**Accepts a Wikipedia link - return/throw an error if the link is not a valid wiki link**

import java.util.Scanner;

This allows us to take input from the user (e.g., type in a link or number).

public class WikipediaLinkValidator {

//Main Method to validate Wikipedia link

public static void validateWikipediaLink(String url) {

String wikipediaPattern = "^https?://([a-z]+\\.)?wikipedia\\.org/wiki/[^\\s]+$";

//I passed the string with a valid url and used the if and else statement to valid and invalid link.

if (!url.matches(wikipediaPattern)) {

throw new IllegalArgumentException("Invalid Wikipedia link.");

} else {

System.out.println("Valid Wikipedia link.");

}

}

**Accepts a valid integer between 1 to 3 - call it n**

import java.util.Scanner;

public class WikipediaInputTest {

// Method to validate integer between 1 and 3

public static int getValidInteger(Scanner scanner) {

System.out.print("Enter an integer between 1 and 3: ");

if (!scanner.hasNextInt()) {

throw new IllegalArgumentException("Input is not a valid integer.");

}

int n = scanner.nextInt();

if (n < 1 || n > 3) {

throw new IllegalArgumentException("Integer must be between 1 and 3.");

}

return n;

}

**Scrape the link provided in Step 1, for the first 10 unique (not previously added already) wiki links embedded in the page and store them in a data structure of your choice.**

Connects to the Wikipedia page using Jsoup.

Finds all links starting with wiki (these are internal Wikipedia links).

Filters out links that contain because those are usually special pages (like categories, files).

Collects the first 10 unique links into a Set (no duplicates)

import java.io.IOException;

import java.util.HashSet;

import java.util.Scanner;

import java.util.Set;

import org.jsoup.Jsoup;

import org.jsoup.nodes.Document;

import org.jsoup.nodes.Element;

import org.jsoup.select.Elements;

public class WikipediaScraper {

public static Set<String> scrapeFirst10WikiLinks(String url) throws IOException {

Set<String> uniqueLinks = new HashSet<>();

// Fetch the Wikipedia page

Document doc = Jsoup.connect(url).get();

// Select all links that start with /wiki/ (Wikipedia article links)

Elements links = doc.select("a[href^=/wiki/]");

for (Element link : links) {

String href = link.attr("href");

// Ignore links with ':' (like /wiki/File:, /wiki/Category:)

if (!href.contains(":")) {

String fullLink = "https://en.wikipedia.org" + href;

uniqueLinks.add(fullLink);

if (uniqueLinks.size() >= 10) {

break;

}

}

}

return uniqueLinks; }

**Repeat Step 3 for all newly found links and store them in the same data structure**

Set<String> allDiscoveredLinks = new HashSet<>();

allDiscoveredLinks.add(url);

System.out.println("\nScraping first 10 unique wiki links from the initial page...");

Set<String> firstLevelLinks = scrapeFirst10WikiLinks(url, allDiscoveredLinks);

int count = 1;

System.out.println("\nLevel 1 links:");

for (String link : firstLevelLinks) {

System.out.println(count + ". " + link);

count++;

}

System.out.println("\nNow scraping first 10 unique wiki links from each Level 1 link...");

for (String link : firstLevelLinks) {

System.out.println("\nScraping links from: " + link);

Set<String> secondLevelLinks = scrapeFirst10WikiLinks(link, allDiscoveredLinks);

int subCount = 1;

for (String sLink : secondLevelLinks) {

System.out.println(" " + subCount + ". " + sLink);

subCount++;

}

}

System.out.println("\nTotal unique links discovered (including initial URL): " + allDiscoveredLinks.size());

**This process should terminate after n cycle**

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

try {

System.out.print("Enter Wikipedia URL: ");

String url = scanner.nextLine();

validateWikipediaLink(url);

System.out.println("✅ Valid Wikipedia link.");

int n = getValidInteger(scanner);

System.out.println("✅ You entered a valid integer: " + n);

Set<String> allDiscoveredLinks = new HashSet<>();

allDiscoveredLinks.add(url);

Set<String> currentFrontier = new LinkedHashSet<>();

currentFrontier.add(url);

for (int cycle = 1; cycle <= n; cycle++) {

System.out.println("\n=== Cycle " + cycle + " ===");

Set<String> nextFrontier = new LinkedHashSet<>();

int count = 1;

for (String pageUrl : currentFrontier) {

System.out.println("\nScraping page: " + pageUrl);

Set<String> newLinks = scrapeFirst10WikiLinks(pageUrl, allDiscoveredLinks);

if (newLinks.isEmpty()) {

System.out.println(" No new links found.");

} else {

for (String link : newLinks) {

System.out.println(" " + count + ". " + link);

count++;

}

nextFrontier.addAll(newLinks);

}

}

if (nextFrontier.isEmpty()) {

System.out.println("\nNo more new links found; stopping early.");

break;

}

currentFrontier = nextFrontier;

}

System.out.println("\nTotal unique links discovered: " + allDiscoveredLinks.size());

## **Program 1: Validate a Wikipedia Link**

* Asks the user to type in a URL.
* Checks whether the URL is a real Wikipedia link.
* If it’s not, it shows an error.

## **Program 2: Accept a Number Between 1 and 3**

* Asks the user to enter a number.
* Checks if the number is 1, 2, or 3.
* Shows an error if the number is outside this range or not a number.

## **Program 3: Scrape First 10 Unique Wikipedia Links**

* Takes the Wikipedia link from the user.
* Opens that Wikipedia page.
* Collects the first 10 internal article links (like /wiki/Java, /wiki/Python, etc.).
* Skips things like files or categories (e.g., /wiki/File:Logo.png).

## Program 4: Repeat the Scraping for Each of Those 10 Links

### **After scraping the first 10 links from the starting page,**

* It visits each of those 10 links,
* And scrapes their own first 10 links,
* All links are saved without duplicates.

## **Program 5: Repeat Scraping for ‘n’ Cycles (Depth Levels)**

* Accepts n (between 1 to 3) from the user.
* Repeats the scraping process for n levels deep.
  + Level 1 = original page
  + Level 2 = 10 links from that page
  + Level 3 = 10 links from each of those 10 links, and so on
* Stops after n rounds or if no more new links are found.