# HW8

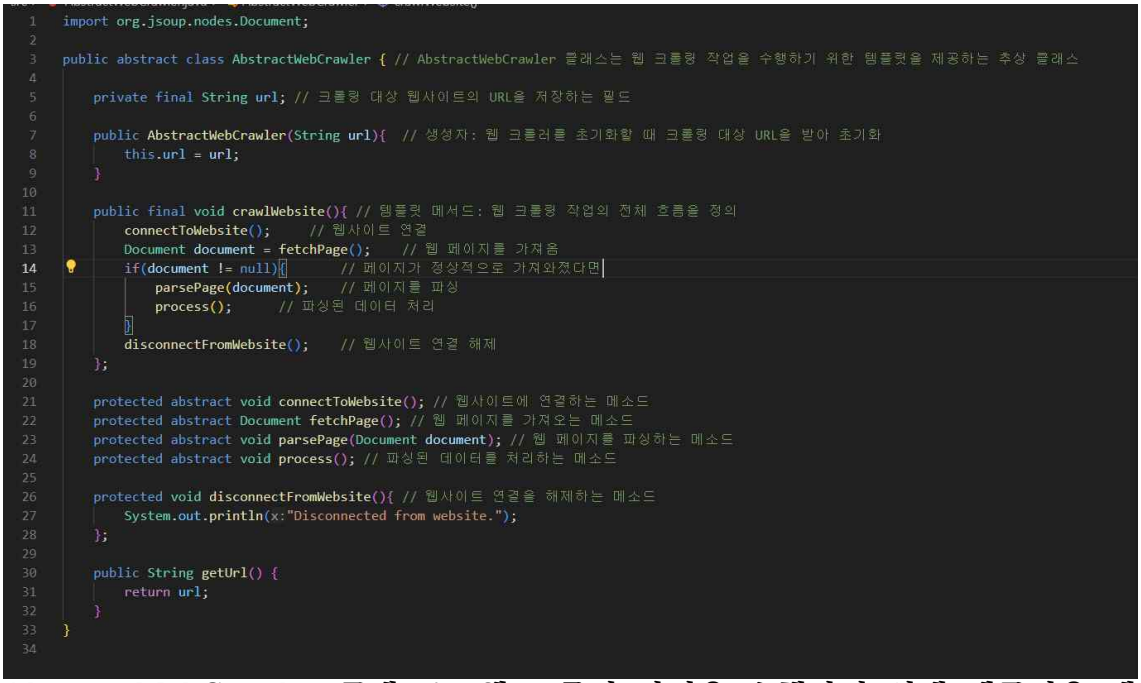
1분반

2024년 11월 27일

32211792

박재홍

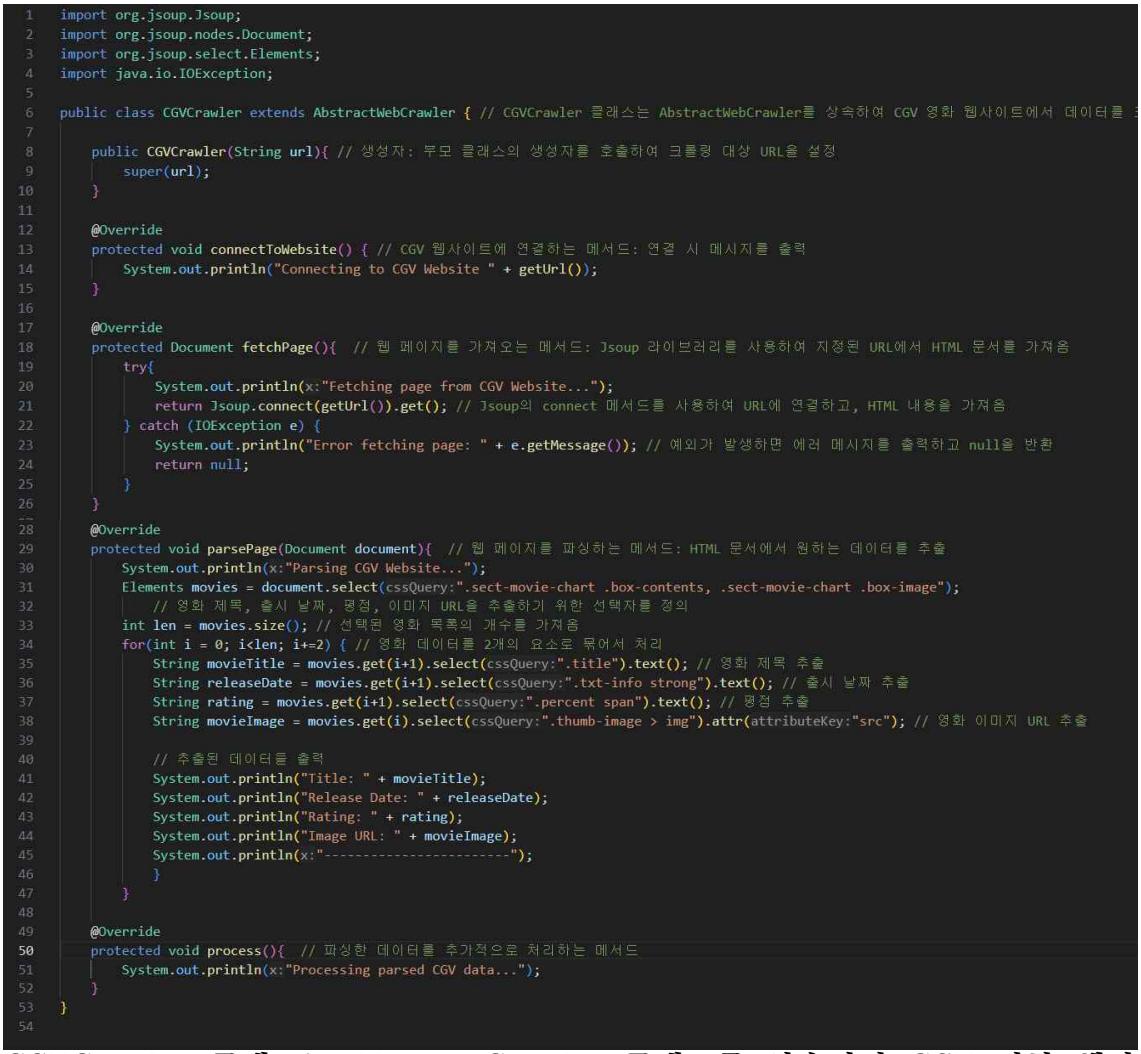
# AbstractWebCrawler 클래스)



```java
import org.jsoup.nodes.Document;

public abstract class AbstractWebCrawler { // AbstractWebCrawler 클래스는 웹 크롤링 작업을 수행하기 위한 템플릿을 제공하는 추상 클래스

    private final String url; // 크롤링 대상 웹사이트의 URL을 저장하는 필드

    public AbstractWebCrawler(String url){  // 생성자: 웹 크롤러를 초기화할 때 크롤링 대상 URL을 받아 초기화
        this.url = url;
    }

    public final void crawlWebsite(){ // 템플릿 메서드: 웹 크롤링 작업의 전체 흐름을 정의
        connectToWebsite();     // 웹사이트 연결
        Document document = fetchPage();    // 웹 페이지를 가져옴
        if(document != null){       // 페이지가 정상적으로 가져와졌다면
            parsePage(document);    // 페이지를 파싱
            process();      // 파싱된 데이터 처리
        }
        disconnectFromWebsite();    // 웹사이트 연결 해제
    };

    protected abstract void connectToWebsite(); // 웹사이트에 연결하는 메소드
    protected abstract Document fetchPage(); // 웹 페이지를 가져오는 메소드
    protected abstract void parsePage(Document document); // 웹 페이지를 파싱하는 메소드
    protected abstract void process(); // 파싱된 데이터를 처리하는 메소드

    protected void disconnectFromWebsite(){ // 웹사이트 연결을 해제하는 메소드
        System.out.println(x:"Disconnected from website.");
    };

    public String getUrl() {
        return url;
    }
}
```

AbstractWebCrawler 클래스는 웹 크롤링 작업을 수행하기 위해 템플릿을 제공하는 추상 클래스이다. 우선적으로 크롤링 대상 웹사이트의 URL을 저장하는 필드를 선언해주고 생성자를 생성해 웹 크롤러를 초기화할 때 크롤링 대상의 URL을 받아 초기화 할 수 있게 해주었다. 그 다음 crawlwebsite 메소드를 선언해 웹 크롤링 작업의 전체 흐름을 정의해주었다. 그 안에는 웹사이트를 연결해주고 웹 페이지를 가져오고 페이지가 정상적으로 가져와졌다면 페이지를 파싱하도록 하고 그 다음 파싱된 데이터를 처리할 수 있게 해주었고 마지막에는 웹사이트의 연결을 해제해주었다. 그 다음 웹사이트에 연결하는 메소드를 선언, 웹 페이즈를 가져오는 메소드 선언, 웹 페이즈를 파싱하는 메소드 선언, 파싱된 데이터를 처리하는 메소드, 웹사이트 연결을 해제하는 메소드를 선언해주었다.

# CGVCrawler 클래스)



```java
import org.jsoup.Jsoup;
import org.jsoup.nodes.Document;
import org.jsoup.select.Elements;
import java.io.IOException;

public class CGVCrawler extends AbstractWebCrawler { // CGVCrawler 클래스는 AbstractWebCrawler를 상속하여 CGV 영화 웹사이트에서 데이터를 

    public CGVCrawler(String url){ // 생성자: 부모 클래스의 생성자를 호출하여 크롤링 대상 URL을 설정
        super(url);
    }

    @Override
    protected void connectToWebsite() { // CGV 웹사이트에 연결하는 메서드: 연결 시 메시지를 출력
        System.out.println("Connecting to CGV Website " + getUrl());
    }

    @Override
    protected Document fetchPage(){  // 웹 페이지를 가져오는 메서드: Jsoup 라이브러리를 사용하여 지정된 URL에서 HTML 문서를 가져옴
        try{
            System.out.println(x:"Fetching page from CGV Website...");
            return Jsoup.connect(getUrl()).get(); // Jsoup의 connect 메서드를 사용하여 URL에 연결하고, HTML 내용을 가져옴
        } catch (IOException e) {
            System.out.println("Error fetching page: " + e.getMessage()); // 예외가 발생하면 에러 메시지를 출력하고 null을 반환
            return null;
        }
    }
    @Override
    protected void parsePage(Document document){  // 웹 페이지를 파싱하는 메서드: HTML 문서에서 원하는 데이터를 추출
        System.out.println(x:"Parsing CGV Website...");
        Elements movies = document.select(cssQuery:".sect-movie-chart .box-contents, .sect-movie-chart .box-image");
            // 영화 제목, 출시 날짜, 평점, 이미지 URL을 추출하기 위한 선택자를 정의
        int len = movies.size(); // 선택된 영화 목록의 개수를 가져옴
        for(int i = 0; i<len; i+=2) { // 영화 데이터를 2개의 요소로 묶어서 처리
            String movieTitle = movies.get(i+1).select(cssQuery:".title").text(); // 영화 제목 추출
            String releaseDate = movies.get(i+1).select(cssQuery:".txt-info strong").text(); // 출시 날짜 추출
            String rating = movies.get(i+1).select(cssQuery:".percent span").text(); // 평점 추출
            String movieImage = movies.get(i).select(cssQuery:".thumb-image > img").attr(attributeKey:"src"); // 영화 이미지 URL 추출

            // 추출된 데이터를 출력
            System.out.println("Title: " + movieTitle);
            System.out.println("Release Date: " + releaseDate);
            System.out.println("Rating: " + rating);
            System.out.println("Image URL: " + movieImage);
            System.out.println(x:"------------------------");
            }
        }

    @Override
    protected void process(){  // 파싱한 데이터를 추가적으로 처리하는 메서드
        System.out.println(x:"Processing parsed CGV data...");
    }
}

```

CGVCrawler 클래스는 AbstractCrawler 클래스를 상속받아 CGV 영화 웹사이트에서 데이터를 크롤링 한다. 생성자를 생성해 부모 클래스의 생성자를 호출하여 크롤링 대상 URL을 설정해주었고 connectToWebsite메소드를 통해 cgv 웹사이트에 연결해 연결시 메시지를 출력하도록 설정해주었다. 그 다음 fetch 메소드를 선언해 jsoup 라이브러리를 사용해 지정된 URL에서 HTML 문서를 가져오도록 했다. 다음 웹 페이지를 파싱하는 메소드인 parsePage 메소드를 선언해 영화제목, 출시 날짜, 평점, 이미지 url을 추출하기 위한 선택자를 정의했고 선택된 영화 목록의 개수를 가져오도록 했고 영화 데이터를 2개의 요소로 묶어서 처리해 영화제목, 출시 날짜, 평점, 이미지url을 추출시켰다.

# Recipe10000Crawler 클래스)

```java
import org.jsoup.Jsoup;
import org.jsoup.nodes.Document;
import org.jsoup.select.Elements;

import java.io.IOException;

public class Recipe10000Crawler extends AbstractWebCrawler {
// Recipe10000Crawler 클래스는 AbstractWebCrawler를 상속받아 '10000 Recipe' 웹사이트에서 레시피 데이터를 크롤링
    public Recipe10000Crawler(String url){  // 생성자: 부모 클래스의 생성자를 호출하여 크롤링 대상 URL을 설정
        super(url);
    }

    @Override
    protected void connectToWebsite() { // '10000 Recipe' 웹사이트에 연결하는 메서드: 연결 시 메시지를 출력
        System.out.println("Connecting to Recipe10000 website: " + getUrl());
    }

    @Override
    protected Document fetchPage(){ // 웹 페이지를 가져오는 메서드: Jsoup 라이브러리를 사용하여 HTML 데이터를 가져옴
        try{
            System.out.println(x:"Fetching page from Recipe10000 website...");
            return Jsoup.connect(getUrl()).get(); // Jsoup 라이브러리를 사용하여 URL에서 HTML 문서를 가져옴
        } catch (IOException e) {
            System.out.println("Error fetching page: "+ e.getMessage()); // 예외 발생 시 에러 메시지를 출력하고 null을 반환
            return null;
        }
    }

    @Override
    protected void parsePage(Document document){ // 웹 페이지를 파싱하는 메서드: HTML 문서에서 레시피 데이터를 추출
        System.out.println(x:"Parsing Recipe10000 website...");
        Elements recipes = document.select(cssQuery:".common_sp_list_ul li"); // 레시피 목록 요소를 CSS 선택자로 선택함

        for (var recipe : recipes) {
            String recipeTitle = recipe.select(cssQuery:".common_sp_caption_tit").text(); // 레시피 제목 추출
            String recipeImage = recipe.select(cssQuery:"img").attr(attributeKey:"src"); // 레시피 이미지 URL 추출
            String recipeLink = recipe.select(cssQuery:"a").attr(attributeKey:"href"); // 레시피 상세 이미지 링크 추출

            // 추출된 데이터 출력
            System.out.println("Recipe Title: " + recipeTitle);
            System.out.println("Image URL: " + recipeImage);
            System.out.println("Link: " + recipeLink);

            try{
                Document recipePage = Jsoup.connect("https://www.10000recipe.com" + recipeLink).get(); // 상세 페이지를 가져옴
                String recipeDescription = recipePage.select(cssQuery:".view2_summary_in").text(); // 레시피 설명 추출
                Elements ingredients = recipePage.select(cssQuery:".ready_ingre3 ul li"); // 재료 목록 추출
                System.out.println("Description: " + recipeDescription);
                System.out.println(x:"Ingredients: ");
                for (var ingredient : ingredients){
                    System.out.println("- "+ ingredient.text());
                }

            Elements cookingSteps = recipePage.select(cssQuery:".view_step_cont"); // 조리 단계 추출
            System.out.println(x:"Cooking steps: ");
            for(var step : cookingSteps){
                System.out.println("- "+ step.text());
            }
            } catch (IOException e){
                System.out.println("Error fetching recipe details: "+ e.getMessage());
            }
            System.out.println(x:"----------------");
        }
    }

        @Override
        protected void process(){ // 파싱한 데이터를 추가적으로 처리하는 메서드
            System.out.println(x:"Processing parsed Recipe10000 data...");
        }



}

```

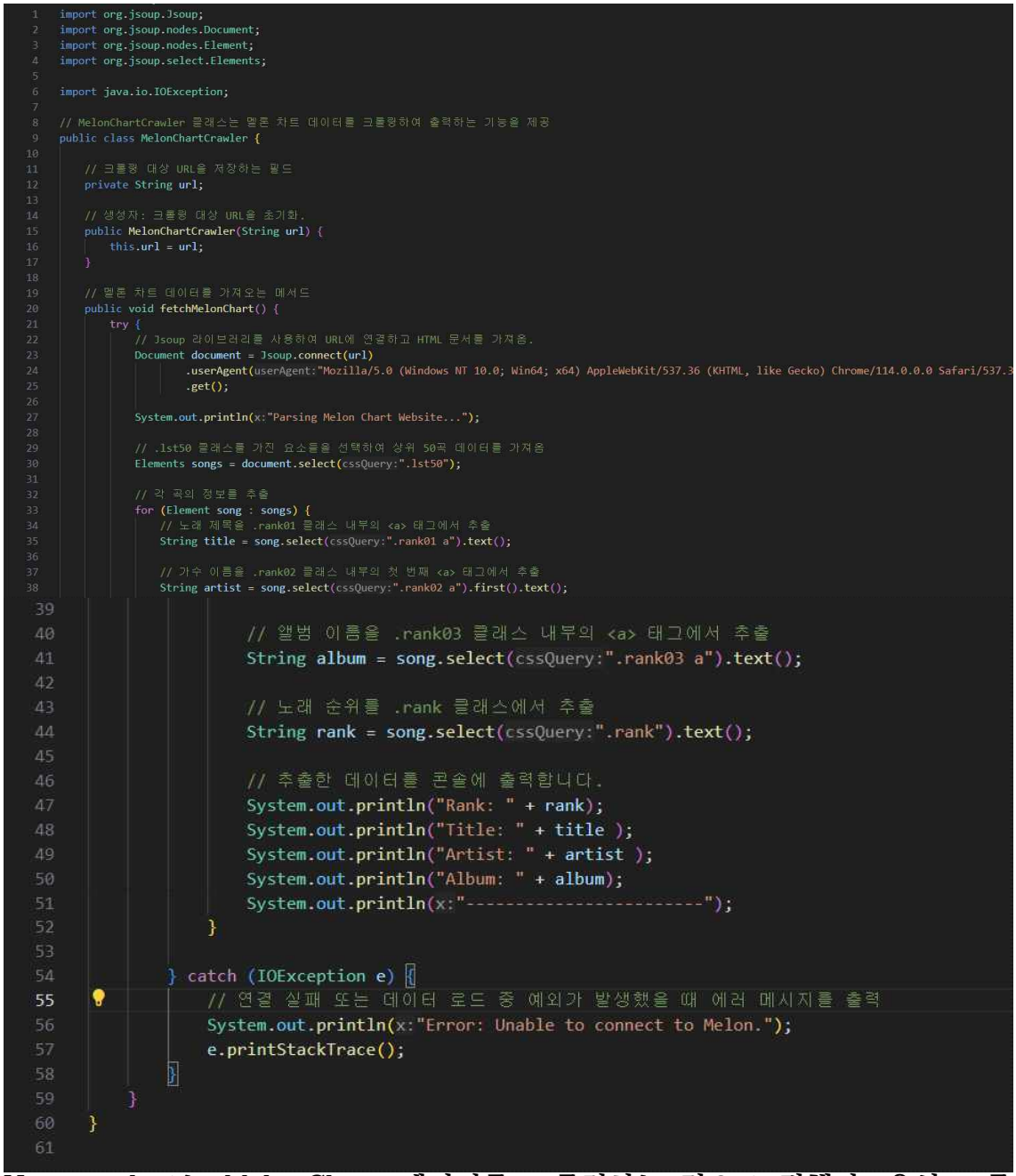
Recipe10000Crawler 클래스는 AbstractCrawler클래스를 상속받아 10000recipe 웹사이트에서 레시피 데이터를 크롤링 하는 클래스이다. 우선 생성자를 생성해 부모클래스의 생성자를 호출하여 크롤링 대상 URL을 설정해주었다. 그 다음은 connectToWebsite 메소드를 통해 웹 사이트에 연결해서 연결시 메시지를 출력할 수 있게 해주었다. 그 다음은 fetch 메소드를 통해 웹 페이지를 가져왔다. jsoup 라이브러리를 사용해서 HTML 데이터를 가져왔다. 다음 parsePage메소드를 통해 웹 페이지를 파싱했다. 레시피 목록 요소를 CSS 선택자로 선택했고 for-each문을 통해 레시피 제목 추출, 레시피 이미지 URL 추출, 레시피 상세 이미지 정보 추출을 해주었고 try-catch 문을 통해 상세 페이지를 가져오고 레시피 설명 추출, 재료 목록 추출, 조리단계 추출을 해주었다.

# MainTest 클래스)

```java
/* HW8
자바프로그래밍2_1분반
2024/11/27
32211792
박재홍 */


public class MainTest {
    Run | Debug
    public static void main(String[] args) {
        AbstractWebCrawler crawler = new Recipe10000Crawler(url:"https://www.10000recipe.com/recipe/list.html");
        crawler.crawlWebsite();

        System.out.println(x:"\nSwitching to CGV Crawler...\n");

        crawler = new CGVCrawler(url:"http://www.cgv.co.kr/movies/?lt=1&ft=0");
        crawler.crawlWebsite();

        System.out.println(x:"\n Switching to MelonChartCrawler (Your Code)");

        // (Your Code)
        String url = "https://www.melon.com/chart/index.htm"; // 멜론 차트 URL
        MelonChartCrawler crawler1 = new MelonChartCrawler(url);
        crawler1.fetchMelonChart();
    }

}
```

Maintest 클래스에서는 각 주소를 받아와 크롤링 할 수 있도록 해주었다.

# Your Code)



```java
import org.jsoup.Jsoup;
import org.jsoup.nodes.Document;
import org.jsoup.nodes.Element;
import org.jsoup.select.Elements;

import java.io.IOException;

// MelonChartCrawler 클래스는 멜론 차트 데이터를 크롤링하여 출력하는 기능을 제공
public class MelonChartCrawler {

    // 크롤링 대상 URL을 저장하는 필드
    private String url;

    // 생성자: 크롤링 대상 URL을 초기화.
    public MelonChartCrawler(String url) {
        this.url = url;
    }

    // 멜론 차트 데이터를 가져오는 메서드
    public void fetchMelonChart() {
        try {
            // Jsoup 라이브러리를 사용하여 URL에 연결하고 HTML 문서를 가져옴.
            Document document = Jsoup.connect(url)
                    .userAgent(userAgent:"Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/114.0.0.0 Safari/537.3
                    .get();

            System.out.println(x:"Parsing Melon Chart Website...");

            // .lst50 클래스를 가진 요소들을 선택하여 상위 50곡 데이터를 가져옴
            Elements songs = document.select(cssQuery:".lst50");

            // 각 곡의 정보를 추출
            for (Element song : songs) {
                // 노래 제목을 .rank01 클래스 내부의 <a> 태그에서 추출
                String title = song.select(cssQuery:".rank01 a").text();

                // 가수 이름을 .rank02 클래스 내부의 첫 번째 <a> 태그에서 추출
                String artist = song.select(cssQuery:".rank02 a").first().text();

                // 앨범 이름을 .rank03 클래스 내부의 <a> 태그에서 추출
                String album = song.select(cssQuery:".rank03 a").text();

                // 노래 순위를 .rank 클래스에서 추출
                String rank = song.select(cssQuery:".rank").text();

                // 추출한 데이터를 콘솔에 출력합니다.
                System.out.println("Rank: " + rank);
                System.out.println("Title: " + title );
                System.out.println("Artist: " + artist );
                System.out.println("Album: " + album);
                System.out.println(x:"----------------------");
            }

        } catch (IOException e) {
            // 연결 실패 또는 데이터 로드 중 예외가 발생했을 때 에러 메시지를 출력
            System.out.println(x:"Error: Unable to connect to Melon.");
            e.printStackTrace();
        }
    }
}
```

Your code로는 MelonChart 데이터를 크롤링하는 것으로 정했다. 우선 크롤링 대상 URL을 저장하는 필드를 선언해주었고 생성자를 생성해 크롤링 대상 URL을 초기화시켰고 fetchMelonchart 메소드를 통해 멜론차트 데이터를 가져왔다. jsoup 라이브러리를 사용해 url에 연결하고 HTML 문서를 가져왔다. 그 후 .lst50 클래스를 가진 요소들을 선택하여 상위 50곡 데이터를 가져왔고 for-each 문을 통해 각 곡의 정보들을 추출할 수 있도록 했다. 노래 제목, 가수 이름, 앨범 이름, 노래 순위를 추출해주었다.

# 실행결과)

## 출력문이 너무 길어 각 한두개씩만 캡처함.

```
Recipe Title: 감자 계란국 끓이는 법 계란 감자국 간단한 맑은 국 요리
Image URL: https://recipe1.ezmember.co.kr/cache/recipe/2024/06/26/a41a276d2f96b4d00e24c14b8a81d1661_m.jpg
Link: /recipe/7029194
Description: 감자와 계란 평범한 재료로 달큼하면서 고소한 맛. 간편하게 만든 든든한 맑은 감자 계란국이에요.
Ingredients:
- 감자 1개 구매
- 계란 1개 구매
- 양파 1/4개 구매
- 대파 약간 구매
- 물 700ml 구매
- 코인육수 1개 구매
- 다진마늘 0.5T 구매
- 국간장 0.5T 구매
- 소금 0.3T 구매
- 후추 약간 구매
- 감자칼 구매
- 도마 구매
- 조리용나이프 구매
- 볼 구매
- 채반 구매
- 냄비 구매
- 요리스푼 구매
- 국자 구매
- 대접 구매
Cooking steps:
- 재료를 준비해 주세요. 감자 1개, 계란 1개, 양파 1/4개, 대파 약간
- 감자 1개는 껍질을 감자 필러로 벗겨주시고 약 1cm 정도 두께로 썰어 찬물에 3번 정도 씻어 전분을 제거해 주세요. 감자칼 , 도마 , 조리용나이프 , 볼 , 채반 감자 손질법
- 양파는 감자와 비슷한 크기로 썰어주세요. 양파 손질법
- 대파는 쫑쫑쫑 썰어주세요. 대파 손질법
- 계란은 알끈을 제거하고 풀어주세요. 볼
- 물 700ml에 코인 육수 1개를 넣어 주세요. 감자를 넣고 찬물에서부터 5분 끓여주세요. 냄비
- 양파를 넣어주시고 양념을 해주세요.
- 다진 마늘 0.5, 국간장 0.5를 넣고 간을 보신 후 부족한 간은 소금으로 해주세요. 저는 소금 0.3 정도 넣어주었어요. 요리스푼
- 계란물을 냄비 가에 쪽으로 둘러서 넣어주세요. 끓어오를 때까지 잠시 두신 후 저어주세요. 계란이 얇고 모양이 예쁘게 익어요.
- 대파를 넣어주시고 후추도 약간 넣어주세요.
- 저어주고 불 끄고 국그릇에 담아 맛있게 드시면 됩니다. 국자 , 대접
Recipe Title: 우삼겹 콩나물 볶고기
Image URL: https://recipe1.ezmember.co.kr/cache/recipe/2024/10/22/f7c6ee76721f7664ac107f0a49450bc21_m.png
Link: /recipe/7036904
Description: 우삼겹은 기름이 많은 부위입니다. 처음에는 맛있지만 먹을수록 느끼합니다 우삼겹에 매콤한 양념 그리고 콩나물의 아삭함으로 요리를 한층 맛있게 하며, 마지막 볶음밥을 만들어 먹으면 행복함을 느끼실 수 있습니다
Ingredients:
- 우삼겹 300g 구매
- 콩나물 400g 구매
- 고추장 4T 구매
- 진간장 3T 구매
- 맛술 3T 구매
- 설탕 3T 구매
- 간마늘 3T 구매
- 다시다 1T 구매
- 후추 톡톡 2꼬집 구매
- 트레이 구매
- 전자저울 구매
- 프라이팬 구매
- 요리집게 구매
- 조리주걱 구매
Cooking steps:
- 콩나물 400g 세척하여 준비하고, 우삼겹 300g 준비합니다 전자저울 , 트레이
- 가열된 팬에 우삼겹 300g 넣고 구워줍니다 프라이팬 , 요리집게
- 우삼겹이 앞뒤로 익었을때, 콩나물 400g 넣어줍니다
- 양념을 콩나물 위에 올려줍니다 (양념은 미리 섞어서 준비합니다_ 고추장 4T, 진간장 3T, 맛술 3T, 설탕 3T, 간마늘 3T, 다시다 1T, 후추 2꼬집 )
- 우삼겹, 콩나물이 양념과 잘 섞이도록 뒤집어 주면서 조려 줍니다 조리주걱 대파, 양파, 청양고추 등 필요시 추가하시면 됩니다
- 우삼겹 콩나물 볶고기 완성입니다
----------
Processing parsed Recipe10000 data...
Disconnected from website.

Switching to CGV Crawler...
```

```
Switching to CGV Crawler...

Connecting to CGV Website http://www.cgv.co.kr/movies/?lt=1&ft=0
Fetching page from CGV Website...
Parsing CGV Website...
Title: 위키드
Release Date: 2024.11.20 개봉
Rating: 32.5%
Image URL: https://img.cgv.co.kr/Movie/Thumbnail/Poster/000088/88076/88076_320.jpg
------------------------
Title: 모아나 2
Release Date: 2024.11.27 개봉 D-5
Rating: 11.4%
Image URL: https://img.cgv.co.kr/Movie/Thumbnail/Poster/000088/88381/88381_320.jpg
------------------------
Title: 히든페이스
Release Date: 2024.11.20 개봉
Rating: 8.2%
Image URL: https://img.cgv.co.kr/Movie/Thumbnail/Poster/000088/88920/88920_320.jpg
------------------------
Title: 나의 히어로 아카데미아 더 무비: 유어 넥스트
Release Date: 2024.11.20 개봉
Rating: 6.8%
Image URL: https://img.cgv.co.kr/Movie/Thumbnail/Poster/000089/89066/89066_320.jpg
------------------------
Title: 글래디에이터 II
Release Date: 2024.11.13 개봉
Rating: 5.0%
Image URL: https://img.cgv.co.kr/Movie/Thumbnail/Poster/000088/88459/88459_320.jpg
------------------------
Title: 백현: 론스달라이트 닷 인 시네마
Release Date: 2024.11.27 개봉 D-5
Rating: 3.9%
Image URL: https://img.cgv.co.kr/Movie/Thumbnail/Poster/000089/89065/89065_320.jpg

Processing parsed CGV data...
Disconnected from website.

 Switching to MelonChartCrawler (Your Code)
Parsing Melon Chart Website...
Rank: 1
Title: APT.
Artist: 로제 (ROSÉ)
Album: APT.
------------------------
Rank: 2
Title: Whiplash
Artist: aespa
Album: Whiplash - The 5th Mini Album
------------------------
Rank: 3
Title: POWER
Artist: G-DRAGON
Album: POWER
------------------------
Rank: 4
Title: UP (KARINA Solo)
Artist: aespa
Album: SYNK : PARALLEL LINE - Special Digital Single
------------------------
Rank: 5
Title: HAPPY
Artist: DAY6 (데이식스)
Album: Fourever
------------------------
Rank: 6
Title: Mantra
Artist: 제니 (JENNIE)
Album: Mantra
```