МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
КАФЕДРА САПР

Практическая работа №1

по дисциплине

«Объектно-ориентированное программирование»

на тему: «Консольная программа для поиска в Википедии»

Студенты гр. 4351; 4353 _____ **Петухова В.С.**
Сизых П.В.

Преподаватель _____ **Кулагин М.В.**

Санкт-Петербург
2025

1. Задание

Требуется разработать программу, которая с консоли считывает поисковый запрос пользователя, и выводит результат поиска по Википедии. После выбора нужной статьи программа должна открывать ее в браузере. Программа должна реагировать корректно на любой пользовательский ввод.

2. Спецификация программы

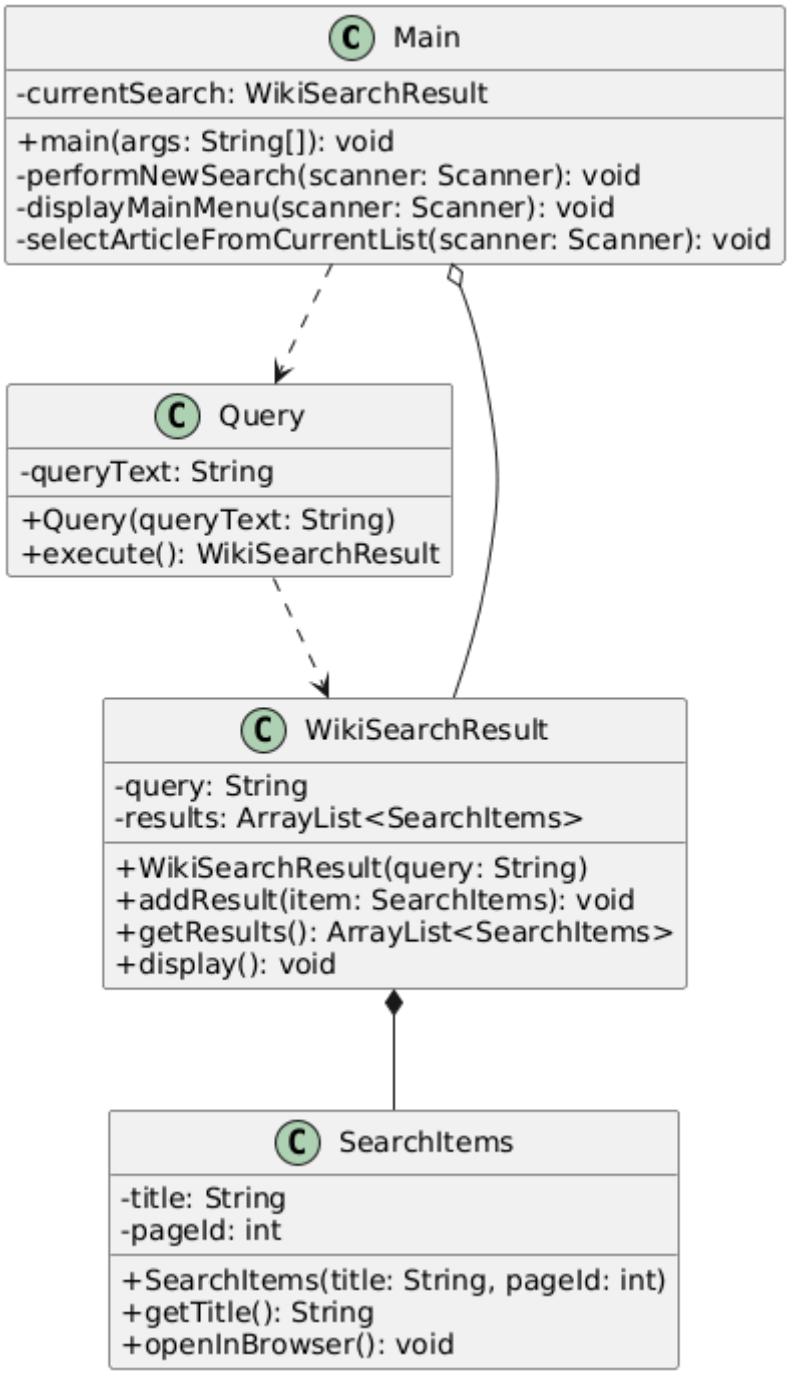

Рисунок 1. Диаграмма классов

3. Описание интерфейса пользователя программы

Ниже на рисунках 2-4 приведены примеры работы программы.

```
Введите поисковый запрос: электротехнический университет

=== Результаты поиска: 'электротехнический университет' ===
1. Санкт-Петербургский государственный электротехнический университет
2. Нагиев, Дмитрий Владимирович
3. Московский технический университет связи и информатики
4. Дьяченко, Александр Станиславович
5. Московский политехнический университет
6. Бачинский, Геннадий Николаевич
7. Мишин, Михаил Анатольевич
8. Новосибирский государственный технический университет
9. Вишневский, Борис Лазаревич
10. Колкер, Александр Наумович

=== Главное меню ===
1. Выбрать статью из текущего списка
2. Новый поиск
3. Завершить работу
Выберите действие (1-3):
```

Рисунок 2. Пример работы программы

```
=== Главное меню ===
1. Выбрать статью из текущего списка
2. Новый поиск
3. Завершить работу
Выберите действие (1-3): 2
Введите поисковый запрос: программирование

=== Результаты поиска: 'программирование' ===
1. Программирование
2. Программирование (значения)
3. Язык программирования
4. Объектно-ориентированное программирование
5. Парадигма программирования
6. Нейролингвистическое программирование
7. Структурное программирование
8. Эволюционное программирование
9. Бесточечное программирование
10. Динамическое программирование
```

Рисунок 3. Пример работы программы

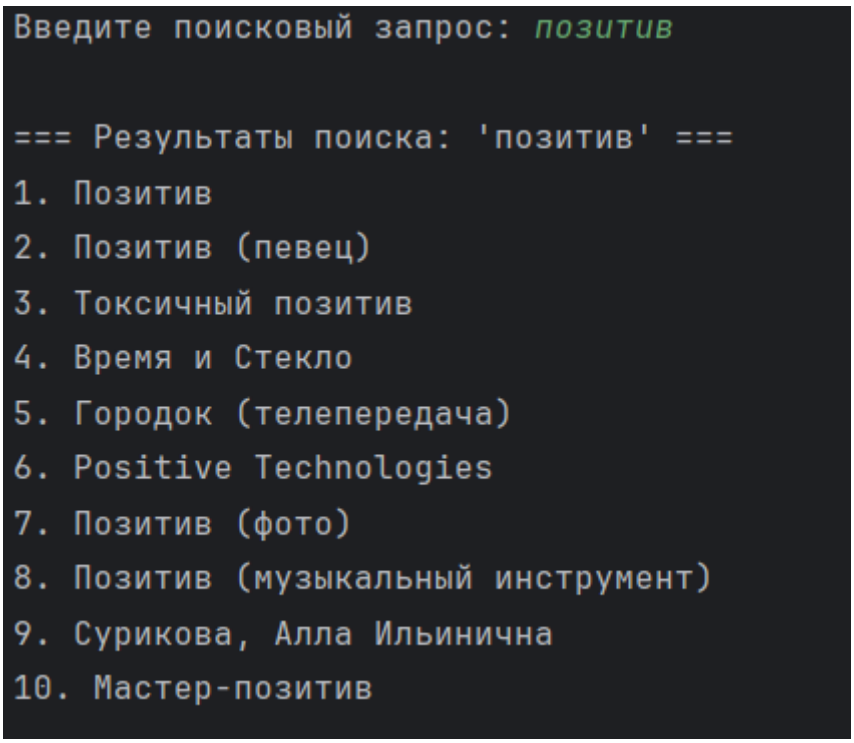

Рисунок 4. Пример работы программы

Описание интерфейса пользователя программы.

O1: "=== Поиск в Википедии ==="

O2: "Введите поисковый запрос: "

I3: Ввод запроса

O4: Вывод результатов поиска:

"=== Результаты поиска: 'запрос пользователя' ==="

"1. …"

"2. …"

"3. …"

...

O5: "=== Главное меню ==="

"1. Выбрать статью из текущего списка"

"2. Новый поиск"

"3. Завершить работу"

"Выберите действие (1-3): "

O6: Выбор действия 1-3

I7: "1"

О8: "Список статей:"

Вывод текущего списка статей (аналогично О4)

"Введите номер статьи для открытия (1-10): "

О9: Выбор номер статьи 1-10

I10: "3"

О11: "Открываю статью: 'запрос пользователя' "

"Статья открыта в браузере!"

О12: "Нажмите Enter для возврата в меню..."

О13: [Нажатие Enter]

О14: Возврат к главному меню (О5)

Альтернативные сценарии:

Сценарий А (пустой запрос):

О2: "Введите поисковый запрос: "

О3: ""

О4: "Пустой запрос. Попробуйте снова."

Сценарий Б (ничего не найдено):

О2: "Введите поисковый запрос: "

О3: "абвгдеёжзик"

О4: "Ничего не найдено. Попробуйте другой запрос."

Сценарий В (неверный выбор в меню):

О5: Главное меню

О6: "5"

О7: "Неверный выбор. Пожалуйста, введите 1, 2 или 3."

Сценарий Г (неверный номер статьи):

О7: "Введите номер статьи для открытия (1-10): "

О8: "15"

О9: "Неверный номер статьи. Пожалуйста, введите число от 1 до 10"

Сценарий Д (выход из программы):

О5: Главное меню

О6: "3"

О7: "Выход из программы. До свидания!"

4. Текст программы

```
import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.io.IOException;
import java.net.HttpURLConnection;
import java.net.URLEncoder;
import java.net.URL;
import java.net.URI;
import java.awt.Desktop;
import java.util.Scanner;
import com.google.gson.JsonArray;
import com.google.gson.JsonObject;
import com.google.gson.JsonParser;

public class Main {
    private static WikiSearchResult currentSearch = null;

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.println("=== Поиск в Википедии ===");

        while (true) {
            try {
                if (currentSearch == null) {
```

```
                performNewSearch(scanner);
            } else {
                displayMainMenu(scanner);
            }
        } catch (Exception e) {
            System.out.println("Ошибка: " + e.getMessage());
        }
    }
}

private static void performNewSearch(Scanner scanner) throws IOException {
    System.out.print("Введите поисковый запрос: ");
    String query = scanner.nextLine().trim();

    if (query.isEmpty()) {
        System.out.println("Пустой запрос. Попробуйте снова.");
        return;
    }

    Query searchQuery = new Query(query);
    currentSearch = searchQuery.execute();

    if (currentSearch.getResults().size() == 0) {
        System.out.println("Ничего не найдено. Попробуйте другой запрос.");
        currentSearch = null;
        return;
    }

    currentSearch.display();
}
```

```
    private static void displayMainMenu(Scanner scanner) throws IOException {
        System.out.println("\n=== Главное меню ===");
        System.out.println("1. Выбрать статью из текущего списка");
        System.out.println("2. Новый поиск");
        System.out.println("3. Завершить работу");
        System.out.print("Выберите действие (1-3): ");

        String choice = scanner.nextLine().trim();

        switch (choice) {
            case "1":
                selectArticleFromCurrentList(scanner);
                break;
            case "2":
                currentSearch = null;
                break;
            case "3":
                System.out.println("Выход из программы. До свидания!");
                scanner.close();
                System.exit(0);
                break;
            default:
                System.out.println("Неверный выбор. Пожалуйста, введите 1, 2 или 3.");
        }
    }

    private static void selectArticleFromCurrentList(Scanner scanner) throws IOException {
```

```
        if (currentSearch == null) return;

        System.out.println("\nСписок статей:");
        currentSearch.display();

        System.out.print("\nВведите номер статьи для открытия (1-" +
currentSearch.getResults().size() + "): ");
        String input = scanner.nextLine().trim();

        try {
            int choice = Integer.parseInt(input);
            if (choice >= 1 && choice <= currentSearch.getResults().size()) {
                SearchItems article = currentSearch.getResults().get(choice - 1);
                article.openInBrowser();

                System.out.println("\nНажмите Enter для возврата в меню...");
                scanner.nextLine();
            } else {
                System.out.println("Неверный номер статьи. Пожалуйста, введите
число от 1 до " + currentSearch.getResults().size());
            }
        } catch (NumberFormatException e) {
            System.out.println("Пожалуйста, введите корректный номер.");
        } catch (Exception e) {
            System.out.println("Ошибка при открытии статьи: " + e.getMessage());
        }
    }
}

class Query {
```

```
    private String queryText;

    public Query(String queryText) {
        this.queryText = queryText;
    }

    public WikiSearchResult execute() throws IOException {
        String encodedQuery = URLEncoder.encode(queryText, "UTF-8");
        String apiUrl = "https://ru.wikipedia.org/w/api.php?" +
                "action=query&list=search&srsearch=" + encodedQuery +
                "&srlimit=10&utf8=&format=json";

        URL url = new URL(apiUrl);
        HttpURLConnection conn = (HttpURLConnection) url.openConnection();
        conn.setRequestMethod("GET");
        conn.setRequestProperty("User-Agent", "WikiSearchApp/1.0");

        BufferedReader in = new BufferedReader(new
InputStreamReader(conn.getInputStream(), "UTF-8"));
        StringBuilder response = new StringBuilder();
        String line;
        while ((line = in.readLine()) != null) {
            response.append(line);
        }
        in.close();

        String json = response.toString();
        JsonObject jsonResponse = JsonParser.parseString(json).getAsJsonObject();
        JsonObject queryObj = jsonResponse.getAsJsonObject("query");
        JsonArray searchResults = queryObj.getAsJsonArray("search");
```

```
        WikiSearchResult result = new WikiSearchResult(queryText);
        for (int i = 0; i < searchResults.size(); i++) {
            JsonObject article = searchResults.get(i).getAsJsonObject();
            String title = article.get("title").getAsString();
            int pageId = article.get("pageid").getAsInt();
            result.addResult(new SearchItems(title, pageId));
        }

        return result;
    }
}

class WikiSearchResult {
    private String query;
    private java.util.ArrayList<SearchItems> results;

    public WikiSearchResult(String query) {
        this.query = query;
        this.results = new java.util.ArrayList<>();
    }

    public void addResult(SearchItems item) {
        results.add(item);
    }

    public java.util.ArrayList<SearchItems> getResults() {
        return results;
    }
```

```
    public void display() {
        System.out.println("\n=== Результаты поиска: '" + query + "' ===");
        for (int i = 0; i < results.size(); i++) {
            System.out.println((i + 1) + ". " + results.get(i).getTitle());
        }
    }
}

class SearchItems {
    private String title;
    private int pageId;

    public SearchItems(String title, int pageId) {
        this.title = title;
        this.pageId = pageId;
    }

    public String getTitle() {
        return title;
    }

    public void openInBrowser() throws Exception {
        String articleUrl = "https://ru.wikipedia.org/w/index.php?curid=" + pageId;
        System.out.println("Открываю статью: " + title);

        if (Desktop.isDesktopSupported()) {
            Desktop.getDesktop().browse(new URI(articleUrl));
            System.out.println("Статья открыта в браузере!");
        } else {
            System.out.println("Открытие браузера не поддерживается.");
```

```
        }
    }
}
```

5. Выводы

В результате выполнения практической работы было создано консольное приложение для поиска материалов в Wikipedia с возможностью открытия выбранной статьи в браузере. Программа принимает от пользователя поисковый запрос и выводит пронумерованный список из 10 найденных статей. Пользователь имеет возможность выбрать номер статьи для просмотра в браузере, перейти к следующим результатам текущего поиска, выполнить новый поиск или завершить работу приложения. При выборе статьи приложение отображает сообщения о запуске процесса открытия и его успешном завершении, что обеспечивает понятную обратную связь.

В ходе разработки были изучены основные принципы работы с сетевыми запросами в Java: выполнение HTTP-запросов к API Wikipedia, обработка JSON-ответов с применением библиотеки Gson, кодирование параметров URL и автоматизированное открытие веб-страниц в браузере с использованием Desktop API.

Разработанный программный код собирался с помощью системы автоматизированной сборки Maven. Результаты были выложены на Github.

https://github.com/wkinax