

**Федеральное агентство связи**  
**Ордена Трудового Красного Знамени**  
**Федеральное государственное бюджетное образовательное учреждение**  
**высшего образования**  
**«Московский технический университет связи и информатики»**

Кафедра математической кибернетики и информационных технологий

**Отчет по лабораторной работе №8**  
по дисциплине «Технологии разработки программного обеспечения»

Выполнил: студент группы  
БВТ1801

Протасова Елена Сергеевна

Руководитель:

Мосева Марина Сергеевна

Москва 2020

**Цель работы:** расширить сканер для использования поточной обработки Java так, чтобы несколько веб-страниц можно было сканировать параллельно.

**Выполнение:**

**URLPool:**

```
import java.util.LinkedList;

public class URLPool {
    private LinkedList<URLDepthPair> listNotRun = new LinkedList<>();
    private int busyThreads=0;
    private int waitingThreads=0;
    private int count;
    private LinkedList<URLDepthPair> resultlist = new LinkedList<>();
    public URLPool(int count){
        this.count = count;
    }
    public synchronized URLDepthPair get(){
        if (listNotRun.size() == 0) {
            try {
                waitingThreads++;
                this.wait();
            }
            catch (InterruptedException e) {
                System.err.println("MalformedURLException: " + e.getMessage());
                return null;
            }
        }
        resultlist.add(listNotRun.getFirst());
        return listNotRun.removeFirst();
    }
    public synchronized int getSizeList(){
        return listNotRun.size();
    }
    public synchronized int getWaitingThreads(){
        return waitingThreads;
    }
    public synchronized LinkedList<URLDepthPair> getResultlist(){
        return resultlist;
    }
    public synchronized void set(LinkedList<URLDepthPair> newLinks){
        if (newLinks.size()>0) {
            if (newLinks.getFirst().depth()>=count){
                resultlist.addAll(newLinks);
            }
            else{
                listNotRun.addAll(newLinks);
                for (int countSite=listNotRun.size(); countSite!=0 &&
waitingThreads!=0;countSite-- , waitingThreads--){
                    this.notify();
                }
            }
        }
    }
}
```

## CrawlerTask:

```
import java.io.IOException;
import java.util.LinkedList;

public class CrawlerTask implements Runnable {
    private URLDepthPair urlDepthPair;
    private URLPool urlPool;
    private LinkedList<URLDepthPair> newLinks;
    private int count;
    public CrawlerTask(URLPool urlPool, int count){
        this.urlPool = urlPool;
        this.count = count;
    }
    @Override
    public void run() {

        while (true) {
            urlDepthPair = urlPool.get();
            try {
                CrawlerTask crawlers = new CrawlerTask(urlDepthPair.getURL(), 80,
urlDepthPair.depth());
                newLinks = crawlers.StartScanner(urlDepthPair.depth() + 1);
                urlPool.set(newLinks);
            }
            catch (IOException e) {
                System.out.println(e);
            }
        }
    }
}
```

## Результаты работы:

```
C:\Users\Elena\IdeaProjects\KTPlab8\src>java CrawlerTask http://www.google.com 2 4
URL = http://www.google.com Depth = 0
URL = http://www.google.ru/imghp?hl=ru&tab=wi Depth = 1
URL = http://maps.google.ru/maps?hl=ru&tab=w1 Depth = 1
URL = http://www.youtube.com/?gl=RU&tab=w1 Depth = 1
URL = http://news.google.ru/nwshp?hl=ru&tab=wn Depth = 1
URL = http://translate.google.ru/?hl=ru&tab=wT Depth = 1
URL = http://www.blogger.com/?tab=wj Depth = 1
URL = http://video.google.ru/?hl=ru&tab=wv Depth = 1
URL = http://www.google.com/ Depth = 1
URL = http://www.google.ru/imghp?hl=ru&tab=wi Depth = 2
URL = http://maps.google.ru/maps?hl=ru&tab=w1 Depth = 2
```

**Выводы:** в ходе работы была изучена работы с потоками.