

Evgen959 /
Advanced_Backend[Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Wiki](#) [Security](#) [Insights](#) [Settings](#)[Advanced_Backend](#) / [Lesen 050](#) / [code](#) / [hw_na_les049_1](#) / [src](#) / [Main2.java](#)

Evgen959 newLes50

21943f6 · 10 minutes ago



72 lines (61 loc) · 3.43 KB

[Code](#) [Blame](#)

Raw



```
1  /*По ссылке https://drive.google.com/file/d/1XZrLLuJlHK3n35NwQAnY9t1nSwZ3-piq/view?usp=sharing
2  находится файл file.dat . Сохраните этот файл на диск (просто "ручками", не из программы).
3  Ваша программа должна записать в отдельный файл первые 601 байт,
4  затем в отдельный файл записать следующие 57053 байта
5  и оставшиеся 22494 байта записать в следующий файл. Если все сделано правильно,
6  у вас должно получиться 3 файла.
7
8  В одном из этих файлов лежит gif картина, в другом jpg картинка,
9  еще в одном скомпилированный .class файл java программы. Вам нужно написать программу,
10 которая определит, в каком файле что лежит. Сделать это можно используя так называемые сигнатуры файлов:
11 gif должен начинаться с шестнадцатеричной последовательности байт 47 49 46 38 39 61
12 jpeg с последовательности ff d8 java .class файл с шестнадцатеричной последовательности ca fe ba be
13 Сохраните файл, опознанный как .class файл под именем Main.class и запустите его из командной строки.
14 Прочитайте кодовое слово.*/
15
16 import java.io.*;
17
18 public class Main2 {
19     public static void main(String[] args) {
20         String inputFile = "../file.dat";
21
22         File inputData = new File(inputFile);
23         if (!inputData.exists()){
24             System.out.println("Отсутствует файл данных");
25             System.exit(-1);
26         }
27
28         String[] outputFile = {
29             "../test_temp/file1",
30             "../test_temp/file2",
31             "../test_temp/file3"
32         };
33
34         int[] fileSize = {601,57053,22494};
35
36         try (FileInputStream inputStream = new FileInputStream(inputFile);
37             FileOutputStream outputStream1 = new FileOutputStream(outputFile[0]);
38             FileOutputStream outputStream2 = new FileOutputStream(outputFile[1]);
39             FileOutputStream outputStream3 = new FileOutputStream(outputFile[2]);
40         ) {
41             OutputStream[] os = {outputStream1,outputStream2,outputStream3};
42
43             for (int i = 0; i < 3; i++) {
44                 byte[] buffer = new byte[fileSize[i]];
45                 int bytes = inputStream.read(buffer);
46                 if (bytes!=-1){
47                     os[i].write(buffer, 0, bytes);
48                 }
49             }
50         }
```

```
50      /*byte[] buffer = new byte[filesSize[0]];
51      int bytes = inputStream.read(buffer);
52      if (bytes!=-1){
53          os[0].write(buffer, 0, bytes);
54      }
55      buffer = new byte[filesSize[1]];
56      bytes = inputStream.read(buffer);
57      if (bytes!=-1){
58          os[1].write(buffer, 0, bytes);
59      }
60      buffer = new byte[filesSize[2]];
61      bytes = inputStream.read(buffer);
62      if (bytes!=-1){
63          os[2].write(buffer, 0, bytes);
64      }*/
65
66      } catch (IOException e) {
67          e.printStackTrace();
68      }
69  }
70
71
72  }
```