

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

Практическая работа №1
по дисциплине
«Объектно-ориентированное программирование»

Студент гр. 4351

Чурзин Г.В.

Преподаватель

Кулагин М.В.

Санкт-Петербург

2025

Задание

Напишите 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.

Задача разбивается на 5 этапов:

1. Считать введенные пользователем данные
2. Сделать запрос к серверу
3. Распарсить ответ
4. Вывести результат поиска
5. Открыть нужную страницу в браузере

Использовать готовые библиотеки для работы с Википедией нельзя.

Спецификация программы

Класс Main:

Методы:

1. private static String getRequest() принимает запрос от пользователя и возвращает ссылку на json с ответами
2. private static List<PageType> parseFromUrl(String request) получает json по ссылке и преобразует его в список статей для открытия.
3. private static int getChoice(List<PageType> Pages) выводит заголовки статей и предлагает пользователю выбрать следующее действие (открыть статью в браузере, повторить запрос или выйти)
4. public static void main(String[] args) обеспечивает зацикленность выполнения процессов выше.

Класс PageType:

Поля:

1. public String title — заголовок страницы Википедии
2. public int pageid — id страницы

Методы:

1. public PageType(String title, int pageid) инициализирует поля статьи.

UML-диаграмма классов представлена на рисунке 1.

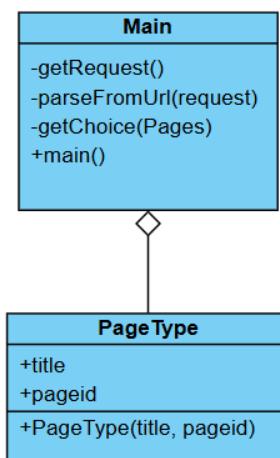


Рисунок 1

Описание интерфейса пользователя программы

Примеры работы приведены на рисунках 2 и 3. Вначале нужно ввести поисковый запрос или 0 для выхода из программы. После выведенных вариантов статей предлагается выбрать цифрами одну из них, повторить запрос вводом 0 и выйти из программы вводом -1. После открытия статьи предлагается открыть следующую. При сетевой ошибке программа завершает работу.

```
Введите запрос для Википедии или 0, чтобы выйти: солнечная система
1. Солнечная система
2. Малые тела Солнечной системы
3. Плутино
4. Поздняя тяжёлая бомбардировка
5. Спутник астероида
6. Рассеянный диск
7. Орбитальная скорость
8. Кольца Земли
9. Комета
10. Облако Оорта
Введите номер страницы для открытия в браузере
или введите 0, чтобы повторить запрос, и -1, чтобы выйти: 7
Введите запрос для Википедии или 0, чтобы выйти: 0

Process finished with exit code 0
```

Рисунок 2

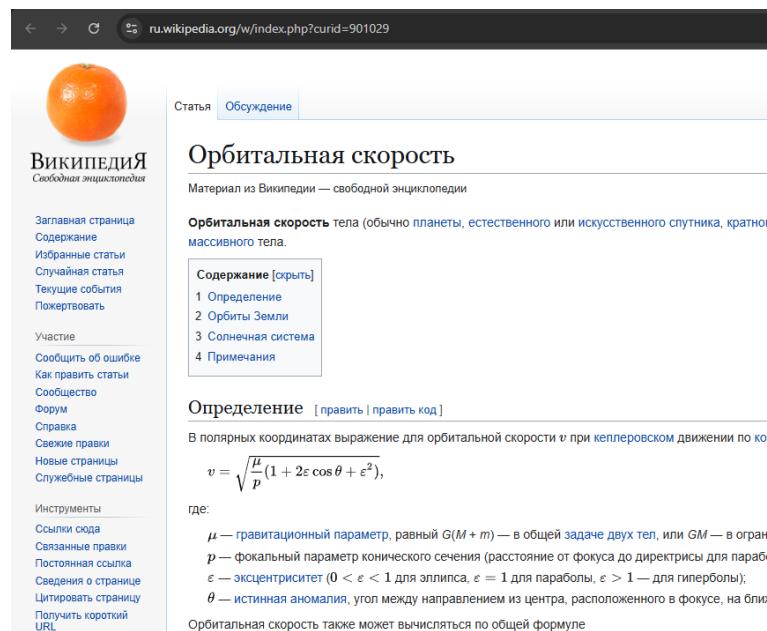


Рисунок 3

Текст программы

«Main.java»

```
import com.google.gson.Gson;
import com.google.gson.reflect.TypeToken;
import java.io.BufferedReader;
import java.lang.reflect.Type;
import java.net.*;
import java.nio.charset.StandardCharsets;
import java.util.List;
import java.util.Scanner;
import java.io.IOException;
import java.io.InputStreamReader;
import java.awt.Desktop;
import java.net.HttpURLConnection;

public class Main {

    private static String getRequest() {
        Scanner in = new Scanner(System.in);
        System.out.print("Введите запрос для Википедии или 0, чтобы выйти: ");
        String question = in.nextLine();
        if (question.equals("0")) {
            return null;
        }
        question = "\"" + question + "\"";
        question = URLEncoder.encode(question, StandardCharsets.UTF_8);
        return
"https://ru.wikipedia.org/w/api.php?action=query&list=search&utf8=&format=json&srsearch=" + question;
    }

    private static List<PageType> parseFromUrl(String request) {
        try {
            if (request == null) {
                return null;
            }
            URL url = new URL(request);
            HttpURLConnection connect = (HttpURLConnection)
url.openConnection();
            connect.setRequestMethod("GET");
            connect.setRequestProperty("User-Agent", "WikiSearchApp/1.0");
            BufferedReader readJson = new BufferedReader(new
InputStreamReader(connect.getInputStream(), "UTF-8"));

```

```

StringBuilder jsonStr = new StringBuilder();
String line;
while ((line = readJson.readLine()) != null) {
    jsonStr.append(line);
}
readJson.close();
connect.disconnect();
Type listOfPages = new TypeToken<List<PageType>>() {
}.getType();
Gson gson = new Gson();
StringBuilder jsonPages = new StringBuilder();
for (int i = 0; i < jsonStr.length(); i++) {
    if (jsonStr.charAt(i) == '[') {
        while (jsonStr.charAt(i) != ']') {
            jsonPages.append(jsonStr.charAt(i));
            i++;
        }
        jsonPages.append(jsonStr.charAt(i));
    }
}
return gson.fromJson(jsonPages.toString(), listOfPages);
} catch (java.net.MalformedURLException e) {
    System.out.println("Ошибка в ссылке");
    return null;
} catch (IOException e) {
    System.out.println("Ошибка открытия json по ссылке");
    return null;
}
}

private static int getChoice(List<PageType> Pages) {
try {
    if (Pages == null) {
        return -1;
    }
    Scanner in = new Scanner(System.in);
    int i = 1;
    for (PageType page : Pages) {
        System.out.println(i + ". " + page.title);
        i++;
    }
    System.out.print("Введите номер страницы для открытия в браузере\n"
+
"или введите 0, чтобы повторить запрос, и -1, чтобы выйти: ");
    int j = in.nextInt();
}

```

```

        while (j < -1 || j > 10) {
            j = in.nextInt();
        }
        if (j > 0) {
            String UrlToOpen = "https://ru.wikipedia.org/w/index.php?curid=" +
Pages.get(j - 1).pageid;
            URI uri = new URI(UrlToOpen);
            Desktop.getDesktop().browse(uri);
        }
        return j;
    } catch (IOException e) {
        System.out.println("Ошибка открытия браузера");
        return -1;
    } catch (URISyntaxException e) {
        System.out.println("Ошибка в ссылке Википедии");
        return -1;
    }
}

public static void main(String[] args) {
    int choice = 0;
    while(choice != -1) {
        choice = getChoice(parseFromUrl(getRequest()));
    }
}
}

```

«PageType.java»

```

public class PageType {

    public String title;
    public int pageid;

    public PageType(String title, int pageid) {

        this.title = title;
        this.pageid = pageid;
    }
}

```

Выводы

В результате выполнения работы была разработана без использования специальных библиотек программа на Java, открывающая статьи Википедии в браузере по запросу пользователя. Программа учитывает возможные ошибки соединения. Разработанный программный код собирался с помощью системы автоматизированной сборки Maven. Результаты были выложены на Github:

<https://github.com/ChursStudent/OOPprograms>