**Министерство образования Республики Беларусь   
Учреждение образования «Брестский государственный   
технический университет»   
Кафедра интеллектуальных информационных технологий**

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**К КУРСОВОМУ ПРОЕКТУ ПО ДИСЦИПЛИНЕ**

**«Объектно-ориентированно программирование»**

**Тема: «Создание приложения ToDoList»**

**КП.АС-64.200093-01 81-00**

**Листов:**

**Выполнила:**Студентка 2-го курса,   
Группы АС-64   
Будник А.А.

**Проверил:**

**Старший преподаватель кафедры ИИТ**Соловчук А.М.

**Брест 2023**

Содержание

[Введение 3](#_Toc166593507)

[1. Анализ предметной области 6](#_Toc166593508)

[2. Проектирование 9](#_Toc166593509)

[2.1. Архитектура приложения 9](#_Toc166593510)

**Введение**

**1. Анализ предметной области**

Целью проекта было создание полнофункционального и интуитивно понятного приложения ToDoList, которое позволяет пользователям эффективно организовывать свои задачи и управлять делами.

**2. Объектно-ориентированное программирование**

Для реализации поставленной задачи по сбору данных с площадок интернет торговли был выбран язык программирования Python. Python - это высокоуровневый, интерпретируемый и универсальный динамический язык программирования, который акцентирует внимание на читаемости кода. Он был основан в 1991 году разработчиком Гвидо Ван Россумом. Python занимает высокие позиции среди самых популярных и быстрорастущих языков в мире [1].

Python является отличным инструментом для процесса извлечения данных с веб-сайтов. Библиотека BeautifulSoup в Python предоставляет мощные инструменты для этой задачи. Она предлагает множество простых методов и идиом Python для навигации, поиска и изменения DOM-дерева [3].

Использование Python для сбора данных с площадок интернет-торговли позволяет эффективно автоматизировать процесс мониторинга цен, что значительно экономит время и ресурсы, необходимые для анализа рынка и принятия обоснованных бизнес-решений. Преимущества языка Python следующие: Простота изучения и использования: Python отличается простотой, что делает его особенно привлекательным для начинающих программистов. Идеален для быстрой разработки: Python поддерживает интерактивную разработку, что позволяет быстро тестировать и экспериментировать с фрагментами кода. Высокая производительность: Python демонстрирует высокую производительность, благодаря умному управлению памятью и динамическому переводу кода Python в машинный код [2].

Процесс работы парсера осуществляет сбор данных о продуктах с нескольких веб-страниц. Выполняется запросы к каждой странице из предоставленного списка ссылок, затем с помощью библиотеки BeautifulSoup извлекает информацию о продуктах, такую как название и цена. Полученные данные добавляются в базу данных, а также регистрируется текущее время и количество полученных продуктов (см. рис. 2.1).

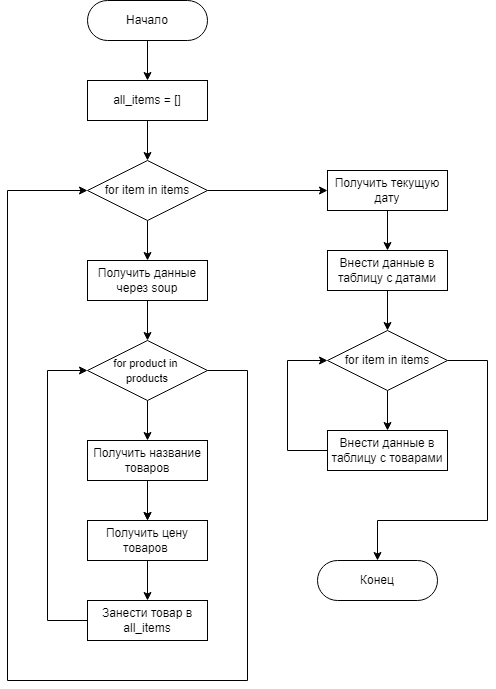


Рисунок 2.1 — блок-схема работы парсера

Для синхронного сбора данных была использована библиотека Requests. Этот инструмент предоставляет удобные средства для отправки HTTP-запросов к веб-серверам и получения ответов от них. В сочетании с другими инструментами, такими как упомянутый выше BeautifulSoup, предназначенным для анализа HTML-кода страниц, Requests становится мощным инструментом для получения данных с веб-сайтов. Для выполнения процесса получения синхронной обработки данных была написана функция sync\_parsing(). Эта функция предназначена для извлечения информации о названии и цене товара. Полученные данные затем записываются в базу данных для последующего использования. С помощью синхронного подхода и сочетания различных инструментов, обеспечивается эффективный и надежный процесс сбора и обработки данных. (см. рис. 2.2)



Рисунок 2.2 — функция sync\_parsing()

Процесс асинхронного парсинга начинается с инициализации пустого списка all\_items для хранения извлеченных данных. Цикл проходит по коллекции ссылок, выполняя асинхронные GET-запросы к каждому URL с использованием ClientSession из aiohttp. Библиотека aiohttp в Python предоставляет асинхронные HTTP-клиент и сервер. Она поддерживает как клиентскую, так и серверную сторону HTTP-протокола, а также поддерживает WebSockets. aiohttp особенно полезна в сценариях, где необходимо обрабатывать множество одновременных соединений [4]. HTML-контент, полученный от каждого запроса, затем парсится с помощью BeautifulSoup. Внутри распарсенного HTML-контента определенные элементы нацелены предположительно информация о продукте на основе их HTML-тегов и атрибутов. Название продукта и цена извлекаются из этих элементов, причем цена преобразуется в целое число после удаления всех нечисловых символов. Каждая извлеченная пара название продукта и цена добавляется в виде кортежа в список all\_items. После обработки всех ссылок текущая дата и время форматируются в виде строки и сохраняются в cur\_date, которые вместе с общим числом извлеченных элементов используются для получения идентификатора выборки из другой функции. Последний цикл проходит по всем элементам в списке all\_items, передавая каждый элемент вместе с идентификатором выборки еще одной функции предположительно для дальнейшей обработки или хранения (см. рис. 2.3).

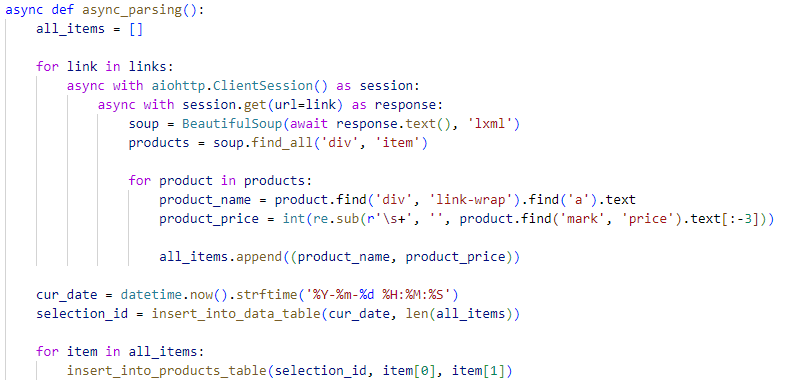


Рисунок 2.2 — функция async\_parsing()

В качестве базы данных был выбран SQL. Данный язык используется для работы с базами данных в виде таблиц. Он позволяет составлять запросы так, чтобы находить среди большого объёма информации ту, что нужна для конкретных целей, сортировать её, структурировать и представлять в наиболее простом и понятном виде [5]. SQL обеспечивает удобный доступ к данным через различные операторы, позволяющие создавать, изменять, удалять и извлекать информацию из базы данных. Он также поддерживает различные уровни безопасности данных, что является важным аспектом при работе с конфиденциальной информацией. Кроме того, SQL позволяет оптимизировать процессы доступа к данным через использование индексов и представлений, что повышает производительность при выполнении запросов. Для хранения результатов сбора информации было написано две таблицы, которые связаны между собой. Первая таблица хранит результаты отбора, где хранятся имеются поля с уникальным идентификатором отбора, датой отбора и количеством продуктов. Вторая таблица имеет поля с уникальным идентификатором, порядковым номером отбора, который берется из первой таблицы, а также имя и цену товара (см. рис. 2.3).



Рисунок 2.3 — схема базы данных

Источники (ес че, это набросок, чтобы просто потом не мучавться)

1. <https://www.geeksforgeeks.org/python-language-advantages-applications/>

2. <https://www.geeksforgeeks.org/advantages-of-python/>

3. <https://www.freecodecamp.org/news/web-scraping-python-tutorial-how-to-scrape-data-from-a-website/>

4. <https://www.geeksforgeeks.org/install-aiohttp-in-python/>

5. https://practicum.yandex.ru/blog/chto-takoe-sql/