Gebze Technical University
Computer Engineering

CSE 222
2017 Spring

HOMEWORK 01 REPORT

GULZADA IISAEVA
131044085

- Ödevde istenildiği üzere bir kütüphane sistemi tasarlandı. Kullanacak olduğumuz interface ve abstract class'ların yapıları tasarlandı. Tasarım aşamasında gerekli requirementlar belirlenerek classlar arasındaki hiyerarşi kuruldu.
- Proje Moodle'da verildiği CSE222-VM İntellijde yapıldı

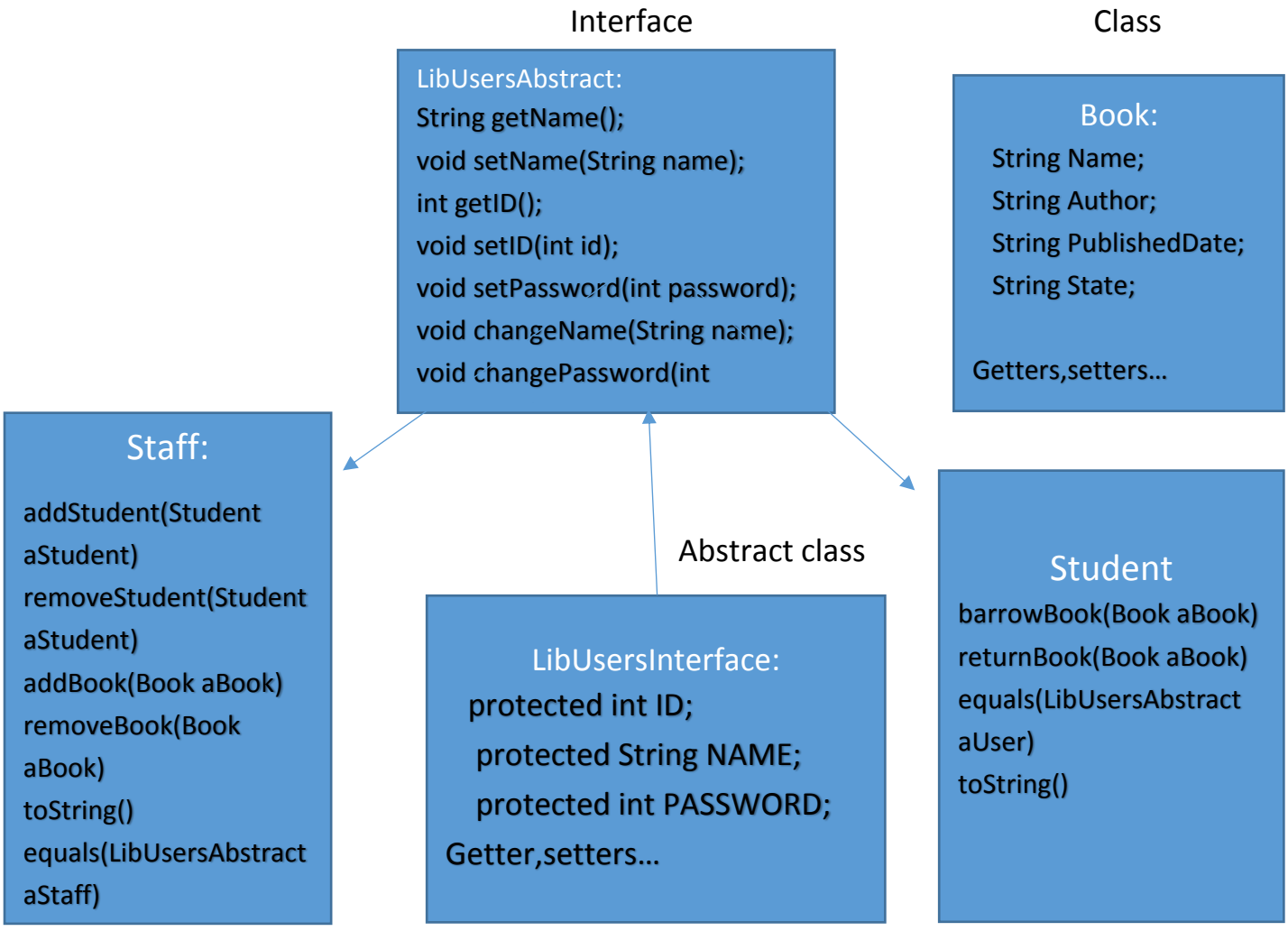
- Object Oriented'ın temel prensiplerine ve clean code standartlarına uyulmaya dikkat edildi.

- Grup dataları bir arada tutmak için sadece **ArrayList** yapısı kullanılmıştır.

- **Javadoc** kullanılarak class içerisinde yer alan methodların implementasyonları gerçekleştirilmiştir.

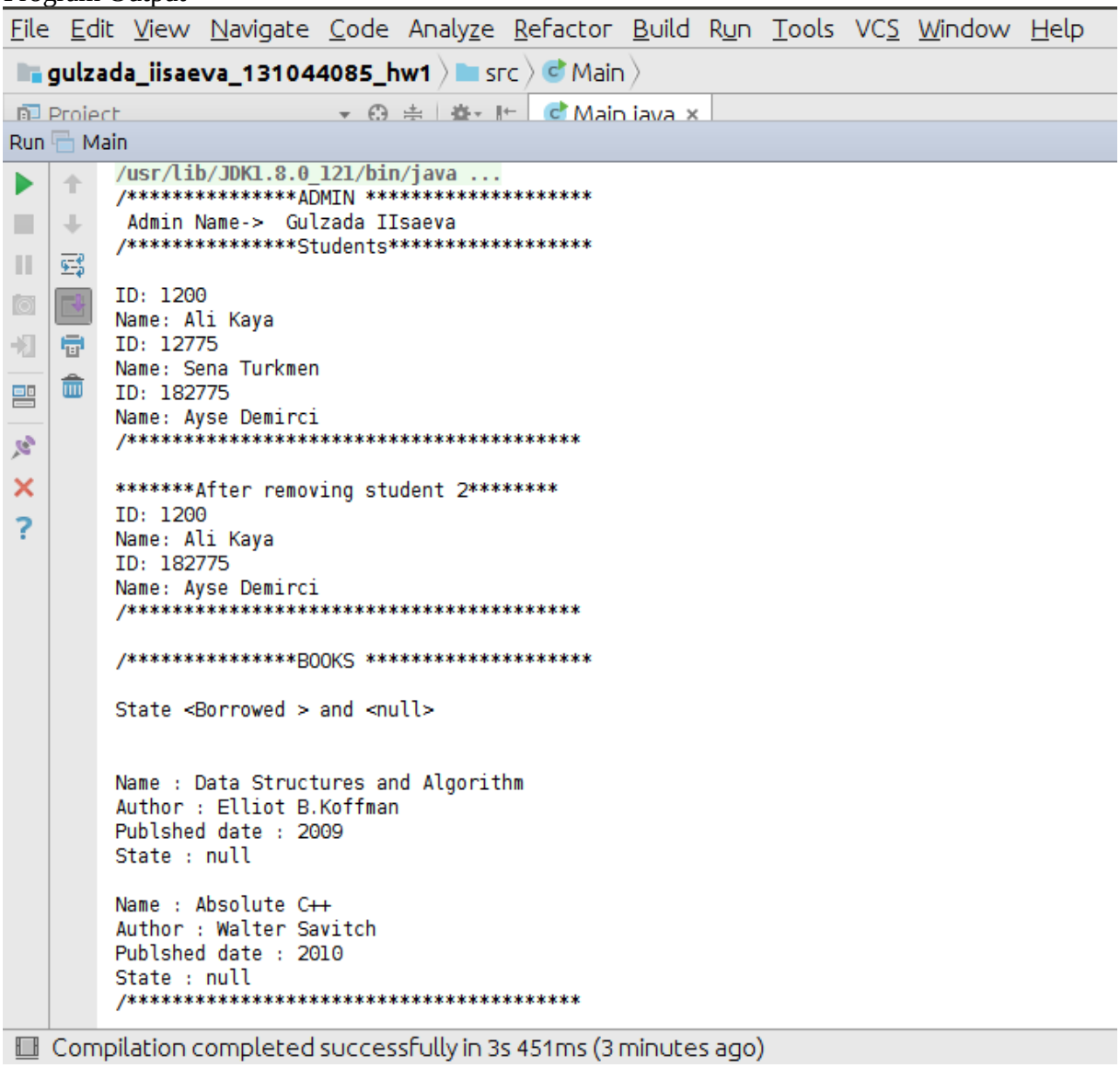
- **Exception Handling** gerekli methodlarda uygulanmıştır.

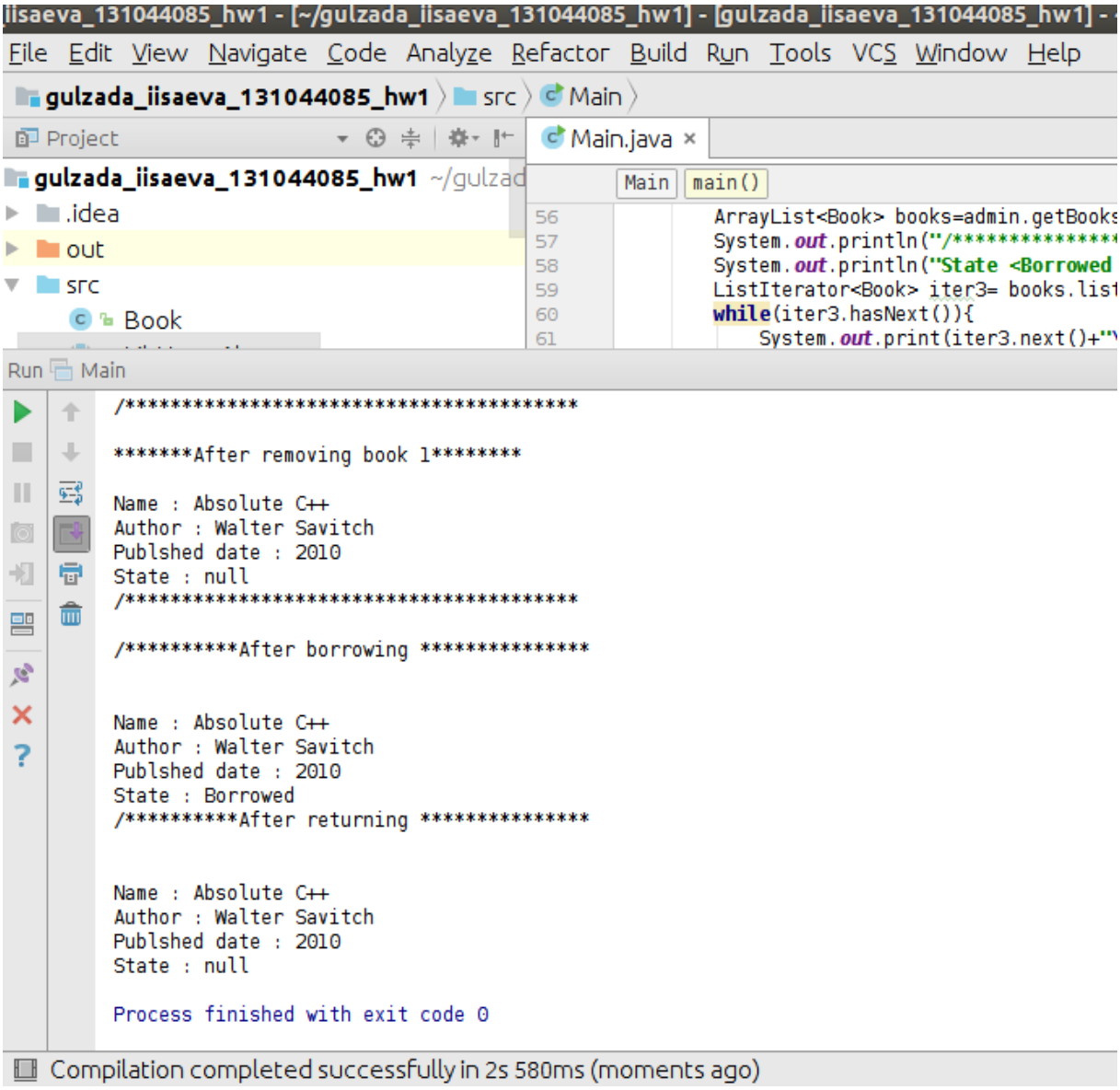
Classes diagram:



- Tüm kullanıcılar Users **interface**'inden implement olmuşlardır. Bu bize polymorphism ve inheritance sağlamıştır. Ayrı olarak Book classı var.
- Kütüphanenin yönetmeni (staff) ve kullanıcıları vardır. Staff ve kullanıcıların sisteme giriş için id ve şifresi vardir. Yönetici hem kullanıcıları (students) hem kitapları ekleyebilir ve silebilir. Kullanıcılar kitap ödünç alabilir ve geri verebilir.
- Ödevde istenildiği gibi csv dosyasına kayıt etme kısmını zamanımın kısıtlı olmasından dolayı yetiştiremedim.
- GitHub Link https://github.com/gulzadaiisaeva/CSE222_HW1
- İntellij de Main.java ana class seçilerek çalıştırlır.

Program Output





Q2) Encapsulation is possible in non-object-oriented languages. In C, for example, a structure can be declared in the public API (i.e., the header file) for a set of functions that operate on an item of data containing data members that are not accessible to clients of the API:

```
// Header file "api.h"

struct Entity;          // Opaque structure with hidden members

// API functions that operate on 'Entity' objects
extern struct Entity *  open_entity(int id);
extern int              process_entity(struct Entity *info);
extern void             close_entity(struct Entity *info);
```

Clients call the API functions to allocate, operate on, and deallocate objects of an opaque data type. The contents of this type are known and accessible only to the implementation of the API functions; clients cannot directly access its contents. The source code for these functions defines the actual contents of the structure:

```
// Implementation file "api.c"

#include "api.h"

// Complete definition of the 'Entity' object
struct Entity {
    int     ent_id;         // ID number
    char    ent_name[20];   // Name
    ... and other members ...
};

// API function implementations
struct Entity * open_entity(int id)
{ ... }

int process_entity(struct Entity *info)
{ ... }

void close_entity(struct Entity *info)
{ ... }
```