

DA.

Name - Mainpal

Subject - Java Programming

Assignment - 04

Roll No - 2401730097

\_/\_/\_

```

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.comparing(
        b -> b.title)).forEach((Book b) : 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 (BufferedWriter w = new BufferedWriter(new
        File Writer("members.txt"))) {
        for (Member m : map.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"))) {
        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.issued = Boolean.parseBoolean(p[4]);
            bmap.put(b.id, b); b = Math.max(b, bmap.get(b.id));
        }
    } catch (Exception e) {}
}

```



case 2 -> {

s.nextLine();  
Scrd("Name:"); String n = s.nextLine();  
Scrd("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();  
T.issue(bid, mid);

}

case 4 -> {

Scrd("Bid:"); int bid = s.nextInt();  
Scrd("Mid:"); int mid = s.nextInt();  
l.set(bid, mid);

}

case 5 -> {

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

}

case 6 -> l.sort();

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

}

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

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

}

}

}

Assignment - 4

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

```
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 auth, cat; boolean issued;
    Book(int id, String t, String a, String c) { super(id, t);
        auth = a; cat = c; }
    void issue() { issued = true; }
    void ret() { issued = false; }
    public void show() { System.out.println(id + "|" + title +
        "|" + auth + "|" + cat + "|" + issued); }
}
```

```
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) { list.add(id); }
    void rem(int id) { list.remove(Integer.valueOf(id)); }
    public void show() { System.out.println(mid + "|" +
        name + "|" + email + "|" + list); }
}
```

```
class BookErr extends Exception {
    BookErr(String m) { super(m); }
}
```



```

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 = Math.max(m, 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 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 Issue 4 Return
            5 Search 6 Sort 7 Exit");

```

```

try {
    int c = s.nextInt();
    switch(c) {
        case 1 -> {
            s.nextLine();
            Sout("Title:"); String t = s.nextLine();
            Sout("Auth:"); String au = s.nextLine();
            Sout("Cat:"); String cl = s.nextLine();
            l.addBook(t, au, cl);
        }
    }
}

```



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) throws 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 search(String k) {