

Netrapakshi

240173012:L

26/10/2019

### Assignment - 4

```
import java.io.*;  
import java.util.*;
```

```
interface ShowFunctions();
```

```
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, n, c); this.id = id, a,  
        author = a; Cat = c; }
```

```
    void issue() { issued = true; }
```

```
    public void display() { System.out.println(id + "/" + title +
```

```
        "/" + author + "/" + issued + "/" + Cat); }
```

```
}
```

```
class Member implements Show {
```

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

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

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

```
    public void display() { System.out.println(mid + "/" +  
        name + "/" + Email + "/" + list); }
```

```
}
```

```
class BookError extends Exception {
```

```
    BookError(String m) { super(m); }
```

```
}
```

classlib {

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

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

int bc = 100, mc = 200;

lib() { 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 BookErr {

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

Book b = bmap.get(bid);

if (b.issued) throw new BookErr("Issued");

b.issue();

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

System.out.println("Done");

}

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 slash(String k) {



\_/\_/\_

```

bmap.values().stream().filter(b -> b.title.contains(k)
    || b.auth.contains(k) || b.cat.contains(k)).forEach
(Book::show);
}

```

```

void sort() {

```

```

    bmap.values().stream().sorted(Comparator.comparing
        (b -> b.title)).forEach(Book::show);
}

```

```

void save() {

```

```

    try (BufferedWriter w = new BufferedWriter(new File
        Writer("books.txt"))) {

```

```

        for (Book b : bmap.values()) w.write(b.id + ", " + b.
            title + ", " + b.auth + ", " + b.cat + ", " + b.issued
            + "\n");

```

```

    } catch (Exception e) {}

```

```

    try (BufferedReader w = new BufferedReader(new
        FileReader("members.txt"))) {

```

```

        for (Member m : mmap.values()) w.write(m.mid + ", "
            + m.name + ", " + m.email + ", " + m.list + "\n");

```

```

    } catch (Exception e) {}
}

```

```

void load() {

```

```

    try (BufferedReader r = new BufferedReader(new File
        Reader("books.txt"))) {

```

```

        Strings: while ((s = r.readLine()) != null) {

```

```

            String p[] = s.split(",");

```

```

            Book b = new Book(Integer.parseInt(p[0]), p[1], p[2],
                p[3]);

```

```

            b.issued = Boolean.parseBoolean(p[4]);

```

```

            bmap.put(b.id, b); b = Math.max(b, b.id);

```

```

        }
    } catch (Exception e) {}
}

```

```

try(BufferedReader 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); m.c = Math.max(m.c, m.mid);
    }
} catch (Exception e) {}
}

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

```

```

public class Library System {
    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 Issue 4 Return
            5 Search 6 Sort 7 Exit");
            try {
                int c = s.nextInt();
                switch(c) {
                    case 1 -> {
                        s.nextLine();
                        Sent ("Title:"); String t = s.nextLine();
                        Sent ("Auth:"); String au = s.nextLine();
                        Sent ("Cat:"); String cl = s.nextLine();
                        l.add Book(t, au, cl);
                    }
                }
            }
        }
    }
}

```



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();

Issue(bid, mid);

}

case 4 -> {

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

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

l.ret(bid, mid);

}

case 5 -> {

s.nextLine();

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

}

case 6 -> l.save();

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

}

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

catch (Exception e) { System.out.printLn("Err"); s.nextLine(); }

}

}

}