы* обучения по *-источнику передачи восприятия информации:*

словесные методы:

- *Рассказ:* Краткое, эмоционально окрашенное изложение фактов, истории создания языка Python, биографий известных программистов или примеров из реальной жизни, где применяется программирование.
- *Объяснение:* Толкование изучаемых понятий, принципов работы алгоритмов, синтаксических конструкций языка. Это основной метод при введении новой темы (например, объяснение того, как работает цикл `for`).
- *Беседа* (вводная, эвристическая): Вопросно-ответный метод. Вводная беседа используется для актуализации знаний перед изучением новой темы. Эвристическая (поисковая) беседа строится так, чтобы ученики с помощью наводящих вопросов педагога сами «открыли» для себя новое правило или закономерность.

- характеру деятельности:

- *Объяснительно-иллюстративный*- педагог объясняет новую тему, иллюстрируя ее основные положения средствами мультимедийных презентаций, обучающими программами.
- *Практический* (практические задания; анализ и решение проблемных ситуаций и т. д.).
- *Практическая работа:* выполнение задания по инструкции (алгоритму) для закрепления материала.
- *Метод инженерного изложения* (постановка задачи и решение её самостоятельно или группой. Перед учениками ставится задача, которую необходимо реализовать на языке Python, определив метод и алгоритм ее решения).
- *Проектная работа:* выполнение комплекса заданий, направленных на создание уникального творческого продукта (проекта) за определенный срок.
- *Решение ситуационных задач* (кейс-метод): предложение учащимся реальной или смоделированной проблемной ситуации, которую нужно решить с помощью программирования. Например: «В школьной столовой нужно автоматизировать расчет сдачи. Напишите программу- калькулятор».

При реализации программы используются следующие *педагогические технологии:*

- **Группового обучения.**
- **проектная технология.** Учащиеся не просто изучают команды языка, а работают над созданием собственного проекта в течение всего курса. Например: разработка текстовой игры-квеста. Это формирует навыки планирования, командной работы и применения знаний на практике.
- **Технология проблемного обучения.** Педагог ставит перед учениками конкретную проблему или задачу (например, "Как отсортировать список игроков в игре по их очкам?"), а ученики сами ищут пути решения, исследуя возможности языка. Это стимулирует исследовательскую активность.
- **Игровые технологии.** Внедрение элементов игры в учебный процесс для повышения мотивации. Это может быть система достижений за решенные задачи, командные соревнования по написанию самого короткого или быстрого кода.
- **Технология сотрудничества.** Организация работы в парах или малых группах над общим проектом. Это развивает коммуникативные навыки и учит распределять задачи в команде.

- **Адаптивное обучение.** Использование онлайн-платформ, которые подстраивают сложность задач под уровень конкретного ученика, позволяя сильным ученикам двигаться быстрее, а тем, кто испытывает трудности, - получать больше практики на базовых заданиях.
- **Здоровьесберегающие** технологии. Способствуют предотвращению состояний переутомления, гиподинамии учащихся.

Формы учебных занятий

Форма организации образовательного процесса- групповая.

Формы проведения занятий

Курс «Программирование на Python» предусматривает групповую работу школьников, а также предоставляет им возможность проявить и развить свою самостоятельность.

Основной формой проведения учебных занятий является *практическое занятие*.

Однако в ходе реализации программы, педагог вправе применять любую из доступных форм организации учебного занятия: беседа, встреча с интересными людьми, выставка, лекция, обсуждения, дискуссии, решение инженерных кейсов, викторины.

По дидактической цели: занятие по углублению знаний, практическое занятие, по контролю знаний, комбинированные формы.

Алгоритм учебного занятия.

Учебное занятие любого типа как модель можно представить в виде последовательности следующих этапов: *организационного, проверочного, подготовительного, основного, контрольного, рефлексивного (самоанализ), итогового, информационного*. Каждый этап отличается от другого сменой видов деятельности, содержанием и конкретной задачей.

2.5.Рабочая программа воспитания

Характеристика объединения: объединение «Программист» создано и действует на базе МБОУ ДО ЦДО с 2026 г., в объединении занимаются обучающиеся в возрасте от 11 до 17 лет.

Принципы воспитательного процесса, используемые в объединении: объединение педагогического руководства с инициативой и самостоятельностью обучающихся. Деятельность объединения строится на *принципах*:

- равенства всех участников;
- добровольного привлечения к различным видам деятельности;
- развития духа соревнований, товарищества, взаимовыручки.

Методы воспитания: мотивация, убеждение, поощрение, упражнение, стимулирование, создание ситуации успеха и др.

Целевой ориентир программы воспитания объединения «Программист» — воспитание у обучающихся информационной культуры и безопасного поведения при работе с ИКТ, а также формирование уважительного отношения к своему и чужому труду, толерантности, умения планировать работу, ориентироваться в информационном пространстве и решать общественно значимые задачи с применением ИКТ-технологий.

Цель: воспитание у обучающихся информационной культуры и безопасного поведения при работе с ИКТ.

Ключевым фактором в формировании информационной культуры и воспитания уважительного отношения к своему и чужому труду, толерантности к людям, независимо от возраста, национальности, вероисповедания является проведение мероприятий с привлечением родителей, общественных организаций, СМИ, работа в команде.

Модуль «Ключевые дела внутри объединения»

Это традиционные дела в объединении, в которых принимает участие большая часть обучающихся, которые обязательно планируются, готовятся и проводятся, а также анализируются совместно педагогом и детьми.

На уровне объединения:

- Ежегодные творческие мероприятия технической направленности, согласно плану работы объединения.

- Социальные акции.

- Церемония награждения по итогам года за активное участие в жизни объединения.

За пределами объединения:

- Социальный проект.

- Привлечение обучающихся к совместной работе с учреждениями соцсферы, общественных организаций и т.п. (помощь по благоустройству территории Центра дополнительного образования, участие в городских ярмарках, проектах).

Модуль «Образовательное занятие»

Реализация педагогом воспитательного потенциала занятия предполагает:

- Установление доверительных отношений между педагогом и обучающимся.

- Побуждение обучающихся соблюдать на занятии общепринятые нормы поведения (правила общения с педагогом и сверстниками).

- Привлечение внимания обучающихся к ценностному аспекту изучаемых на занятии явлений (моделирование инженерных задач).

- Включение игровых ситуаций в образовательный процесс с целью поддержания мотивации к получению знаний в области программирования

Модуль «Работа с родителями»

Работа с родителями или законными представителями обучающихся осуществляется для более эффективного достижения цели воспитания в виде следующих форм:

- Родительские собрания (информирование родителей по вопросам воспитания и образовательного процесса в объединении).

- Родительский чат в Махи Telegram (обсуждение актуальной информации в режиме онлайн).

- Индивидуальные консультации (индивидуальный маршрут развития обучающихся).

Планируемые результаты:

- Происходит формирование умения планировать свою работу и распределять время для выполнения поставленных информационных задач, самостоятельно оценивать уровень безопасности информационно насыщенной среды для выработки собственной ценностно-поведенческой линии в сфере жизнедеятельности.

- Развивается умение ориентироваться в информационном пространстве.

- Формируется умение комплексного видения проблем современного общества, поиска оптимального способа их решения с применением ИКТ технологий.

Календарный план воспитательной работы объединения «Программист»

<i>№пп</i>	<i>Направления воспитательной работы</i>	<i>Название мероприятия (события) и форма его проведения</i>	<i>Краткое содержание</i>	<i>Сроки проведения</i>	<i>Ответственный исполнитель</i>
1.	Быть со всеми, оставаться самим собой (социальная адаптация и социальная автономизация личности)	День открытых дверей Мастер-классы, консультации для детей и родителей	Информирование родителей об учебно-воспитательном процессе в объединении, консультации для родителей	Сентябрь, октябрь	Яголови ч В.И.
2.	Познавательное	Конкурс по программированию	Участие в конкурсе по программированию на python	Февраль	

		ию «Я могу»	между учащимися		
3.	Познавательное	Турнир по питону «Битва Алгоритмов»	Воспитание ценностного отношения к своему труду, повышение познавательного интереса к предмету.	Март	
4.	ЗОЖ	«Зарядка для всех»	Формирование основ здорового образа жизни у обучающихся	Октябрь Декабрь Март Апрель	
5.	Познавательное	Конкурс «Техностарт»	Участие в городском конкурсе «Техностарт» Воспитание ценностного отношения к своему труду, повышение познавательного интереса к предмету.	ноябрь	
6.	Познавательное, досуговое	Профильная смена «Эрудит. Программирование на Python»	Организация и проведение познавательной, творческой, досуговой и спортивной деятельности	Июнь	

3.Список литературы

3.1. Список литературы для педагогических работников

1. Бизли, Д. М. *Python. Подробный справочник* [Электронный ресурс] / Д. М. Бизли. — URL: www.python.org (дата обращения: 28.04.2026).
2. Кириенко, Д. П. *Основы языка программирования Python: легко и просто* [Электронный курс] / Д. П. Кириенко. — URL: <http://informatics.msk.ru/course/view.php?id=156> (дата обращения: 28.04.2026).
3. Лутц, М. *Язык программирования Python* [Электронный курс] / М. Лутц. — URL: <http://www.intuit.ru/studies/courses/49/49/info> (дата обращения: 28.04.2026).
4. Лутц, М. *Введение в * [Электронный курс] / М. Лутц. — URL: <http://www.intuit.ru/studies/courses/12179/1172/info> (дата обращения: 28.04.2026).
5. Мейерс, К. *Серия практических примеров на Python* [Электронный ресурс] / К. Мейерс. — URL: openbookproject.net (дата обращения: 28.04.2026).
6. Поляков, К. Ю. *Материалы для подготовки к итоговой аттестации по информатике в форме ОГЭ* [Электронный ресурс] / К. Ю. Поляков. — URL: <http://kpolyakov.spb.ru/school/oge.htm> (дата обращения: 28.04.2026).
7. Босова Л.Л. Комбинированные уроки информатики / Л.Л Босова // Информатика и образование. - 2000. - № 3. - С. 36-41.
8. Босова Л.Л. О методике проведения уроков информатики с младшими школьниками : статья // <http://www.ito.su/2001/ito/I/2/I-2-11.html>.
- 9.
10. Волкова Н.П. Педагогика / Н.П. Волкова. - Киев : Академия, 2001. - 384 с.

11. Давыдова Е.В. Преподавание информатики в школе с углубленным изучением предмета / Е.В. Давыдова, Л.В. Дашкова // <http://www.psyedu/view.php?id=157>.

3.2.Список литературы для обучающихся

1. Поляков, К. Ю., Еремин, Е. А. *Информатика* : учебник для углублённого уровня / К. Ю. Поляков, Е. А. Еремин. — М.: Бином : Лаборатория знаний, 2013.
2. Лутц, М. *Изучаем Python* / М. Лутц ; пер. с англ.— СПб.: Символ-Плюс, 2011.
3. Семакин, И.Г., Хеннер, Е.К., ред.-сост.. *Задачник-практикум по информатике* : учебное пособие для средней школы / под ред.: И.Г.Семакина, Е.К.Хеннера.— М.: БИНОМ : Лаборатория знаний.
4. Поляков, К.Ю.. *Компьютерный практикум в электронном виде с комплектом электронных учебных средств* [Электронный ресурс] / К.Ю.Поляков.— URL: <http://kpolyakov.spb.ru/school/probook.htm> (дата обращения: 28.04.2026).

**Календарно-тематическое планирование
«Программирование на Python» I год обучения – 4 часа в неделю**

Учебная неделя	Название раздела, темы	Количество часов			Форма занятий	Формы аттестации/контроля
		Всего	Теория	Практика		
Первый год обучения						
1 неделя	Введение. Техника безопасности и правила работы на компьютере.	2	1	1	Беседа, наглядные пособия	Опрос, викторина
Раздел 1. Алгоритмы(46часов)						
1 неделя	Понятие алгоритма, свойства алгоритмов, виды алгоритмов	2	2	0	Лекция, наглядные пособия	Опрос
2 неделя	Практическая работа по созданию и описанию алгоритмов. Блок-схемы	4		4	Практическое занятие	Практическое задание

3 неделя	Практическая работа по созданию и описанию алгоритмов. Блок-схемы	2		2	Практическое занятие	Практическое задание
3 неделя	Способы описания алгоритмов, исполнители алгоритмов	2	1	1	Комбинированное занятие	Опрос, Практическое задание
4 неделя	Понятие линейного алгоритма, примеры линейных алгоритмов	2	2		лекция	Опрос
4 неделя	Практическая работа по созданию и описанию линейных алгоритмов. Блок-схемы	2		2	Практическое занятие	Работа по карточкам
5 неделя	Виды и анализ линейных алгоритмов и алгоритмов с ветвлениями, исполнители алгоритмов	2	1	1	Комбинированное занятие Разбор ошибок	Опрос, Практическое задание
5 неделя	Практическая работа по созданию и описанию алгоритмов с ветвлением. Блок-схемы	2		2	Практическое занятие, задание по карточкам	текущий контроль
6 неделя	Практическая работа по созданию и описанию алгоритмов с ветвлением. Блок-схемы	2		2	Практическое занятие Разбор ошибок	Опрос, Практическое задание
6 неделя	Простые и составные условия, вложенные ветвления	2	2		лекция	Опрос
7 неделя	Простые и составные условия, вложенные ветвления	2	2		Практическое занятие	Опрос, Практическое задание

7 неделя	Линейные алгоритмы, программы	2		2	Практическое занятие	текущий контроль
8 неделя	Алгоритмы с ветвлениями, программы	2		2	Практическое занятие Разбор ошибок	текущий контроль Опрос
8 неделя	Самостоятельная работа по созданию и описанию линейных, алгоритмов с ветвлением	2		2	Практическое занятие	Тестовые задания
9 неделя	Простые и составные условия, ветвления, вложенные ветвления	2	1	1	Комбинированное занятие, практическое занятие	Опрос Практическое задание
9 неделя	Практическая работа по созданию и описанию алгоритмов с простыми и составными условиями	2		2	Практическое занятие	текущий контроль
10 неделя	Алгоритмы с вложенными ветвлениями, программы	2		2	Практическое занятие Разбор ошибок	Тестовые задания
10 неделя	Самостоятельная работа по созданию линейных алгоритмов и алгоритмов с ветвлением	2		2	Практическое занятие	Практическое задание
11 неделя	Самостоятельная работа по созданию линейных алгоритмов и алгоритмов с ветвлением	2		2	Практическое занятие Разбор ошибок	Практическое задание Опрос
11-12 неделя	Тестовые задания по всем темам	4		4	Практическое занятие	Практическое задание
12 неделя	резерв	2	1	1	Практическое занятие	Практическое задание
	Итого	48	13	35		

Раздел 2. Знакомство с языком Python(32час)

13-13 неделя	Знакомство с историей развития языков программирования, способами трансляции программного кода Знакомство с Python и его особенностями, простейшая программа	4	2	2	Беседа, наглядные пособия Практическое занятие	Опрос, викторина, задания по карточкам
14 неделя	Практическая работа. Установка программы Python	2	1	1	Комбинированное занятие Практическое занятие	Практическое задание
14 неделя	Среда разработки IDLE.	2	1	1	Комбинированное занятие Практическое занятие	Практическое задание
15 неделя	Структура программы. Комментарии. Этапы решения задач на ПК	2	1	1	Комбинированное занятие Практическое занятие	Опрос Практическое задание
15-16 неделя	Основы синтаксиса Python и арифметические выражения	4	2	2	Комбинированное занятие Практическое занятие	Практическое задание
16 неделя	Ввод/вывод данных	2		2	Практическое занятие	Практическое задание
17-17 неделя	Создание простейших программ Работа с числами в Python	4	1	3	Практическое занятие Разбор ошибок	Практическое задание Опрос
18-18 неделя	Практическая работа на выполнение арифметических действий	4		4	Практическое занятие	Практическое задание
19-19 неделя	Задачи на элементарные действия с числами	4		4	Практическое занятие Разбор ошибок,	Практическое задание
20-20 неделя	Резерв	4		4	Практическое занятие Разбор ошибок	Практическое задание
	итого	32	8	24		

Раздел 3.Переменные(36 часов)						
21 неделя	Переменные. Правила именования переменных. Допустимые и не допустимые символы в обозначении переменных	2	1	1	Комбинированное занятие.	Опрос, Практическое задание
21 неделя	Объявление переменной, ,	2	1	1	Практическое занятие-разработка программ	Практическое задание
22 неделя	Инициализация значений, множественное присваивание	2	1	1	Комбинированное занятие	Практическое задание
22-23 неделя	Разработка программ с использованием операций присваивания	4	2	2	Практическое занятие	Практическое задание
23-24 неделя	Практическая работа с использованием множественного присваивания	4		4	Занятие закрепления знаний, выработки умений и навыков	Работа по карточкам, текущий контроль
24-25 неделя	Основные типы данных переменных в Python, знакомство с числовыми, строковыми, логическими типами данных	4	2	2	Комбинированное занятие	Практическая работа
25-26 неделя	Создание программ с использованием переменных различных типов	4		4	Практическое занятие. Занятие закрепления знаний	Практическая работа Опрос
26-27 неделя	Практическая работа: задачи с арифметическими выражениями	4		4	Практическое занятие. Занятие закрепления знаний	Практическая самостоятельная работа. Подведение итогов,
27-28 неделя	Практическая работа: задачи на ввод и вывод данных	4		4	Практическое занятие	Практическая самостоятельная работа. Подведение итогов
28-29 неделя	Тестовые задания	4		4	Практическое занятие	Работа по тестам

29 неделя	резерв	2		2	Практическое занятие	Самостоятельная работа
	Итого	36	7	29		
Раздел 4. Вычисления(32 часа)						
30 неделя	Понятие целых и вещественных чисел, преобразователи при вводе данных разных типов	2	1	1	Комбинированное занятие	Работа по тестам
30 неделя	Понятие деления нацело и остаток от целочисленного деления	2	1	1	Комбинированное занятие	Опрос
31 неделя	Создание программ с использованием простейших арифметических действий	2	1	1	Практическое занятие	Практическое задание
31-32 неделя	Практическая работа с использованием математических операторов ://, % и **	4	2	2	Комбинированное занятие	Работа по карточкам
32 неделя	Практическая работа с использованием вещественных значений и целых чисел	2	1	1	Комбинированное занятие	Работа по карточкам
33-33 неделя	Стандартные функции модуля math, Встроенные числовые функции	4	2	2	Комбинированное занятие	Практическое задание
34 неделя	Практическая работа с использованием функций модуля math	2	1	1	Практическое занятие	Работа по тестам
34 неделя	Повторение материала.Практическая работа с использованием математических операторов ://, % и **	2	1	1	Практическое занятие	Практическое задание, опрос, обсуждение
35-35 неделя	Комбинированная работа по карточкам	4		4	Практическое занятие	Практическое задание, опрос, обсуждение
36	Форматирование чисел для вывода	2	1	1	Практическое занятие	Практическое задание

неделя	данных					
36 неделя	Задачи на использование форматированного вывода	2		2	Практическое занятие	Практическое задание
37 неделя	Итоговая практическая работа	2		2	Практическое занятие	Практическое задание
37 неделя	Резерв	2	1	1	Комбинированное занятие повторение наиболее сложных тем	Опрос, беседа
	Итого	32	12	20		
	Всего за год	148	40	108		
Второй год обучения						
38 неделя	Техника безопасности и правила работы на компьютере.	2	1	1	Беседа, наглядные пособия	Опрос, тест на знание правил техники безопасности
	Раздел 1 .Основные алгоритмические конструкции(48 часов)					
38-39 неделя	1.1 Виды и анализ алгоритмов – линейных, условных и циклических , и их блок-схемы	4	1	3	Практическое занятие	Практическое занятие
	1. 2 Условные алгоритмы в Python					
39 неделя	Ветвление в Python. Оператор if- else.	2	1	1	Комбинированный урок Практическое занятие	Практическое задание, опрос, обсуждение
40-40 неделя	Практическое задание с использованием оператора if-else.	4	1	3	Практическое занятие	Практическое задание, опрос, обсуждение
41 неделя	Множественное ветвление. Оператор if-elif-else	2	1	1	Практическое занятие	Практическое задание
41-42 неделя	Практическое задание с использованием оператора if-elif- else.	4	1	3	Систематизация знаний по изученным ранее темам	Практическое задание, опрос

	1.3 Циклы(24 часа)					
42-43 неделя	Цикл с параметром FOR	4	1	3	Комбинированный урок	Практическое задание
43-44 неделя	Цикл с условием WHILE. Вложенные циклы.	4	1	3	Комбинированный урок	Практическое задание
44-45 неделя	Использование логики True, False, флагов. Операторы управления циклом: break, continue, else, решение задач	4	2	2	Практическое занятие Комбинированный занятие	Практическое задание карточки
45-46 неделя	Решение задач с использованием операторов if-elif-else, for, while	4	1	3	Комбинированное занятие	Практическое задание карточки
46-47 неделя	Решение задач с использованием логики True, False, флагов	4	1	3	Систематизация знаний по изученным ранее темам	Опрос Практическое задание
47-48 неделя	Решение задач с использованием Операторов управления циклом: break, continue, else,	4	1	3	Систематизация знаний по изученным ранее темам	Опрос Практическое задание
	1.4 Логические выражения и операторы в Python					
48-49 неделя	Логические операторы в Python: and, or и not. Операции сравнения. Вычисление суммы последовательностей	4	2	2	Лекция , презентация «Логика в Python» 1 часть	Опрос Практическое задание
49 неделя	Логическое выражение. Простые и сложные логические выражения. Результат вычисления логического выражения	2	1	1	Лекция , презентация «Логика в Python»2 часть	Опрос Практическое задание
50 неделя	Практическое занятие с	2	1	1	Комбинированное	Практическое занятие

	использованием логических выражений				занятие	«работа по карточкам»
	итого	50	17	33		
Раздел 2 .Строки (30часов)						
50-51 неделя	Что такое строка. Создание строки Специальные символы. Строковые величины. Операции со строками	4	2	2	Комбинированный урок Практическое занятие	Практическое задание
51-52 неделя	Действия над строками: обращение к элементу по индексу, срез	4	1	3	Практическое занятие Комбинированный урок	Практическое задание
52-53 неделя	Практическое занятие по теме : обращение к элементу по индексу, срезы	4	1	3	Комбинированный урок	Практическое задание, опрос, обсуждение
53 неделя	Длина строки и отрицательные индексы	2	1	1	Практическое занятие	Практическое задание
54 неделя	Практическое занятие по теме: проверка на вхождение, повтор, функция len()	2		2	Практическое занятие	Практическое задание
54 неделя	Срезы строк, сравнение строк	2	1	1	Комбинированный урок	Практическое задание
55 неделя	Оператор in, модуль string	2	1	1	Комбинированный урок	Практическое задание
55 неделя	Операторы для всех типов последовательностей(строки , списки, кортежи)	2	1	1	Комбинированный урок	Практическое задание
56 неделя	Поиск и замена в строке	2	1	1	Практическое занятие	Практическое задание
56 неделя	Поиск и замена символа в строке, нахождение подстроки в строке	2		2	Практическое занятие	Практическое задание, опрос, обсуждение
57-57 неделя	Практические занятия по всем темам	4		4	Систематизация знаний по изученным ранее темам	Практическое задание, опрос, обсуждение

	Итого	30	9	21		
Заддел 3.Списки (34 часа)						
58-58 Неделя	Создание списков	4	1	3	Комбинированный урок, практическое занятие	Практическое задание, опрос, обсуждение
59-59 неделя	Операции над списками:индексация, изменение элементов, объединение списков, разложение списка, перебор элементов списка с помощью циклов <u>for</u> и <u>while</u> . Сравнение списков	4	1	3	Комбинированный урок, практическое занятие	Практическая самостоятельная работа. Подведение итогов,
60-60 неделя	Встроенные функции для списков len(),sorted(),min() и max()	4	1	3	Комбинированный урок, практическое занятие	Практическое задание, опрос, обсуждение
61-61 неделя	МетодысписковPython:append(),extend(),insert(),remove(),clear(),index(),pop(),count(),sort(), reverse(),copy()	4	1	3	Комбинированный урок, практическое занятие	Практическое задание, опрос, обсуждение
62 неделя	Поиск элемента в списке, добавление и удаление элементов в списке	2		2	практическое занятие	Практическая самостоятельная работа. Подведение итогов,
62 неделя	Перемешивание элементов списка и выбор случайного элемента	2	1	1	Комбинированный урок,	Практическое задание, опрос, обсуждение
63-63 неделя	Сортировка списков, преобразование списка в строку	4	1	3	Систематизация знаний по изученным ранее темам	Практическое задание, опрос, обсуждение
64-64 неделя	Кортежи: понятие кортежей, создание кортежей, методы кортежей	4	1	3	Систематизация знаний по изученным ранее темам	Практическое задание, опрос, обсуждение

65 неделя	Перебор элементов кортежа	2	1	1	Комбинированный урок	Практическое задание
65 неделя	Итоговое практическое занятие	4		4	Практическое занятие	Самостоятельная итоговая работа
	итого	34	8	26		
Раздел 4.Случайные числа. (26 часов)						
66 неделя	Для чего нужен модуль Random.Как работает генерация случайных чисел	2	2	--	Лекция, презентация «Random»	Опрос,беседа
66-67 неделя	Загрузка модуля Random,Случайные целые числа (int):функция randint (a, b), функция randrange (start, stop[, step])	4	2	2	Комбинированный урок	Практическое задание
68 неделя	Практическое занятие: по теме использование функций randint(a, b),randrange(start, stop[, step])	2		2	Систематизация знаний по изученным ранее темам	Практическое задание
68-69 неделя	Случайные вещественные числа (float): функция random(), функция uniform(a, b)	4	2	2	Комбинированный урок,	Практическое задание
69-70 неделя	Практическое занятие по теме: использование случайных вещественных чисел (float):random(),uniform(a, b)	4	2	2	Комбинированный урок,	Практическое задание
70-71 неделя	Функции для работы с последовательностями: choice(seq), choices(seq [, weights, cum_weights, k]), shuffle(x[, random])	4	2	2	Комбинированный урок,	Практическое задание

71 неделя	Функции для работы с последовательностями:sample(x, k), triangular(a, b, mode)	2	1	1	Комбинированный урок	Практическое задание
72 неделя	Практическое занятие по теме: функции для работы с последовательностями:choice(seq), choices(seq [, weights, cum_weights, k]), shuffle(x[, random])	2		2	Систематизация знаний по изученным ранее темам Практическое занятие	Практическое задание опрос, обсуждение
72 неделя	random.seed, отладка, тестирование Практическое занятие с использованием функции random.seed	2	1	1	Комбинированный урок	опрос, обсуждение тесты
	итого	26	12	14		
73 неделя	Резерв Повторение и углубление тем	2	1	1	Комбинированное занятие	Опрос, Работа по карточкам
73 неделя	Концепция проекта, понятия, постановка задачи, подготовка презентации проекта	2	2	0	Лекция,	Беседа, обсуждение проекта
74 неделя	Разработка программы проекта	2		2	Практическое занятие	Пробный запуск программы
74 неделя	Защита проекта	2		2	Практическое занятие	Итоговая практическая работа
	Итого	8	3	5		
	Всего часов за год	148	48	100		
	Всего за 2 года	296	87	209		

**Календарно-тематическое планирование
«Программирование на Python» 2 год обучения – 2 часа в неделю**

38 неделя	Техника безопасности и правила работы на компьютере.	2	1	1	Беседа, наглядные пособия	Опрос, тест на знание правил техники безопасности
	Раздел 1 .Основные алгоритмические конструкции(18 часов)					
39 неделя	1.1 Виды и анализ алгоритмов – линейных, условных и циклических и их блок-схемы	2	1	1	Практическое занятие	Практическое занятие
	1. 2 Условные алгоритмы в Python					
40 неделя	Ветвление в Python. Оператор if-else.Практическое задание с использованием оператора if-elif-else.	2		2	Комбинированный урок Практическое занятие	Практическое задание, опрос
	1.3 Циклы(24 часа)					
41 неделя	Цикл с условием WHILE. Цикл с параметром FOR.Вложенные циклы.	2	1	1	Комбинированный урок	Практическое задание
42 неделя	Использование логики True, False, флагов. Операторы управления циклом: break, continue, else, решение задач	2	1	1	Практическое занятие Комбинированный занятие	Практическое задание карточки
43 неделя	Решение задач с использованием операторов if-elif-else, for, while	2		2	Комбинированное занятие	Практическое задание карточки
44 неделя	Решение задач с использованием логики True, False, флагов	2	1	1	Систематизация знаний по изученным ранее темам	Опрос Практическое задание

45 неделя	Решение задач с использованием Операторов управления циклом: break, continue, else,	2		2	Систематизация знаний по изученным ранее темам	Опрос Практическое задание
	1.4 Логические выражения и операторы в Python					
46 неделя	Логические операторы в Python: and, or и not. Операции сравнения. Вычисление суммы последовательностей	2	1	1	Лекция , презентация «Логика в Python» 1 часть	Опрос Практическое задание
47 неделя	Практическое занятие с использованием логических выражений	2	1	1	Комбинированное занятие	Практическое занятие «работа по карточкам»
	итого	20	7	13		
Раздел 2 .Строки (16 часов)						
48 неделя	Что такое строка. Создание строки Специальные символы. Строковые величины. Операции со строками	2	1	1	Комбинированный урок Практическое занятие	Практическое задание
49 неделя	Действия над строками: обращение к элементу по индексу, срез	2	1	1	Практическое занятие Комбинированный урок	Практическое задание
50 неделя	Длина строки и отрицательные индексы	2	1	1	Практическое занятие	Практическое задание
51 неделя	Практическое занятие по теме: проверка на вхождение, повтор, функция len()	2		2	Практическое занятие	Практическое задание
52 неделя	Срезы строк, сравнение строк	2	1	1	Комбинированный урок	Практическое задание
53 неделя	Оператор in, модуль string	2	1	1	Комбинированный урок	Практическое задание
54 неделя	Операторы для всех типов	2	1	1	Комбинированный урок	Практическое задание

	последовательностей(строки , списки, кортежи)					
55 неделя	Поиск и замена символа в строке, нахождение подстроки в строке	2		2	Практическое занятие	Практическое задание, опрос, обсуждение
	Итого	16	6	10		
Задел 3.Списки (14 часов)						
56 неделя	Создание списков. Операции над списками:индексация, изменение элементов, объединение списков, разложение списка, перебор элементов списка с помощью циклов <u>for</u> и <u>while</u> . Сравнение списков	2	2		Комбинированный урок, практическое занятие	Практическая самостоятельная работа. Подведение итогов,
57 неделя	Встроенные функции для списков len(),sorted(),min() и max()	2	1	1	Комбинированный урок, практическое занятие	Практическое задание, опрос, обсуждение
58 неделя	МетодысписковPython:append(),ext end(),insert(),remove(),clear(),index() ,pop(),count(), sort(), reverse(),copy()	2	1	1	Комбинированный урок, практическое занятие	Практическое задание, опрос, обсуждение
59 неделя	Поиск элемента в списке, добавление и удаление элементов в списке, перемешивание элементов списка и выбор случайного элемента	2		2	практическое занятие	Практическая самостоятельная работа. Подведение итогов,
60 неделя	Сортировка списков, преобразование списка в строку	2	1	1	комбинированный урок	Практическое задание, опрос, обсуждение
61 неделя	Кортежи: понятие кортежей, создание кортежей, методы	2	1	1	комбинированный урок	Практическое задание, опрос, обсуждение

	кортежей. Перебор элементов кортежа					
62 неделя	Итоговое практическое занятие	2		2	Практическое занятие	Самостоятельная итоговая работа
	итого	14	6	8		
Раздел 4.Случайные числа. (16 часов)						
63 неделя	Загрузка модуля Random,Случайные целые числа (int):функция randint (a, b), функция randrange (start, stop[, step])	2	1	1	Комбинированный урок	Практическое задание
64 неделя	Практическое занятие: по теме использование функций randint(a, b),randrange(start, stop[, step])	2		2	Систематизация знаний по изученным ранее темам	Практическое задание
65 неделя	Случайные вещественные числа (float): функция random(), функция uniform(a, b)	2	1	1	Комбинированный урок,	Практическое задание
66 неделя	Практическое занятие по теме: использование случайных вещественных чисел (float):random(), uniform(a, b)	2	1	1	Комбинированный урок,	Практическое задание
67 неделя	Функции для работы с последовательностями: choice(seq), choices(seq[, weights, cum_weights, k]), shuffle(x[, random])	2	1	1	Комбинированный урок,	Практическое задание
68 неделя	Функции для работы с последовательностями:sample(x, k), triangular(a, b, mode)	2	1	1	Комбинированный урок	Практическое задание

69 неделя	Практическое занятие по теме: функции для работы с последовательностями:choice(seq), choices(seq [, weights, cum_weights, k]), shuffle(x[, random])	2		2	Систематизация знаний по изученным ранее темам Практическое занят ие	Практическое задание опрос, обсуждение
70 неделя	random.seed, отладка, тестирование Практическое занятие с использованием функции random.seed	2	1	1	Комбинированный урок	опрос, обсуждение тесты
	итого	16	6	10		
71 неделя	Резерв Повторение и углубление тем	2		2	Комбинированное занятие	Опрос, Работа по карточкам
72 неделя	Концепция проекта, понятия, постановка задачи, подготовка презентации проекта	2	2	0	Лекция,	Беседа, обсуждение проекта
73 неделя	Разработка программы проекта	2		2	Практическое занятие	Пробный запуск программы
74 неделя	Защита проекта	2		2	Практическое занятие	Итоговая практическая работа
	Итого	8	2	6		
	Всего часов за год	74				

Оценочные материалы промежуточной аттестации

Первый год обучения

Предметные

Компьютерный тест

1. Что необходимо сделать перед началом работы за компьютером в классе?

- а) Сразу включить компьютер.
- б) Проверить, нет ли на рабочем месте посторонних предметов и жидкости.
- в) Громко поздороваться с одноклассниками.
- г) Начать настраивать программу.

Ответ: б

1. Что категорически запрещено приносить и употреблять в компьютерном классе?

- а) Учебники.
- б) Еду и напитки.
- в) Тетради.
- г) Ручки.

Ответ: б

3. "Что такое алгоритм?",

- "Это ошибка в программе",
- "Это набор команд для решения задачи",
- "Это название языка", "Это компьютер",
- "Это набор команд для решения задачи"

4. Как называется алгоритм, в котором действия выполняются строго друг за другом, без повторений и ветвлений?

- а) Циклический.
- б) Разветвляющийся.
- в) Линейный.
- г) Рекурсивный.

Ответ: в

5. Знакомство с Python

3. Какую команду нужно написать, чтобы вывести на экран фразу «Привет, мир!»?

- а) `print(«Привет, мир!»)`.
- б) `write(«Привет, мир!»)`.
- в) `output(«Привет, мир!»)`.
- г) `show(«Привет, мир!»)`.

Ответ: а

6. С помощью какого символа начинаются однострочные комментарии в коде Python?

- а) `//`
- б) `#`
- в) `/*`
- г) `<!--`

Ответ: б

7. Какой тип данных будет у переменной после выполнения команды `x = 10.5`?

- а) `int` (целое число).
- б) `str` (строка).
- в) `float` (вещественное число).

г) bool (логический).

Ответ: в

8. Какое имя переменной написано с ошибкой (не по правилам Python)?

а) age

б) user_name

в) 2nd_number

г) myVar

Ответ: в

9. Чему будет равно значение выражения $10 + 5 * 2$?

а) 30

б) 20

в) 17

г) Ошибка

Ответ: б

10. Какой оператор используется для возведения числа в степень (например, «два в кубе»)?

а) ^

б) **

в) pow

г) *

Ответ: б

11. Что выведет этот код?

```
x = 15
```

```
if x < 10:
```

```
    print("Меньше")
```

```
else:
```

```
    print("Больше или равно")
```

а) Меньше

б) Больше или равно

в) Ошибка

г) Ничего не выведет

Ответ: б

12. (Практическое задание)

Напишите программу, которая выводит на экран квадраты чисел от 1 до 5 (1, 4, 9, 16, 25).

Примерответа:

```
for i in range(1, 6):
```

```
    print(i ** 2)
```

13. (Практическое задание)

Напишите программу, которая запрашивает у пользователя пароль до тех пор, пока он не введет правильный ("secret"). После правильного ввода программа должна вывести «Доступ разрешен».

Пример ответа:

```
password = ""
```

```
while password != "secret":
    password = input("Введите пароль: ")
    print("Доступ разрешен")
```

14. (Практическое задание)

Напишите программу, которая вычисляет сумму всех нечетных чисел от 1 до N (N вводится с клавиатуры).

Пример ответа:

```
n = int(input("Введите N: "))
total = 0
for i in range(1, n+1):
    if i % 2 != 0:
        total += i
print("Сумма:", total)
```

15. Самая старшая логическая операция (выполняется первой)

- A) and B) or B) not
 B) not

16. Какой тип должен быть у переменной **a** для оператора присваивания $a = 25 / 3$?

- A) int B) bool B) float Г) str
 B) float

17. Укажите синтаксически неправильную запись операции присваивания:

- A) $x = (y + 1) / 3$ Б) $x = (y) / 3 + 1$ В) $x + 1 = (y) / 3$ Г) $x = (y) / (3 + 1)$
 В) $x + 1 = (y) / 3$

18. В результате выполнения фрагмента программы $x = 4$

```
y = 5
y = y + x ** 0.5
print('y =', y) на экран будет выведено
```

A) $y = 7$ Б) $y = y$ В) $y = 21$ Г) $y = 7.0$ Д) $y = 9$
 А) $y = 7$

19. В результате выполнения фрагмента программы

```
x = 3
while x < 9:
    print('УРА')
    x = x + 2
```

слово «УРА» будет напечатано:

- A) 0 раз Б) 1 раз В) 2 раза Г) 3 раза Д) 4 раза
 Г) 3 раза

20. В результате выполнения фрагмента программы

```
for a in range(5):
    print('РОССИЯ')
```

слово «РОССИЯ» будет напечатано:

- A) 1 раз Б) 0 раз В) 4 раза Г) 5 раз Д) 3 раза
 В) 4 раза

21. Укажите, какими способами значения переменных **a** и **b** поменяются местами

- A) $a = b$ $b = a$

Б) a, b = b, a

В) b = a

c = a

a = b

Г) c = a

a = b

b = c

Д) c = b

c = a

b = a

Б) a, b = b, a ,

Г) c = a

a = b

b = c

22. Какой ряд чисел образуется после выполнения следующего алгоритма:

```
for i in range(1,10):
```

```
print(i)
```

a) 1 2 3 4 5 6 7 8 9 10

b) 1 2 3 4 5 6 7 8 9

c) 0 1 2 3 4 5 6 7 8

b) 1 2 3 4 5 6 7 8 9

23. Что будет в результате выполнения следующего алгоритма программы: a = int(input())

```
b = int(input())
```

```
if a % 10 == 0 or b % 10 == 0:
```

```
print('YES') else:
```

```
print('NO')
```

если a = 15, b = 45?

a) YES

b) NO

b) NO

24. Определите, что будет напечатано в результате работы следующей программы

```
s = 0
```

```
for i in range(8, 13):
```

```
s = s + 12
```

```
print(s)
```

Ответ: 60

25. Напишите программу, которая запрашивает у пользователя число и проверяет, является ли оно положительным, отрицательным или нулём.

Пример ответа:

```
x = int(input("Введите число: "))
```

```
if x > 0:
```

```
print("Число положительное")
```

```
elif x < 0:
```

```
print("Число отрицательное")
```

```
else:
```

```
print("Число равно нулю")
```

Критерии оценки:

Каждый правильный теоретический ответ оценивается в один балл, практический в два балла. Максимальное количество баллов – 30. Набранные баллы переводятся в уровень освоения программы за первый год обучения по следующей шкале:

10 – 15 баллов - низкий уровень;

16-24 баллов - средний уровень;

25 – 30 баллов - высокий уровень

Метапредметные результаты: Оценочные материалы и примеры

Планируемый результат	Инструмент оценки (что используем)	Пример конкретного задания/критерия
1 год обучения		
Применять нестандартные подходы при решении учебных задач.	Практическая работа с открытым заданием.	Задание: «Написать программу, которая приветствует пользователя. Не использовать input() для ввода имени. Имя должно браться из системных данных или генерироваться случайно». Оценка: 1 балл за работающую программу, 1 балл за нестандартный подход.
Осуществлять самоанализ своей деятельности, оценивать правильность выполнения заданий и вносить коррективы.	Практическая работа «Отладка кода» (Debugging).	Задание: Дан код, который должен считать среднее арифметическое чисел в списке. Результат неверный. numbers = [10, 20, 30] average = sum(numbers) / len(numbers) - 1 print(average) Критерий: 1 балл за нахождение ошибки, 1 балл за исправление (- 1 убрать), 1 балл за комментарий в коде # Исправил формулу.
Эффективно работать в команде: обсуждать идеи, аргументировать свою точку зрения.	Карта наблюдения за работой в паре/группе.	Критерий для учителя: «Учащийся не просто пишет свой код, а объясняет партнеру логику своей функции. Умеет выслушать аргументированное предложение партнера и внедрить его в общий проект».
Самостоятельно находить, отбирать и использовать необходимую информацию.	Практическая работа «Исследователь».	Задание: «Создайте программу, которая определяет, является ли год високосным. Вам нужно самостоятельно найти в интернете или документации алгоритм/формулу для определения високосного года и реализовать ее на Python».

Результаты выполнения этих заданий заносятся в электронный журнал или бумажный журнал в виде баллов

Журнал полученных баллов

№ вопроса Ф.И. учащегося	1	2	3	4	5	6	Количество набранных баллов

Оценочные материалы для Личностных результатов

Карта наблюдения за деятельностью ученика

Ф.И.О. ученика	Критерий: Ответственность (выполнение ДЗ)	Критерий: Работа в команде	Критерий: Инициатива	Примечание
	Всегда сдает в срок, просит доп. задания.	Активно участвует в обсуждении, помогает другим.	Предложил использовать новую библиотеку.	
	Сдает нерегулярно, часто не доделывает.	Склонен работать в одиночку, не делится кодом.	Ждет инструкций, не проявляет инициативу.	

Оценивается (1-3)балла в конце отчетного периода.

Приложение №3

Оценочные материалы промежуточной аттестации

2 год обучения предметные

1. Что из перечисленного является обязательным правилом техники безопасности в компьютерном классе?

- а) Работать за компьютером в верхней одежде.
- б) Отключать питание системного блока мокрыми руками.
- в) Не касаться проводов и разъемов мокрыми руками.
- г) Приносить еду и напитки к рабочему месту.

Ответ: в

2. Какое действие запрещено при возникновении запаха гари или дыма от компьютера?

- а) Сообщить преподавателю.
- б) Самостоятельно отключить компьютер от сети.
- в) Продолжать работу, игнорируя запах.
- г) Эвакуироваться из помещения, если ситуация критическая.

Ответ: в

3. Какой оператор используется для создания цикла с предусловием в Python?

- а) for
- б) while
- в) repeat
- г) do

Ответ: б

4. Что выведет следующий код?

```
```python
x = 5
if x > 10:
 print("Больше")
elif x > 0:
 print("Положительное")
else:
 print("Ноль или меньше")
```
```

- а) Больше
- б) Положительное
- в) Ноль или меньше
- г) Ошибка

Ответ: б

5. Что будет выведено на экран в результате выполнения следующего фрагмента кода?

```
python
i = 1
while i < 5:
    print(i, end=' ')
    i += 1
```

- а) 1 2 3 4
- б) 1 2 3 4 5
- в) Бесконечный цикл
- г) Ошибка

Ответ: а

6. (Практическое задание)**

Напишите программу, которая вычисляет сумму всех нечетных чисел от 1 до N (N вводится с клавиатуры).

Пример ответа:

```
python
n = int(input("Введите N: "))
total = 0
for i in range(1, n+1):
    if i % 2 != 0:
        total += i
print("Сумма:", total)
```

7. (Практическое задание)

Напишите программу, которая запрашивает у пользователя пароль. Если пароль неверный, программа должна предлагать ввести его снова (использовать цикл `while`). Верный пароль — qwerty.

Пример ответа:

```
python
password = ""
while password != "qwerty":
    password = input("Введите пароль: ")
print("Доступ разрешен")
```

8. (Практическое задание)

Напишите фрагмент кода, который выводит все числа от 1 до 10, кратные 3.

Пример ответа:

```
python
for i in range(1, 11):
    if i % 3 == 0:
        print(i)
```

9. Как получить третий символ строки `s = "Привет"`?

- а) `s[3]`
- б) `s(3)`
- в) `s[2]`
- г) `s.get(3)`

Ответ: в

10. Каков результат выражения `len("Python")`?

- а) 5
- б) 6
- в) 7
- г) Ошибка

Ответ: б

11. (Практическое задание)

Напишите программу, которая запрашивает у пользователя слово и выводит его в обратном порядке.

Пример ответа:

```
python
word = input("Введите слово: ")
print(word[::-1])
```

12. Как преобразовать строку "123" в целое число?

- а) `int("123")`
- б) `str(123)`
- в) `to_int("123")`
- г) `number("123")`

Ответ: а

13. Какой метод вернет строку "helloworld" в верхнем регистре?

- а) `"hello world".upper()`
- б) `"hello world".toUpper()`
- в) `upper("hello world")`
- г) `"hello world".capitalize()`

Ответ: а

14. (Практическое задание)

Напишите программу, которая проверяет, является ли введенное пользователем слово палиндромом (читается одинаково слева направо и справа налево, например, «шалаш», «казак»).

Пример ответа:

```
python
word = input("Введите слово: ").lower()
if word == word[::-1]:
    print("Этопалиндром")
else:
    print("Это не палиндром")
```

15. Что делает метод list.sort()?

- а) Создаст новый отсортированный список, не изменяя старый.
- б) Отсортирует список по убыванию.
- в) Отсортирует список по возрастанию на месте.
- г) Перемешает элементы списка.

Ответ: в

16. Как получить длину списка fruits = ['apple', 'banana', 'cherry']?

- а) fruits.length()
- б) len(fruits)
- в) length(fruits)
- г) fruits.size()

Ответ: б

17. (Практическое задание)

Дан список чисел. Напишите код, который создает новый список, содержащий только четные числа из исходного.

Примерответа:

```
python
numbers = [1, 2, 3, 4, 5, 6]
even_numbers = [x for x in numbers if x % 2 == 0]
print(even_numbers)
```

18. Как добавить элемент 10 в конец списка my_list?

- а) my_list.add(10)
- б) my_list.append(10)
- в) my_list.insert(10)
- г) my_list.push(10)

Ответ: б

19. Как удалить второй элемент из списка numbers = [1, 2, 3, 4]?

- а) numbers.remove(2)
- б) numbers.pop(2)
- в) numbers.delete(2)
- г) numbers.erase(2)

Ответ: б

20. (Практическое задание)

Напишите код для создания списка квадратов чисел от 1 до 5 с помощью генератора списков.

Примерответа:

```
python
```

```
squares = [x**2 for x in range(1, 6)]
print(squares)
```

21.С помощью какого модуля в Python работают с генератором случайных чисел?

- а) math
- б) random
- в) numpy
- г) sys

Ответ: б

22. Как сгенерировать случайное целое число от 1 до 10 включительно?

- а) random.rand(1, 10)
- б) random.randint(1, 10)
- в) random.random(1, 10)
- г) random.range(1, 10)

Ответ: б

23.(Практическое задание)

Напишите программу, которая имитирует бросок шестигранного кубика (выводит случайное число от 1 до 6).

Примерответа:

```
python
import random
print(random.randint(1, 6))
```

24.Как сгенерировать случайное вещественное число в диапазоне от 0 до 1?

- а) random.randint(0, 1)
- б) random.random()
- в) random.uniform(0, 1)
- г) random.range(0, 1)

Ответ: б

25.Какой метод позволяет перемешать элементы списка случайным образом?

- а) random.shuffle(list)
- б) list.randomize()
- в) mix(list)
- г) random.mix(list)

Ответ: а

Критерии оценки:

Каждый правильный теоретический ответ оценивается в один балл, практический в два балла. Максимальное количество баллов – 33. Набранные баллы переводятся в уровень освоения программы за второй год по следующей шкале:

- 10 – 19 баллов- низкий уровень;
- 20 – 28 баллов: средний уровень;
- 29 – 33 баллов: высокий уровень.

Метапредметные результаты: Оценочные материалы и примеры

| Планируемый результат | Инструмент оценки (что используем) | Пример конкретного задания/критерия |
|-----------------------|------------------------------------|-------------------------------------|
| 2-й год обучения | | |
| Применять | Проектная задача | 1. Умение разбить сложную |

| | | |
|--|--|---|
| комплексные методы анализа для решения нестандартных задач повышенной сложности. | «Разработка алгоритма».
<i>Задание:</i> Написать программу для решения головоломки (например, «Ханойские башни», лабиринт) или реализовать алгоритм сортировки (не sort(), а вручную). | задачу на простые шаги (декомпозиция).
2. Корректность работы алгоритма на разных наборах входных данных.
3. Наличие комментариев, объясняющих логику сложного алгоритма. |
| Планировать и организовывать самостоятельную работу над проектом, распределяя задачи в команде. | Итоговый групповой проект с защитой.
<i>Задание:</i> Разработать полезное приложение (например, планировщик дел, калькулятор скидок, простой чат-бот). | Оценивается:
- Наличие плана проекта (сроки, ответственные)..
- Готовая презентация с демонстрацией работы продукта.
- Умение каждого члена команды рассказать о своей части работы. |
| Владеть навыками проектной деятельности: от постановки задачи до презентации готового продукта. | Защита индивидуального или группового проекта. | Оценивается по критериям
- Постановка задачи: Четко ли определена проблема и цель проекта?
- Реализация: Работает ли программа стабильно? Соответствует ли код стандартам?
- Презентация: Убедительно ли ученик представил свой продукт? Отвечает ли на вопросы по существу? |

Результаты выполнения этих заданий заносятся в электронный журнал или бумажный журнал в виде баллов

Журнал полученных баллов

| № вопроса
Ф.И. учащегося | 1 | 2 | 3 | 4 | 5 | 6 | Количество набранных баллов |
|-----------------------------|---|---|---|---|---|---|-----------------------------|
| | | | | | | | |
| | | | | | | | |
| | | | | | | | |

Оценочные материалы для Личностных результатов

Карта наблюдения за деятельностью ученика

| Ф.И.О. ученика | Критерий:
Ответственность
(выполнение ДЗ) | Критерий:
Работа в команде | Критерий:
Инициатива | Примечание |
|----------------|---|-------------------------------|-------------------------|------------|
|----------------|---|-------------------------------|-------------------------|------------|

| Ф.И.О. ученика | Критерий: Ответственность (выполнение ДЗ) | Критерий: Работа в команде | Критерий: Инициатива | Примечание |
|----------------|---|--|--|------------|
| | Всегда сдает в срок, просит доп. задания. | Активно участвует в обсуждении, помогает другим. | Предложил использовать новую библиотеку. | |
| | | | | |

Приложение №4

Оценочные материалы аттестация по итогам освоения ДОП

Аттестация по итогам освоения ДОП реализуется в форме защиты проектов.

Использование защиты проектов позволяет обеспечить условия для развития у обучающихся навыков инженерной практики (самостоятельной постановки задач, разработки алгоритма выполнения, выбора оптимального варианта решения, анализа полученных результатов с точки зрения решения поставленной задачи.)

Критерии оценки проекта:

| Ф.И. учащегося: | | | | | |
|------------------------|---|---------------------------|-------------|--------|--|
| Тема/Название проекта: | | | | | |
| Оценка: | | Комментарий преподавателя | | | |
| № | Критерий | Да(2) | Частично(1) | Нет(0) | |
| 1 | Проект соответствует заявленной теме | | | | |
| 2 | Программа запускается и выполняет основные функции без ошибок | | | | |
| 3 | Архитектура проекта логична, код разбит на модули | | | | |
| 4 | Используются сторонние библиотеки или модули (если требуется) | | | | |
| 5 | Проект демонстрирует самостоятельность и инженерный подход | | | | |
| 6 | Итого | | | | |

Каждый критерий оценивается по шкале (0/1/2 балла).

Итоговая оценка выводится как сумма баллов

Сумма баллов результатов *промежуточной* аттестации каждого года и *аттестации по итогам освоения ДОП(защиты проекта)* переводится в один из уровней освоения программы

Низкий-30%-40% от общего количества баллов ;

Средний -40%-70% от общего количества баллов;

Высокий-70%-100% от общего количества баллов.

Итоговый уровень освоения общеобразовательной общеразвивающей программы технической направленности «Программирование на Python»

| Фамилия обучающегося | Вид аттестации | | | Общая сумма баллов | Уровень освоения программы |
|----------------------|---------------------|---------------------|--|--------------------|----------------------------|
| | Промежуточная 1 год | Промежуточная 2 год | Аттестация по итогам освоения ДОП (Проект) | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |
| | | | | | |

После окончания курса «Программирование на Python» обучающимся выдается сертификат об окончании курса, для пополнения личного портфолио

Приложение № 6.

Динамика уровня сформированности познавательного интереса

«Лист наблюдения» по методике Щукиной Г.И.

| Ф.И. учащегося | | Краткая характеристика уровней сформированности познавательного интереса | | | Способы диагностики |
|----------------|----------------------------|--|---|---|---|
| | | Низкий | Средний | Высокий | |
| | Интерес к программированию | Нет интереса | Возникает иногда, объясняется несущественными причинами | Всегда или часто интересно, что объясняется существенными причинами | Методика «Лист наблюдений», Г.И.Щукина. |
| | | | | | |

Щукина Г. И. Проблема познавательного интереса в педагогике. – М.: Педагогика, 1971. – 315 с.

Низкий уровень сформированности устойчивого учебно-познавательного интереса характеризуется эпизодическим включением школьников в направление учебной деятельности и слабым проявлением качественных характеристик данного интереса.

Средний уровень сформированности устойчивого учебно-познавательного интереса школьников определяется включением их, в рефлексивно-корректировочное и частично в творческое направления учебной деятельности и более активным проявлением ведущих характеристик данного интереса.

Высокий уровень сформированности устойчивого учебно-познавательного интереса школьников характеризуется включением школьников во все направления учебной деятельности и высокой степенью проявления качественных характеристик данного