**Отчет по самостоятельной работе №3**

**по дисциплине МДК 01.02 “Инструментальные средства разработки программного обеспечения”.**

Выполнил: студент

группы 319

Меркушин Евгений Романович

Дата 17.11.2024

**Цель работы**

Изучение и применение системы контроля версий на практике, включая настройку типов импортируемых файлов, путей, фильтров и других параметров для эффективного управления репозиториями.

**Основная структура задания**

Для создания базы данных интернет-магазина можно использовать SQLite, так как это простая и удобная для работы СУБД, которая не требует установки отдельного сервера баз данных.

# Шаг 1: Создание таблицы товаров

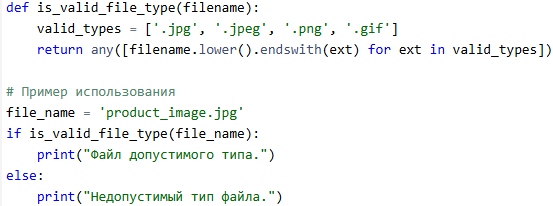
Предположим, что у нас есть таблица products, где хранятся данные о товарах. Вот пример кода на Python с использованием библиотеки sqlite3:



# Шаг 2: Определение типов файлов

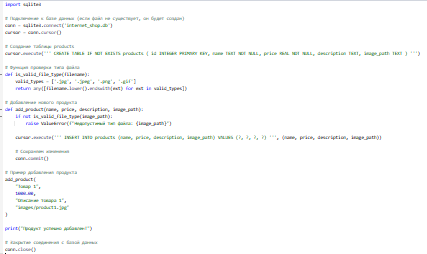
Теперь определим список допустимых типов файлов для изображений товаров. Допустимые форматы могут быть, например, .jpg, .png, .gif. Мы можем хранить эти файлы непосредственно в базе данных в виде бинарных данных (BLOB), но чаще их хранят на диске, а в базе данных указывают путь к файлу.

Пример функции проверки типа файла по его расширению:



# Полный код

Вот полный пример программы, включающий создание базы данных и проверку типов файлов:

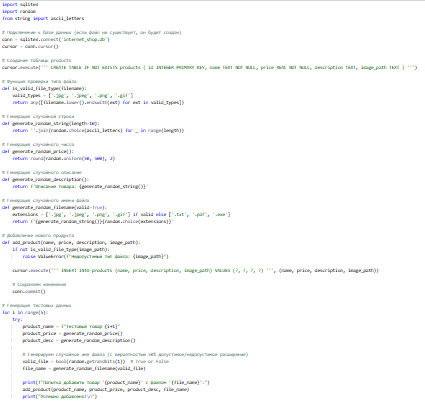


Этот скрипт создает таблицу products в базе данных internet\_shop.db, добавляет новый продукт и проверяет корректность типа файла перед добавлением записи.

Дополним код, чтобы добавить несколько тестовых файлов и убедиться, что недопустимые типы файлов действительно исключаются при попытке добавить товар.

# Обновленный код

В этом примере мы создадим функцию для генерации случайного набора продуктов с разными типами файлов, включая допустимые и недопустимые. Затем попробуем добавить каждый из этих продуктов в базу данных и убедимся, что те продукты, у которых тип файла недопустим, вызывают исключение.



# Объяснение обновленного кода

1. **Генерация случайных данных**:
   * Используются функции для генерации случайных строк, чисел и описаний.
   * Функция generate\_random\_filename() генерирует случайные имена файлов с допустимыми и недопустимыми расширениями.
2. **Цикл для тестирования**:
   * В цикле создается пять тестовых продуктов, для каждого из которых случайно выбирается допустимость расширения файла.
   * Если файл имеет недопустимое расширение, то вызывается исключение ValueError.

# Результат выполнения

При запуске этого скрипта вы увидите примерно такой вывод:

Попытка добавить товар 'Тестовый товар 1' с файлом 'qUodjWmQ.pdf':

Ошибка: Недопустимый тип файла: qUodjWmQ.pdf

Попытка добавить товар 'Тестовый товар 2' с файлом 'pHxLmOwD.png':

Успешно добавлено!

Попытка добавить товар 'Тестовый товар 3' с файлом 'kZKmuoSw.exe':

Ошибка: Недопустимый тип файла: kZKmuoSw.exe

Попытка добавить товар 'Тестовый товар 4' с файлом 'FvczUpVb.gif':

Успешно добавлено!

Попытка добавить товар 'Тестовый товар 5' с файлом 'AaxgXhWh.txt':

Ошибка: Недопустимый тип файла: AaxgXhWh.txt

Как видно из вывода, попытки добавить товары с недопустимыми типами файлов завершились ошибкой, тогда как допустимые файлы были успешно добавлены в базу данных.

Начало формы