|  |
| --- |
|  |
| МИНОБРНАУКИ РОССИИ |
| *Федеральное государственное бюджетное образовательное учреждение высшего образования* ***«МИРЭА – Российский технологический университет»***  **РТУ МИРЭА** |

**Институт информационных технологий (ИТ)**

**Кафедра инструментального и прикладного программного обеспечения (ИиППО)**

**Дисциплина «Программирование на языке Джава»**

**ОТЧЕТ**

**ПО ПРАКТИЧЕСКОМУ ЗАНЯТИЮ №27-28**

Выполнил студент группы ИНБО-02-20 Маркарянц Р.Д.

Принял Степанов П.В.

Практические работы выполнены «\_\_\_»\_\_\_\_\_\_\_2021г.

«\_\_\_\_\_\_\_\_\_\_\_\_\_\_» «\_\_\_»\_\_\_\_\_\_\_2021г.

Отметка о выполнении

**Москва – 2021 г.**

## **Задание**

Напишите программу, которая уменьшает изображения на основе кода который приведен в данном пакете задания. Она должна запускать число потоков, равное количеству ядер процессора вашего компьютера.

## **Ход Работы**

В ходе выполнения работы были получены следующие исходные коды:

package ru.mirea.inbo0220.markaryants.pr27;  
  
import javax.imageio.ImageIO;  
import java.awt.image.BufferedImage;  
import java.io.File;  
import java.nio.file.Files;  
import java.nio.file.Paths;  
import java.util.ArrayList;  
  
public class Test implements Runnable {  
 public static void main(String[] args) {  
 Test image = new Test();  
 int cores = Runtime.*getRuntime*().availableProcessors();  
 ArrayList<Thread> threads = new ArrayList<>();  
 for(int i = 0; i < cores; i++) {  
 Thread thread = new Thread(image);  
 thread.setName(String.*valueOf*(i));  
 threads.add(thread);  
 }  
 for (Thread thread : threads){  
 thread.start();  
 }  
 }  
  
 @Override  
 public void run() {  
 resize(Integer.*parseInt*(Thread.*currentThread*().getName()));  
 }  
  
 private void resize(int i) {  
 int cores = Runtime.*getRuntime*().availableProcessors();  
 String srcFolder = "src/ru/mirea/inbo0220/markaryants/pr27/images";  
 String dstFolder = "src/ru/mirea/inbo0220/markaryants/pr27/new\_images";  
  
 File srcDir = new File(srcFolder);  
  
 long start = System.*currentTimeMillis*();  
  
 File[] files = srcDir.listFiles();  
 assert files != null;  
 cores=files.length/cores;  
 try {  
 if (!Files.*exists*(Paths.*get*(dstFolder))) {  
 Files.*createDirectories*(Paths.*get*(dstFolder));  
 }  
 for (int l=i;l<cores\*(i+1);l++) {  
 File file = files[l];  
 BufferedImage image = ImageIO.*read*(file);  
 if (image == null) {  
 continue;  
 }  
  
 int newWidth = image.getWidth() / 4;  
 int newHeight = (int) Math.*round*(  
 image.getHeight() / (image.getWidth() / (double) newWidth)  
 );  
 BufferedImage newImage = new BufferedImage(  
 newWidth, newHeight, BufferedImage.*TYPE\_INT\_RGB* );  
  
 int widthStep = image.getWidth() / newWidth;  
 int heightStep = image.getHeight() / newHeight;  
  
 for (int x = 0; x < newWidth; x++) {  
 for (int y = 0; y < newHeight; y++) {  
 int rgb = image.getRGB(x \* widthStep, y \* heightStep);  
 newImage.setRGB(x, y, rgb);  
 }  
 }  
  
 File newFile = new File(dstFolder + "/" + file.getName());  
 ImageIO.*write*(newImage, "jpg", newFile);  
 }  
 } catch (Exception ex) {  
 ex.printStackTrace();  
 }  
 System.*out*.println("Время работы: " + (System.*currentTimeMillis*() - start));  
 }  
}

## **Вывод**

Выполнив практическую работу, я получил навыки работы со сжатием изображений на языке Java.