

Пример решения задания 1 от эксперта

1. Выбранные библиотеки Python:

- requests
- BeautifulSoup4.

2. Программа содержится в файле **code_parsing.py**:

- В верхней части файла выполняется импорт библиотек **requests** и **Beautiful Soup**. Также создаем функцию с названием **main**, которая будет вызываться в момент запуска программы.
- В переменной **url** сохраняется адрес страницы, с которой будет поступать информация.
- Переменная **url** передается функции **requests.get()** для получения кода ответа от сервера и самого ответа в виде данных с HTML-страницы. Результат функции **requests.get()** присваивается переменной **page**.
- Проверяем статус ответа сервера, и если он не равен (!=) 200, то выводим сообщение из вложения 1 и завершаем выполнение программы.
- Используем конструктор **BeautifulSoup()**, чтобы создать экземпляр класса для дальнейшей работы по парсингу и поместить его в переменную **soup**. Для парсинга выбираем **html.parser**.
- Выводим переменную **all_cards** с помощью **print**.
- Для поиска карточки на странице используем метод из BeautifulSoup, который называется **findall** для поиска всех карточек на сайте с названием класса **'search-registry-entry-block box-shadow-search-input'**. Это нужно для того, чтобы не сразу искать номер карточки и сумму, а сначала определить, что карточка существует на странице.
- Создаем пустой список для сохранения карточек **all_cards_saved**. Это также необходимо для корректной инициализации переменной по списку.
- Создаем переменные **number_card** и **amount_card** и инициализируем их значениями **None** в цикле для того, чтобы избежать дублирования данных по номеру карточки и по сумме. В **class_** указываем html-данные из вложения 1.
- Если карточка на сайте существует, то с помощью выбранных данных присваиваем номер карточки и суммы в соответствующие переменные (**number_card, amount_card**).
- С помощью **append** вставляем номер карточки и сумму по карточке в список **all_cards_saved**, который затем печатаем на экране.
- Для того чтобы программа запускалась, необходимо прописать соответствующий код для корректного запуска программы с указанием функции **main** для запуска:

```
if __name__ == '__main__':  
    main()
```



Пример решения задания 2 от эксперта

Код содержится в файле `test_unit`

Пример решения задания 3 от эксперта

1. Код для создания БД содержится в файле `base_create.py`

В коде выполняются следующие операции:

- Импорт библиотеки `sqlite3`.
- Создание файла `cards.db` (это стандартный способ управления базой SQLite) с помощью функции `connect`. За соединение будет отвечать переменная `conn`.
- После создания объекта соединения с БД создадим объект `cursor` – используем переменную `cur` для его хранения.
- За выполнение запроса отвечает функция `execute`, в которой передается запрос SQL для создания таблицы `cards`. Здесь **IF NOT EXISTS** позволит не создать таблицу, если таблица уже создана в базе данных.
- Создаем две колонки: `number` и `amount`. В случае если у нас есть несколько значений, то лучше сначала создать переменную `val` для их хранения. Затем использовать конструкцию `cur.executemany("INSERT INTO cards VALUES(?, ?);", val)`. Здесь два вопросительных знака, так как у нас в таблице будет две колонки.
- Сохраняем изменения с помощью функции `commit` для объекта соединения.
- Для получения данных из базы для печати используем функцию `fetchall`. Создадим переменную `results` для хранения всех результатов.