РЕГИОНАЛЬНЫЙ ЦЕНТР ВЫЯВЛЕНИЯ, ПОДДЕРЖКИ И РАЗВИТИЯ СПОСОБНОСТЕЙ И ТАЛАНТОВ ДЕТЕЙ И МОЛОДЁЖИ СТАВРОПОЛЬСКОГО КРАЯ «СИРИУС 26»

СОГЛАСОВАНО
Экспертным советом регионального центра выявления, поддержки и развития способностей и талантов детей и молодёжи Ставропольского края «Сириус 26»,
протокол № 1/2025 от 03.02.2025 г.

УТВЕРЖДЕНО
Директором Центра «Поиск»
Томилиной О.А.

приказ № 13/1 от 04.02.2025 г.

ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА

«ЯЗЫК ПРОГРАММИРОВАНИЯ PYTHON»

Направленность:	техническая
Возраст обучающихся:	13-16 лет
Объем программы:	90 часов
Срок освоения:	2 месяца
Форма обучения:	очная с использованием дистанционных образовательных технологий
Автор программы:	Решетняк Ольга Владимировна, педагог дополнительного образования центра «Поиск»; Пономаренко Елена Александровна, руководитель СП МО ИТ ГАОУ ДО «Центр для одаренных детей «Поиск»

Ставрополь
2025

ОГЛАВЛЕНИЕ

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

1. Основные характеристики программы

В настоящее время, в условиях экономических санкций, российская экономика переживает значительные изменения, особенно в сфере информационных технологий. С уходом зарубежных разработчиков программного обеспечения возникает необходимость в квалифицированных специалистах, способных эффективно решать задачи государства. В данной ситуации особую актуальность приобретает быстрое наращивание числа программистов, владеющих языком Python, который зарекомендовал себя как один из наиболее востребованных языков программирования.

Программирование занимает центральное место в подготовке кадров в области информационных технологий. В рамках федерального проекта «Развитие кадрового потенциала ИТ-отрасли» в национальной программе «Цифровая экономика РФ» подчеркивается исключительная значимость изучения языков программирования, включая Python. Следовательно, обучение программированию, в частности языку Python, следует начинать уже на уровне школьного образования.

Школьники, которые умеют структурировать данные, разрабатывать алгоритмы и писать код на Python, обычно легче осваивают пользовательские навыки, так как осознают, как функционирует управление компьютером. Это знание способствует повышению успеваемости и по другим предметам, поскольку учащиеся развивают критическое мышление и запас методов для решения различных задач. В этой связи важно прививать навыки алгоритмического мышления, необходимого для работы с языком Python.

1.1. Направленность программы

Дополнительная общеобразовательная общеразвивающая программа «Язык программирования Python» имеет техническую направленность и охватывает два ключевых аспекта изучения:

1) Технологический: обучение языку Python рассматривается как способ формирования образовательного потенциала, необходимого для освоения современных цифровых технологий.

2) Общеобразовательный: программа нацелена на развитие основных познавательных процессов, таких как анализ, выявление закономерностей, создание инструкций и логические выводы.

1.2. Адресат программы

Программа предназначена для учащихся 13-16 лет. Она ориентирована на одаренных школьников 8-10 классов, проявляющих интерес к информатике и желающих развить навыки программирования, получая углубленные

теоретические и практические знания по актуальным темам в сфере цифровой экономики.

1.3. Актуальность программы

Совершенствование технологий и программного обеспечения привело к сокращению часов, выделяемых для изучения программирования в школьных учебных планах. Современные визуальные и мультимедийные инструменты часто отвлекают школьников от изучения языков программирования. Простые инструменты для поиска информации в интернете подразумевают более легкий подход, что снижает интерес к углубленному пониманию программных концепций. В большинстве школ отсутствует систематическое обучение языкам программирования, таким как Python.

Дополнительная общеобразовательная общеразвивающая программа «Основы программирования на Python» предоставляет учащимся возможность освоить актуальный язык, формируя аналитические способности и навыки для работы с данными, а также облегчая процесс обучения за счет наглядности и доступности.
Вот текст про язык программирования Python в аналогичном стиле:

1.4. Отличительные особенности/новизна программы

Содержание программы «Язык программирования Python» разработано с учетом современных требований к подготовке школьников к жизни и работе в цифровом обществе, в рамках ключевых направлений федерального проекта «Развитие кадрового потенциала ИТ-отрасли» национальной программы «Цифровая экономика РФ».
Новизна программы «Язык программирования Python» заключается в предоставлении учащимся возможности изучать один из самых популярных языков программирования, который активно используется в различных областях, таких как веб-разработка, анализ данных, искусственный интеллект и автоматизация. Программа обучает основам синтаксиса языка, принципам объектно-ориентированного программирования и дает возможность применять полученные знания на практике через создание собственных компьютерных приложений и проектов. По окончании курса учащиеся будут готовы продолжить свое образование на более углубленном уровне в области программирования.

Уровень освоения программы – углубленное изучение языка программирования Python.

1.5. Объем и срок освоения программы

Объем программы – 90 часов.
Срок реализации программы – 2 месяца.

1.6. Цели и задачи программы

Цель программы – выявление, развитие и продвижение одаренных детей Ставропольского края в сфере программирования, а также подготовка их к участию в государственных инициативах по поддержке таланта. Программа формирует у школьников базовые знания и навыки в программировании на Python, а также осознание фундаментальных концепций и технологий, необходимых для создания современных программных решений.

Задачи программы

1) Обучающие:
- изучение основ синтаксиса и структуры языка Python;
- знакомство с принципами объектно-ориентированного программирования;
- обучение работе с библиотеками и фреймворками, используемыми в Python;
- изучение встраивания Python в различные среды и приложения;
- практическое освоение разработки собственных проектов, включая написание скриптов и приложений.

2) Развивающие:
- развитие самостоятельности в приобретении знаний, умений и навыков;
- развитие способности к применению полученных знаний в различных контекстах;
- формирование критического мышления, умения ставить и проверять гипотезы, разрабатывать стратегии решения проблем и анализировать результаты своей деятельности;
- стимулирование рефлексивной активности, позволяющей оценивать и анализировать собственные действия и успехи.

3) Воспитательные:
- формирование мировоззрения, связанного с этическими нормами и принципами работы в информационном обществе;
- освоение информационной культуры: ответственного отношения к информации и соблюдения правовых и этических норм в работе с данными;
- приобщение к системам ценностей, принципам и правилам, связанным с цифровыми технологиями;
- формирование умений работать в группе и сотрудничать с окружающими;
- ранняя профориентация школьников через знакомство с востребованными профессиями, связанными с программированием и цифровыми технологиями.

1.7. Планируемые результаты освоения программы

1. Предметные результаты:
- знание основ программирования на языке Python;
- владение базовыми конструкциями языка, такими как переменные, операторы, функции и классы;

- понимание принципов работы с библиотеками и фреймворками, а также навыки создания простых программ, игр, Telegram-ботов и проектов на Python.

2. Метапредметные результаты:
- способность соотносить и оценивать результаты своей деятельности с поставленными задачами;
- использование цифровых технологий как инструмента для реализации программных решений;
- осознание связи языка Python с другими областями науки и технологий;
- осуществление саморефлексии и анализа работы группы, что позволит экспериментировать с новыми подходами и методами в совместной деятельности.

3. Личностные результаты:
- понимание и адекватная оценка собственных возможностей в программировании;
- развитие навыков межличностного общения и способности работать в команде;
- обучение эффективному распределению времени при выполнении заданий;
- формирование навыков совместного решения проблем и эффективного распределения ролей во время групповой работы.

2. Организационно-педагогические условия реализации программы

2.1. Язык реализации программы
Реализация дополнительной общеобразовательной общеразвивающей программы «Язык программирования Python» осуществляется на государственном языке Российской Федерации (русском языке).

2.2. Форма обучения
Форма обучения – очная с применением дистанционных образовательных технологий.

2.3. Особенности реализации программы
Программа реализуется по модульному принципу с интеграцией дистанционных образовательных технологий.

2.4. Условия набора и формирования групп
К обучению допускаются учащиеся 8-10 классов общеобразовательных школ Ставропольского края:
1) по результатам конкурсного отбора успешного прохождения учебно-отборочного курса «Введение в Python»;
2) на основании участия в олимпиадах и других интеллектуальных конкурсах регионального и всероссийского уровней.

Условия конкурсного отбора обеспечивают соблюдение прав обучающихся в сфере дополнительного образования и гарантируют зачисление наиболее способных и подготовленных учащихся для освоения программы.
Состав групп формируется по принципу одновозрастности.

2.5. Формы организации и проведения занятий
Занятия организуются в дистанционном формате, в аудиториях (под непосредственным руководством преподавателя) и внеаудиторно (самостоятельная подготовка учащихся к олимпиадам вне учебного плана).
Формы проведения занятий включают комбинированные, теоретические, практические, самостоятельные, репетиционные и контрольные.
Организация деятельности учащихся может принимать различные формы:
- фронтальная: работа преподавателя с группой всех учащихся одновременно;
- групповая: организация работы в малых группах, включая пары, для выполнения конкретных задач, где виден вклад каждого участника;
- коллективная: проблемно-ориентированное взаимодействие, где все участники задействованы одновременно;
- индивидуальная: для коррекции пробелов в знаниях и отработки отдельных навыков.

Режим занятий. Программа реализуется в г. Ставрополе в очной форме пять дней в неделю, по восемь учебных часов.

Основные методы и формы реализации содержания программы

Метод двумерной дидактики
В рамках программы обучения широко используется метод двумерной дидактики. Данный метод становится актуальным, когда знаний и навыков, полученных обучающимися на уроках в школе, недостаточно для освоения курса «Язык программирования Python». Этот подход предполагает выбор таких форм обучения, которые позволяют не только решать задачи, но и глубоко понимать терминологию, технологию, а также осмысливать суть и смысл изучаемого материала. Учащиеся учатся оценивать преимущества и недостатки различных решений, предлагать альтернативные варианты и делают всё это в сжатые сроки. Суть метода двумерной дидактики заключается в организации результативного учебного процесса в соответствии с уровнем подготовки учащихся. Систематическое применение данного метода способствует более глубокому усвоению сложного материала, что может привести к опережающему усвоению знаний на несколько лет. Это достигается благодаря спиральному повторению информации, расширению понятийного поля и применению знаний в различных контекстах и с разных перспектив.

Проблемный метод

Проблемный метод охватывает ряд приемов, используемых для выполнения заданий с неоднозначными вариантами решения, особенно в условиях недостатка или избытка информации. Основная цель применения этого метода заключается в развитии аналитических навыков у обучающихся, таких как сравнение, аргументация и выводы, через активное участие в решении сложных задач. Все задания основываются на имеющихся знаниях, однако требуют самостоятельного поиска новых фактов и сведений. Осознание недостатка собственных знаний стимулирует учащихся к поиску новых знаний, что является важным условием для развития творческого мышления. Ключевым достоинством проблемного метода является овладение технологиями принятия решений в ситуациях неопределенности или неоднозначности, что подразумевает разработку различных стратегий решения задач с учетом ограничения информации и ресурсов.

Словесные методы

Лекция с обратной связью представляет собой один из словесных методов, при котором преподаватель периодически задает вопросы во время изложения теоретического материала, чтобы оценить усвоение содержания. Вопросы заранее формулируются для контроля понимания ключевых аспектов.

Эвристическая беседа, вопросно-ответная форма, основывается на греческом слове «эвристика», что означает «отыскать» или «открыть». Суть метода заключается в том, что преподаватель формирует последовательность вопросов, направляющих мысли учащихся в нужное русло. Эвристическая беседа основывается на интуитивных и неявных знаниях, которые учащиеся приобрели на основе самостоятельного опыта. Этот метод может быть особенно эффективен в качестве мотивационной беседы при знакомстве с новой темой.

Метод дизайн-мышления

Метод дизайн-мышления фокусируется на создании продуктов и услуг, которые ориентированы на потребности пользователей. Каждая идея представляет собой решение конкретной потребности человека.

Принципы дизайн-мышления:

1) Ошибайся раньше и чаще.
2) Создавай прототипы вместо того, чтобы просто рассказывать о продукте.
3) Первостепенно фиксируй пожелания пользователей.
4) Разрабатывай продукт совместно с пользователем.

Основные компоненты дизайн-мышления:

1) Процесс: присутствует алгоритм работы, интерактивность и формирование смешанных команд.

2) Подход: акцент на человекоцентричности, эмпатии и культуре быстрого допуска ошибок.
3) Среда: наглядный мыслительный процесс и возможность «думать руками».

УЧЕБНЫЙ ПЛАН

№ темы	Наименование модуля, учебного курса	Контактная работа обучающихся с преподавателем, часов			Формы контроля / аттестации
		Теория	Практика	Всего	
1.	Дистанционный учебно-отборочный курс «Введение в разработку на Python»	3	3	6	Тест с самопроверкой. Разработка программы.
2.	Учебный курс «Язык программирования Python»	20	60	80	Разработка и дизайн десктопного приложения, разработка игры, разработка Telegram-бота.
3.	Учебно-тренинговый курс «Архитектура проложения. Тестирование»	2	2	4	Тест с самопроверкой.
Итого:		**25**	**65**	**90**	

КАЛЕНДАРНЫЙ УЧЕБНЫЙ ГРАФИК

Наименование модуля, учебного курса	Дата начала обучения	Дата окончания обучения	Количество учебных недель	Количество учебных дней	Количество учебных часов	Режим занятий
Дистанционный учебно-отборочный курс «Введение в разработку на Python»	02.03.2025	19.03.2025	2	14	6	Дистанционное обучение
Учебный курс «Язык программирования Python»	13.04.2025	24.04.2025	2	10	80	Очное обучение, 5 раз в неделю по 8 часов
Учебно-тренинговый курс «Архитектура проложения. Тестирование»	24.04.2025	18.05.2025	3	21	4	Дистанционное обучение

РАБОЧАЯ ПРОГРАММА УЧЕБНОГО КУРСА «Язык программирования Python»

Курс «Язык программирования Python» предназначен для обучающихся 8-10 классов. Освоение языка Python и связанных с ним технологий является важным этапом для тех, кто стремится развивать навыки в программировании, создании приложений и разработке игр.

В рамках курса «Язык программирования Python» рассматриваются следующие темы:

1. Вводно-отборочный курс:
   - Синтаксис языка Python и его основные конструкции.
   - Типы данных и работа с ними.
   - Вложенные конструкции.
   - Циклы и функции для построения логики программ.

В результате успешного освоения курса обучающийся должен:

Знать:
– основы синтаксиса и конструкции языка Python;
– ключевые типы данных и способы их использования;

Уметь:
– формулировать задачи и разрабатывать алгоритмы;

Тематический план курса

№	Наименование раздела, темы	Контактная работа обучающихся с преподавателем, час.		
		Теория	Практика	Всего
1	Введение в программирование.	1		1
2	Синтаксис языка Python и его основные конструкции	1		1
3	Типы данных и работа с ними		1	2
4	Вложенные конструкции		1	2
5	Циклы		1	2
6	Функции для построения логики программ		1	2

Итого:	**2**	**4**	**6**

РАБОЧАЯ ПРОГРАММА УЧЕБНО-ОТБОРОЧНОГО КУРСА «Введение в разработку на Python»

Курс «Введение в разработку на Python» предназначен для обучающихся 8-10 классов, желающих освоить основы программирования на языке Python и научиться работать с его базовыми конструкциями.

СОДЕРЖАНИЕ КУРСА:

1. Тема 1: Синтаксис языка Python и его основные конструкции
   - Теория: Знакомство с синтаксисом Python, структура программы, правила написания кода.
   - Практика: Написание простых программ с использованием основных конструкций Python.

2. Тема 2: Типы данных и работа с ними
   - Теория: Обзор основных типов данных в Python: целые числа, вещественные числа, строки, списки, кортежи, множества и словари.
   - Практика: Работа с различными типами данных, выполнение операций и манипуляции с ними.

3. Тема 3: Вложенные конструкции
   - Теория: Понимание и использование вложенных структур: условные операторы и циклы.
   - Практика: Написание программ с использованием вложенных конструкций для решения задач.

4. Тема 4: Циклы
   - Теория: Изучение циклов `for` и `while`, их применение в программировании.
   - Практика: Опыт написания программ с различными типами циклов для автоматизации задач.

5. Тема 5: Функции для построения логики программ
   - Теория: Определение функций, передача аргументов, возврат значений. Понимание важности использования функций для организации кода.

- Практика: Создание и использование собственных функций в программах для повышения их читаемости и модульности.

Форма подведения итогов: тестирование с возможностью самопроверки и написание программного проекта, в котором используются изученные темы.

Ожидаемые результаты:

По окончании курса обучающиеся смогут:
- Понимать синтаксис языка Python и использовать его основные конструкции.
- Работать с различными типами данных и выполнять операции над ними.
- Применять вложенные конструкции, циклы и функции для организации логики своих программ.

РАБОЧАЯ ПРОГРАММА КУРСА «ЯЗЫК ПРОГРАММИРОВАНИЯ PYTHON»

Курс «Язык программирования Python» предназначен для обучающихся 8-10 классов. В рамках курса школьники научатся использовать язык Python для решения различных задач, разрабатывать Telegram-ботов, создавать графические приложения и игровые проекты. Участники получат навыки работы с популярными библиотеками Python и основы проектирования приложений. В курсе рассматриваются следующие темы:

1. Обзор библиотек Python:
    - Введение в модули и пакеты.
    - Визуализация данных с использованием соответствующих библиотек.

2. Введение в Telegram-ботов:
    - Основные команды и обработка сообщений.
    - Работа с клавиатурой и кнопками.

3.Основы работы с API
    - Определение API и его использование в программировании
    - Примеры API: REST, JSON

4.Работа с библиотекой `requests`:
    - Отправка запросов (GET, POST)

- Обработка ответов (JSON)
- Примеры использования API (например, получение данных о погоде)

5.Создание Telegram-бота
- Регистрация бота через BotFather
- Получение API-токена

6.Знакомство с Tkinter и создание первого окна:
- Введение в GUI-программирование.
- Обзор Tkinter: преимущества, структура, основные виджеты (Label, Button, Entry).

7. Виджеты ввода и компоновка интерфейса:
- Виджеты Entry, Text, Listbox.
- Менеджеры компоновки grid() и pack().
- Валидация данных (например, проверка на пустые поля).

8.Работа с файлами CSV и диалоговыми окнами
- Основы работы с CSV-файлами: чтение и запись (модуль csv).
- Диалоговые окна: filedialog (открытие/сохранение файлов), messagebox (уведомления).

9. Табличное отображение данных (Treeview) и меню
- Виджет ttk.Treeview для отображения табличных данных.
- Создание меню: верхнее меню (Menu) и контекстное меню.

10. Редактирование контактов и валидация
- Редактирование данных в Treeview: получение выделенной строки, изменение значений.
- Валидация полей (например, email должен содержать "@").

ОЖИДАЕМЫЕ РЕЗУЛЬТАТЫ

По окончании курса обучающийся должен:

Знать:

- Основные библиотеки Python, применяемые для научных расчетов и визуализации данных.
- Принципы разработки Telegram-ботов, обработки сообщений и взаимодействия с внешними API.
- Основы создания графических приложений и работы с пользовательскими интерфейсами.

- Архитектурные паттерны для десктопных приложений, такие как MVC и MVVM.
- Основы разработки игр с использованием библиотеки Pygame и обработки звуковых эффектов.
- Принципы обработки исключений и ведения отладки в приложениях.
- Способы упаковки и развертывания приложений, включая создание исполняемых файлов.
- Создание и настройка графических элементов и пользовательских интерфейсов.
- Основы тестирования приложений, включая юнит-тестирование и тестирование графических интерфейсов.

Уметь:
- Подбирать соответствующие библиотеки и инструменты для решения задач разной сложности.
- Разрабатывать Telegram-ботов, реализуя функционал обработки сообщений и взаимодействия с пользователями.
- Создавать простые и сложные графические приложения, работать с данными и файлами.
- Реализовывать простейшие и полноценные игры, включая интеграцию звука и музыки, а также тестировать их работоспособность.
- Применять архитектурные паттерны для структурирования кода и повышения его читаемости.
- Обрабатывать исключения и выполнять отладку кода для повышения надежности приложений.
- Упаковывать приложения в исполняемые файлы для удобства распространения и использования.
- Создавать и настраивать пользовательские интерфейсы, разрабатывать собственные виджеты и компоненты.
- Проводить тестирование своих приложений, используя соответствующие инструменты и методы.

Тематический план

№ темы	Наименование раздела, темы	Контактная работа обучающихся с преподавателем, часов		
		Теория	Практика	Всего
1	Обзор библиотек Python.	2	6	8
2	Введение в Telegram-ботов.	2	6	8
3	Основы работы с API.	2	6	8

4	Работа с библиотекой `requests`.	2	6	8
5	Создание Telegram-бота.	2	6	8
6	Знакомство с Tkinter и создание первого окна.	2	6	8
7	Виджеты ввода и компоновка интерфейса.	2	6	8
8	Работа с файлами CSV и диалоговыми окнами.	2	6	8
9	Табличное отображение данных (Treeview) и меню.	2	6	8
10	Редактирование контактов и валидация.	2	6	8
	Итого	**20**	**60**	**80**

СОДЕРЖАНИЕ КУРСА «ЯЗЫК ПРОГРАМИРОВАНИЯ

Тема 1. Обзор библиотек Python
Теория:
- Понятие библиотек и модулей в Python.
- Зачем использовать библиотеки: ускорение разработки, доступ к готовым функционалам.
- Популярные библиотеки: NumPy, Pandas, Matplotlib, Requests, BeautifulSoup и др.
- Установка библиотек с помощью `pip` и использование `requirements.txt` для управления зависимостями.

Практика:
- Установка нескольких библиотек (`requests`, `beautifulsoup4`, `numpy`) через `pip`.
- Простой пример использования каждой библиотеки:
  - `requests`: выполнение GET-запроса.
  - `beautifulsoup4`: парсинг HTML-страницы.
  - `numpy`: создание и работа с массивами.

Тема 2. Введение в Telegram-ботов
Теория:
- Что такое Telegram-боты и их назначения.
- Как зарегистрировать бота в Telegram через BotFather.
- Основные функции ботов (отправка сообщений, реакции на команды и т.д.).
- Обзор архитектуры ботов: webhook, long polling.

Практика:
- Регистрация бота с помощью BotFather и получение токена.
- Создание простейшего скрипта на Python, который взаимодействует с Telegram API, используя `requests`.
- Отправка приветственного сообщения пользователю.

Тема 3. Основы работы с API
Теория:
- Что такое API и зачем оно нужно.
- Различия между REST и SOAP.
- Основные методы HTTP (GET, POST, PUT, DELETE).
- JSON как формат обмена данными.
Практика:
- Написание простого клиента для взаимодействия с публичным API.
- Выполнение GET и POST запросов с использованием `requests`.
- Обработка и вывод полученных данных в формате JSON.

Тема 4. Работа с библиотекой `requests`
Теория:
- Обзор библиотеки `requests`: возможность работы с HTTP-запросами в Python.
- Основные функции: GET, POST, PUT, DELETE.
- Работа с параметрами и заголовками запроса.
- Обработка ответов (статус коды, JSON, текст).
Практика:
- Выполнение различных типов HTTP-запросов к публичному API (например, API погоды или новостей).
- Обработка полученного ответа, вывод на экран интересующей информации (например, температура, заголовок новости).
- Практическое задание: написать программу, которая получает и отображает данные из выбранного API.

Тема 5. Создание Telegram-бота
Теория:
- Обзор библиотеки `python-telegram-bot` и ее возможности.
- Структура кода для создания Telegram-бота на Python.
- Основные обработчики: текстовые команды, кнопки, inline-режим и т.д.
Практика:
- Установка библиотеки `python-telegram-bot`.

- Создание Telegram-бота, который обрабатывает команды и отправляет сообщения.
  - Команда `/start` — приветственное сообщение.
- Команда `/help` — список доступных команд.
- Расширение функциональности бота: например, добавление кнопок, реакции на текстовые сообщения или интеграция с API для получения данных.

Тема 6: Знакомство с Tkinter и создание первого окна
Теория: Введение в GUI-программирование.
Обзор Tkinter: преимущества, структура, основные виджеты (Label, Button, Entry).
Создание главного окна: Tk(), настройка заголовка и размеров (title(), geometry()).
Практика: Написать программу с окном, полем ввода имени и кнопкой "Добавить".
Выводить введенное имя в консоль при нажатии кнопки.

Тема 7: Виджеты ввода и компоновка интерфейса
Теория: Виджеты Entry, Text, Listbox.
Менеджеры компоновки grid() и pack().
Валидация данных (например, проверка на пустые поля).
Практика: Создать форму для ввода контакта: имя, телефон, email.
Добавить кнопку "Добавить контакт", которая сохраняет данные в список (Listbox).

Тема 8: Работа с файлами CSV и диалоговыми окнами
Теория: Основы работы с CSV-файлами: чтение и запись (модуль csv).
Диалоговые окна: filedialog (открытие/сохранение файлов), messagebox (уведомления).
Практика: Реализовать загрузку контактов из CSV-файла в Listbox.
Добавить кнопку "Сохранить в CSV", которая записывает данные из Listbox в файл.

Тема 9: Табличное отображение данных (Treeview) и меню
Теория: Виджет ttk.Treeview для отображения табличных данных.
Создание меню: верхнее меню (Menu) и контекстное меню.
Практика: Заменить Listbox на Treeview с колонками: "Имя", "Телефон", "Email".
Добавить меню "Файл" с опциями "Загрузить CSV", "Сохранить CSV".

Тема 10: Редактирование контактов и валидация
Теория: Редактирование данных в Treeview: получение выделенной строки, изменение значений.
Валидация полей (например, email должен содержать "@").
Практика: Добавить кнопку "Редактировать": открывать окно с заполненными полями выбранного контакта.
Сохранять изменения в Treeview и CSV.

РАБОЧАЯ ПРОГРАММА УЧЕБНО-ТРЕНИНГОВОГО КУРСА «Архитектура десктопных приложений и тестирование программного обеспечения»

Курс «Архитектура десктопных приложений и тестирование программного обеспечения» предназначен для обучающихся, заинтересованных в разработке качественных программных решений.

В курсе «Архитектура десктопных приложений и тестирование программного обеспечения» рассматриваются основные принципы проектирования десктопных приложений и основы тестирования программного обеспечения.

В результате освоения учебного курса обучающийся должен:

Знать:
- Основные архитектурные паттерны десктопных приложений (такие как MVC, MVVM и т.д.).
- Этапы жизненного цикла разработки программного обеспечения.
- Принципы организации кода при разработке приложений.
- Основы тестирования программного обеспечения, включая различные методологии тестирования и написание тестов.
- Виды тестирования: юнит-тестирование, интеграционное тестирование, функциональное тестирование.

Уметь:
- Проектировать архитектуру десктопных приложений с использованием выбранных архитектурных паттернов.
- Разрабатывать функциональные модули, соблюдая лучшие практики программирования.
- Проводить тестирование программного обеспечения, применяя различные методы и инструменты.

- Писать юнит-тесты для проверки отдельных модулей и компонентов приложения.
- Анализировать и исправлять ошибки на основе результатов тестирования, обеспечивая качество конечного продукта.

Тематический план

№ темы	Наименование раздела, темы	Контактная работа обучающихся с преподавателем, часов		
		Теория	Практика	Всего
1	ООП-структура приложения	1	1	2
2	Тестирование приложений	1	1	2
	Итого	2	2	4

Содержание курса «Архитектура десктопных приложений и тестирование программного обеспечения»

Тема 1. ООП-структура приложения
Теория. Обзор основных архитектурных паттернов (MVC, MVVM и др.), а также этапы жизненного цикла разработки программного обеспечения. Знакомство с основами организации кода и принципами SOLID.
Практика. Проектирование архитектуры простого десктопного приложения с использованием одного из паттернов и разработка функционального модуля в соответствии с принятыми стандартами.
Форма подведения итогов: тестирование реализации архитектуры и разработки кода с самопроверкой.

Тема 2. Основы тестирования приложений
Теория. Введение в тестирование ПО, виды тестирования (юнит-тестирование, интеграционное тестирование, функциональное тестирование) и методологии тестирования. Обзор инструментов для тестирования и отладки кода.
Практика. Написание юнит-тестов для проверки функциональности разработанного модуля, анализ и отладка ошибок.
Форма подведения итогов: оценка качества написанных тестов и проведенной отладки с самопроверкой.

МЕТОДИЧЕСКИЕ МАТЕРИАЛЫ

№ п/п	Название раздела, темы	Формы учебного занятия	Формы, методы, приемы обучения. Педагогические технологии	Материально-техническое оснащение, дидактико-методический материал	Форма подведения итогов
1	Тема 1. Введение в программирование.Синтаксис языка Python и его основные конструкции	Комбинированная	Объяснительно-иллюстративный. Частично-поисковый. Исследовательский..	Проекционное оборудование, ПК. Доступ к сети Интернет.	Тестирование
2	Тема 2. Типы данных и работа с ними	Комбинированная	Объяснительно-иллюстративный. Частично-поисковый. Исследовательский.	Проекционное оборудование, ПК. Доступ к сети Интернет.	Тестирование
3	Тема 3. Вложенные конструкции	Комбинированная	Объяснительно-иллюстративный. Частично-поисковый. Исследовательский.	Проекционное оборудование, ПК. Доступ к сети Интернет.	Тестирование
4	Тема 4. Циклы и функции	Комбинированная	Объяснительно-иллюстративный. Частично-поисковый. Исследовательский.	Проекционное оборудование, ПК. Доступ к сети Интернет.	Тестирование

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ

В процессе обучения проводятся разные виды контроля результативности усвоения программного материала.

1) Входной контроль

Входной контроль проводится целью выявления первоначального уровня знаний и умений, возможностей обучающихся. Входной контроль проводится с каждым обучающимся индивидуально по следующим параметрам – теоретическим и практическим.

Формы: тестирование и проектная деятельность.

Отборочный тест проводится в рамках дистанционного учебно-отборочного курса с целью отбора участников очной профильной смены.

1. Отборочный тест состоит из 25 заданий разного уровня сложности из разделов «Введение в программирование на Python» и «Язык программирования Python», направлен на проверку основных понятий, рассматриваемых тем.
2. Разработка творческого проекта с целью определения уровня умений и навыков самостоятельной работы, критического мышления и творческого подхода обучающихся на основе имеющихся знаний.

Во время проведения входной диагностики педагог заполняет информационную карточки «Результаты входной диагностики», пользуясь шкалой «Оценка параметров входного контроля».

Оценка параметров входного контроля

Наименование уровня	Результат диагностики, %
Элементарный уровень	0 - 54%
Низкий уровень	55 - 69%
Средний уровень	70 - 84%
Высокий уровень	85 - 100%

Примерные задания:

1. Какое из следующих утверждений корректно объявляет переменную в Python?
   - A) int x
   - B) x = 10
   - C) let x = 10
   - D) var x: int

2. Какой тип данных используется для хранения последовательности символов в Python?
   - A) int
   - B) str
   - C) list
   - D) dict

3. Какой из приведённых операторов используется для сравнения значений в Python?
   - A) =
   - B) ==
   - C) !=
   - D) <>

4. Какой из приведённых вариантов правильно объявляет функцию в Python?
   - A) def my_function():
   - B) function my_function():
   - C) my_function() =>
   - D) define my_function():

5. Какой метод используется для получения длины списка в Python?
   - A) length()
   - B) size()
   - C) count()
   - D) len()

6. Какой из следующих типов данных является неизменяемым?
   - A) list
   - B) dict
   - C) tuple
   - D) set

7. Какой результат выполнения следующего кода?

```
result = 0
for i in range(3):
    result += i
```

8. Какое значение будет иметь `x` после выполнения следующего кода?

```
x = 1
while x < 5:
    x += 1
```

9. Какое значение будет выведено, если `x` равно 0 и выполняется следующий код:

```
if x:
    print("Истина")
else:
    print("Ложь")
```

- A) "Истина"
- B) "Ложь"

10. Какой оператор используется для проверки неравенства?
- A) ==
- B) !=
- C) <>
- D) <>

11. Какой из этих операторов логического И используется в Python?
- A) &&
- B) &
- C) and
- D) ||

12. Какой из приведённых конструкций указывает на начало цикла `for` в Python?
- A) for i in range(5):
- B) for (i: 0; i < 5; i++)
- C) foreach i in range(5):

- D) loop i from 0 to 5

13. Какой из следующих параметров позволяет сделать аргумент функции необязательным?
- A) Использование `default`
- B) Присвоение значения по умолчанию
- C) Использование `optional`
- D) Использование `def`

14. Завершите фразу: В Python разница между `append()` и `extend()` заключается в том, что `append()` _____.
- A) Добавляет элемент в конец списка
- B) Объединяет два списка
- C) Удаляет элемент из списка
- D) Создаёт новый список

15. Какой метод используется для удаления элемента из списка по значению?
- A) remove()
- B) del()
- C) pop()
- D) delete()

16. Какой оператор используется для преобразования типа данных в Python?
- A) convert
- B) as
- C) cast
- D) type()

17. Завершите фразу: Если мы хотим, чтобы функция возвращала несколько значений, мы можем использовать _____.
- A) список
- B) множество
- C) кортеж
- D) ни одно из вышеперечисленных

18. Что возвращает функция, если в ней не указана инструкция `return`?
- A) 0
- B) None
- C) Пустую строку
- D) Значение, равное 1

19. Какой из этих циклов следует использовать, когда необходимо пройтись по элементам коллекции?
- A) while
- B) for
- C) do while
- D) repeat

20. Как выведет следующий код, если `x` равно 10?

```
if x > 5:
    print("больше пяти")
else:
    print("пять или меньше")
```

- A) "больше пяти"
- B) "пять или меньше"

21. Какой выходной результат имеет следующий код?

```
x = [1, 2, 3]
print(x[1])
```

22. Каково значение переменной `result`, после выполнения следующего кода:

```
def add(a, b=5):
    return a + b

print(add(10))
```

23. Выберите все возможные структуры данных в Python:
- A) list
- B) tuple
- C) array
- D) set

24. Какое значение будет выведено на экран в результате выполнения следующей программы?

```
for i in range(3):
    print(i)
```

- A) 0, 1, 2
- B) 1, 2, 3
- C) 1, 2
- D) 0, 1, 2, 3

25. Какой из приведённых операторов используется для преобразования типа данных в Python?
- A) int()
- B) float()
- C) str()
- D) Все вышеперечисленные

2) *Текущий контроль* проводится в рамках очной профильной смены на занятиях в виде наблюдения за успехами каждого обучающегося, процессом формирования компетенций. Текущий контроль успеваемости служит для определения педагогических приемов и методов для индивидуального подхода к каждому обучающемуся, корректировки плана работы с группой.

Формы:

- устные и письменные работы;
- индивидуальный опрос.

Практические задания, домашние работы, учащиеся выполняют в форме устной или письменной речи. Оценка основывается на ясности выражения мыслей и использовании предметных знаний.

Текущий контроль успеваемости служит для определения педагогических приемов и методов для индивидуального подхода к каждому обучающемуся, корректировки плана работы с группой. Осуществляется в форме наблюдения, тестирования, контрольного опроса (устного или письменного), собеседования, психологического мониторинга.

Варианты примерных заданий.

1. Что такое библиотека в Python и как она используется?
2. Как импортировать библиотеку в Python? Примеры использования.
3. Какие библиотеки вы знаете для работы с данными?
4. Как создать нового Telegram-бота и получить токен?
5. Что такое webhook в контексте Telegram-ботов?

6. Как использовать библиотеку `python-telegram-bot` для обработки сообщений?
7. Что такое графический интерфейс пользователя (GUI) и зачем он нужен?
8. Как создать основное окно приложения с помощью Tkinter?
9. Что такое виджеты в Tkinter? Примеры основных виджетов.
10. Как организовать структуру вашего приложения с использованием классов?
11. Как сохранить и загрузить данные с помощью текстового файла в вашем приложении?
12. Как обработать пользовательский ввод от GUI-элемента?
13. Что такое игровая петля (game loop) и для чего она необходима?
14. Как создать простую игру с использованием Pygame? Основные шаги.
15. Как добавить обработку нажатий клавиш в вашей игре?
16. Как реализовать систему очков в игре?
17. Как создавать и использовать спрайты в Pygame?
18. Как организовать уровни в игре и переход между ними?
19. Что такое исключения в Python и как их обрабатывать с помощью `try` и `except`?
20. Как использовать оператор `finally` в блоке обработки исключений?
21. Как создать исполняемый файл для приложения с помощью PyInstaller?
22. Что такое виртуальное окружение и для чего оно используется?
23. Каковы основные методы тестирования программного обеспечения?
24. Как написать простой юнит-тест с помощью библиотеки `unittest`?
25. Основные принципы проектирования программного обеспечения и как они помогают в разработке приложений?

Промежуточная аттестация. Проводится в форме теста с самопроверкой.

Итоговая аттестация. Завершает второй модуль, проводится в виде индивидуального итогового тестирования и создание творческого проекта.

Формы отслеживания результатов: наблюдение, тестирование, контрольный опрос (устный или письменный), психологический мониторинг.

Формы фиксации результатов: аналитическая справка, оценочные материалы, результаты психологического мониторинга, отчёт.

Документальной формой подтверждения итогов реализации отдельного курса программы является документ об обучении «Сертификат» (без оценки) установленного Центром «Поиск» образца.

КАДРОВОЕ ОБЕСПЕЧЕНИЕ ПРОГРАММЫ

Обеспечение реализации программы, нацеленной на предоставление высокого качества обучения, планируется за счет штата, состоящего из высококвалифицированных специалистов, обладающих определенными компетенциями и выполняющими определенный функционал. Из них:

- учитель информатики высшей квалификационной категории - 2 чел.;
- педагог-психолог высшей квалификационной категории - 1 чел.;
- педагог-организатор высшей квалификационной категории - 1 чел.

МАТЕРИАЛЬНО-ТЕХНИЧЕСКАЯ БАЗА, НЕОБХОДИМАЯ ДЛЯ ОСУЩЕСТВЛЕНИЯ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ПО ПРОГРАММЕ

Требования к зданию/помещению

Для реализации программы помещение должно удовлетворять строительным, санитарным и противопожарным нормам.

Учебные кабинеты укомплектованы удобными рабочими местами за ученическими столами в соответствии с ростом обучающихся, состоянием их зрения и слуха.

Кабинеты информатики оборудованы в соответствии с гигиеническими требованиями, предъявляемыми к видеодисплейным терминалам, персональным электронно-вычислительным машинам и организации работы с ними. Используемые цифровые образовательные ресурсы, инструменты учебной деятельности (программные средства) лицензированы для использования во всём учреждении или на необходимом количестве рабочих мест. В работе используются комплекты лицензионного или свободно распространяемого программного обеспечения.

В целях организации антитеррористической защищённости охрана здания учреждения должна быть обеспечена системой наружного видеонаблюдения, пропускным режимом и штатными охранниками. Территория учреждения должна иметь периметральное ограждение и наружное освещение в темное время суток.

Учебно-методическое и информационное обеспечение программы

Аудитории:

- аудитория для теоретических занятий с необходимой ученической мебелью, пластиковой доской;
- компьютерный класс на 12 ученических и 1 учительское место;
- коворкинг-зона.

Технические средства и оборудование:

- проекционное оборудование;
- персональные компьютеры с выходом в сеть интернет и необходимым для стандартного функционирования программным обеспечением;
- обучающие и демонстрационные файлы;
- черно-белый лазерный принтер;
- белая бумага для стандартной печати формата А4;
- маркеры для пластиковой доски;
- сплитсистема.

Лицензионное программное обеспечение:

- Операционная система Linux;
- Среда разработки Visual Studio Code;
- Офисный пакет LibreOffice.

Средства защиты:

- антибактериальные салфетки;
- антибактериальный спрей;
- огнетушитель;
- рециркулятор.

УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ПРОГРАММЫ

1. Перечень литературы, необходимой для освоения программы

1.1. Перечень литературы, использованной при написании программы

1. Лутц, М. (2013). «Изучаем Python» (4-е изд.). – Москва: Издательство «Вильямс». – 1648 с.
Оригинальное название: Learning Python.

2. Шульженко, С. (2019). *Python. Книга для начинающих*. – Санкт-Петербург: Питер. – 416 с.

3. Миллингтон, И. (2018). «Python. Краткий курс». – Санкт-Петербург: Питер. – 304 с.
Оригинальное название: Python in a Nutshell.

4. МакКинни, У. (2013). Python для анализа данных. – Москва: О’Рэйли. – 504 с.
Оригинальное название: Python for Data Analysis.

5. Рамальо, Л. (2015). Fluent Python. – Санкт-Петербург: Питер. – 552 с.
Оригинальное название: Fluent Python.

6. Свейгарт, А. (2015). Automate the Boring Stuff with Python. – Москва: Издательство «Вильямс». – 400 с.
Оригинальное название: Automate the Boring Stuff with Python.

1.2. Перечень литературы, рекомендованной обучающимся

1. Рэш, А. (2020). «Python для научных вычислений». – Москва: Издательство «Вильямс». – 432 с.
Оригинальное название: Python for Scientific Computing.

2. Гудич, М. (2021). «Изучаем сценарии на Python». – Санкт-Петербург: Питер. – 480 с.

3. Роули, К. (2018). «Python и создание игр». – Москва: Бомбора. – 400 с.
Оригинальное название: Python Crash Course.

4. Ван Россум, Г. (2018). «Python. Подробное руководство». – Москва: Издательство «БХВ-Петербург». – 600 с.
Оригинальное название: Python Essential Reference.

1.3. Перечень литературы, рекомендованной родителям

1) Кови С. «7 навыков высокоэффективных людей. Мощные инструменты развития личности» - Альпина Паблишер, 2019

2) Ицхак Пинтусевич «Действуй! 10 заповедей успеха» изд. Эксмо 2018 г.

3) Стивен Кови «Восьмой навык. От эффективности к величию» «Альпина Паблишер», 2020 г.

2. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения программы:

1. Документация по Python
   Сайт: pythonworld.ru
   Описание: Русскоязычная документация и материалы по языку Python, включая справочники и примеры кода.

2. Stepik - Изучение Python
   Сайт: [stepik.org](https://stepik.org/course/67)
   Описание: Бесплатные курсы по Python, охватывающие основы и более сложные концепции, с интерактивными заданиями.

3. GeekBrains - Python для начинающих
   Сайт: [geekbrains.ru](https://gb.ru/courses/python)
   Описание: Онлайн-курсы по языку Python, предлагающие структурированные материалы и проекты для практики.

4. Хабр - Python
   Сайт: [habr.com](https://habr.com/ru/hub/python/)
   Описание: Комьюнити и статьи по Python на русском языке, где опытные разработчики делятся знаниями, полезными материалами и решениями задач.

Вот обновлённый список литературы с указанием авторов, даты издания, издательств и оригинальных названий:

Список литературы:

1. "Изучаем Python", Автор: Марк Лутц , Дата издания: 2013 (4-е издание), Издательство: "Вильямс", (Оригинальное название: "Learning Python")

2. "Python для анализа данных", Автор: Уэс Маккинни, Дата издания: 2012 (1-е издание),Издательство: "Вильямс". (Оригинальное название: "Python for Data Analysis")

3. "Python. Книга для начинающих", Автор: Майк О'Салливан, Дата издания: 2015, Издательство: "Питер". (Оригинальное название: "Python. A Beginner's Guide")
4. "Автоматизация рутины с помощью Python", Автор: Ала Свигарт, Дата издания: 2015, Издательство: "Питер", (Оригинальное название: "Automate the Boring Stuff with Python")

Интернет-ресурсы:

1. Документация по Python
Сайт: pythonworld.ru
Описание: Русскоязычная документация по Python.

2. Stepik - Изучение Python
Сайт: stepik.org
Описание: Бесплатный курс по Python с интерактивными заданиями.

3. GeekBrains - Python для начинающих
Сайт: geekbrains.ru
Описание: Онлайн-курс по языку Python с практическими заданиями.

4. Хабр - Python
Сайт: habr.com
Описание: Сообщество с множеством статей, туториалов и решений задач по Python.

5. Курс "Программирование на Python" от Coursera
Сайт: coursera.org
Описание: Онлайн-курс, предлагающий основы программирования на Python и его применение.

6. Учебный ресурс "Python для начинающих" на сайте "Программирование на Python"
Сайт: programming-book.ru
Описание: Учебные материалы, примеры и задачи по Python для начинающих программистов.