Министерство науки и высшего образования Российской Федерации

**Муромский институт (филиал)**

федерального государственного бюджетного образовательного учреждения высшего образования

**«Владимирский государственный университет   
имени Александра Григорьевича и Николая Григорьевича Столетовых»**

(МИВлГУ)

Факультет ИТР



Кафедра ИС

КУРСОВАЯ

РАБОТА

по Прикладная разработка на Java

(наименование дисциплины)

Тема Разработка парсера сайтов для выбора принтера

Руководитель

ФИО РУКОВОДИТЕЛЯ

(оценка) (фамилия, инициалы)

Члены комиссии

(подпись) (дата)

Студент ИСз-121

(подпись) (Ф.И.О.) (группа)

Баранцева А. Л.

(подпись) (Ф.И.О.) (фамилия, инициалы)

(подпись) (дата)

Муром 2024

В курсовой работе согласно задания разработан парсер сайта avito.ru для выбора принтера. В ходе работы сделан анализ технического задания, сформированы требования к парсеру, выбраны средства реализации, а также язык программирования. Парсер реализован на языке Java 17, среда разработки VSCode со всеми необходимыми плагинами. Для автоматизации сборки использован mvn, в итоге парсер запаковывается в автономный jar архив.

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 4](#_Toc155692589)

[1. Анализ технического задания 5](#_Toc155692590)

[1.1 Анализ предметной области 5](#_Toc155692591)

[1.2 Формирование требований к системе 6](#_Toc155692592)

[2. Разработка алгоритма 7](#_Toc155692593)

[3. Реализация программы 9](#_Toc155692594)

[3.1 Настройка среды разработки. 9](#_Toc155692595)

[3.2 Реализация программы 11](#_Toc155692596)

[4. Руководство по программному продукту 14](#_Toc155692597)

[4.1 Руководство пользователя 14](#_Toc155692598)

[4.2 Руководство администратора 16](#_Toc155692599)

[5. Тестирование программного продукта 17](#_Toc155692600)

[ЗАКЛЮЧЕНИЕ 18](#_Toc155692601)

[Список литературы 19](#_Toc155692602)

ВВЕДЕНИЕ

В настоящее время происходит бурное развитие информационных технологий. Благодаря информационным технологиям и сети Интернет появляется возможность овладеть знаниями. Интернет – это глобальная сеть компьютерных ресурсов с коллективным доступом на основе использования единой стандартной схемы адресации, высокопроизводительной магистрали и высокоскоростных линий связи с главными сетевыми компьютерами (серверами). На сегодняшний день, Интернет нечто большее, чем просто способ коммуникации между людьми, это еще и самый большой источник информации в мире. И этот источник растет на глазах, ведь каждый день создаются многочисленные веб-сайты и интернет-порталы, где располагается важный и интересный контент. Однако овладеть таким количеством информации человеку не под силу, и тут на помощь приходит компьютер, в частности технология парсинга сайтов.

Парсинг сайтов – это метод извлечения большого объема данных с различных веб-сайтов. Термин «парсинг» относится к получению информации из другого источника (веб-страницы) и сохранению ее в локальном файле или базе данных. Парсинг позволяет собрать в одном месте данные из различных источников, преобразовать их в удобную для понимания человека форму, а затем использовать в своих целях.

Темой данной курсовой работы является проектирование и разработка парсера сайта avito.ru для выбора принтера на языке программирования Java.

Целью курсовой работы является изучение технологии парсинга, а также приобретение практических навыков программирования на языке Java.

Основными задачами работы являются:

- выбор средств разработки парсера;

- проектирование парсера;

- реализация парсера;

- парсер должен сохранять полученные данные, удобные для понимания человеком.

1. Анализ технического задания

1.1 Анализ предметной области

В данном курсовой работе необходимо разработать парсер сайта avito.ru, для выбора принтера. Для того чтобы найти нужный товар (например принтер) пользователю сайта нужно зайти на сайт, выбрать нужную категорию, задать параметры поиска, выбрать нужный город. Если это разовый поиск, то можно пользоваться только сайтом. Но для анализа информации нужно постоянно заходить на сайт производить поиск, записывать каким-то образом найденные данные и это нужно делать циклически. Парсер поможет автоматизировать процесс поиска и сохранения информации, для последующего анализа.

Пользователь для исследования динамики цен на определенный товар, запускает парсер, который периодически опрашивает сайт avito.ru и сохраняет полученные данные. В итоге пользователь получит необходимые отчеты, полученные с различным интервалом времени.

У парсинга сайтов есть преимущества и недостатки.

Преимущества:

- Скорость;

- Сбор большого объема данных;

- Экономическая эффективность;

- Надежность;

- Низкие затраты на техническое обслуживание.

Недостатки:

- Парсинг веб-страниц требует постоянного обслуживания;

- Парсеры могут быть заблокированы.

1.2 Формирование требований к системе

Исходя из приведенного выше анализа предметной области, сформулированы следующие основные требования к разрабатываемому парсеру:

- разрабатываемый парсер должен быть реализован на языке Java;

- парсер должен быть собран в автономный jar архив, со всеми необходимыми зависимостями;

- парсер должен собирать данные для выбора принтера в г. Кулебаки:

- название товара;

- описание товара;

- цену;

- денежную единицу.

- после произведенного поиска, результаты должны быть сохранены в CSV файл.

- при работе парсера должны быть учтены исключительные ситуации:

- отсутствие ответа от сайта avito.ru;

- ошибка записи в CSV файл.

- для получения данных с сайта используется библиотека Jsoup;

- записи должны быть отсортированы по возрастанию цены;

- для запуска приложения у пользователя должна установлена только Java Runtime Environment – минимальная (без компилятора и других средств разработки) реализация виртуальной машины, необходимая для исполнения Java-приложений.

2. Разработка алгоритма

Алгоритм парсера, представленный в виде блок – схемы:

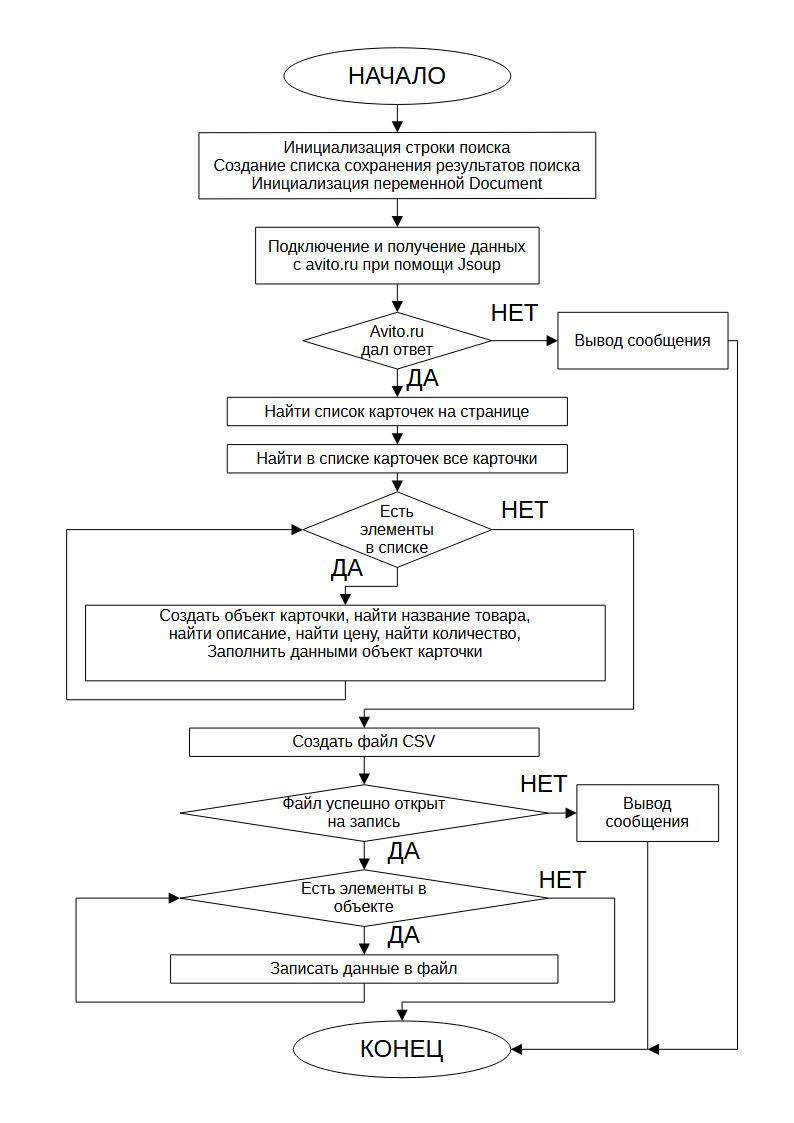


Рисунок 1 – Блок-схема алгоритма парсера

После запуска приложения парсера, инициализируются:

- строка поиска;

- объект Document;

Строка поиска — это поисковый запрос на сайт avito.ru, в строке можно указать нужные нам параметры.

После выполнения запроса Jsoup создаст виртуальную копию полученной страницы в объекте Document.

Поиск нужных данных будет производится в объекте Document, а результаты будут сохранятся в список объектов, при старте список объектов объявляется и инициализируется пустым.

При помощи инструмента Jsoup подключаемся к avito.ru получаем данные. При получении avito.ru может не дать ответа, нужно обработать данную ситуацию (вывод сообщения и выход из программы).

После того как объект Document будет заполнен данными производим поиск списка карточек, для параметров поиска используем CSS селекторы и дата-атрибуты.

В списке карточке находим необходимые нам данные в виде карточек товара.

В цикле обходим найденные карточки и для каждой создаем объект карточки, заполняем объект данными и сохраняем в списке.

Для вывода результатов работы необходимо сохранить и представить данные в удобной для человека форме. Для этого используем формат CSV. Создаем файл (пустой) и открываем его на запись. На этом шаге возможно получить исключительную ситуацию (ошибка ввода-вывода), обрабатываем ее (вывод сообщения и завершение программы). Заносим строковое представление каждого объекта в файл (в цикле). Завершаем программу. В итоге у нас будет CSV файл с нужными нам данными.

3. Реализация программы

3.1 Настройка среды разработки.

- на сайте разработчика скачиваем устанавливаем Java Development Kit LTS 17 версии <https://www.oracle.com/java/technologies/downloads/#jdk17>

- устанавливаем и проверяем версию java --version. В консоли получаем информацию о установленной версии java.

- в разработке используем сборщик Maven, он позволит установить все зависимости в проект, а также собрать автономный jar архив. Для установки скачиваем сам сборщик, распаковываем и добавляем в переменную PATH пусть к папке bin. Это позволит запускать сборщик без указания полного пути.

- для разработки используем IDE VSCode, а также плагин Extension Pack for Java + все его зависимости.

- после установки VSCode с плагинами, после запуска появляется вкладка Maven, выбираем в этой вкладке Создать проект.

- во время создания проекта нужно указать параметры:

- maven-archetype-quickstart;

- версия 1.4 (последняя);

- group id указываем com.asupstudent (группа проектов);

- artifact id указываем avito название проекта (папка с проектом);

- указываем папку с проектом на диске;

- после создания проекта скачивается все необходимые зависимости;

- по ходу скачивания указываем версию проекта 1.0;

- проверяем все что задавали в процессе и подтверждаем Y;

- VSCode предложит открыть проект, открываем;

- Проект создан.

В текущем состоянии создан пустой проект, нужно добавить нужные нам зависимости в файле pom.xml нужно добавить в зависимость Jsoup:

<dependency>

<groupId>org.jsoup</groupId>

<artifactId>jsoup</artifactId>

<version>1.17.1</version>

</dependency>

Для создания автономного jar архива добавляем в секцию build 2 плагина:

- maven-assembly-plugin;

- maven-jar-plugin;

В итоге секция build будет выглядеть следующим образом:

<build>

<plugins>

<plugin>

<artifactId>maven-assembly-plugin</artifactId>

<configuration>

<archive>

<manifest>

<mainClass>com.asupstudent.App</mainClass>

</manifest>

</archive>

<descriptorRefs>

<descriptorRef>jar-with-dependencies</descriptorRef>

</descriptorRefs>

</configuration>

<executions>

<execution>

<id>make-assembly</id>

<phase>package</phase>

<goals>

<goal>single</goal>

</goals>

</execution>

</executions>

</plugin>

<plugin>

<artifactId>maven-jar-plugin</artifactId>

<configuration>

<archive>

<manifest>

<addClasspath>true</addClasspath>

<classpathPrefix>lib/</classpathPrefix>

<mainClass>com.asupstudent.App</mainClass>

</manifest>

</archive>

</configuration>

</plugin>

</plugins>

</build>

Важно указать (2 вхождения):

<mainClass>com.asupstudent.App</mainClass>

Для сборки jar архива используем команду:

mvn clean compile assembly:single

Сборщик очищает, компилирует и воздает jar архив.

Чтобы запустить собранный jar архив используем команду:

java -jar avito-1.0-jar-with-dependencies.jar

В итоге получаем в консоли HelloWorld, на этом шаге сборка настроена.

3.2 Реализация программы

Парсер состоит из 2 классов:

- Точка входа в приложения (сам парсер) App.java;

- Класс карточки товара Card.java.

Реализация App.java:

package com.asupstudent;

import java.io.File;

import java.io.IOException;

import java.io.PrintWriter;

import java.nio.charset.StandardCharsets;

import java.util.ArrayList;

import java.util.List;

import org.jsoup.Jsoup;

import org.jsoup.nodes.\*;

import org.jsoup.select.Elements;

public class App

{

public static void main( String[] args )

{

String baseUrl = "https://www.avito.ru/kulebaki/orgtehnika\_i\_rashodniki/printery-ASgBAgICAUSoAoQK?s=1";

List<Card> cards = new ArrayList<>();

Document document = null;

try {

document = Jsoup.connect(baseUrl)

.timeout(10 \* 1000)

.userAgent("Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/118.0.0.0 YaBrowser/23.11.0.0 Safari/537.36")

.referrer("http://www.google.com")

.get();

}

catch(IOException ex) {

System.out.println("Авито не дал ответ, нужно запустить еще раз");

return;

}

// Нашли список карточек

Element itemsList = document.selectFirst("div[data-marker=catalog-serp]");

//Нашли в списке все карточки

Elements items = itemsList.select("div[data-marker=item]");

for(Element item: items) {

Card card = new Card();

/\*НАЗВАНИЕ\*/

Element title = item.selectFirst("h3[itemprop=name]");

/\*ОПИСАНИЕ\*/

Element description = item.selectFirst("meta[itemprop=description]");

/\*ЦЕНА + ДЕНЕЖНАЯ ЕДИНИЦА\*/

Element priceContainer = item.selectFirst("p[data-marker=item-price]");

Element priceCurrency = priceContainer.selectFirst("meta[itemprop=priceCurrency]");

Element priceCount = priceContainer.selectFirst("meta[itemprop=price]");

/\*СОЗДАЕМ ОБЪЕКТ Card\*/

card.setTitle(title.text());

card.setDescription(description.attr("content"));

card.setPrice(priceCount.attr("content"));

card.setCurrency(priceCurrency.attr("content"));

cards.add(card);

/\*ВЫВОД В КОНСОЛЬ\*/

// System.out.println(title.text());

// System.out.println(description.attr("content"));

// System.out.println(priceCount.attr("content") + " " + priceCurrency.attr("content"));

}

File csvFile = new File("output.csv");

try (PrintWriter printWriter = new PrintWriter(csvFile, StandardCharsets.UTF\_8)) {

// Обработать BOM

printWriter.write('\ufeff');

for (Card card : cards) {

List<String> row = new ArrayList<>();

row.add("\"" + card.getTitle() + "\"");

row.add("\"" + card.getDescription() + "\"");

row.add("\"" + card.getPrice() + "\"");

row.add("\"" + card.getCurrency() + "\"");

// Вывод строки в CSV файл

printWriter.println(String.join(",", row));

}

}

catch(IOException ex) {

System.out.println("Ошибка записи в CSV файл");

return;

}

}

}

Реализация Card.java:

package com.asupstudent;

public class Card {

private String title;

private String description;

private String price;

private String currency;

public String getTitle() {

return title;

}

public void setTitle(String title) {

this.title = title;

}

public String getDescription() {

return description;

}

public void setDescription(String description) {

this.description = description;

}

public String getPrice() {

return price;

}

public void setPrice(String price) {

this.price = price;

}

public String getCurrency() {

return currency;

}

public void setCurrency(String currency) {

this.currency = currency;

}

}

4. Руководство по программному продукту

4.1 Руководство пользователя

Для запуска парсера нужно:

- запустить консоль, выполнить команду cmd.exe;

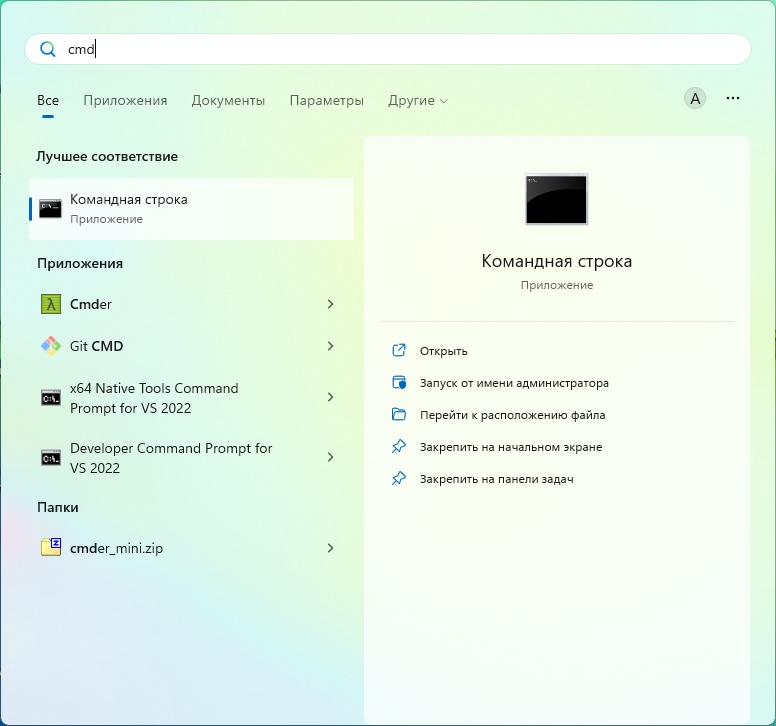


Рисунок 2 – Запуск консоли

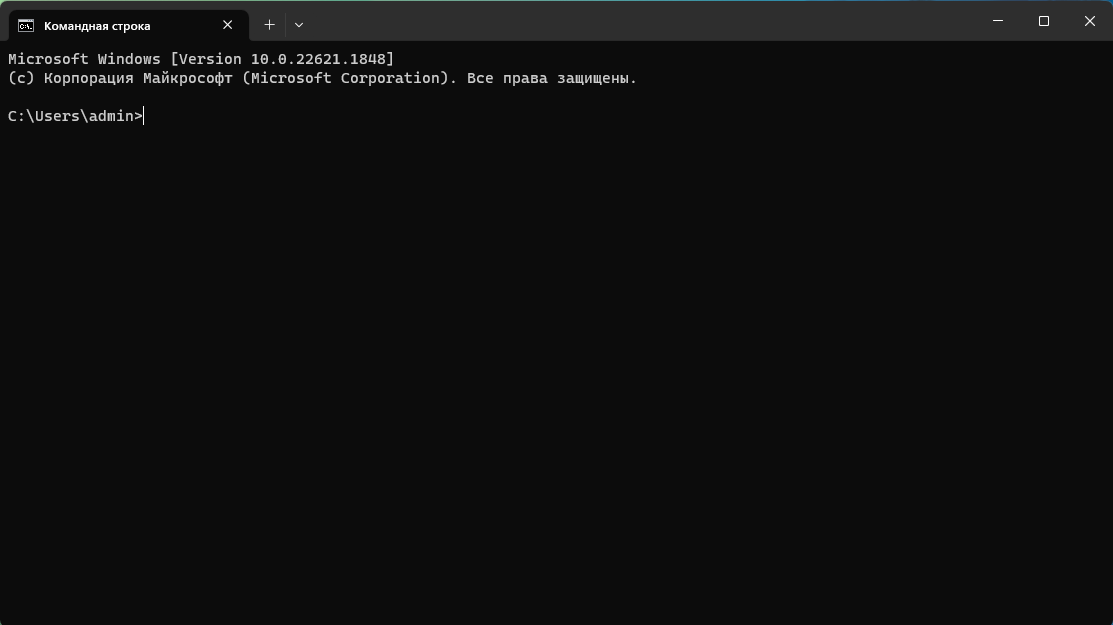


Рисунок 3 – Консоль

- перейти в каталог, где располагается jar архив с парсером;

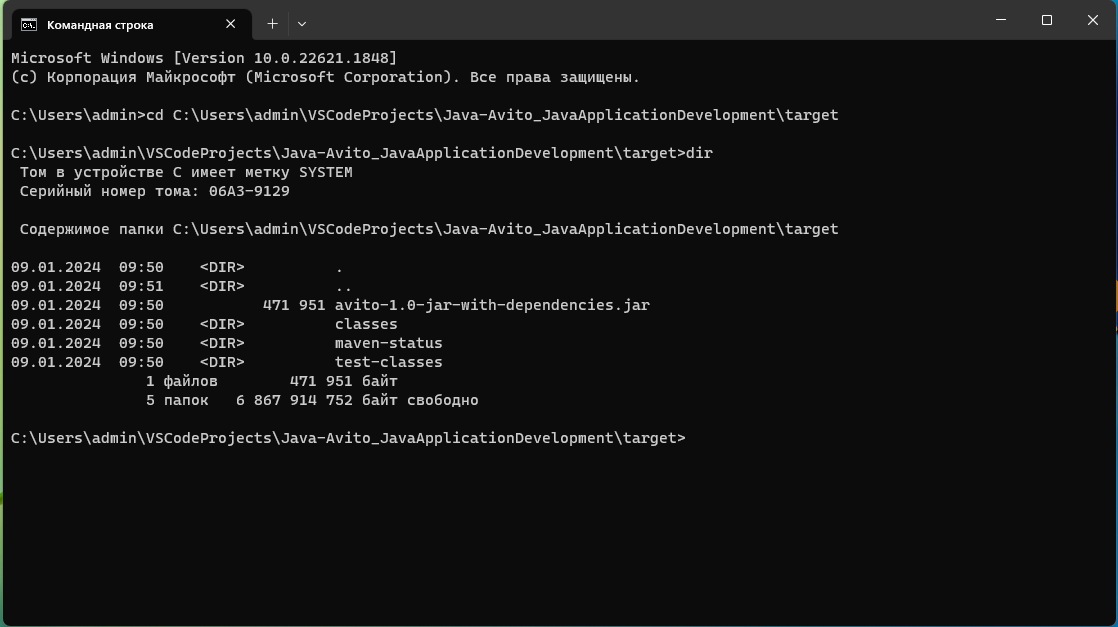


Рисунок 4 – Каталог с приложением

- запуск парсера командой:

java -jar avito-1.0-jar-with-dependencies.jar

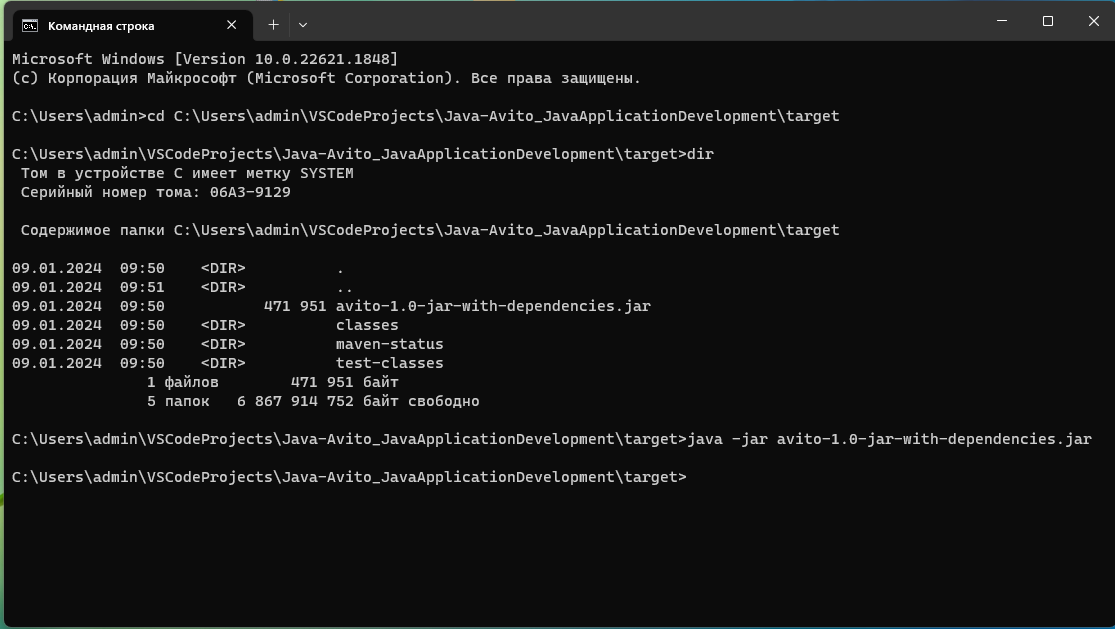


Рисунок 5 – запуск парсера

В каталоге с парсером появится файл output.csv:

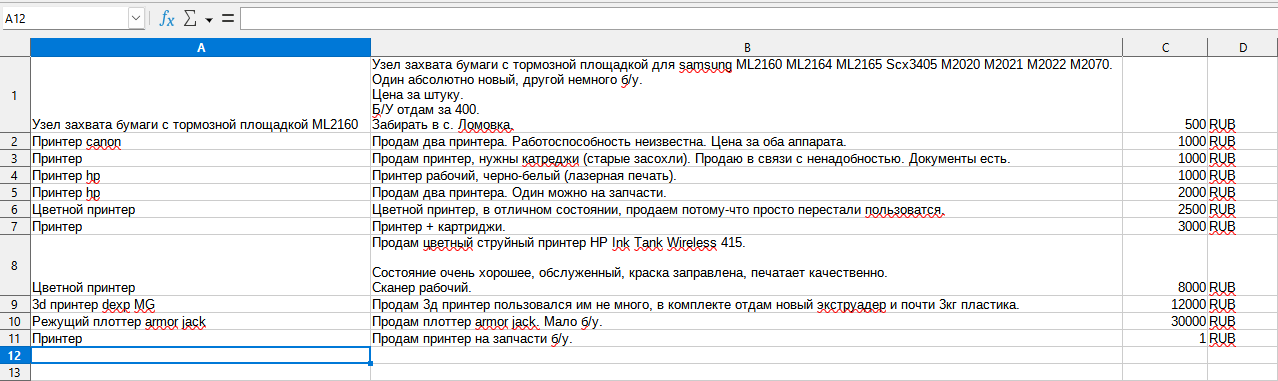


Рисунок 6 – Данные полученные парсером

4.2 Руководство администратора

Для установки парсера на компьютере пользователя администратору необходимо на клиентской машине операционная система:

- Windows 7;

- Windows 8.1;

- Windows 10;

- Windows 11.

Для работы парсера нужно установить JRE (Java Runtime Enviroment v17). Дистрибутив Java можно скачать с официального сайта:

<https://www.oracle.com/java/technologies/downloads/#jdk17-windows>

Администратору нужно перенести на ПК пользователя jar архив, создав предварительно директорию, а также сообщить пользователю, где располагается архив с приложением. Парсеру нужен доступ в интернет

5. Тестирование программного продукта

Тестирование разработанного программного продукта выполняется с целью проверки соответствия реализованного функционала требованиям технического задания, а также проверки надежности работы и отсутствия ошибок в реализации функционала. Ниже приведена методика тестирования программного продукта.

Таблица 1 Методика тестирования программного продукта

|  |  |
| --- | --- |
| Выполняемая функция | Требуемый результат |
| 1. Запуск парсера | Программа завершит работу без вывода сообщений |
| 2. Перейти в каталог парсера | Должен появится файл output.csv |
| 3. Открыть файл | В файле должны быть необходимые данные |

Результаты, полученные в ходе тестирования, позволяют сделать заключение в том, что реализация программного продукта соответствует требованиям технического задания.

ЗАКЛЮЧЕНИЕ

В данной курсовой работе в соответствии с заданием разработан парсер для сайта avito.ru для выбора принтера.

В ходе работы над курсовой работой, были выполнены следующие задачи:

- анализ технического задания;

- анализ предметной области;

- выбор средств разработки;

- проектирование парсера;

- реализация парсера;

Парсер для сайта avito.ru разработан на языке программирования Java 17 версии. Разработанная программа позволяет выполнять следующие задачи:

- получать данные о продаже принтеров в г. Кулебаки;

- название принтеров;

- описание принтеров;

- цена принтеров;

- денежная единица.

Парсер сохраняет данные в CSV файл.

Разработанный программный продукт может быть полезен для анализа цен на принтеры в г. Кулебаки.

В курсовой работе есть ряд недостатков, среди которых:

- сайт avito.ru может не всегда ответить, поэтому парсер анализирует только 1 страницу пагинации;

- в парсере нельзя выбрать другой город или другой товар;

- SCV файл создается каждый раз новый, а не дописывается в конец существующего.

Список литературы

1. Альфред, В. Ахо Компиляторы. Принципы, технологии и инструментарий / Альфред В. Ахо и др. – М.: Вильямс, 2015. – 689 с.

2. Савитч, Уолтер Язык Java. Курс программирования /Уолтер Савитч. – М.: Вильямс, 2015. – 928 с.

3. Шилдт, Герберт Java Руководство для начинающих / Герберт Шилдт. – М.: Вильямс, 2015. – 720 с.

4. Эккель, Брюс Философия Java / Брюс Эккель. – М.: Питер, 2016. – 809 с.

5. Блох, Дж. Java: эффективное программирование / Дж. Блох. - М.: Диалектика, 2019. - 464 c.