ФЕДЕРАЛЬНОЕ Государственное АВТОНОМНОЕ образовательное УЧРЕЖДЕНИЕ ВЫСШЕГО образования

**БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ НАЦИОНАЛЬНЫЙ**

**ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ»**

**(НИУ «БелГУ»)**

**ИНСТИТУТ ИНЖЕНЕРНЫХ И ЦИФРОВЫХ ТЕХНОЛОГИЙ**

КАФЕДРА ПРИКЛАДНОЙ ИНФОРМАТИКИ И ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

**Отчет**

по дисциплине «Разработка программных приложений на языках высокого уровня»

специальности 09.03.03 «Прикладная информатика»

**Лабораторная работа № 4**

студента очного отделения

3 курса 12002005 группы

Огородникова Льва Олеговича

Проверил:

старший преподаватель

Лифиренко Максим Вячеславович

БЕЛГОРОД 2023

**Цель работы:** получить практические навыки работы с Collections в Java, изучить работу с регулярными выражениями.

**Вариант №15**

На вход программы подается произвольный текстовый файл, необходимо извлечь из него все пароли, которые удовлетворяют условию от 8 до 15 символов с минимум одной цифрой, одной заглавной и одной строчной буквой. Например A32DjodpL. Напротив каждого подходящего пароля указать число повторений. Результат сохранить в файл.

Пример:

На входе: A32DjodpL Авфыа Ва P232kfPfkdkfl A32DjodpL

На выходе: A32DjodpL 2 P232kfPfkdkfl 1

**Листинг**

import java.io.\*;  
import java.util.\*;  
import java.util.regex.\*;  
import java.awt.\*;  
  
*/\*\*  
 \* Класс для извлечения паролей из текстового файла.  
 \*/*public class PasswordExtractor  
{  
 */\*\*  
 \* Главный метод, который читает файл, ищет пароли, записывает результаты в файл, а затем открывает результирующий файл  
 \*  
 \** ***@param*** *args аргументы командной строки (не используются)  
 \*/* public static void main(String[] args)  
 {  
 try  
 {  
 String content = *readFile*("password.txt");  
 Map<String, Integer> passwordCount = *findPasswords*(content);  
 *writeToFile*("output.txt", passwordCount);  
 File outputFile = new File("output.txt");  
 Desktop.*getDesktop*().open(outputFile);  
 }  
 catch (FileNotFoundException e)  
 {  
 e.printStackTrace();  
 } catch (IOException e)  
 {  
 e.printStackTrace();  
 }  
 }  
  
 */\*\*  
 \* Читает содержимое файла с заданным именем и возвращает его в виде строки.  
 \*  
 \** ***@param*** *filename имя файла для чтения  
 \** ***@return*** *содержимое файла в виде строки  
 \** ***@throws*** *FileNotFoundException если файл не найден  
 \*/* public static String readFile(String filename) throws FileNotFoundException  
 {  
 File file = new File(filename);  
 Scanner scanner = new Scanner(file);  
 String content = scanner.useDelimiter("\\Z").next();  
 scanner.close();  
 return content;  
 }  
  
 */\*\*  
 \* Извлекает пароли из заданной строки и возвращает карту паролей и количество их повторений.  
 \*  
 \** ***@param*** *content текстовая строка для поиска паролей  
 \** ***@return*** *карта паролей и их количества повторений  
 \*/* public static Map<String, Integer> findPasswords(String content)  
 {  
 Pattern pattern = Pattern.*compile*("(?=.\*\\d)(?=.\*[a-z])(?=.\*[A-Z]).{8,15}");  
 Matcher matcher = pattern.matcher(content);  
 Map<String, Integer> passwordCount = new HashMap<>();  
 while (matcher.find())  
 {  
 String password = matcher.group();  
 int count = passwordCount.getOrDefault(password, 0);  
 passwordCount.put(password, count + 1);  
 }  
 return passwordCount;  
 }  
  
 */\*\*  
 \* Записывает карту паролей и количество их повторений в файл с заданным именем.  
 \*  
 \** ***@param*** *filename имя файла для записи результатов  
 \** ***@param*** *passwordCount карта паролей и количества их повторений  
 \** ***@throws*** *IOException если произошла ошибка записи в файл  
 \*/* public static void writeToFile(String filename, Map<String, Integer> passwordCount) throws IOException  
 {  
 FileWriter writer = new FileWriter(filename);  
 for (Map.Entry<String, Integer> entry : passwordCount.entrySet())  
 {  
 String password = entry.getKey();  
 int count = entry.getValue();  
 writer.write(password + " " + count + "\n");  
 }  
 writer.close();  
 }  
}

На рисунке 1 представлен файл с входными данными.

Изображение выглядит как текст

Автоматически созданное описание

Рисунок 1 – Входной файл

На рисунке 2 представлен результат работы программы, сохраненный в файл.

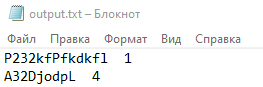


Рисунок 2 – Файл с результатом

**Вывод:** были получены практические навыки работы с Collections в Java, и изучена работа с регулярными выражениями.