Министерство образования и науки Российской Федерации

Севастопольский государственный университет

Институт информационных технологий и управления в технических системах

Кафедра ИС

ОТЧЕТ

по лабораторной работе №3

ИССЛЕДОВАНИЕ КОЛЛЕКЦИЙ И ИТЕРАТОРОВ JAVA

Выполнил:

ст. гр. ИС/б-21-о

Куркчи А. Э.

Проверил:

Кузнецов А. С.

Севастополь

2016

1. ЦЕЛЬ РАБОТЫ

В ходе выполнения данной лабораторной работы необходимо ознакомиться с организацией коллекций объектов на языке Java, приобрести практические навыки использования списков, очередей, хеш-таблиц при создании Java программ.

1. ПОСТАНОВКА ЗАДАЧИ

2.1. В соответствии с вариантом задания (см. таблицу 4.1) реализовать класс для представления требуемой информации.

2.2. Реализовать коллекцию типа Т1(см. таблицу 4.1) объектов разработанного в п. 2.1. класса с возможностью ввода элементов из файла, вывода на консоль, проверки членства по введенному с консоли значению поля 1. Имя файла вводить параметром командной строки –i.

2.3. Реализовать коллекцию типа LinkedList объектов разработанного в п. 2.1. с возможностью: упорядочивания по полю 1 (использовать Collections.sort(list)); с возможностью упорядочивания по полю P (см. таблицу 4.1) в направлении U класса (использовать Collections.sort(list, myComp), где myComp – экземпляр разработанного класса, реализующего интерфейс Comparator); с возможностью ввода элементов из файла, вывода на консоль и сохранения в файл. Имена файлов вводить параметрами командной строки –i и –o.

2.4. Реализовать коллекцию типа Т2(см. таблицу 4.1) объектов разработанного в п. 2.1. класса c ключом по значению поля 1, с возможностью ввода элементов из файла, вывода на консоль в виде «Ключ -> Значения» (значения остальных полей), вывода значения полей по введенному с консоли значению поля 1. Имя файла вводить параметром командной строки –i.

2.5. Реализовать класс Lab3Java, в методе main которого реализовать работу с объектами классов из п. 2.1-2.4:

1. Ввести записи из файла, заданного параметром командной строки –i в коллекцию T1.

2. Отобразить записи в консоли.

3. Предложить пользователю ввести значение поля 1.

4. Отобразить в консоли результат проверки наличия записи по введенному значению поля 1.

5. Ввести записи из файла, заданного параметром командной строки –i в коллекцию LinkedList.

6. Отобразить записи в консоли. Отсортировать по полю 1. Отобразить записи в консоли. Отсортировать по полю P в направлении U. Отобразить записи в консоли.7. Вывести записи в файл, заданный параметром командной строки –o.

8. Ввести записи из файла, заданного параметром командной строки –i в коллекцию T2.

9. Отобразить записи в консоли.

10. Предложить пользователю ввести значение поля 1.

11. Отобразить в консоли значения остальных полей по введенному значению поля 1.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A | 2 | Возрастание | LinkedHashSet | LinkedHashMap |

Вариант № 6

1. ТЕКСТ ПРОГРАММЫ

Файл Main.java

package ru**.**justnero**.**study**.**sevsu**.**java**.**lab03**;**

**import** java**.**io**.**File**;**

**import** java**.**util**.**Collections**;**

**import** java**.**util**.**Scanner**;**

public class Main **{**

protected static String inFile **=** ""**;**

protected static String outFile **=** ""**;**

public static void main**(**String**[]** args**)** **{**

**if(!**init**(**args**))** **{**

**return;**

**}**

Scanner scn **=** **new** Scanner**(**System**.**in**);**

FirstList list1 **=** **new** FirstList**();**

list1**.**read**(**inFile**);**

\_log**(**"\tFirst list:\n"**);**

list1**.**print**();**

\_log**(**"Input book author to be found\n"**);**

**if(**list1**.**contains**(**scn**.**next**()))** **{**

\_log**(**"Book found\n"**);**

**}** **else** **{**

\_log**(**"No such book\n"**);**

**}**

\_log**(**"\n"**);**

SecondList list2 **=** **new** SecondList**();**

list2**.**read**(**inFile**);**

\_log**(**"\tSecond list unsorted:\n"**);**

list2**.**print**();**

\_log**(**"\tSecond list sorted by 1st field:\n"**);**

Collections**.**sort**(**list2**);**

list2**.**print**();**

\_log**(**"\tSecond list sorted by 2nd field:\n"**);**

Collections**.**sort**(**list2**,new** BookComparator**());**

list2**.**print**();**

list2**.**write**(**outFile**);**

\_log**(**"\n"**);**

ThirdList list3 **=** **new** ThirdList**();**

list3**.**read**(**inFile**);**

\_log**(**"\tThird list:\n"**);**

list3**.**print**();**

\_log**(**"Input book author to be found\n"**);**

list3**.**print**(**scn**.**next**());**

\_log**(**"\n"**);**

scn**.**close**();**

**}**

private static boolean init**(**String**[]** args**)** **{**

int i **=** 0**;**

**while(**i **<** args**.**length**)** **{**

**if(**args**[**i**].**equalsIgnoreCase**(**"-i"**))** **{**

inFile **=** args**[++**i**];**

**}** **else** **if(**args**[**i**].**equalsIgnoreCase**(**"-o"**))** **{**

outFile **=** args**[++**i**];**

**}** **else** **{**

\_log**(**"Unknown argument: "**,**args**[**i**]);**

**}**

i**++;**

**}**

**if(**inFile**.**isEmpty**())** **{**

\_log**(**"Input file is unset, use -i argument"**);**

**return** **false;**

**}** **else** **if(!(new** File**(**inFile**)).**exists**())** **{**

\_log**(**"Input file does not exists"**);**

**return** **false;**

**}**

**if(**outFile**.**isEmpty**())** **{**

\_log**(**"Output file is unset, use -o argument"**);**

**return** **false;**

**}**

**return** **true;**

**}**

private static void \_log**(**String**...** args**)** **{**

StringBuilder sb **=** **new** StringBuilder**();**

**for(**String str **:** args**)** **{**

sb**.**append**(**str**);**

**}**

System**.**out**.**print**(**sb**.**toString**());**

**}**

**}**

Файл Book.java

package ru**.**justnero**.**study**.**sevsu**.**java**.**lab03**;**

**import** java**.**util**.**Scanner**;**

public class Book **implements** Comparable**<**Book**>** **{**

protected final String author**;**

protected final int publishYear**;**

protected final int pageCount**;**

protected final String publisher**;**

public Book**(**String author**,** int publishYear**,** int pageCount**,** String publisher**)** **{**

**this.**author **=** author**;**

**this.**publishYear **=** publishYear**;**

**this.**pageCount **=** pageCount**;**

**this.**publisher **=** publisher**;**

**}**

public String getAuthor**()** **{**

**return** author**;**

**}**

public int getPublishYear**()** **{**

**return** publishYear**;**

**}**

public int getPageCount**()** **{**

**return** pageCount**;**

**}**

public String getPublisher**()** **{**

**return** publisher**;**

**}**

public static Book read**(**Scanner scn**)** **{**

**return** **new** Book**(**scn**.**next**(),**scn**.**nextInt**(),**scn**.**nextInt**(),**scn**.**next**());**

**}**

@Override

public String toString**()** **{**

StringBuilder sb **=** **new** StringBuilder**();**

sb**.**append**(**"Book author: "**);**

sb**.**append**(**author**);**

sb**.**append**(**"; year: "**);**

sb**.**append**(**publishYear**);**

sb**.**append**(**"; pages: "**);**

sb**.**append**(**pageCount**);**

sb**.**append**(**"; publisher: "**);**

sb**.**append**(**publisher**);**

sb**.**append**(**";"**);**

**return** sb**.**toString**();**

**}**

public String toKeyValueString**()** **{**

StringBuilder sb **=** **new** StringBuilder**();**

sb**.**append**(**author**);**

sb**.**append**(**" -> "**);**

sb**.**append**(**publishYear**);**

sb**.**append**(**"; "**);**

sb**.**append**(**pageCount**);**

sb**.**append**(**"; "**);**

sb**.**append**(**publisher**);**

sb**.**append**(**";"**);**

**return** sb**.**toString**();**

**}**

public String toRawString**()** **{**

StringBuilder sb **=** **new** StringBuilder**();**

sb**.**append**(**author**);**

sb**.**append**(**" "**);**

sb**.**append**(**publishYear**);**

sb**.**append**(**" "**);**

sb**.**append**(**pageCount**);**

sb**.**append**(**" "**);**

sb**.**append**(**publisher**);**

sb**.**append**(**" "**);**

**return** sb**.**toString**();**

**}**

@Override

public int compareTo**(**Book o**)** **{**

**return** author**.**compareTo**(**o**.**author**);**

**}**

**}**

Файл BookComparator.java

package ru**.**justnero**.**study**.**sevsu**.**java**.**lab03**;**

**import** java**.**util**.**Comparator**;**

public class BookComparator **implements** Comparator**<**Book**>** **{**

@Override

public int compare**(**Book b1**,** Book b2**)** **{**

**return** b1**.**getPublishYear**()** **-** b2**.**getPublishYear**();**

**}**

**}**

Файл FirstList.java

package ru**.**justnero**.**study**.**sevsu**.**java**.**lab03**;**

**import** java**.**io**.**FileInputStream**;**

**import** java**.**io**.**FileNotFoundException**;**

**import** java**.**util**.**LinkedHashSet**;**

**import** java**.**util**.**Scanner**;**

public class FirstList **extends** LinkedHashSet**<**Book**>** **{**

public void read**(**String filename**)** **{**

**this.**clear**();**

**try(**Scanner scn **=** **new** Scanner**(new** FileInputStream**(**filename**)))** **{**

**while(**scn**.**hasNext**())** **{**

add**(**Book**.**read**(**scn**));**

**}**

**}** **catch** **(**FileNotFoundException e**)** **{**

System**.**out**.**println**(**"File not found"**);**

e**.**printStackTrace**();**

**}**

**}**

public void print**()** **{**

**for(**Book b **:** **this)** **{**

System**.**out**.**println**(**b**.**toString**());**

**}**

**}**

public boolean contains**(**String author**)** **{**

**for(**Book b **:** **this)** **{**

**if(**b**.**getAuthor**().**equalsIgnoreCase**(**author**))** **{**

**return** **true;**

**}**

**}**

**return** **false;**

**}**

**}**

Файл SecondList.java

package ru**.**justnero**.**study**.**sevsu**.**java**.**lab03**;**

**import** java**.**io**.**FileInputStream**;**

**import** java**.**io**.**FileNotFoundException**;**

**import** java**.**io**.**FileOutputStream**;**

**import** java**.**io**.**PrintWriter**;**

**import** java**.**util**.**LinkedList**;**

**import** java**.**util**.**Scanner**;**

public class SecondList **extends** LinkedList**<**Book**>** **{**

public void read**(**String filename**)** **{**

**this.**clear**();**

**try(**Scanner scn **=** **new** Scanner**(new** FileInputStream**(**filename**)))** **{**

**while(**scn**.**hasNext**())** **{**

add**(**Book**.**read**(**scn**));**

**}**

**}** **catch** **(**FileNotFoundException e**)** **{**

System**.**out**.**println**(**"File not found"**);**

e**.**printStackTrace**();**

**}**

**}**

public void print**()** **{**

**for(**Book b **:** **this)** **{**

System**.**out**.**println**(**b**.**toString**());**

**}**

**}**

public void write**(**String filename**)** **{**

**try(**PrintWriter wrt **=** **new** PrintWriter**(new** FileOutputStream**(**filename**)))** **{**

**for(**Book b **:** **this)** **{**

wrt**.**write**(**b**.**toRawString**());**

wrt**.**write**(**'\n'**);**

**}**

**}** **catch** **(**FileNotFoundException e**)** **{**

System**.**out**.**println**(**"File not found"**);**

e**.**printStackTrace**();**

**}**

**}**

**}**

Файл ThirdList.java

package ru**.**justnero**.**study**.**sevsu**.**java**.**lab03**;**

**import** java**.**io**.**FileInputStream**;**

**import** java**.**io**.**FileNotFoundException**;**

**import** java**.**util**.**LinkedHashMap**;**

**import** java**.**util**.**Scanner**;**

public class ThirdList **extends** LinkedHashMap**<**String**,**Book**>** **{**

public void read**(**String filename**)** **{**

**this.**clear**();**

Book b**;**

**try(**Scanner scn **=** **new** Scanner**(new** FileInputStream**(**filename**)))** **{**

**while(**scn**.**hasNext**())** **{**

b **=** Book**.**read**(**scn**);**

put**(**b**.**getAuthor**(),**b**);**

**}**

**}** **catch** **(**FileNotFoundException e**)** **{**

System**.**out**.**println**(**"File not found"**);**

e**.**printStackTrace**();**

**}**

**}**

public void print**()** **{**

**for(**Book b **:** **this.**values**())** **{**

System**.**out**.**println**(**b**.**toKeyValueString**());**

**}**

**}**

public void print**(**String author**)** **{**

Book b **=** get**(**author**);**

**if(**b **!=** **null)** **{**

System**.**out**.**println**(**b**.**toKeyValueString**());**

**}** **else** **{**

System**.**out**.**println**(**"No such book"**);**

**}**

**}**

**}**

Вывод

В ходе данной лабораторной работы были изучены особенности организации коллекций объектов на языке Java, приобретены практические навыки использования списков, очередей, хеш-таблиц при создании Java программ.