Министерство науки и высшего образования Российской Федерации

ФГАОУ ВО «УрФУ имени первого Президента России Б.Н.Ельцина»

ИРИТ-РТФ

Центр ускоренного обучения

**Отчёт по лабораторной работе №11**

**«Работа с текстовыми файлами. Файлы с произвольным доступом и файлы для записи объектов»**

Руководитель ст. преподаватель Н.А. Архипов

Студент гр. РИЗ-100028у И. С. Арсентьев

Екатеринбург 2021

Содержание

[Постановка задачи 3](#_Toc74511417)

[ОПИСАНИЕ ЗАДАЧИ 4](#_Toc74511418)

[ХОД ВЫПОЛНЕНИЯ 5](#_Toc74511419)

[ВЫВОДЫ 26](#_Toc74511420)

# Постановка задачи

Цель: Получение навыков ввода/вывода данных файла через символьные потоки. Получение навыков работы с файлами с произвольным доступом, содержащими данные какого-то одного примитивного типа или данные разных примитивных типов. Знакомство с механизмом сериализации и десериализации объектов собственных разработанных классов.

Учебные вопросы:

1. Классы иерархии символьных потоков;

2. Посимвольный ввод/вывод;

3. Буферизованный ввод/вывод данных текстового файла;

4. Преобразование байтовых потоков в символьные;

5. Использование класса PrintWriter;

6. Файлы с произвольным доступом;

7. Понятия сериализации и десериализации;

8. Пример записи объектов в файл и чтения из файла;

9. Задания для самостоятельной работы;

10. Описание результата выполнения лабораторной работы.

# ОПИСАНИЕ ЗАДАЧИ

Задание 1. В отдельных проектах выполнить примеры 1 – 7 (8 и 9 не надо). Протестировать программы с помощью отладчика. Выявить различие в работе программ в примерах 2 и 3. Заменить тип данных переменных int на другие числовые типы по выбору и ознакомиться с методами их чтения/записи и определения положения указателя.

Задание 2. Создать проект, позволяющий из одного текстового файла, содержащего несколько строк (тип String) заранее подготовленного текста на русском языке (обратитесь к классике), построчно переписать в другой текстовый файл слова, отвечающие условию.

Условие:

Переписать в результирующий файл слова, которые начинаются с той же буквы, что и первое слово.

Требования:

–слова из предложения выделять методом split();

–в новом файле следует указать номер строки, в которой искомые слова находились в исходном файле;

–для каждой строки указать количество выбранных слов.

Задание 3. Используя рассмотренные в данной работе примеры 8 и 9, выполнить задание согласно условию в виде двух проектов:

1-й проект – работа с файлом с произвольным доступом;

2-й проект – работа через сериализацию.

Условие:

Записать в исходный файл информацию о фильмах: Название фильма, год выпуска, страна, жанр, стоимость проката. Количество фильмов задать с клавиатуры. Создать программным способом другой файл и переписать в него информацию о фильмах, выпущенных в России.

# ХОД ВЫПОЛНЕНИЯ

1. Код программы File\_RW\_byByte - чтение из одного файла и запись в другой файл данных посимвольно.

/\* Код программы File\_RW\_byByte - чтение из одного файла и запись в другой файл данных посимвольно.\*/

import java.io.\*;

public class File\_RW\_byByte { //Метод main генерирует исключение

public static void main(String[] args) throws IOException {

Reader in=null; // можно сразу записать FileReader in=null; Writer out=null;

// можно сразу записать FileWriter out =null;

Writer out=null; // можно сразу записать FileWriter out =null;

try {

in = new FileReader("С:\\MyFile1.txt"); // файл для чтения

out= new FileWriter("С:\\MyFile2txt", true); // файл для записи разрешено добавление

// Данные считываются и записываются побайтно, как и для // InputStream/OutputStream

int oneByte; // переменная, в которую считываются данные

while ((oneByte = in.read()) != -1) {

// out.write((char)oneByte); // запись с уничтожением ранее

// существующих данных out.append((char)oneByte); // запись с добавлением данных в конец System.out.print((char)oneByte);

}

} catch (IOException e) { System.out.println("Ошибка!!!! "); }

finally{

in.close();

out.close();

} }}//конец программы

2. Код программы Buf\_RW\_3 - чтение из одного файла и запись в другой файл данных построчно с использованием буфера в 1 килобайт.

/\*Код программы Buf\_RW\_3 - чтение из одного файла и запись в другой файл данных построчно с использованием буфера в 1 килобайт.\*/

import java.io.\*;

public class Buf\_RW\_3 {

public static void main(String[] args) throws IOException { BufferedReader br = null;

BufferedWriter out=null;

try { // Создание файловых символьных потоков для чтения и записи

br = new BufferedReader( new FileReader("C:\\Lab11\\пример2.txt" ), 1024);

out = new BufferedWriter( new FileWriter( "C:\\Lab11\\пример21.txt" ));

int lineCount = 0; // счетчик строк

String s; // Переписывание информации из одного файла в другой

while ((s = br.readLine()) != null) { lineCount++;

System.out.println(lineCount + ": " + s);

out.write(s);

out.newLine(); // переход на новую строку размер буфера

}

} catch (IOException e) { System.out.println("Ошибка !!!!!!!!");

}

finally{

br.close();

out.flush();

out.close();

} }}

3. Код программы InConvertInText - чтение и вывод на экран информации из трех источников: файла на диске, интернет-страницы и массива данных типа byte.

/\*Код программы InConvertInText - чтение и вывод на экран информации из трех источников: файла на диске, интернет-страницы и массива данных типа byte.\*/

import java.io.\*;

import java.net.URL;

public class InConvertInText {

public static void readAllByByte( Reader in) throws IOException { while (true) { int oneByte = in.read(); // читает 1 байт

if (oneByte != -1) { // признак конца файла

System.out.print((char) oneByte);

} else {

System.out.print("\n" + " конец "); break;

}}}

public static void main(String[] args) {

try { // С потоком связан файл

InputStream inFile = new FileInputStream("E:\\MyFile1.txt"); // байтовый поток

Reader rFile= new InputStreamReader(inFile,"cp1251"); // символьный поток

readAllByByte(rFile); System.out.print("\n\n\n"); inFile.close(); rFile.close(); // С потоком связана интернет-страница, передается «русская» кодировка

InputStream inUrl = new URL("http://google.com").openStream(); // байтовый поток

Reader rUrl=new InputStreamReader(inUrl, "cp1251"); // символьный поток

readAllByByte(rUrl); System.out.print("\n\n\n"); inUrl.close();

rUrl.close(); // С потоком связан массив типа byte

InputStream inArray = new ByteArrayInputStream( new byte[] {5, 8, 3, 9, 11}); Reader rArray=new InputStreamReader(inArray,"cp1251" ); // символьный

// поток

readAllByByte(rArray); System.out.print("\n\n\n"); inArray.close(); rArray.close();

} catch (IOException e) { System.out.println("Ошибка: "+ e);

}}}

4. Код программы Buf\_WR\_IO\_4 - чтение из одного файла и запись в другой файл данных построчно с использованием буферизации символьных потоков, основанных на байтовых файловых потоках.

/\*Код программы Buf\_WR\_IO\_4 - чтение из одного файла и запись в другой файл данных построчно с использованием буферизации символьных потоков, основанных на байтовых файловых потоках.\*/

import java.io.\*;

public class Buf\_WR\_IO\_4 {

public static void main(String[] args) throws IOException {

BufferedReader br = null; BufferedWriter bw=null; try { // Создание потоков для чтения и записи с нужной кодировкой

br = new BufferedReader(

new InputStreamReader(

new FileInputStream("E:\\MyFile1.txt"),"cp1251"));

bw = new BufferedWriter(

new OutputStreamWriter(

new FileOutputStream("E:\\MyFile2.txt"),"cp1251"));

// Переписывание информации из одного файла в другой

int lineCount = 0; // счетчик строк

String s;

while ((s = br.readLine()) != null) { lineCount++; System.out.println(lineCount + ": " + s);

bw.write(lineCount + ": " + s); // запись без перевода строки

bw.newLine(); // принудительный переход на новую строку

}

} catch (IOException e) { System.out.println("Ошибка !!!!!!!!");

}

finally{

br.close();

bw.flush();

bw.close();

}}}

5. Код программы Buf\_RW\_2 - чтение из одного файла и запись в другой файл с использованием класса PrintWriter.public class Buf\_RW\_2 {

/\*Код программы Buf\_RW\_2 - чтение из одного файла и запись в другой файл с использованием класса PrintWriter.\*/

import java.io.\*;

public class Buf\_RW\_2 {

public static void main(String[] args) {

BufferedReader br = null;

PrintWriter out=null;

try { // Создание потоков

br = new BufferedReader(

new InputStreamReader(

new FileInputStream("C:\\Test\\MyFile1.txt"),"UTF-8"));

out = new PrintWriter("C:\\Test\\MyFile2.txt", "UTF-8");

int lineCount = 0; String s;

while ((s = br.readLine()) != null) { lineCount++; out.println(lineCount + ": " + s);

}

} catch (IOException e) { System.out.println("Ошибка !!!!!!!!"); }

finally{

try {

br.close();

} catch (IOException e) {

e.printStackTrace();

}

out.flush();

out.close();

}}}

//PrintWriter out = new PrintWriter(System.out);

//int lineCount = 0; String s; // Вывод информации из файла на монитор

//while ((s = br.readLine()) != null) { lineCount++; out.println(lineCount + ": " + s);

//}

***6. Код программы RandAccNumb - работа с числовыми данными в файле с произвольным доступом.***/\*Код программы RandAccNumb - работа с числовыми данными в файле с произвольным доступом.\*/

import java.io.File;

import java.io.IOException;

import java.io.RandomAccessFile;

import java.util.Scanner;

public class RandAccNumb{

public static void main(String[] args) {

try {

File folder = new File("Lab11"); if (!folder.exists())

folder.mkdir();

//Создание папки «My» на диске, если она не существует

File f1 = new File(folder+"\\Lab23.txt"); f1.createNewFile();

Scanner sc = new Scanner(System.in, "cp1251");

//Создание файла в папке

System.out.print("Сколько чисел надо записать в файл? \n => "); int count = sc.nextInt(); // ввести количество чисел // Открыть файл одновременно для чтения и записи "rw"

RandomAccessFile rf = new RandomAccessFile(f1, "rw");

System.out.println("Исходный размер файла в байтах =" + rf.length()

+ ", указатель стоит на " + rf.getFilePointer() + "-м байте");

System.out.println("Введите числа:"); for (int i = 0; i < count; i++) {

rf.writeInt(sc.nextInt()); // Записать числа в файл. На каждое

} // число типа int приходится 4 байта

System.out.println("Новый размер файла в байтах =" + rf.length()

+ ", указатель стоит на " + rf.getFilePointer() + "-м байте"); System.out.println("Количество байт на 1 число = " + rf.length() / count); rf.close();

// Открыть файл только для чтения "r"

rf = new RandomAccessFile(f1, "r");

// Прочитать числа из файла и вывести на экран

System.out.println("\n Числа в файле:");

for (int i = 0; i < count; i++) { // i – текущий номер числа

rf.seek(i \* 4); // перевод указателя на текущее число  i\* 4 байта

// в данной ситуации при последовательном считывании операцию seek() можно было не использовать

System.out.println("Число" + i + ": " + rf.readInt());

}

System.out.println("Числа в обратном порядке:");

for (int i = count - 1; i >= 0; i--) {

rf.seek(i \* 4); // операцию использовать обязательно!

System.out.println("Число" + i + ": " + rf.readInt());

}

rf.seek(rf.getFilePointer() - 4); // перевод указателя на последнее число

System.out.println(" Количество чисел в файле= " + rf.length()/4

+ ", последнее число= " + rf.readInt());

// Поиск заданного числа в файле и определение его номера (номеров)

System.out.print("\nВведите число, которое нужно найти в файле => ");

int kol = 0; // количество искомых чисел в файле

for (int i = 0; i < count; i++) { rf.seek(i \* 4);

int number = rf.readInt();

int x =0;

if (number == x) { kol++;

System.out.print("номер " + i+ " ");

}

}

System.out.println(" количество искомых чисел = " + kol);

rf.close();

int number1 = 0;

int number2 = 0;

// СОРТИРОВКА ЧИСЕЛ В ФАЙЛЕ МЕТОДОМ «ПУЗЫРЬКА»

rf = new RandomAccessFile(f1, "rw"); // открыт для чтения и записи

for (int k = 0; k < count; k++) { // k – номер просмотра

for (int i = 0; i < count - k -1; i++) { // i – номер числа

rf.seek(i \* 4); // переход к i-тому числу

number1 = rf.readInt();

number2 = rf.readInt();

rf.seek(i \* 4);

rf.writeInt(number2);

rf.writeInt(number1);

//чтение i-того и(i+1)-го чисел в переменные

if (number1 > number2) { // условие для сортировки по возрастанию

} } }

//возврат указателя на i-тое число и перестановка (запись чисел в обратном порядке)

System.out.println("\n Числа, отсортированные в файле:"); for (int i = 0; i < count; i++) {

rf.seek(i \* 4);

System.out.print(" " + rf.readInt());

}

rf.close();

} catch (IOException e) { System.out.println("End of file " + e);

}

}}

***7. Код программы RandFtxt - запись строк и чтение их из файла с произвольным доступом.***/\*Код программы RandFtxt - запись строк и чтение их из файла с произвольным доступом.\*/

import java.io.File;

import java.io.IOException;

import java.io.RandomAccessFile;

import java.util.Scanner;

//Выполнить запись строк и чтение их из файла с произвольным доступом.

public class RandFtxt{

public static void main(String[] args) {

try {

File folder = new File("E:\\My");

if (!folder.exists())

folder.mkdir();

File f1 = new File("E:\\My\\strokiRand.txt");

f1.createNewFile();

//При вводе строк символ абзаца (нажатие Enter) имеет такой же размер в байтах, как и другие символы, и учитывается при вычислении размера файла

Scanner sc = new Scanner(System.in, "UTF-8");

System.out.print("Сколько строк надо записать в файл? \n =>");

int count = sc.nextInt();

sc.nextLine(); // очистка буфера после ввода числа

RandomAccessFile rf = new RandomAccessFile(f1, "rw"); // чтение/запись

rf.setLength(0);

long len = rf.length();

System.out.println("Открыт файл размером " + len + " байт");

System.out.println("Введите строки:");

int kol = 0; // счетчик букв

// Записать строки в файл

for (int i = 0; i < count; i++) {

String s = sc.nextLine();

rf.writeUTF(s);

kol += s.length();

}

len = rf.length();

System.out.println("Размер файла в байтах = " + len);

rf.close();

// Открыть файл для чтения "r"

rf = new RandomAccessFile(f1, "r");

// Вывод строк из файла на экран

System.out.println("Строки из файла:");

rf.seek(0); // перевести указатель в начало файла (на первое слово)

for (int i = 0; i < count; i++)

System.out.println("Строка " + i + " начинается с байта "

+ rf.getFilePointer() + " - " + rf.readUTF() + " - заканчивается байтом " + (rf.getFilePointer() - 1));

rf.close();

} catch (IOException e) {

System.out.println("End of file " + e);

}

} }

8. Записать в файл с произвольным доступом данные о заданном количестве сотрудников и считать информацию из файла.

Данные о сотрудниках:

– фамилия, должность – записывается в формате строки UTF-8;

– оклад – число типа int.

Один символ в формате UTF – 1 байт, число типа int – 4 байта. Разное количество байтов для фамилии и должности дополним с помощью «дозаписи» любых чисел типа byte до общего размера, например, 20. Значение выбрано из расчета того, что фамилия и должность будут иметь длину не более 20 символов.

/\*Записать в файл с произвольным доступом данные о

заданном количестве сотрудников и считать информацию из файла.

Данные о сотрудниках:

– фамилия, должность – записывается в формате строки UTF-8;

– оклад – число типа int.

Один символ в формате UTF – 1 байт, число типа int – 4 байта.

Разное количество байтов для фамилии и должности дополним с

помощью «дозаписи» любых чисел типа byte до общего размера,

например, 20. Значение выбрано из расчета того, что фамилия и

должность будут иметь длину не более 20 символов. \*/

import java.io.File;

import java.io.IOException;

import java.io.RandomAccessFile;

import java.util.Scanner;

public class RandAccF\_record {

public static void main(String[] args) {

try {

File folder = new File("C:\\Test\\My");

if (!folder.exists())

folder.mkdir();

File f1 = new File("C:\\Users\\My\\rec\_RAF.txt");

if (!f1.exists())

f1.createNewFile();

RandomAccessFile rf = new RandomAccessFile(f1,"rw"); // чтение и запись

long fSize = rf.length(); // размер файла

//System.out.println(rf.length());

Scanner sc = new Scanner(System.in, "UTF-8");

System.out.print("Введите количество сотрудников для записи в файл\n"

+ "=> ");

int kol = sc.nextInt();

sc.nextLine(); // очистка буфера после ввода числа

String fam, doljnost;

int oklad;

//----ЗАПИСЬ ИНФОРМАЦИИ О СОТРУДНИКАХ В ФАЙЛ----

for (int i = 0; i < kol; i++) {

System.out.print("Введите фамилию " + (i + 1) + " сотрудника => ");

fam = sc.next();

rf.seek(rf.length()); // поиск конца файла

rf.writeUTF(fam); // запись фамилии

for (int j = 0; j < 20 - fam.length(); j++)

rf.writeByte(1); // добавление байтов до 20-ти любой цифрой (=1)

System.out.print("Введите его должность => ");

doljnost = sc.next();

rf.seek(rf.length()); // поиск конца файла

rf.writeUTF(doljnost); // запись должности

for (int j = 0; j < 20 - doljnost.length(); j++)

rf.writeByte(1); // добавление байтов до кол=20 любым числом

System.out.print("Введите его оклад => ");

oklad = sc.nextInt();

sc.nextLine(); // очистка буфера

rf.seek(rf.length()); // поиск конца файла

rf.writeInt(oklad); // запись оклада

}

rf.close();

//----ЧТЕНИЕ ИНФОРМАЦИИ О СОТРУДНИКАХ ИЗ ФАЙЛА----

rf = new RandomAccessFile(f1, "r");

rf.seek(0); // перемещение в начало файла

System.out.println("Информация о сотрудниках");

System.out.println("Фамилия \t\t Должность \t\t Оклад");

for (int i = 0; i < kol; i++) {

fam = rf.readUTF();

for (int j = 0; j < 20 - fam.length(); j++)

rf.readByte();

//System.out.println(fam);

doljnost = rf.readUTF();

for (int j = 0; j < 20 - doljnost.length(); j++)

rf.readByte();

oklad = rf.readInt();

//System.out.println(fam);

//System.out.println(doljnost);

//System.out.println(oklad);

System.out.println(fam + "\t\t\t" + doljnost + "\t\t\t" + oklad);

System.out.println(doljnost=="oklad");

}

rf.close();

} catch (IOException e) {

System.out.println("End of file " + e);

} }}

***Самостоятельные задания***

1. Создать проект, позволяющий из одного текстового файла, содержащего несколько строк (тип String) заранее подготовленного текста на русском языке (обратитесь к классике), построчно переписать в другой текстовый файл слова, отвечающие условию.

Условие:

Переписать в результирующий файл слова, которые начинаются с той же буквы, что и первое слово.

Требования:

–слова из предложения выделять методом split();

–в новом файле следует указать номер строки, в которой искомые слова находились в исходном файле;

–для каждой строки указать количество выбранных слов.

Создать проект, позволяющий из одного текстового файла, содержащего несколько строк (тип String) заранее подготовленного текста на русском языке (обратитесь к классике), построчно переписать в другой текстовый файл слова, отвечающие условию.

Условие:

Переписать в результирующий файл слова, которые начинаются с той же буквы, что и первое слово.

Требования:

–слова из предложения выделять методом split();

–в новом файле следует указать номер строки, в которой искомые слова находились в исходном файле;

–для каждой строки указать количество выбранных слов.

/\*Создать проект, позволяющий из одного текстового файла, содержащего несколько строк (тип String) заранее подготовленного текста на русском языке (обратитесь к классике), построчно переписать в другой текстовый файл слова, отвечающие условию.

Условие:

Переписать в результирующий файл слова, которые начинаются с той же буквы, что и первое слово.

Требования:

–слова из предложения выделять методом split();

–в новом файле следует указать номер строки, в которой искомые слова находились в исходном файле;

–для каждой строки указать количество выбранных слов.\*/

import java.io.BufferedReader;

import java.io.File;

import java.io.FileInputStream;

import java.io.IOException;

import java.io.InputStreamReader;

import java.io.PrintWriter;

public class zadacha2 {

public static void main(String[] args) throws IOException {

BufferedReader in = null;

PrintWriter out = null;

int i = 0;

try {

in = new BufferedReader( new InputStreamReader( new FileInputStream(GetFileByName("Task3Input.txt")),"cp1251"));

out = new PrintWriter(GetFileByName("Task3Output.txt"), "cp1251");

String line;

while ((line = in.readLine()) != null) {

i++;

int schet = 0;

String[] words = line.split(" ");

for (String word : words) {

if (StartsWithСonstant(word.toLowerCase(),String.valueOf(line.charAt(0)).toLowerCase())) {

System.out.println("строка "+i+": "+word);

out.println("строка "+i+": "+word);

schet = schet +1;

}

}

System.out.println("строка "+i+" слов: "+schet);

System.out.println();

out.println("строка "+i+" слов: "+schet);

out.println();

}}

catch (Exception e) {

System.out.println("Error: " + e.getMessage());

}

finally {

in.close();

out.flush();

out.close();

}

}

private static String DirPath = "C:\\Test\\zadacha\\%s";

public static File GetFileByName(String filename) throws IOException {

File f1=new File(String.format(DirPath, filename));

return f1;

}

private static boolean StartsWithСonstant(String inStr,String k)

{

return k.contains(inStr.subSequence(0, 1));

}

}

Результат работы программы представлен на рисунке 1.

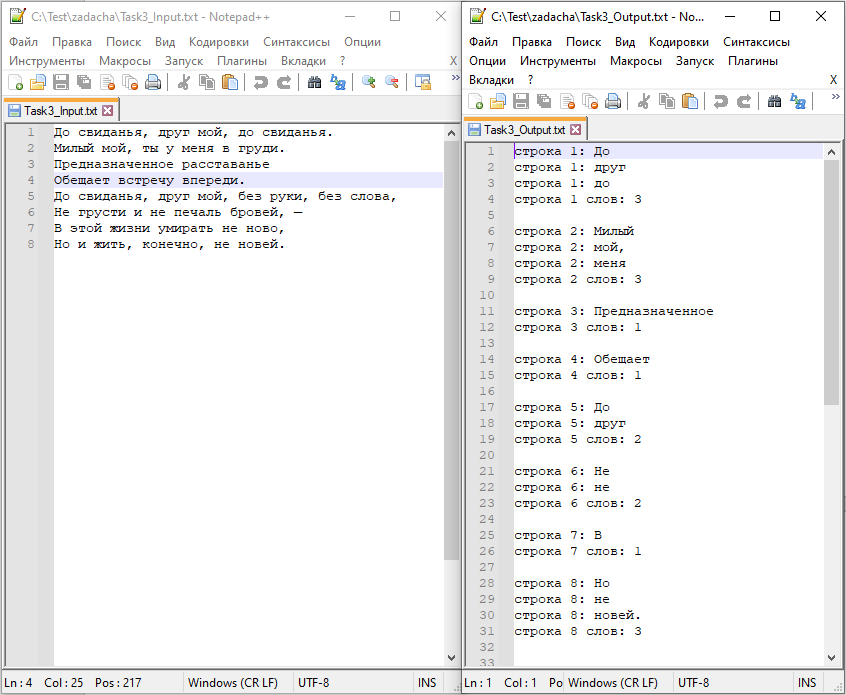


Рисунок 1. Результат работы программы zadacha2

9. Код программы zadacha3

/\*Используя рассмотренные в данной работе примеры 8 и 9, выполнить задание согласно условию в виде двух проектов:

1-й проект – работа с файлом с произвольным доступом;

2-й проект – работа через сериализацию.

Условие:

Записать в исходный файл информацию о фильмах:

Название\_фильма, год\_выпуска, страна, жанр, стоимость\_проката

Количество фильмов задать с клавиатуры.

Создать программным способом другой файл и переписать в него

информацию о фильмах, выпущенных в России. \*/

import java.io.\*;

import java.util.Objects;

import java.util.Scanner;

class Film implements Serializable {

String nazvanie, god, strana, zhanr;

int stoimost;

}

public class zadacha3 {

public static void main(String[] args) throws IOException, ClassNotFoundException {

Scanner sc = new Scanner(System.in, "UTF-8");

PrintWriter out = null;

// создается файл на диске

File f = new File("C:\\Test\\My\\MyFileSer1.txt");

f.createNewFile();

// -------------ЗАПИСЬ ОБЪЕКТА В ФАЙЛ-------------

// Создается поток для записи объекта

FileOutputStream fos = new FileOutputStream(f);

ObjectOutputStream oos = new ObjectOutputStream(fos);

System.out.print("Введите количество фильмов для записи в файл\n"

+ "=> ");

int kol = sc.nextInt();

sc.nextLine(); // очистка буфера после ввода числа

for (int i = 0; i < kol; i++) {

// Вводится информация об объекте

Film film = new Film();

System.out.println("Введите название фильма" + (i + 1) + ": ");

System.out.print("Название фильма => ");

film.nazvanie = sc.nextLine();

System.out.print("Год выпуска=> ");

film.god = sc.nextLine();

System.out.print("Страна выпуска=> ");

film.strana = sc.nextLine();

System.out.print("Жанр=> ");

film.zhanr = sc.nextLine();

System.out.print("Введите стоимость его проката => ");

film.stoimost = sc.nextInt();

sc.nextLine(); // очистка буфера после ввода числа

// Объект записывается в файл

oos.writeObject(film);}

// Дописывается информация и закрывается файловый поток

oos.flush();

oos.close();

// -------------ЧТЕНИЕ ОБЪЕКТА ИЗ ФАЙЛА-------------

out = new PrintWriter(GetFileByName("rec\_RAF\_res1.txt"), "UTF-8");

// Создается поток для чтения объекта из файла

FileInputStream fis = new FileInputStream(f);

ObjectInputStream oin = new ObjectInputStream(fis);

// Объект считывается, и на экран выводится требуемая информация

System.out.println("Название \t\t Год выпуска \t\t Страна \t\t Жанр \t\t Стоимость проката");

for (int i = 0; i < kol; i++) {

Film film = new Film();

film = (Film) oin.readObject();

System.out.println(film.nazvanie + "\t\t\t" + film.god + "\t\t\t" + film.strana + "\t\t\t" + film.zhanr + "\t\t\t" + film.stoimost);

String kk = "Russia";

if (Objects.equals(kk,film.strana.toLowerCase())) {

out.println("Название \t\t Год выпуска \t\t Страна \t\t Жанр \t\t Стоимость проката");

out.println(film.nazvanie + "\t\t\t" + film.god + "\t\t\t" + film.strana + "\t\t\t" + film.zhanr + "\t\t\t" + film.stoimost);

}

}

// Поток закрывается

oos.close();

out.flush();

out.close();

}

private static String DirPath = "C:\\Test\\My\\%s";

public static File GetFileByName(String filename) throws IOException {

File f1 = new File(String.format(DirPath, filename));

return f1;

}}

Результат работы программы представлен на рисунке 2.

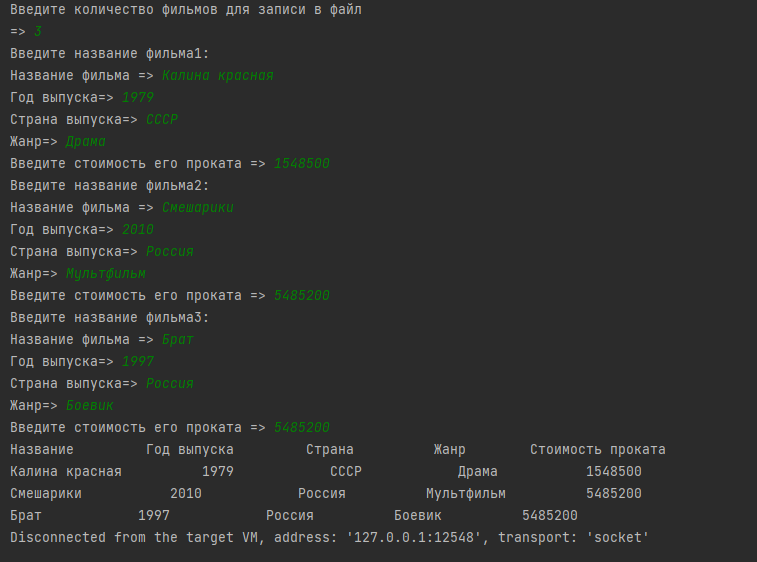


Рисунок 2. Результат работы программы zadacha3

10. Код программы zadacha3\_1

/\*Используя рассмотренные в данной работе примеры 8 и

9, выполнить задание согласно условию в виде двух проектов:

1-й проект – работа с файлом с произвольным доступом;

2-й проект – работа через сериализацию.

Условие:

Записать в исходный файл информацию о фильмах:

Название\_фильма, год\_выпуска, страна, жанр, стоимость\_проката

Количество фильмов задать с клавиатуры.

Создать программным способом другой файл и переписать в него

информацию о фильмах, выпущенных в России.\*/

import java.io.\*;

import java.util.Objects;

import java.util.Scanner;

class Film implements Serializable {

String nazvanie, god, strana, zhanr;

int stoimost;

}

public class zadacha3\_1{

public static void main(String[] args) throws IOException, ClassNotFoundException {

Scanner sc = new Scanner(System.in, "UTF-8");

PrintWriter out = null;

// создается файл на диске

File f = new File("C:\\Test\\My\\3.1.txt");

f.createNewFile();

// -------------ЗАПИСЬ ОБЪЕКТА В ФАЙЛ-------------

// Создается поток для записи объекта

FileOutputStream fos = new FileOutputStream(f);

ObjectOutputStream oos = new ObjectOutputStream(fos);

System.out.print("Введите количество фильмов для записи в файл\n"

+ "=> ");

int kol = sc.nextInt();

sc.nextLine(); // очистка буфера после ввода числа

for (int i = 0; i < kol; i++) {

// Вводится информация об объекте

Film film = new Film();

System.out.println("Введите название фильма" + (i + 1) + ": ");

System.out.print("Название фильма => ");

film.nazvanie = sc.nextLine();

System.out.print("Год выпуска=> ");

film.god = sc.nextLine();

System.out.print("Страна выпуска=> ");

film.strana = sc.nextLine();

System.out.print("Жанр=> ");

film.zhanr = sc.nextLine();

System.out.print("Введите стоимость его проката => ");

film.stoimost = sc.nextInt();

sc.nextLine(); // очистка буфера после ввода числа

// Объект записывается в файл

oos.writeObject(film);

}

// Дописывается информация и закрывается файловый поток

oos.flush();

oos.close();

// -------------ЧТЕНИЕ ОБЪЕКТА ИЗ ФАЙЛА-------------

out = new PrintWriter(GetFileByName("rec\_RAF\_res1.txt"), "UTF-8");

// Создается поток для чтения объекта из файла

FileInputStream fis = new FileInputStream(f);

ObjectInputStream oin = new ObjectInputStream(fis);

// Объект считывается, и на экран выводится требуемая информация

System.out.println("Название \t\t Год выпуска \t\t Страна \t\t Жанр \t\t Стоимость проката");

for (int i = 0; i < kol; i++) {

Film film = new Film();

film = (Film) oin.readObject();

System.out.println(film.nazvanie + "\t\t\t" + film.god + "\t\t\t" + film.strana + "\t\t\t" + film.zhanr + "\t\t\t" + film.stoimost);

String kk = "Россия";

if (Objects.equals(kk,film.strana.toLowerCase())) {

out.println("Название \t\t Год выпуска \t\t Страна \t\t Жанр \t\t Стоимость проката");

out.println(film.nazvanie + "\t\t\t" + film.god + "\t\t\t" + film.strana + "\t\t\t" + film.zhanr + "\t\t\t" + film.stoimost);

}

}

// Поток закрывается

oos.close();

out.flush();

out.close();

}

private static String DirPath = "C:\\Test\\My\\%s";

public static File GetFileByName(String filename) throws IOException {

File f1 = new File(String.format(DirPath, filename));

return f1;

}

}

Результат работы программы представлен на рисунке 3.

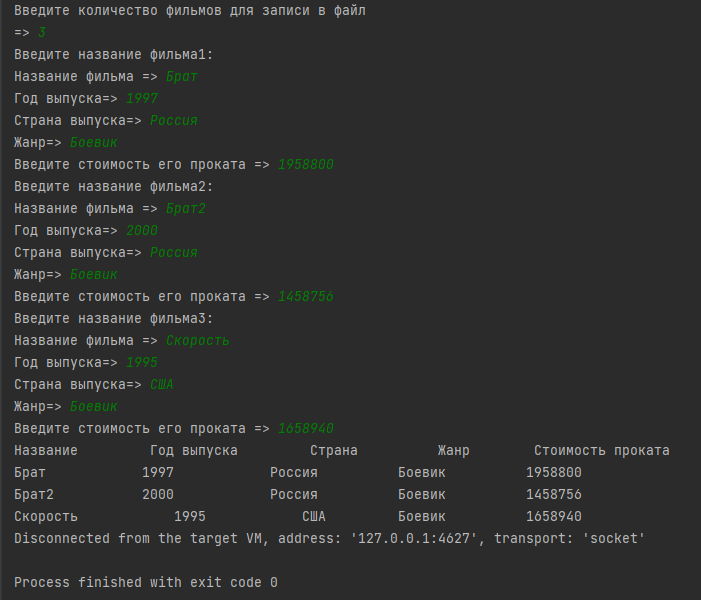


Рисунок 3. Результат работы программы zadacha3\_1

# ВЫВОДЫ

В ходе выполнения лабораторной работы были получены навыки ввода/вывода данных файла через символьные потоки, навыки работы с файлами с произвольным доступом, содержащими данные какого-то одного примитивного типа или данные разных примитивных типов. Ознакомлен с механизмом сериализации и десоциализации объектов собственных разработанных классов.