МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра систем автоматизированного проектирования

ОТЧЕТ

по практической работе №1

по дисциплине «ООП»

ТЕМА: Консольная программа для поиска в Википедии.

Студент гр. 4354 _____ Шупряков В.В.

Преподаватель _____ Кулагин М.В.

Санкт-Петербург

2025

Цель работы.

Разработка консольного приложения для интерактивного поиска и просмотра статей в Википедии, обеспечивающего удобное взаимодействие с пользователем и устойчивость к некорректному вводу.

Задача.

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.

Спецификация программы

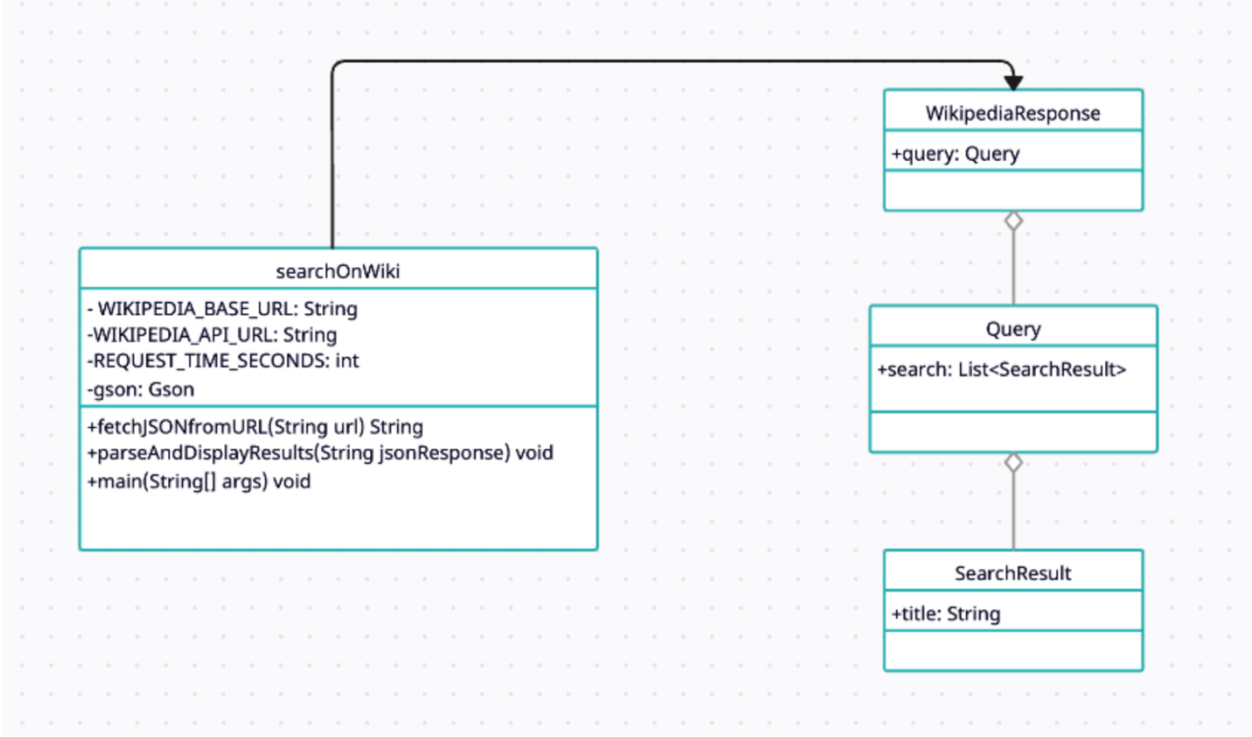
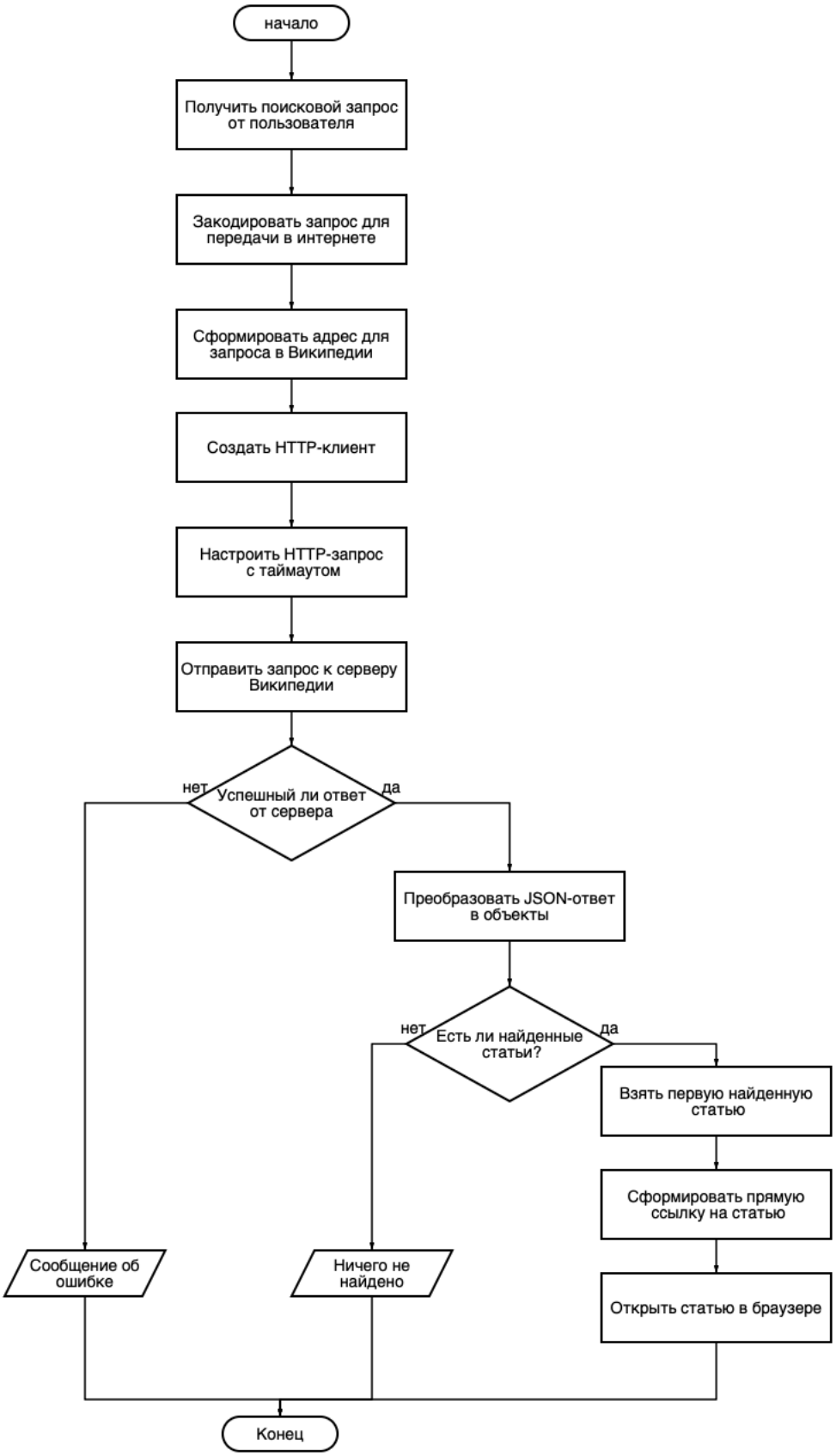


Рисунок 1 – Диаграмма классов

Блок-схема


начало
Получить поисковой запрос от пользователя
Закодировать запрос для передачи в интернете
Сформировать адрес для запроса в Википедии
Создать HTTP-клиент
Настроить HTTP-запрос с таймаутом
Отправить запрос к серверу Википедии
Успешный ли ответ от сервера
нет
да
Преобразовать JSON-ответ в объекты
Есть ли найденные статьи?
нет
да
Взять первую найденную статью
Сформировать прямую ссылку на статью
Открыть статью в браузере
Сообщение об ошибке
Ничего не найдено
Конец

Код программы

searchOnWiki

```
import java.io.IOException;
import java.net.URI;//класс для отправки url
import java.util.Scanner;//класс для считывания данных пользователя
import java.net.http.HttpClient;//класс для создания HTTP клиента и отправки запросов
import java.net.http.HttpRequest;//класс для построения HTTP запроса
import java.net.http.HttpResponse;//класс для представления HTTP ответа от сервера
import java.time.Duration;//класс для работы с временными промежутками
import java.net.URLEncoder;//класс для кодирования строк в URL-формат
import java.nio.charset.StandardCharsets;//Класс со стандартными кодировками
import java.util.List;//интерфейс для работы со списками (упорядоченными коллекциями)
import java.awt.Desktop;//класс для просмотра в браузере в данном случае
import com.google.gson.Gson;//библиотека для преобразования JSON в Java-объекты и обратно
public class searchOnWiki {

    private static final String WIKIPEDIA_BASE_URL = "https://ru.wikipedia.org/wiki/";
    private static final String WIKIPEDIA_API_URL = "https://ru.wikipedia.org/w/api.php?action=query&list=search&utf8=&format=json&srsearch=";
    private static final int REQUEST_TIMEOUT_SECONDS = 15;//время таймаута

    private static final Gson gson = new Gson();
```

```
    public static String fetchJSONfromURL(String url){

        // 1. Создаем HTTP клиент
        HttpClient client = HttpClient.newHttpClient();

        // 2. Создаем запрос
        HttpRequest request = HttpRequest.newBuilder()//создаем "строителя для удобства/читаемости"                .uri(URI.create(url))
                .uri(URI.create(url))

.timeout(Duration.ofSeconds(REQUEST_TIMEOUT_SECONDS))//таймаут - "обход" вечного ожидания
                .header("Accept", "application/json")//"просим" сервер ответить файлом json
                .header("User-Agent", "WikiSearchBot/1.0")
                .GET() //указываем метод GET
                .build();

        // 3. Отправляем запрос и получаем ответ
       HttpResponse<String>/*ответ строка*/ response = null;
       try {
          response = client.send(
                request,//запрос
                HttpResponse.BodyHandlers.ofString()//преобразовываем тело ответа в строку
          );
       } catch (IOException e) {
          System.out.println(e.getMessage());
       } catch (InterruptedException e) {
          System.out.println(e.getMessage());
```

```
        }

        // 4. Проверяем статус и обрабатываем ответ
            if (response.statusCode() >= 200 && response.statusCode()<300) {//если успешный код (200-299), то...
                return response.body();//возвращаем тело ответа
            }
            else {
                throw new RuntimeException("HTTP ошибка: " + response.statusCode() + " - " + response.body());//обработка исключений
            }

    }

    public static void parseAndDisplayResults(String jsonResponse){
        try {
            // Превращаем JSON в Java объект
            WikipediaResponse response = gson.fromJson(jsonResponse, WikipediaResponse.class);
            // Проверяем есть ли результат
            if (response.query.search.isEmpty()) {
                System.out.println("Ничего не найдено");
                return;//выходим, если нет
            }
            // Берем первую найденную статью
            SearchResult firstResult = response.query.search.get(0);
            // Создаем ссылку на статью Wikipedia
            String articleUrl = WIKIPEDIA_BASE_URL +
                    URLEncoder.encode(firstResult.title, StandardCharsets.UTF_8);
```

```
            // Открываем в браузере
            Desktop.getDesktop().browse(new URI(articleUrl));

        }catch(Exception e) {
            // Выводим ошибку, если что-то пошло не так
            System.out.println("Ошибка: " + e.getMessage());
        }
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        String Text = scanner.nextLine();
        try {
            String new_str = URLEncoder.encode(Text, StandardCharsets.UTF_8);
            String apiurl = WIKIPEDIA_API_URL+new_str;

            String jsonResponse = fetchJSONfromURL(apiurl);
            // Просто передаем JSON в Gson - он всё сделает сам
            parseAndDisplayResults(jsonResponse);

        }catch(Exception e){
            System.out.println("Ошибка: " + e.getMessage());
        }finally{
            scanner.close(); // Закрываем scanner
        }
    }
    //-----------------------------------
    // Классы для автоматического парсинга

    // Класс для хранения всего ответа от Wikipedia
```

```
    static class WikipediaResponse {
        public Query query; // Объект, содержащий результаты запроса
    }
    // Класс для хранения раздела с поиском
    static class Query {
        public List<SearchResult> search; // Список найденных статей в
результате поиска
    }
    // Класс для хранения информации об одной статье
    static class SearchResult {
        public String title; // Заголовок найденной статьи Wikipedia
    }
}
```

Код программы

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://maven.apache.org/POM/4.0.0
http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <groupId>org.example</groupId>
    <artifactId>untitled</artifactId>
    <version>1.0-SNAPSHOT</version>
    <properties>
        <maven.compiler.source>24</maven.compiler.source>
        <maven.compiler.target>24</maven.compiler.target>
        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    </properties>
    <repositories>
        <repository>
            <id>central</id>
            <url>https://repo.maven.apache.org/maven2</url>
        </repository>
    </repositories>
    <dependencies>
        <dependency>
            <groupId>com.google.code.gson</groupId>
            <artifactId>gson</artifactId>
            <version>2.13.2</version>
        </dependency>
    </dependencies>
</project>
```

Пример работы программы

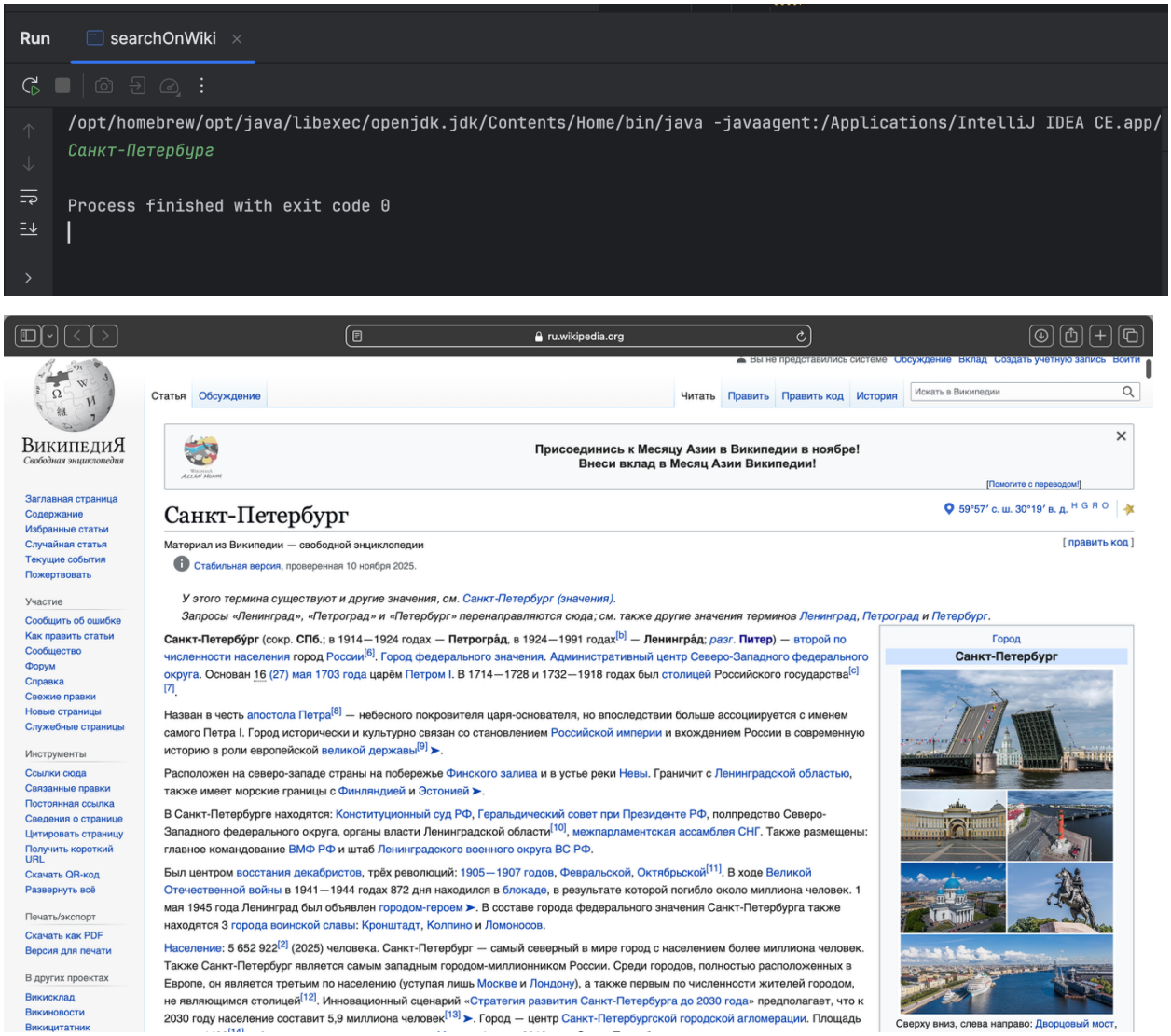


Рисунок 1-2 – Пример работы программы

Вывод:

В результате выполнения практической работы №1 было разработано консольное приложения для поиска в Википедии с последующим открытием выбранной статьи в браузере. Программа корректно реагирует на любой пользовательский ввод

В процессе выполнения практической работы были получены навыки по работе с запросом к серверу, парсингу ответа, вывода результата и открытию нужной страницы в браузере на языке Java. Разработанный программный код собирался с помощью системы автоматизированной сборки Maven.

Результаты были выложены на Github:

https://github.com/diwkos/OOP_discipline