

Управление образования администрация Верхнебуреинского
муниципального района Хабаровского края
Муниципальное бюджетное учреждение дополнительного образования
Центр развития творчества детей и юношества
городского поселения «Рабочий поселок Чегдомын»
Верхнебуреинского муниципального района
Хабаровского края

Рассмотрено
на заседании НМС
Протокол №1
05.09.2025 г.



**ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ
ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА
ТЕХНИЧЕСКОЙ НАПРАВЛЕННОСТИ
«ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ PYTHON»**

Уровень усвоения: базовый
Возраст обучающихся: 12-15 лет
Срок реализации программы: 1 год

Составитель: Глызина О.И., педагог
дополнительного образования

рп. Чегдомын,
2025 г.

I. Комплекс основных характеристик

Пояснительная записка

Программа составлена в соответствии с нормативными документами:

- Федеральный закон от 29 декабря 2012 г. № 273-ФЗ «Об образовании в Российской Федерации»;
- Распоряжение Правительства Российской Федерации от 31.03.2022г. № 678-р «Концепция развития дополнительного образования детей до 2030 года»;
- Приказ Минпросвещения Российской Федерации от 27 июля 2022 г. № 629 «Об утверждении порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам»;
- Постановление Главного государственного санитарного врача Российской Федерации от 28 сентября 2020 года № 28«Об утверждении санитарных правил СП 2.4.3648-20 «Санитарно-эпидемиологические требования к организации воспитания и обучения, отдыха и оздоровления детей и молодежи»;
- Постановление Главного государственного санитарного врача Российской Федерации от 28 января 2021 года № 2«Об утверждении санитарных правил и норм СанПиН 1.2.3685-21 «Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания»;
- Распоряжение Министерства образования и науки Хабаровского края от 26.09.2019 г. № 1321 «Об утверждении методических рекомендаций «Правила персонифицированного финансирования дополнительного образования детей в городском округе, муниципальном районе Хабаровского края»;
- Методические рекомендации по проектированию дополнительных общеразвивающих программ (включая разноуровневые программы) (Письмо Министерства образования и науки Российской Федерации от 18 ноября 2015 г. № 09-3242);
- Методические рекомендации по формированию механизмов обновления содержания, методов и технологий обучения в системе дополнительного образования детей, направленных на повышение качества дополнительного образования детей, в том числе включение компонентов, обеспечивающих формирование функциональной грамотности и компетентностей, связанных с эмоциональным, физическим, интеллектуальным, духовным развитием человека, значимых для вхождения Российской Федерации в число десяти ведущих стран мира по качеству общего образования, для реализации приоритетных направлений научно-технологического и культурного развития страны (Письмо Минпросвещения Российской Федерации от 29 сентября 2023 г. № АБ-3935/06);

- Приказ краевого государственного автономного образовательного учреждения дополнительного образования «Центр развития творчества детей (региональный модельный центр дополнительного образования детей Хабаровского края)» от 27.05.2025 г. №220 П «Об утверждении Положения о дополнительной общеобразовательной общеразвивающей программе, реализуемой в Хабаровском крае»;

-Устав муниципального бюджетного учреждения дополнительного образования Центр развития творчества детей и юношества городского поселения «Рабочий поселок Чегдомын» Верхнебуреинского муниципального района Хабаровского края (далее - ЦРТДиЮ).

- Приказ Министерства образования и науки Российской Федерации и министерства просвещения Российской Федерации от 05.08.2020г. № 882/391 «Об организации и осуществлении образовательной деятельности по сетевой форме реализации образовательных программ».

Направленность ДООП: техническая.

Тип ДООП: одноуровневая.

Уровень освоения: базовый.

Актуальность программы обусловлена широким распространением информационно-коммуникационных технологий в обществе и необходимостью обеспечивать связанную с этим инфраструктуру специалистами. Всё большее значение приобретает умение человека грамотно обращаться с компьютером, причём на уровне начинающего программиста.

Изучение основных принципов программирование невозможно без регулярной практики написания программ на каком-либо языке программирования. Для обучения в рамках программы выбран язык Python, который является достаточно эффективным и доступным инструментом достижения задач в области создания программных продуктов.

Синтаксис языка достаточно прост и интуитивно понятен, что понижает порог вхождения и позволяет сосредоточиться на алгоритмических аспектах программирования.

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

Новизна программы заключается в том, что Python дает более широкие возможности в области программирования, чем Pascal, который входит в школьный курс информатики. На языке Python можно легко и быстро создавать простые компьютерные игры, трёхмерные модели и программировать роботов. Этот язык быстрее и легче усваивается, чем Pascal. Многие мировые компании такие, как Intel, Cisco, Hewlett-Packard, используют этот язык при реализации своих проектов. Крупнейшие

интернет-ресурсы такие, как Google, YouTube, также разработаны с помощью языка программирования Python.

Адресаты ДООП: учащиеся в возрасте 12-15 лет.

Форма обучения: очная.

Продолжительность академического часа: 45 минут.

Объем и сроки усвоения программы, режим занятий

Период	Продолжительность занятия	Кол-во занятий в неделю	Кол-во часов в неделю	Кол-во недель	Кол-во часов в год
1 год обучения.	2 ч.	4	2 ч.	36	144

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

Задачи программы:

Образовательные (предметные):

- Познакомить с основными алгоритмическими конструкциями и правилами их записи, с основными способами организации данных.
- Научить составлять и записывать алгоритмы с использованием соответствующих алгоритмических конструкций.
- Научить распознавать необходимость применения той или иной алгоритмической конструкции при решении задач.
- Научить организовывать данные для эффективной алгоритмической обработки.
- Научить разрабатывать алгоритмы и реализовывать их на языке программирования Python.
- Научить осуществлять отладку и тестирование программы.

Личностные:

- повышение общекультурного уровня;
- вооружение правильным методологическим подходом к познавательной и практической деятельности;
- выделение и раскрытие роли информационных технологий и компьютеров в развитии современного общества;
- привитие навыков сознательного и рационального использования компьютера в своей учебной, а затем и профессиональной деятельности;
- воспитание стремления к овладению техникой исследования;
- воспитание трудолюбия, инициативности и настойчивости в преодолении трудностей.

Метапредметные:

- владение умениями организации собственной учебной деятельности, включающими: целеполагание как постановку учебной задачи на основе соотнесения того, что уже известно, и того, что требуется установить;
- планирование – определение последовательности промежуточных целей с учетом конечного результата, разбиение задачи на подзадачи, разработка последовательности и структуры действий, необходимых для достижения цели при помощи фиксированного набора средств;
- прогнозирование – предвосхищение результата.

Учебный план

№	Название раздела	Кол-во часов			Формы промежуточного контроля
		Всего	Теория	Практика	
	Модуль 1. Базовые конструкции в Python	64	12	52	
1.1	Вводное занятие. Инструктаж по ТБ. Введение в программу.	2	2		
1.2	Интегрированные среды, исполнение кода и отладка	8	2	6	
1.3	Переменные, основные операторы	8	2	6	
1.4	Базовые типы данных, ветвления	8	2	6	
1.5	Циклы, срезы, списочные выражения	8	2	6	
1.6	Методы списков и строк. Функции	8	2	6	
1.7.	Решение задач по пройденным темам. Практическая работа	20		20	
1.8	Итоговое занятие	2		2	Тест. Решение задач
	Модуль 2. Решение прикладных задач в Python.	80	20	60	
2.1	Функциональный стиль программирования, лямбда функции, модуль functools	6	2	4	
2.2	Хэш-таблицы. Модуль Collections	6	2	4	
2.3	Обработка исключений	6	2	4	
2.4	Принципы устройства и механика создания модулей и пакетов	6	2	4	
2.5	ООП: инкапсуляция, наследование, полиморфизм	6	2	4	
2.6	Основы объектно-ориентированного анализа и дизайна, шаблоны проектирования	6	2	4	
2.7	Итераторы и генераторы	6	2	4	
2.8	Автоматизированное тестирование в python	6	2	4	
2.9	Tkinter: создание графического интерфейса	6	2	4	
2.10	Голосовые помощники и чат боты. Создание навыка для Алисы Яндекса	6	2	4	

2.11	Решение задач по пройденным темам. Практическая работа	18	2	16	
2.12	Итоговое занятие	2		2	Защита творческого проекта
Итого		144 ч	32	112	

Содержание учебного плана

Модуль 1. Базовые конструкции в Python

Тема 1.1 Вводное занятие.

Инструктаж по ТБ. Введение в программу

Теория: Инструктажи по технике безопасности с отметкой в журнале. Введение в программу «Программирование на Python». Понятия кода, интерпретатора, программы. Простейшие программы с выводом на экран. Знакомство с виртуальной средой взаимодействия - регистрация, организация личного кабинета, поиск и выкладывание материалов; знакомство с системой автоматизированной проверки задач и системой Яндекс.

Практика: решение задач.

Тема 1.2 Интегрированные среды, исполнение кода и отладка.

Теория: интегрированные среды, исполнение кода. Основные понятия программирования: исполнитель, система команд, алгоритм, программа, среда разработки.

Практика: решение задач.

Тема 1.3. Переменные, основные операторы

Теория: условный оператор. Переменные и арифметика. Погружение в условия. Ввод-вывод в программе, условный оператор, оператор цикла с предусловием. Простейшие программы с использованием условного оператора if, оператора циклов while и операторов ввода-вывода. Технология разработки программы.

Практика: решение задач.

Тема 1.4. Базовые типы данных, ветвления.

Теория: знакомство со списками, строками, множествами и кортежами в Python.

Практика: разработка алгоритмов и программ, определение работоспособности разработанной программы.

Тема 1.5 Циклы, срезы, списочные выражения.

Теория: понятие о языке высокого уровня Python. Структура программы, переменные и константы, работа с числовыми переменными, арифметические операторы в Python. Основные управляющие конструкции алгоритмов с ветвлением в Python. Знакомство со срезами и диапазонами. Равенство и совпадение объектов. Устройство циклов for. Основные управляющие конструкции циклического алгоритма в Python. Простейшие циклы и циклы с переменными.

Практика: решение задач по теме «Срезы и диапазоны».

Тема 1.6 Методы списков и строк. Функции.

Теория: списочные выражения. Методы `split` и `join`. Другие методы списков и строк. Знакомство с функциями. Области видимости переменных. Работа со

списками, строками, множествами и кортежами в Python. Понятие итератора, подпрограммы, процедуры, функции. Функции в Python.

Практика: решение задач.

Тема 1.7 Решение задач по пройденным темам. Практическая работа

Теория: повторение пройденных тем. Решение задач по пройденным темам. Анализ результатов.

Практика: практическая работа.

Тема 1.8 Итоговое занятие

Подведение итогов по темам первого модуля. Тестирование.

Модуль 2. Решение прикладных задач в Python

Тема 2.1 Функциональный стиль программирования, лямбда-функции, модуль `functools`

Теория: функции, функциональная парадигма программирования.

Понятие лямбда функции и области ее применения. Функции модуля `functools`.

Практика: решение задач.

Тема 2.2 Хэш-таблицы. Модуль `Collections`

Теория: Словари и множества. Подключение и использование модулей стандартной библиотеки Python. Функции модуля `Collections`.

Практика: решение задач.

Тема 2.3 Обработка исключений

Теория: Понятие исключения, синтаксис их обработки.

Применение

исключений при разработке и отладке программ.

Практика: решение задач.

Тема 2.4 Принципы устройства и механика создания модулей и пакетов.

Теория: Понятие модуля и пакета, размещение и импорт модуля.

Практика: создание собственного пакета модулей.

Тема 2.5 ООП: инкапсуляция, наследование, полиморфизм.

Теория: Причины появления, принципы и основные сущности объектно-ориентированного подхода к разработке ПО. Инкапсуляция, полиморфизм, наследование, композиция.

Практика: решение задач.

Тема 2.6 Основы объектно-ориентированного анализа и дизайна, шаблоны проектирования.

Теория: Восходящий и нисходящий способы разработки.

Методика

объектно-ориентированного дизайна приложения. Шаблоны проектирования.

Практика: проектирование и реализация компьютерной игры.

Тема 2.7 Итераторы и генераторы

Теория: Понятие и реализация итераторов и генераторов. Предназначение, особенности устройства и работы, типовые сферы применения.

Практика: решение задач.

Тема 2.8 Автоматизированное тестирование в python.

Теория: Виды тестирования и наиболее распространенные системы и подходы автоматического тестирования. Модульные тесты.

Практика: написание серии модульных тестов для выбранной задачи.

Тема 2.9 Tkinter: создание графического интерфейса.

Теория: Знакомство с библиотекой tkinter, изучение основных команд, виджетов, упаковщиков и приемов работы.

Практика: создание приложения с графическим интерфейсом.

Тема 2.10 Голосовые помощники и чат боты. Создание навыка для Алисы Яндекса.

Теория: Изучение общих принципов работы голосовых помощников.

Знакомство с платформой Яндекс. Диалоги и протоколом работы навыка.

Практика: изучение документации на платформу, создание навыка для Алисы.

Тема 2.11 Решение задач по пройденным темам. Практическая работа.

Теория: Повторение пройденных тем. Решение задач по пройденным темам. Анализ результатов.

Практика: Разработка проектов. Тема 2.12 Итоговое занятие
Подведение итогов по темам второго модуля. Тестирование.

1.4. Планируемые результаты

Реализация программы предполагает достижение обучающимися следующих личностных, метапредметных и предметных результатов:

Предметные:

- умение определять результат выполнения алгоритма при заданных исходных данных, узнавать
- изученные алгоритмы обработки чисел и числовых последовательностей, создавать на их основе несложные программы анализа данных, читать и понимать несложные программы, написанные на выбранном для изучения универсальном алгоритмическом языке высокого уровня;
- формирование представлений об основных предметных понятиях («информация», «алгоритм», «исполнитель», «модель») и их свойствах;
- развитие логических способностей и алгоритмического мышления, умения составить и записать алгоритм для конкретного исполнителя,

знакомство с основными алгоритмическими структурами — линейной, условной и циклической;

- умение выполнять пошагово (с использованием компьютера или вручную) несложные алгоритмы управления исполнителями и анализа числовых и текстовых данных;
- навыки и опыт разработки программ в выбранной среде программирования, включая тестирование и отладку программ;
- умение использовать основные управляющие конструкции объектно-ориентированного программирования и библиотеки прикладных программ, выполнять созданные программы;
- умение разрабатывать и использовать компьютерно-математические модели, оценивать числовые параметры моделируемых объектов и процессов, интерпретировать результаты, получаемые в ходе моделирования реальных процессов, анализировать готовые модели на предмет соответствия реальному объекту или процессу;
- формирование умения соблюдать нормы информационной этики и права.

Метапредметные:

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

Личностные:

- формирование ответственного отношения к учению, способности довести до конца начатое дело аналогично завершённым творческим учебным проектам;
- формирование способности к саморазвитию и самообразованию средствами информационных технологий на основе приобретённой благодаря иллюстрированной среде программирования мотивации к обучению и познанию;

- развитие опыта участия в социально значимых проектах, повышение уровня самооценки благодаря реализованным проектам;
- формирование коммуникативной компетентности в общении и сотрудничестве со сверстниками в процессе образовательной, учебно-исследовательской и проектной деятельности, участия в конкурсах и конференциях различного уровня;
- формирование целостного мировоззрения, соответствующего современному уровню развития информационных технологий;
- формирование осознанного позитивного отношения к другому человеку, его мнению, результату его деятельности;
- формирование ценности здорового и безопасного образа жизни; усвоение правил индивидуального и коллективного безопасного поведения при работе с компьютерной техникой.

II Комплекс организационно-педагогических условий

Календарный учебный график «Программирование на языке Python» техническая направленность

Год обучения (уровень)	Дата начала занятий	Дата окончания занятий	Количество учебных недель	Количество учебных дней	Количество учебных часов	Режим занятий
1 год обучения	01.09. 2025 г.	31.05. 2026 г.	36	72	144	2 раз в неделю по 2 ч.

Формы проведения занятий: лекция, практическое занятие.

Материально-техническое обеспечение

Перечень оборудования и материалов:

- Кабинет/учебный класс,
- Мультимедиа оборудование: Smart панель,
- Столы и стулья,
- Доступ в сеть Интернет,
- Ноутбуки.
- Программное обеспечение.
- Учебная литература.
- Раздаточный материал.

Формы контроля

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

Форма контроля: тестирование.

Текущий контроль осуществляется после изучения отдельных тем, раздела программы. В практической деятельности результативность оценивается качеством выполнения практических

работ, поиску и отбору необходимого материала, умению работать с различными источниками информации. Анализируются положительные и отрицательные стороны работы, корректируются недостатки. Контроль знаний осуществляется с помощью заданий педагога (решение практических задач средствами языка программирования); взаимоконтроля, самоконтроля и др. Они активизируют, стимулируют работу учащихся, позволяют более полно проявлять полученные знания, умения, навыки.

Промежуточный контроль осуществляется в конце I полугодия учебного года в форме теста, решения практических задач средствами языка программирования и в конце учебного года в форме защиты творческого проекта.

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

Оценочные материалы

Входящий контроль: тестирование (Приложение 2)

Промежуточный контроль: тестирование, решение задач (Приложение 3)

Методическое обеспечение

Учебно-воспитательный процесс направлен на развитие природных задатков обучающихся, на реализацию их интересов и способностей. Каждое занятие обеспечивает развитие личности. При планировании и проведении занятий применяется личностноориентированная технология обучения, в центре внимания которой неповторимая личность, стремящаяся к реализации своих возможностей, а также системно-деятельностный метод обучения. Данная программа допускает творческий, импровизированный подход со стороны детей и педагога того, что касается возможной замены порядка раздела, введения дополнительного материала, методики проведения занятий. Руководствуясь данной программой, педагог имеет возможность увеличить или уменьшить объем и степень технической сложности материала в зависимости от состава группы и конкретных условий работы.

Методы обучения:

Выбор метода обучения зависит от содержания занятий, уровня подготовки и опыта обучающихся.

1. Традиционные методы развития интереса обучающихся к техническому творчеству можно объединить в группы, каждая из которых наиболее продуктивна на том или ином этапе работы. 1. Словесный: объяснение нового материала; рассказ обзорный для раскрытия новой темы; беседы с учащимися в процессе изучения темы.

2. Наглядный: применение демонстрационного материала, наглядных пособий, презентаций по теме.

3. Практический: индивидуальная и совместная продуктивная деятельность, выполнение учащимися определенных заданий, решение задач.

4. Интерактивный: создание специальных заданий, моделирующих реальную жизненную ситуацию, из которой учащимся предлагается найти выход.

Педагогические технологии, применяемые в реализации программы:

1. Технология проблемного диалога. Учащимся не только сообщаются готовые знания, но и организуется такая их деятельность, в процессе которой они сами делают «открытия», узнают что-то новое и используют полученные знания и умения для решения жизненных задач.

2. Технология коллективного взаимообучения («организованный диалог», «сочетательный диалог», «коллективный способ обучения (КСО), «работа учащихся в парах сменного состава») позволяет плодотворно развивать у обучаемых самостоятельность и коммуникативные умения.

3. Игровая технология. Игровая форма в образовательном процессе создаётся при помощи игровых приёмов и ситуаций, выступающих как средство побуждения к деятельности. Способствует развитию творческих способностей, продуктивному сотрудничеству с другими учащимися. Приучает к коллективным действиям, принятию решений, учит руководить и подчиняться, стимулирует практические навыки, развивает воображение.

4. Элементы здоровьесберегающих технологий являются необходимым условием снижения утомляемости и перегрузки учащихся.

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

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

Формы организации учебного занятия: комбинированное занятие; беседа; игра; викторина, творческая встреча, соревнование, экскурсия и др.

Алгоритм учебного занятия.

И этап - организационный.

Задача: подготовка детей к работе на занятии.

Содержание этапа: организация начала занятия, создание психологического настроя на учебную деятельность и активизация внимания.

II этап - подготовительный (подготовка к восприятию нового содержания).

Задача: Организовать занимающихся, сообщение задач занятия, подготовка к основной части занятия, к предстоящей работе. Содержание этапа: сообщение темы, цели учебного занятия и мотивация учебной

деятельности детей (пример, познавательная задача, проблемное задание детям).

III этап - основной.

В качестве основного этапа могут выступать следующие:

1. Усвоение новых знаний и способов действий.

Задача: обеспечение восприятия, осмысливания и первичного запоминания связей и отношений в объекте изучения. Целесообразно при усвоении новых знаний использовать задания и вопросы, которые активизируют познавательную деятельность детей.

2. Первичная проверка понимания.

Задача: установление правильности и осознанности усвоения нового учебного материала, выявление неверных представлений, их коррекция. Применяют пробные практические задания, которые сочетаются с объяснением соответствующих правил или обоснованием.

3. Закрепление знаний и способов действий, применяют творческие задания, выполняемые детьми самостоятельно.

4. Обобщение и систематизация знаний.

Задача: формирование целостного представления знаний по теме.

Распространенными способами работы являются беседа и практические задания.

IV этап – контрольный.

Задача: выявление качества и уровня овладения знаниями, их коррекция.

Используются виды устного и письменного опроса, вопросы и задания различного уровня сложности (репродуктивного, творческого, поисково-исследовательского).

V этап - итоговый.

Задача: дать анализ и оценку успешности достижения цели и наметить перспективу последующей работы.

Содержание этапа: педагог сообщает, как работали обучающиеся на занятии, что нового узнали, какими умениями и навыками овладели.

VI этап - рефлексивный.

Задача: мобилизация детей на самооценку. Может оцениваться работоспособность, психологическое состояние, результативность работы, содержание и полезность учебной работы.

Изложенные этапы могут по-разному комбинироваться, какие-либо из них могут не иметь места в зависимости от педагогических целей и учитывая формы занятия.

Дидактические материалы:

- таблицы;
- схемы;
- памятки;
- видеозаписи, мультимедийные материалы;
- упражнения.

Список источников

Литература для обучающихся:

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

Литература для педагога

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

Календарный учебный график

№ п/п	Месяц	Форма проведения	Кол-во часов	Тема занятия	Форма контроля
1	Сентябрь	Лекция	2	Вводное занятие. Инструктаж по ТБ. Введение в программу.	Опрос
2	Сентябрь	Комбинированное занятие	2	Интегрированные среды, исполнение кода и отладка	Опрос, беседа, выполнение мини-проекта
3	Сентябрь	Практическое занятие	2	Интегрированные среды, исполнение кода и отладка	Опрос, беседа, выполнение мини-проекта
4	Сентябрь	Практическое занятие	2	Интегрированные среды, исполнение кода и отладка	Опрос, беседа,
5	Сентябрь	Практическое занятие	2	Интегрированные среды, исполнение кода и отладка	Опрос, беседа, выполнение мини-проекта
6	Сентябрь	Комбинированное занятие	2	Переменные, основные операторы	Опрос, беседа, выполнение мини-проекта
7	Сентябрь	Комбинированное занятие	2	Переменные, основные операторы	Опрос, беседа, выполнение мини-проекта
8	Сентябрь	Комбинированное занятие	2	Переменные, основные операторы	Опрос, беседа, выполнение мини-проекта
9	Октябрь	Комбинированное занятие	2	Переменные, основные операторы	Опрос, беседа, выполнение мини-проекта
10	Октябрь	Комбинированное занятие	2	Базовые типы данных, ветвления	Беседа. Опрос

11	Октябрь	Комбинированное занятие	2	Базовые типы данных, ветвления	Беседа. Опрос
12	Октябрь	Комбинированное занятие	2	Базовые типы данных, ветвления	Беседа. Опрос
13	Октябрь	Комбинированное занятие	2	Базовые типы данных, ветвления	Беседа. Опрос
14	Октябрь	Комбинированное занятие	2	Циклы, срезы, списочные выражения	Опрос, беседа, выполнение мини-проекта
15	Октябрь	Комбинированное занятие	2	Циклы, срезы, списочные выражения	Опрос, беседа, выполнение мини-проекта
16	Октябрь	Комбинированное занятие	2	Циклы, срезы, списочные выражения	Опрос, беседа, выполнение мини-проекта
17	Ноябрь	Комбинированное занятие	2	Циклы, срезы, списочные выражения	Опрос, беседа, выполнение мини-проекта
18	Ноябрь	Комбинированное занятие	2	Методы списков и строк. Функции	Опрос, беседа, выполнение мини-проекта
19	Ноябрь	Комбинированное занятие	2	Методы списков и строк. Функции	Опрос, беседа, выполнение мини-проекта
20	Ноябрь	Комбинированное занятие	2	Методы списков и строк. Функции	Опрос, беседа, выполнение мини-проекта
21	Ноябрь	Комбинированное занятие	2	Методы списков и строк. Функции	Опрос, беседа, выполнение мини-проекта
22	Ноябрь	Комбинированное занятие	2	Решение задач по пройденным темам. Практическая работа	Практическая работа
23	Ноябрь	Комбинированное занятие	2	Решение задач по пройденным темам. Практическая работа	Практическая работа

24	Ноябрь	Комбинированное занятие	2	Решение задач по пройденным темам. Практическая работа	Практическая работа
25	Декабрь	Комбинированное занятие	2	Решение задач по пройденным темам. Практическая работа	Практическая работа
26	Декабрь	Комбинированное занятие	2	Решение задач по пройденным темам. Практическая работа	Практическая работа
27	Декабрь	Комбинированное занятие	2	Решение задач по пройденным темам. Практическая работа	Практическая работа
28	Декабрь	Комбинированное занятие	2	Решение задач по пройденным темам. Практическая работа	Практическая работа
29	Декабрь	Комбинированное занятие	2	Решение задач по пройденным темам. Практическая работа	Практическая работа
30	Декабрь	Комбинированное занятие	2	Решение задач по пройденным темам. Практическая работа	Практическая работа
31	Декабрь	Комбинированное занятие	2	Решение задач по пройденным темам. Практическая работа	Практическая работа
32	Декабрь	Комбинированное занятие	2	Итоговое занятие	Практическая работа
33	Январь	Комбинированное занятие	2	Функциональный	Опрос, беседа, выполнение мини-проекта
34	Январь	Комбинированное занятие	2	Функциональный	Опрос, беседа, выполнение мини-проекта
35	Январь	Комбинированное занятие	2	Функциональный	Лекция, демонстрация, выполнение мини-проекта
36	Январь	Комбинированное занятие	2	Хэш-таблицы. Модуль Collections	Практическая работа
37	Январь	Комбинированное занятие	2	Хэш-таблицы. Модуль Collections	Практическая работа
38	Январь	Комбинированное занятие	2	Хэш-таблицы. Модуль Collections	Практическая работа

39	Январь	Комбинированное занятие	2	Обработка исключений	Практическая работа
40	Январь	Комбинированное занятие	2	Обработка исключений	Практическая работа
41	Февраль	Комбинированное занятие	2	Обработка исключений	Практическая работа
42	Февраль	Комбинированное занятие	2	Принципы устройства и механика создания модулей и пакетов.	Беседа, выполнение мини-проекта
43	Февраль	Комбинированное занятие	2	Принципы устройства и механика создания модулей и пакетов.	Беседа, выполнение мини-проекта
44	Февраль	Комбинированное занятие	2	Принципы устройства и механика создания модулей и пакетов.	Беседа, выполнение мини-проекта
45	Февраль	Комбинированное занятие	2	ООП: инкапсуляция, наследование, полиморфизм.	Практическая работа
46	Февраль	Комбинированное занятие	2	ООП: инкапсуляция, наследование, полиморфизм.	Практическая работа
47	Февраль	Комбинированное занятие	2	ООП: инкапсуляция, наследование, полиморфизм	Практическая работа
48	Февраль	Комбинированное занятие	2	Основы объектноориентированного анализа и дизайна, шаблоны проектирования.	Опрос, беседа, выполнение мини-проекта
49	Март	Комбинированное занятие	2	Основы объектноориентированного анализа и дизайна, шаблоны проектирования.	Опрос, беседа, выполнение мини-проекта
50	Март	Комбинированное занятие	2	Основы объектноориентированного анализа и дизайна, шаблоны проектирования.	Опрос, беседа, выполнение мини-проекта
51	Март	Комбинированное занятие	2	Итераторы и генераторы	Практическая работа
52	Март	Комбинированное занятие	2	Итераторы и генераторы	Практическая работа
53	Март	Комбинированное занятие	2	Итераторы и генераторы	Практическая работа

54	Март	Комбинированное занятие	2	Автоматизированное тестирование в python.	Практическая работа
55	Март	Комбинированное занятие	2	Автоматизированное тестирование в python.	Практическая работа
56	Март	Комбинированное занятие	2	Автоматизированное тестирование в python.	Практическая работа
57	Апрель	Комбинированное занятие	2	Tkinter: создание графического интерфейса.	Опрос, беседа, выполнение мини-проекта
58	Апрель	Комбинированное занятие	2	Tkinter: создание графического интерфейса.	Опрос, беседа, выполнение мини-проекта
59	Апрель	Комбинированное занятие	2	Tkinter: создание графического интерфейса	Опрос, беседа, выполнение мини-проекта
60	Апрель	Комбинированное занятие	2	Голосовые помощники и чат боты. Создание навыка для Алисы Яндекса.	Опрос, беседа, выполнение мини-проекта
61	Апрель	Комбинированное занятие	2	Голосовые помощники и чат боты. Создание навыка для Алисы Яндекса.	Опрос, беседа, выполнение мини-проекта
62	Апрель	Комбинированное занятие	2	Голосовые помощники и чат боты. Создание навыка для Алисы Яндекса.	Опрос, беседа, выполнение мини-проекта
63	Апрель	Комбинированное занятие	2	Решение задач по пройденным темам. Практическая работа	Практическая работа
64	Апрель	Комбинированное занятие	2	Решение задач по пройденным темам. Практическая работа	Практическая работа
65	Май	Комбинированное занятие	2	Решение задач по пройденным темам. Практическая работа	Практическая работа
66	Май	Комбинированное занятие	2	Решение задач по пройденным темам. Практическая работа	Практическая работа
67	Май	Комбинированное занятие	2	Решение задач по пройденным темам. Практическая работа	Практическая работа

68	Май	Комбинированное занятие	2	Решение задач по пройденным темам. Практическая работа	Практическая работа
69	Май	Комбинированное занятие	2	Решение задач по пройденным темам. Практическая работа	Практическая работа
70	Май	Комбинированное занятие	2	Решение задач по пройденным темам. Практическая работа	Практическая работа
71	Май	Комбинированное занятие	2	Решение задач по пройденным темам. Практическая работа	Практическая работа
72	Май	Комбинированное занятие -	2	Итоговое занятие	Практическая работа

Здоровьесберегающие технологии

На занятиях нужно систематически проводить гимнастику для глаз, физкультминутки, выполнять упражнения для пальцев, рук, плеч. Преподаватель должен постоянно напоминать обучающимся о правильной посадке, следить за положением рук, спины, ног.

Примерный комплекс упражнений для глаз:

1. Закрыть глаза, сильно напрягая глазные мышцы, на счет 1 - 4, затем открыть глаза, расслабить мышцы глаз, посмотреть вдаль на счет 1-6. Повторить 4-5 раз.

2. Посмотреть на переносицу и задержать взгляд на счет 1 - 4. До усталости глаза не доводить. Затем посмотреть вдаль на счет 1 - 6. Повторить 4 - 5 раз.

3. Не поворачивая головы, посмотреть направо и зафиксировать взгляд на счет 1 - 4, затем посмотреть вдаль прямо на счет 1 - 6. Аналогичным образом проводятся упражнения с фиксацией взгляда влево, вверх и вниз. Повторить 3 - 4 раза.

4. Перевести взгляд быстро по диагонали: направо вверх - налево вниз, потом прямо вдаль на счет 1 - 6; затем налево вверх - направо вниз и прямо вдаль на счет 1-6. Повторить 4 - 5 раз.

Создание комфортного психологического климата, учет индивидуальных особенностей обучающихся, их темперамента, умение работать с различными группами обучающихся, использовать дифференцированный подход в обучении.

Необходимо создавать благоприятный эмоциональный климат на занятиях. Чувство успеха при выполнении заданий, очень положительно влияет на здоровье обучающихся. Проявление доверительного подхода к обучающимся, наличие на занятиях эмоциональной разрядки, создание ситуации успеха – оптимальные приемы создания положительных эмоций на занятии.

И последним, но немаловажным компонентом здоровье сберегающих технологий является формирование, пропаганда здорового образа жизни. Если показывать обучающимся, как соотносится изучаемый материал с повседневной жизнью, приучать их постоянно заботиться о своем здоровье, систематически закреплять их на занятиях, это должно стать для них совершенно естественным и при работе в домашних условиях.

Усталости и перегрузок на занятиях будет меньше, если будет чередоваться и вид деятельности обучающихся, и способы преподнесения информации (зрительная, слуховая) с физкультурными паузами. А создание благоприятной атмосферы в начале и в конце занятия, через улыбку, будет способствовать хорошему настроению обучающихся, как на самом занятии, так и после него.

Приложение 2

Входящий контроль

Низкий уровень: 0-70% выполненных заданий;

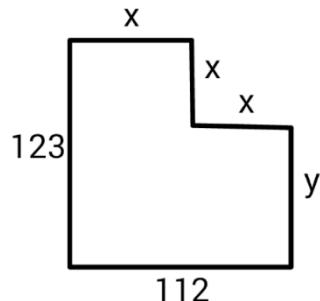
Средний уровень: 70-85% правильно выполненных заданий;

Высокий уровень: 85-100% правильно выполненных заданий.

Для учащихся – 15-18 лет

Фамилия, Имя

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



2. В алфавите племени

мумба-юмба 32 буквы.

Любое слово в языке этого племени состоит из пяти букв и должно одинаково читаться справа налево и слева направо, при этом первые две буквы слова обязательно различаются, а третья совпадает с пятой.

Каково максимальное количество слов в этом языке?

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

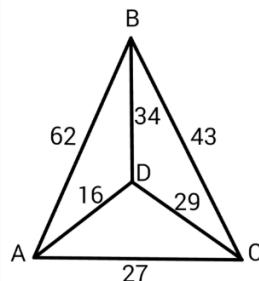
Сколько кусков рулета получится, если разрезать по кольцам всех трёх цветов?

Примечания

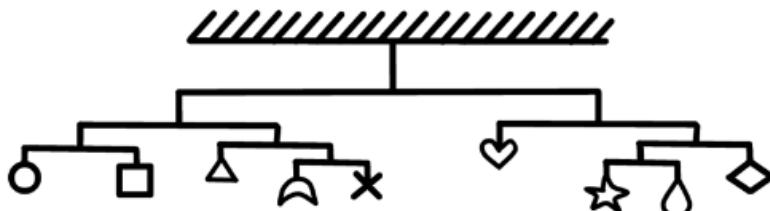
Учтите, что наклеить кольцо одного цвета на кольцо другого нельзя.

4. На рисунке показано расположение городов А, В, С и D и расстояния между ними. Турист выходит из города В и собирается посетить остальные города, побывав в каждом по разу.

Какова наименьшая возможная длина маршрута, если он хочет закончить свой путь в том же городе?



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



6. Люди переезжают в города, за год численность людей удваивается.

Если люди заселят весь город за 12 лет, то сколько лет понадобится, чтобы занять лишь половину города?

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

O/Sanmarco

O/Samnarco

Ф.Wagonerrte

Ф.Wagonertre

A.S.Schmetterling

A.S.Schnetterling

N.V.Murfreesboroque

N.V.Munfreesboroque

P.S.Splendoursec

P.S.Sqlendoursec

Семь человек выясняли, какой сегодня день недели.

Первый сказал: «Послезавтра – воскресенье».

Второй: «Вчера был понедельник».

Третий: «Завтра будет суббота».

Четвертый: «Завтра будет среда».

Пятый: «Вчера был четверг».

Шестой: «Позавчера было воскресенье».

Седьмой: «Позавчера была среда».

Какой сегодня день недели, если трое ошибаются?

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

Согласны ли Вы с этим следствием?

1. Все клёны — растения.

2. Некоторые растения быстро желтеют.

Значит, некоторые клёны быстро желтеют.

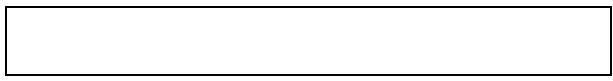
Да

Нет

9. Гусеница прогрызает яблоко диаметром 6 сантиметров насквозь за 16 секунд, вылезая снаружи полностью.

Известно, что середину яблока она начинает грызть уже через 6 секунд после начала пути.

Какова длина гусеницы в сантиметрах?



Для учащихся 14-15 лет

Фамилия, Имя.	
----------------------	--

1. Назовите два числа, у которых количество цифр равно количеству букв, составляющих название каждого из этих чисел.

--	--

2. Собака была привязана к десятиметровой веревке, а прошла двести метров. Как ей это удалось?

--

3. Есть дорога, по которой может проехать только одна машина. По дороге едут две машины: одна с горы, другая под гору. Как им разъехаться?

--

4. Как с помощью только одной палочки образовать на столе треугольник?

--

5. В каком месяце болтливая девочка говорит меньше всего?

--

6. Что становится на треть больше, если его поставить вверх ногами?

--

7. Представьте себе, что вы кондуктор. Поезд везет сто вагонов, в каждом вагоне 10 купе, в каждом купе 4 пассажира. Сколько лет кондуктору?

--

8. Юра разрезал огромную пиццу на 10 кусков. Затем он взял один из кусков и разрезал его еще на 10. После этого из имеющихся кусков он выбрал два и разрезал каждый из них на 10.

Вопрос: Сколько кусков пиццы получилось у Юры.

--

9. В компьютерной игре нужно победить монстра. Изначально у Юры было только 9 выстрелов. Но за каждое попадание он получал дополнительно еще 3 выстрела.

- Вопрос:** Сколько раз Юра попал в цель, если всего он выстрелил 30 раз, израсходовав все выстрелы?

--

Итоговый контроль
Диагностическая карта

№ / п	Имя	Входящая диагностика	Промежуточная диагностика	Итоговая диагностика
1				
2				
3				
+				
	Итого в %	H C B		H C B

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

Низкий уровень

Учащиеся должны знать/ понимать:

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

Учащиеся должны уметь:

- понимать учебную задачу, сохранять ее содержание в процессе ее выполнения под

руководством педагога;

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

Средний уровень

Учащиеся должны знать/ понимать:

- основные положения техники безопасности на занятиях, правила поведения в общественных местах, правила дорожной безопасности;
- основные приемы взаимодействия в группе сверстников;
- понятие программы;
- структуру программы на Python;
- режимы работы с Python.
- типы данных;
- целые, вещественные типы данных и операции над ними;
- оператор присваивания;
- назначение условного оператора;
- способ записи условного оператора;
- логический тип данных;
- логические операторы or, and, not;
- циклы с условием и их виды;
- назначение и особенности использования цикла с параметром;
- формат записи цикла с параметром;
- примеры использования циклов различных типов.
- понятие функции;
- основные способы описания функции;
- принципы структурного программирования;
- понятие локальных переменных подпрограмм;
- понятие формальных и фактических параметров подпрограмм;
- способ передачи параметров.
- назначение строкового типа данных;

- операторы для работы со строками;
- процедуры и функции для работы со строками;
- операции со строками;
- сложные типы данных;
- способ описания списка;
- способ доступа к элементам списка;
- способ описания кортежа;
- способ описания словаря;
- операции, выполняемые со списками, кортежами и словарями;
- понятие множества;
- способы описания множества;
- операторы работы с множествами.
- что такое стиль программирования;
- правила именования объектов;
- основные рекомендации при написании программ;
- основные шаги работы над проектом, его презентации.

Учащиеся должны уметь:

- уважительно относится к преподавателям и сверстникам;
- применять некоторые приемы логического (абстрактное) мышления;
- концентрировать внимание на одном или двух объектах;
- понимать причины успеха/неуспеха с помощью анализа педагога;
- выполнить установку программы;
- выполнить простейшую программу в интерактивной среде;
- написать комментарии в программе;
- решать задачи на элементарные действия с числами;
- использовать условный оператор;
- создавать сложные условия с помощью логических операторов;
- определять вид цикла, наиболее удобный для решения поставленной задачи;
- использовать цикл с условием;
- определять целесообразность применения и использования цикла с параметром для решения поставленной задачи;
- создавать и использовать функции;
- использовать механизм параметров для передачи значений;
- описывать строки;
- соединять строки;
- находить длину строки;
- вырезать часть строки;
- находить подстроку в строке;
- находить количество слов в строке;
- описывать списки;
- вводить элементы списка;
- выводить элементы списка;
- выполнять поиск элемента в списке, поиск минимума и максимума, нахождение суммы элементов списка;
- использовать вложенные списки;
- приводить примеры использования вложенных списков (матриц);
- описывать множества;
- определять принадлежность элемента множеству;
- вводить элементы множества;
- выводить элементы множества.
- определять вид ошибок и находить ошибки в программе.
- составлять алгоритмы для решения задач;

- реализовывать алгоритмы на компьютере в виде программ, написанных на языке Python;
- отлаживать и тестировать программы, написанные на языке Python;
- понимать и сохранять цели и задачи учебной деятельности в ходе занятия;
- планировать свою деятельность с помощью взрослого;
- сотрудничать с взрослыми и сверстниками в процессе выполнения учебной задачи;
- понимать причины успеха/неуспеха учебной деятельности под руководством взрослого;
- делать выводы в ходе полученных заданий; выстраивать логические цепи рассуждений под руководством педагога;
- выражать творческие идеи, разработать творческий проект на основе образца;
- конструктивно взаимодействовать в составе группы в ходе работы над проектом.

Высокий уровень

Учащиеся должны знать/ понимать:

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

- способы описания множества;
- операторы работы с множествами;
- что такое стиль программирования;
- правила именования объектов;
- основные рекомендации при написании программ;
- правила и этапы работы над проектом;
- приемы успешной презентации проекта.

Учащиеся должны уметь:

- выполнить установку программы;
- выполнить простейшую программу в интерактивной среде;
- написать комментарии в программе;
- решать задачи на элементарные действия с числами;
- использовать условный оператор;
- создавать сложные условия с помощью логических операторов;
- определять вид цикла, наиболее удобный для решения поставленной задачи;
- использовать цикл с условием;
- определять целесообразность применения и использования цикла с параметром для решения поставленной задачи;
- создавать и использовать функции;
- использовать механизм параметров для передачи значений;
- описывать строки;
- соединять строки;
- находить длину строки;
- вырезать часть строки;
- находить подстроку в строке;
- находить количество слов в строке;
- описывать списки;
- вводить элементы списка;
- выводить элементы списка;
- выполнять поиск элемента в списке, поиск минимума и максимума, нахождение суммы элементов списка;
- использовать вложенные списки;
- приводить примеры использования вложенных списков (матриц);
- описывать множества;
- определять принадлежность элемента множеству;
- вводить элементы множества;
- выводить элементы множества.
- определять вид ошибок и находить ошибки в программе.
- составлять алгоритмы для решения задач;
- реализовывать алгоритмы на компьютере в виде программ, написанных на языке Python;
- отлаживать и тестировать программы, написанные на языке Python;
- понимать учебную задачу, анализировать достижение результата;
- делать выводы в ходе полученных заданий; самостоятельно выстраивать логические цепи рассуждений;
- понимать причины успеха/неуспеха, конструктивно принимать причины неуспеха, исправлять ошибки с минимальным участием педагога;
- планировать свою деятельность, находить оригинальные способы выполнения поставленной творческой задачи;
- создать проектировочную команду и организовать ее деятельность;
- разработать творческий проект по предложенной или самостоятельно выбранной теме в составе творческой группы;

- самостоятельно подготовить презентацию и защитить проект.

Вопросы для промежуточного контроля по усвоению материала
(Промежуточный контроль проводится в последнюю неделю декабря)

Составьте выражение для вычисления в интерпретаторе Python 3 и вставьте в поле ответа результат вычисления:

11111·1111111

— произведение чисел 11111 (5 единиц) и 1111111 (7 единиц)

Запишите число **1.2345e3** в виде десятичной дроби.

Составьте и запишите выражение для вычисления:

2014.0^14(Возвести в 14 степень)

Обратите внимание на запись числа: это вещественное число.

Приведите к целому типу число 2.99

Расставьте скобки в выражении

a and b or not a and not b

в соответствии с порядком вычисления выражения (приоритетом операций).

Всего потребуется 5 пар скобок (внешние скобки входят в их число).

Найдите результат выражения для заданных значений *a* и *b*

Учитывайте регистр символов при ответе.

a = **True**

b = **False**

a and b or not a and not b

Отметьте выражения, значения которых равны True:

- "239" < "30" and 239 < 30
- "239" < "30" and 239 > 30
- "239" > "30" and 239 < 30
- "239" > "30" and 239 > 30

Укажите результат выражения:

"123" + "42"

Какое значение будет у переменной *i* после выполнения фрагмента программы?

i = 0

while i <= 10:

i = i + 1

if i > 7:

i = i + 2

Сколько итераций цикла будет выполнено в этом фрагменте программы?

i = 0

while i <= 10:

i = i + 1

if i > 7:

i = i + 2

Сколько всего знаков * будет выведено после исполнения фрагмента программы:

i = 0

```
while i <5:  
    print('*')  
    if i % 2 == 0:  
        print('**')  
    if i >2:  
        print('***')  
    i = i + 1
```

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

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

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

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

Задачи для промежуточного контроля по усвоению материала

Напишите простой калькулятор, который считывает с пользовательского ввода три строки: первое число, второе число и операцию, после чего применяет операцию к введённым числам ("первое число" "операция" "второе число") и выводит результат на экран.

Поддерживаемые операции: +, -, /, *, mod, pow, div, где
mod — это взятие остатка от деления,
pow — возведение в степень,
div — целочисленное деление.

Если выполняется деление и второе число равно 0, необходимо выводить строку "Деление на 0!".

Обратите внимание, что на вход программе приходят вещественные числа.

Жители страны Малевии часто экспериментируют с планировкой комнат.

Комнаты бывают треугольные, прямоугольные и круглые. Чтобы быстро вычислять жилплощадь, требуется написать программу, на вход которой подаётся тип фигуры комнаты и соответствующие параметры, которая бы выводила площадь получившейся комнаты.

Для числа π в стране Малевии используют значение 3.14.

Формат ввода, который используют Малевийцы:

Треугольник

a

b

c

где a, b и c — длины сторон треугольника

прямоугольник

a

b

где a и b — длины сторон прямоугольника

круг

r

где r — радиус окружности

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

На ввод могут подаваться и повторяющиеся числа.

Паша очень любит кататься на общественном транспорте, а получая билет, сразу проверяет, счастливый ли ему попался. Билет считается счастливым, если сумма первых трех цифр совпадает с суммой последних трех цифр номера билета.

Однако Паша очень плохо считает в уме, поэтому попросил вас написать программу, которая проверит равенство сумм и выведет "Счастливый", если суммы совпадают, и "Обычный", если суммы различны.

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

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

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

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

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

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

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

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

ДОКУМЕНТ ПОДПИСАН
ЭЛЕКТРОННОЙ ПОДПИСЬЮ

СВЕДЕНИЯ О СЕРТИФИКАТЕ ЭП

Сертификат 143507986500560089701835989304833372774460075073

Владелец Керн Ирина Юсуповна

Действителен С 02.04.2025 по 02.04.2026