Book Web Scraper

.

Overview

book data from the "Books to Scrape" website, which displays a collection of books with details such as title, price, and rating.

Using **Java** and the **Jsoup** library, we built a program that can navigate the website, access book pages one by one, and automatically collect this information.

What makes our project stand out is that we implemented **parallelism** using **Threads** and **ExecutorService**. So instead of processing the pages sequentially (which is slow), the program scrapes multiple pages at the same time – increasing both speed and efficiency.

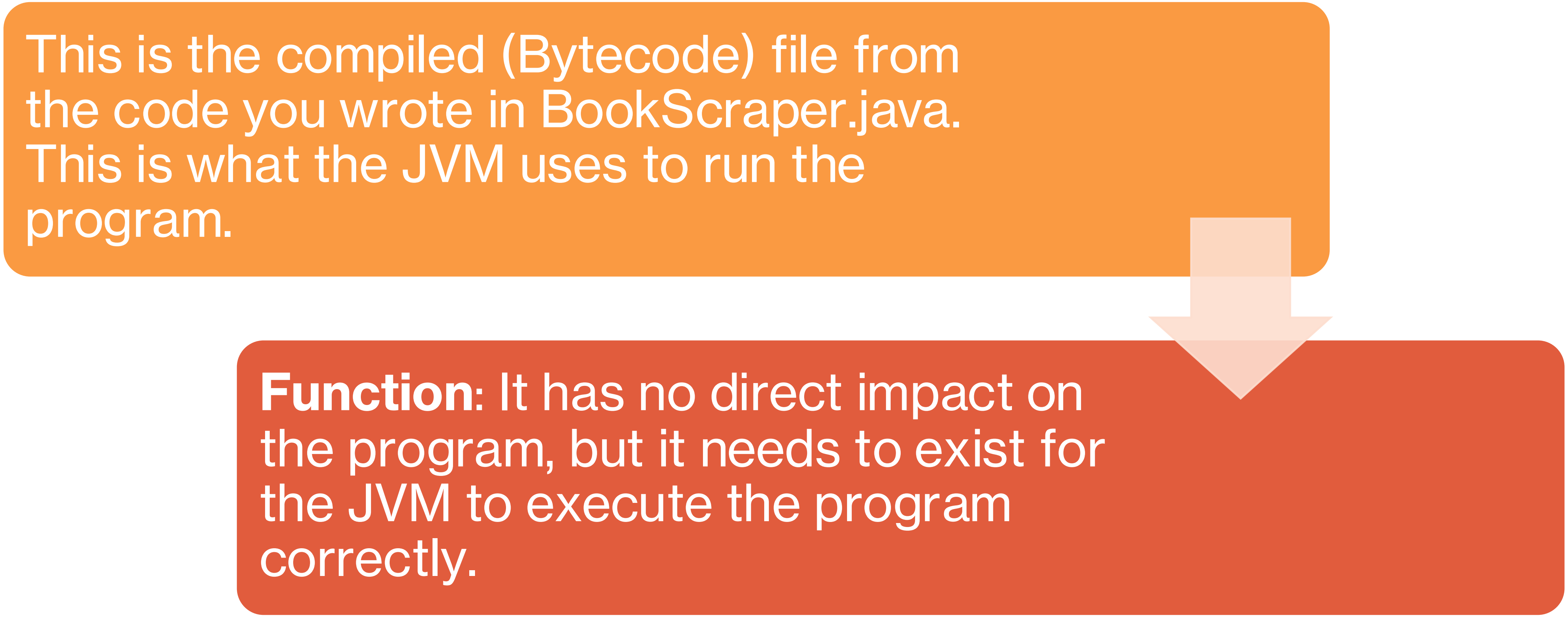
- Finally, all the collected data is saved into a **CSV file**, making it ready for analysis or presentation.

- # BookScraper.java

This is the main file where the program's core code is written. It contains the main logic of the project:

- **Main Function**: The program begins by defining the thread pool (which is a group of threads) using ExecutorService to execute the scraping across multiple pages simultaneously.
- The program starts with page 1 of the site and iterates through the following pages one by one (based on the page number), until it reaches the last page.
- For each page, a new task (ScraperTask) is created to execute in a separate thread and gather data for each book like the title, price, and rating.

BookScraper.class



This is the compiled (Bytecode) file from the code you wrote in BookScraper.java. This is what the JVM uses to run the program.

Function: It has no direct impact on the program, but it needs to exist for the JVM to execute the program correctly.

BookScraper$ScraperTask.class

This file contains the class responsible for the actual scraping process. When you call ExecutorService to start the work, this is the code that gets executed.

- **Function**: It holds the logic for navigating through each page and extracting data like the title, price, and rating.
- Each task (task) runs in a separate thread using Runnable, and it performs the scraping for one page.

books.csv

This is the final output file. All the data the program collects from the site is stored in this file in CSV format (Comma Separated Values), where each line contains information about one book, such as:

- Title
- Price
- Rating

jsoup.jar

This is the Jsoup library file used in the project.
Jsoup is a Java library that helps in reading and parsing HTML pages, such as navigating the page and selecting the elements from which you want to extract data (like the title and price).

Function: Its primary function is to allow you to navigate HTML content and clean it up so that you can easily gather data.

conclusion

In our project, we used Threads to achieve Parallelism, meaning the program can execute more than one task at the same time. Instead of working on one page and then moving to the next, Threads allowed us to scrape multiple pages simultaneously, thus increasing the program's execution speed.

The most important aspect of Threads is synchronization to prevent data issues or conflicts, especially when more than one Thread tries to write at the same time in the same location. Therefore, we used synchronized to ensure that each Thread writes to the data safely and without any conflicts.