Міністерство освіти і науки України

Харківський національний університет ім. В. Н. Каразіна

Факультет комп’ютерних наук

Кафедра безпеки інформаційних систем і технологій

Лабораторна робота №6

*з навчальної дисципліни*

«Спеціалізовані мови програмування та проектування електронних елементів і систем»

Виконав:

Студент групи КБ-41

Кравченко Є.М.

Перевірив:

Олійников Р.В.

Харків – 2020 р.

1.Из текстового файла, содержащего слова (состоящие из латинских букв A-Z в разных регистрах), знаки препинания (,.!?-), пробелы, цифры и спецсимволы (“№;%:\*()+) извлечь все слова и записать в выходной файл, разделенные пробелами.

**private static void** task1(String filename) {  
 **try** (Scanner scanner = **new** Scanner(**new** FileInputStream(filename))) {  
 StringBuilder stringBuilder = **new** StringBuilder();  
 **while** (scanner.hasNextLine()) {  
 stringBuilder.append(scanner.nextLine()).append(**" "**);  
 }  
 String result = String.*join*(**" "**, stringBuilder.toString().split(**"([\\W])"**)).replaceAll(**" +"**, **" "**);  
 System.***out***.println(result);  
 } **catch** (FileNotFoundException ignored) {  
  
 }  
}

2.Запросить у пользователя логин и пароль (пароль - не отображаемый на экране). Выполнить примитивное хэширование (сложить все символы как 16-битовые целые, умножая предыдущий результат на 5 и добавляя к сумме 7 для каждого символа пароля). Сравнить с заранее заданным в программе значением, и по результату напечатать “доступ предоставлен” или “в доступе отказано”.

**private static void** task2() {  
 Console console = System.*console*();  
 console.printf(**"Input login: "**);  
 String login = console.readLine();  
 console.printf(**"Input password: "**);  
 **char**[] password = console.readPassword();  
 **if** (*hash*(password) == *hash*(**"password"**.toCharArray()) && **"login"**.equals(login)) {  
 System.***out***.println(**"Access permit"**);  
 } **else** {  
 System.***out***.println(**"Access deny"**);  
 }  
}  
**private static int** hash(**char**[] chars) {  
 **int** result = 0;  
 **for** (**int** i : chars) {  
 result += i + result \* 5 + 7;  
 }  
 **return** result;  
}

3.Сохранить в одном двоичном файле (последовательно): целое число, булево значение, строку с латинскими и кириллическими буквами, два объекта, в полях которых также присутствуют классы (String, Long и т.п.). Из этого файла прочесть значения и распечатать их на экране (для классов - содержимое их полей).

**private static void** task3() {  
 **class** Person **implements** Serializable {  
 **private** String **name**;  
 **private** Integer **age**;  
  
 **public** Person(String name, Integer age) {  
 **this**.**name** = name;  
 **this**.**age** = age;  
 }  
  
 @Override  
 **public** String toString() {  
 **return "Test{"** +  
 **"name='"** + **name** + **'\''** +  
 **", age="** + **age** +  
 **'}'**;  
 }  
 }  
 **try** (ObjectOutputStream objectOutputStream = **new** ObjectOutputStream(**new** FileOutputStream(**"lab6\_task3.bin"**))) {  
 objectOutputStream.writeInt(123);  
 objectOutputStream.writeBoolean(**true**);  
 objectOutputStream.writeUTF(**"Люблю Java"**);  
 objectOutputStream.writeObject(**new** Person(**"Anna"**, 20));  
 objectOutputStream.writeObject(**new** Person(**"Vita"**, 20));  
 } **catch** (IOException e) {  
 e.printStackTrace();  
 }  
 **try** (ObjectInputStream objectInputStream = **new** ObjectInputStream(**new** FileInputStream(**"lab6\_task3.bin"**))) {  
 System.***out***.println(objectInputStream.readInt());  
 System.***out***.println(objectInputStream.readBoolean());  
 System.***out***.println(objectInputStream.readUTF());  
 System.***out***.println(objectInputStream.readObject());  
 System.***out***.println(objectInputStream.readObject());  
 } **catch** (IOException | ClassNotFoundException e) {  
 e.printStackTrace();  
 }  
}

4.Записать в файл массив из 64 байт, закрыть его. Открыть файл на запись, изменить знак каждого байта с четным индексом (0-го, 2-го, 4-го и т.д.), не перезаписывая весь файл, а только конкретные значения, закрыть файл. Открыть файл на чтение, прочесть массив целиком и распечатать его.

**private static void** task4() {  
 **byte**[] bytes = **new byte**[64];  
 **for** (**int** i = 0; i < bytes.**length**; i++) {  
 bytes[i] = (**byte**) (Math.*random*() \* 256 - 128);  
 }  
 System.***out***.println(Arrays.*toString*(bytes));  
 **try** (FileOutputStream fileOutputStream = **new** FileOutputStream(**"lab6\_task4.txt"**);) {  
 fileOutputStream.write(bytes);  
 } **catch** (IOException e) {  
 e.printStackTrace();  
 }  
 **try** (RandomAccessFile randomAccessFile = **new** RandomAccessFile(**"lab6\_task4.txt"**, **"rw"**)) {  
 **for** (**int** i = 0; i < bytes.**length**; i += 2) {  
 randomAccessFile.seek(i);  
 randomAccessFile.writeByte(-bytes[i]);  
 }  
 } **catch** (IOException e) {  
 e.printStackTrace();  
 }  
 **try** (FileInputStream fis = **new** FileInputStream(**"lab6\_task4.txt"**);) {  
 System.***out***.println(Arrays.*toString*(fis.readAllBytes()));  
 } **catch** (IOException e) {  
 e.printStackTrace();  
 }  
}

5.Запросить у пользователя имя архива, создать такой ZIP-файл в текущем каталоге. Далее итеративно запрашивать имя нового файла в архиве, добавлять данные, введенные пользователем с клавиатуры, в этот файл. Признак того, что ввод текущего файла (внутри архива) завершен - “q!” в самом начале строки. Признак того, что завершен ввод и текущего файла, и всего архива - “Q!” в самом начале строки. Проверить созданных архив (список файлов внутри, и содержание каждого из вложенных файлов) с помощью стандартной программы архивации.

**private static void** task5() {  
 **boolean** write = **true**;  
 System.***out***.println(**"Input archive name"**);  
 String name = **new** Scanner(System.***in***).nextLine();  
 **try** (FileOutputStream fos = **new** FileOutputStream(name + **".zip"**);  
 ZipOutputStream zipOutputStream = **new** ZipOutputStream(fos)) {  
 **while** (write) {  
 File currentFile;  
 System.***out***.println(**"Input file name:"**);  
 currentFile = **new** File(**new** Scanner(System.***in***).nextLine());  
 zipOutputStream.putNextEntry(**new** ZipEntry(currentFile.getName()));  
 StringBuilder stringBuilder = **new** StringBuilder();  
 System.***out***.println(**"Input file content:"**);  
 Scanner scanner = **new** Scanner(System.***in***);  
 **while** (scanner.hasNextLine()) {  
 String line = scanner.nextLine();  
 **if** (line.matches(**"!q.\*"**)) {  
 **break**;  
 }  
 **if** (line.matches(**"!Q.\*"**)) {  
 write = **false**;  
 **break**;  
 }  
 stringBuilder.append(line).append(System.*lineSeparator*());  
 }  
 zipOutputStream.write(stringBuilder.toString().getBytes());  
 }  
 } **catch** (IOException e) {  
 e.printStackTrace();  
 }  
}

6.С помощью стандартной программы архивации создать ZIP-архив, содержащий файлы (текстовые и двоичные), без каталогов. Написать программу, принимающую имя архива как параметр командной строки (запуск - java программа имя\_архива.zip) и извлекающего все файлы в текущий каталог.

**private static void** task6(String zipName) {  
 **try** (ZipInputStream zipInputStream = **new** ZipInputStream(**new** FileInputStream(zipName));  
 ) {  
 ZipEntry zipEntry = zipInputStream.getNextEntry();  
 **while** (Objects.*nonNull*(zipEntry)) {  
 FileOutputStream fileOutputStream = **new** FileOutputStream(**new** File(zipEntry.getName()));  
 **int** a;  
 **while** ((a = zipInputStream.read()) != -1) {  
 fileOutputStream.write(a);  
 }  
 fileOutputStream.close();  
 zipEntry = zipInputStream.getNextEntry();  
 }  
 zipInputStream.closeEntry();  
 } **catch** (IOException e) {  
 e.printStackTrace();  
 }  
}