ФЕДЕРАЛЬНОЕ Государственное АВТОНОМНОЕ образовательное УЧРЕЖДЕНИЕ ВЫСШЕГО образования

**БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ НАЦИОНАЛЬНЫЙ**

**ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ»**

**(НИУ «БелГУ»)**

ИНСТИТУТ ИНЖЕНЕРНЫХ И ЦИФРОВЫХ ТЕХНОЛОГИЙ

КАФЕДРАИНФОРМАЦИОННЫХ СИСТЕМ

**Отчет**

по дисциплине «Разработка программных приложений на языках высокого уровня»

специальности 09.03.03 «Прикладная информатика»

**Лабораторная работа №4**

студентки очного отделения

3 курса 12002005 группы

Хариной Е. Д.

Проверил:

Лифиренко М.В.

БЕЛГОРОД 2023

Цель работы

Получить практические навыки работы с Collections в Java, изучить работу с регулярными выражениями.

## Результаты выполнения работы

1. Изучить пакет java.util.regex (Классы java.util.regex.Pattern, java.util.regex.Matcher, java.util.regex.PatternSyntaxException)

2. Разработать программу и регулярные выражения для реализации указанного

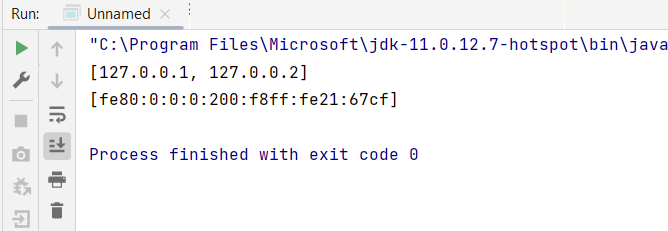
варианта задания.

На вход программы подается произвольный файл, необходимо извлечь из него все ip адреса. В файле могут встречаться ка IPv4 так и Ipv6 адреса например, 127.0.0.1 или fe80:0:0:0:200:f8ff:fe21:67cf. Отсортировать IP-адреса по возрастанию (сначала IPv4, затем IPv6). Сохранить результат в файл.

## Листинг программы:

**import** java.io.\*;  
**import** java.util.ArrayList;  
**import** java.util.Collections;  
**import** java.util.Scanner;  
**import** java.util.regex.Matcher;  
**import** java.util.regex.Pattern;  
  
**public class** Main {  
 **public static void** main(String[] args) {  
  
 File f1 = **new** File(**"Dan.txt"**);  
  
 ArrayList<String> IPv4 = **new** ArrayList<>();  
 ArrayList<String> IPv6 = **new** ArrayList<>();  
  
 *fillCollections*(IPv4,IPv6,f1);  
  
  
 System.***out***.println(IPv4);  
 System.***out***.println(IPv6);  
 }  
  
 */\*\* Данный метод производит заполнения коллекций строк "IPv4" и "IPv6" типа ArrayList значениями IP найденными в файле  
 \** ***@param IPv4*** *- коллекция типа ArrayList<String> для хранения IPv4  
 \** ***@param IPv6*** *- коллекция типа ArrayList<String> для хранения  
 \** ***@param f1*** *- файл в котором будет происходить поиск IPv4 и IPv6\*/* **private static void** fillCollections(ArrayList<String> IPv4, ArrayList<String> IPv6, File f1)  
 {  
 Pattern patternIPv4 = Pattern.*compile*(**"(\\s+|^)\\d{3}\\.\\d{1,3}\\.\\d{1,3}\\.\\d{1,3}(\\s+|$)"**);  
 Pattern patternIPv6 = Pattern.*compile*(**"(\\s+|^)\\w{4}:\\w{1,4}:\\w{1,4}:\\w{1,4}:\\w{1,4}:\\w{1,4}:\\w{1,4}:\\w{1,4}(\\s+|$)"**);  
 Matcher m;  
 Scanner scanner;  
 **try** {  
 scanner= **new** Scanner(f1);  
 }  
 **catch** (Exception e) {  
 **throw new** RuntimeException();  
 }  
 String buffer;  
 **while**(scanner.hasNextLine())  
 {  
 buffer = scanner.nextLine();  
 m = patternIPv4.matcher(buffer);  
  
 **while**(m.find()) {  
 IPv4.add(m.group());  
 }  
  
 m = patternIPv6.matcher(buffer);  
  
 **while** (m.find())  
 {  
 IPv6.add(m.group());  
 }  
 }  
  
 *removeSpace*(IPv4);  
 *removeSpace*(IPv6);  
  
 *printMenedger*(IPv4,IPv6);  
  
 }  
 */\*\* Данный метод производит удаление пробелов в коллеции строк типа ArrayList  
 \** ***@param IP*** *- коллекция строк типа ArrayList для хранения IP\*/* **private static void** removeSpace(ArrayList<String> IP)  
 { String IpByffer;  
 **for** (**int** i = 0; i < IP.size();i++){  
 IpByffer = IP.get(i);  
 IP.set(i,IpByffer.replaceAll(**" "**,**""**));  
 }  
 }  
  
 */\*\*Метод выполняет инициализацию и закрытие PrintWriter и выполняет сортировку коллекций,  
 \* а затем вызывает метод printInFile с переданными в него коллекциями  
 \** ***@param IPv4*** *- коллекция типа ArrayList<String> для хранения IPv4  
 \** ***@param IPv6*** *- коллекция типа ArrayList<String> для хранения\*/* **private static void** printMenedger(ArrayList<String> IPv4, ArrayList<String> IPv6) {  
 PrintWriter writer;  
  
 **try** {  
 writer = **new** PrintWriter(**"Result.txt"**,**"UTF-8"**);  
 } **catch** (FileNotFoundException e) {  
 **throw new** RuntimeException(e);  
 } **catch** (UnsupportedEncodingException e) {  
 **throw new** RuntimeException(e);  
 }  
  
 Collections.*sort*(IPv4);  
 Collections.*sort*(IPv6);  
  
  
 *printInFile*(IPv4,writer);  
 *printInFile*(IPv6,writer);  
 writer.close();  
 }  
  
 */\*\* Метод выполняет заполнение в файл данных коллеции  
 \** ***@param IP*** *- коллекция строк типа ArrayList для хранения IP  
 \** ***@param writer*** *- инициалезцированный PrintWriter который хранит ссылку на файл\*/* **private static void** printInFile(ArrayList<String> IP, PrintWriter writer)  
 {  
 **for** (**int** i = 0; i < IP.size(); i ++){  
 writer.println(IP.get(i));  
 }  
 }  
}

Результат выполнения программы представлен на рисунке 1.



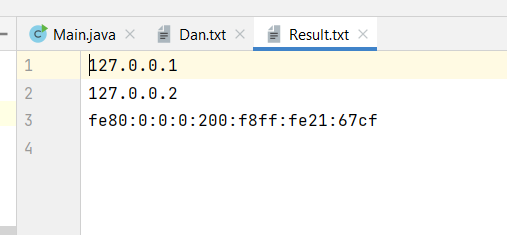


Рисунок 1 – Результат

## Вывод

В ходе выполнения лабораторной работы были получены практические навыки работы с Collections в Java, а также изучена работа с регулярными выражениями.