

Московский Государственный Технический Университет
имени Н.Э. Баумана

Факультет ИУ «Информатика и системы управления»

Кафедра ИУ-3 «Информационные системы и телекоммуникации»

Отчет к лабораторной работе №8
по курсу «Технологии программирования»
“ Улучшенный поисковый робот”

Продолжительность работы: 4 ак. часа

Сдал

Ситниченко С.А.

Принял

Руденкова Ю.С.

Москва, 2022

Лабораторная работа №8

Тема: Улучшенный поисковый робот

Цель работы: В работе требуется написать робота, работающего в многопоточном режиме, чтобы робот смог обрабатывать несколько страниц одновременно. Все это сделано для повышения производительности.

Сделайте класс с именем `URLPool`, который сохраняет список всех найденных URLs, вместе с "уровнем" каждого из этих URL (мы называем это также "глубина поиска").

Для того чтобы поиск можно было выполнять в нескольких потоках, создайте класс `CrawlerTask`, который реализует интерфейс `Runnable`. Каждый экземпляр `CrawlerTask` должен иметь ссылку на единственный экземпляр класса `URLPool` который описан выше.

Так как ваш робот запускает некоторое количество потоков, добавьте в программу третий параметр командной строки, задающий число потоков робота.

`URLPool` – функциональный класс для создания объекта, с которым будут работать все вспомогательные потоки и который будет хранить списки с ссылками, а также иметь методы получения и добавления ссылок.

```
public URLDepthPair getURL() throws InterruptedException {
    synchronized (poolOfURL) {
        if (poolOfURL.isEmpty()) {
            processesCounter++;
            poolOfURL.wait();
            processesCounter--;
        }

        URLDepthPair firstUrlBufferPair = poolOfURL.getFirst();
        synchronized (processedURL) {
            processedURL.add(firstUrlBufferPair);
        }

        poolOfURL.removeFirst();
        return firstUrlBufferPair;
    }
}
```

```

public void putURL(URLDepthPair pair) {
    synchronized (poolOfURL) {
        if (pair.getSearchDepth() < MAX_DEPTH && !processedURL.contains(pair)) {
            poolOfURL.add(pair);
            if (processesCounter > 0)
                poolOfURL.notify();
        } else {
            synchronized (processedURL) {
                processedURL.add(pair);
            }
        }
    }
}
}

```

Для того чтобы поиск можно было выполнять в нескольких потоках, создаем класс CrawlerTask, который реализует интерфейс Runnable.

```

public class CrawlerTask implements Runnable {
    private static URLPool urlPool = null;
    private URLDepthPair currentPair;

    private static final String URL_PREFIX = "https://";
    private static final Pattern prefixPattern = Pattern.compile("^https://");

    private static final Pattern hrefPattern = Pattern.compile("<a href=\"([^\"]+)\"");

    private static final Pattern urlPattern = Pattern.compile("(?<=\") ([^\"]+) (?=\".*)");

    private static final Pattern relPattern = Pattern.compile("/.*");

    CrawlerTask(URLPool gotPool) {
        urlPool = gotPool;
    }
}

```

Output

```
C:\Users\днс\OneDrive\Документы\NetBeansProjects\Main\src>java main.Main https://mos.ru 2 10
java.net.ProtocolException: Server redirected too many times (20)
    at java.base/sun.net.www.protocol.http.HttpURLConnection.getInputStream0(HttpURLConnection.java:2005)
    at java.base/sun.net.www.protocol.http.HttpURLConnection.getInputStream(HttpURLConnection.java:1589)
    at java.base/java.net.HttpURLConnection.getResponseCode(HttpURLConnection.java:529)
    at java.base/sun.net.www.protocol.https.HttpsURLConnectionImpl.getResponseCode(HttpsURLConnectionImpl.java:308)
    at main.CrawlerTask.socketProcess(CrawlerTask.java:35)
    at main.CrawlerTask.run(CrawlerTask.java:116)
    at java.base/java.lang.Thread.run(Thread.java:833)
java.net.SocketTimeoutException: Connect timed out
```

```
https://www.mos.ru/otvet-socialnaya-podderjka/kak-polzovatsya-socialnym-sertifikatom/ 2
https://www.mos.ru/otvet-socialnaya-podderjka/kak-oplachivat-pokupki-i-poluchat-skidki-po-socialnoj-karte/ 2
https://www.mos.ru/pgu/ru/federal/get_dept/ 2
https://service.nalog.ru/inn.html 2
https://service.nalog.ru/addrno.do 2
https://service.nalog.ru/addrno.do 2
https://www.mos.ru/services/centry-gosudarstvennyh-uslug/ 2
https://service.nalog.ru/addrno.do 2
https://service.nalog.ru/inn.html 2
https://service.nalog.ru/addrno.do 2
https://service.nalog.ru/zpufl/ 2
https://www.mos.ru/pgu/ru/md/ 2
https://service.nalog.ru/addrno.do 2
https://service.nalog.ru/gp2.do 2
https://service.nalog.ru/addrno.do 2
https://www.mos.ru/otvet-dokumenty/kak-proverit-i-oplatit-sudebnuyu-zadolzhennost/ 2
https://www.mos.ru/otvet-dokumenty/kak-poluchit-snils/ 2
https://www.mos.ru/otvet-dokumenty/kak-proverit-i-oplatit-nalogovuyu-zadolzhennost/ 2
https://www.mos.ru/otvet-dokumenty/kak-oformit-zagranpasport-novogo-obrazca-dlya-vzroslogo/ 2
https://www.mos.ru/otvet-dokumenty/kak-oformit-zagranpasport-novogo-obrazca-rebenku/ 2
https://www.mos.ru/otvet-dokumenty/kak-oformit-zagranpasport-starogo-obrazca-dlya-vzroslogo/ 2
https://www.mos.ru/otvet-dokumenty/kak-oformit-zagranpasport-starogo-obrazca-rebenku/ 2
https://www.mos.ru/otvet-dokumenty/kak-vpisat-rebenka-v-svoy-zagranpasport/ 2
https://77.xn--b1aew.xn--p1ai/Kontakty 2
https://www.mos.ru/otvet-dokumenty/kak-zaverit-dokumenty-apostilem/ 2
```

Вывод:

В ходе лабораторной работы мы улучшили поискового робота при помощи работы в режиме многопоточности. Также изучили классы Runnable и Thread.