

Netraprakash

240173012:L

26/10/2023

Assignment - 4

import java.util.*;
import java.awt.*;

interface Show {
 void show();

abstract class Item implements Show {

int id; String title;
 Item (int id, String title) { this.id = id; this.title = title; }
}

class Book extends Item {

String author;

Book (int id, String n, String c) { super(id, c), this.id = id, a,
 author = a; Cat = c; }

void issue () { issued = true; }

public void print () { System.out.println ("id: " + id + "
 " + title + " " + author + " " + issued); }

}

class Member implements Show {

int mid; String name, email; List<String> list = new
 ArrayList<String>();

Member (int id, String n, String c) { mid = id; name = n;
 Email = c; }

void add (int id) { return (int id); }

public void value () { System.out.println ("mid: " + mid + "
 " + name + " " + Email + " " + list); }

3

class BookErr extends Exception {

BookErr (String m) { super(m); }

3

Ch 11

```
Map< Integer, Book > bmap = new Hash Map<>();
```

```
Map< Integer, Member > mmap = new Has Map<>();
```

```
int bc = 100, mc = 200;
```

```
bib() { load(); auto(); }
```

```
void addBook(String t, String a, String c) {
```

```
Book b = new Book(++bc, t, a, c);
```

```
bmap.put(b.id, b);
```

```
System.out.println("Book ID: " + b.id);
```

```
}
```

```
void addMem(String n, String e) {
```

```
Member m = new Member(++mc, n, e);
```

```
mmap.put(m.mid, m);
```

```
System.out.println("Member ID: " + m.mid);
```

```
}
```

```
void issue (int bid, int mid) throws Book Err {
```

```
if (!bmap.containsKey(bid) || !mmap.containsKey(mid)) return;
```

```
Book b = bmap.get(bid);
```

```
if (b.issued) throw new Book Err("Issued");
```

```
b.issue();
```

```
mmap.get(mid).add(bid);
```

```
System.out.println("Issued");
```

```
}
```

```
void ret (int bid, int mid) {
```

```
if (!bmap.containsKey(bid) || !mmap.containsKey(mid)) return;
```

```
bmap.get(bid).ret();
```

```
mmap.get(mid).rem(bid);
```

```
System.out.println("Returned");
```

```
}
```

```
void fetch (String k) {
```

```
    / /  
bmap.values().stream().filter(b -> b.title.contains(k)  
    || b.auth.contains(k) || b.cat.contains(k)).forEach  
(Book b: show);  
}  
  
void sort(){  
    bmap.values().stream().sorted(Comparator.compari-  
        ng(b -> b.title)).forEach(Book b: show);  
}  
  
void save(){  
    try(BufferedWriter w = new BufferedWriterWriter(new File  
        Writer("books.txt"))){  
        for(Book b: bmap.values()) w.write(b.id + ", " + b.  
            title + ", " + b.auth + ", " + b.cat + ", " + b.isred  
            + "\n");  
    } catch(NoSuchElementException e){  
        try(BufferedWriter w = new BufferedWriterWriter(new  
            FileWriter("members.txt"))){  
            for(Member m: maph.values()) w.write(m.mid + "  
            " + m.name + " " + m.email + " " + m.list + "\n");  
        } catch(NoSuchElementException e){  
    }  
  
    void load(){  
        try(BufferedReader r = new BufferedReader(new FileReader  
            ("books.txt"))){  
            String s; while((s = r.readLine()) != null){  
                String p[] = s.split(", ");  
                Book b = new Book(Integer.parseInt(p[0]), p[1], p[2],  
                    p[3]);  
                b.isred = Boolean.parseBoolean(p[4]);  
                bmap.put(b.id, b); b = Math.max(b, b.id);  
            }  
        } catch(NoSuchElementException e){  
    }  
}
```

```
try(Buffered Reader r = new BufferedReader(new File  
Reader("members.txt"))){  
    String s; while((s=r.readLine())!=null){  
        String p[] = s.split(",");  
        Member m = new Member(Integer.parseInt(p[0]), p[1],  
            p[2]);  
        mmap.put(m.mid,m); mc=Math.max(mc,m.mid);  
    }  
} catch (Exception e) {}  
}
```

```
void run(){
```

```
    Thread t=new Thread(()->{try {while(true){run();  
        Thread.sleep(3000);}}catch (Exception e){};});  
    t.setDaemon(true); t.start();
```

```
}
```

```
}
```

```
public class LibrarySystem {
```

```
    public static void main (String []a){
```

```
        lib l = new lib();
```

```
        Scanner s = new Scanner (System.in);
```

```
        while(true){
```

```
            System.out.println("1 Add Book 2 Add Mem 3 Show 4 Remove  
                5 Search 6 Sort 7 Exit");
```

```
            try{
```

```
                int c = s.nextInt();
```

```
                switch(c){
```

```
                    case 1 -> {
```

```
                        s.nextLine();
```

```
                        Sout ("Title:"); String t = s.nextLine();
```

```
                        Sout ("Author:"); String au = s.nextLine();
```

```
                        Sout ("Cat:"); String cat = s.nextLine();
```

```
                        l.addBook(t, au, cat);
```

```
}
```

case 2 -> {

s.nextLine();

Send("Name:"); String n = s.nextLine();

Send("Email:"); String e = s.nextLine();

l.addMem(n, e);

}

case 3 -> {

System.out.print("Bid:"); int bid = s.nextInt();

System.out.print("Mid:"); int mid = s.nextInt();

l.insert(bid, mid);

}

case 4 -> {

Send("Bid:"); int bid = s.nextInt();

Send("Mid:"); int mid = s.nextInt();

l.read(bid, mid);

}

case 5 -> {

s.nextLine();

Send("Key:"); l.search(s.nextLine());

}

case 6 -> l.read();

case 7 -> { l.save(); return; }

}

} catch (BookException e) { System.out.print(e.getMessage()); }

Catch (Exception e) { System.out.println("Err"); s.

nextLine(); }

}

}

}