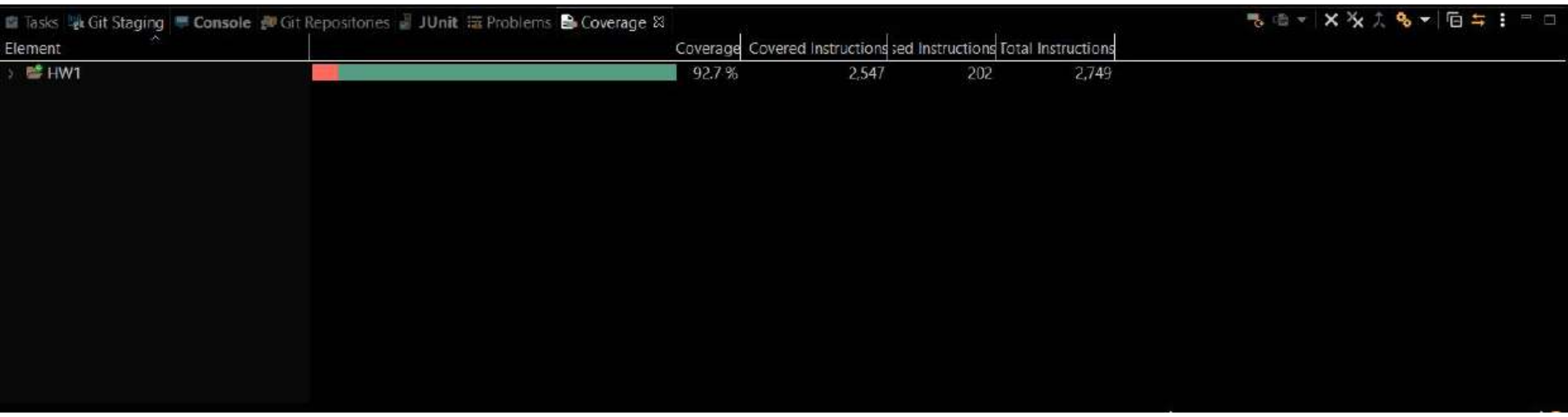
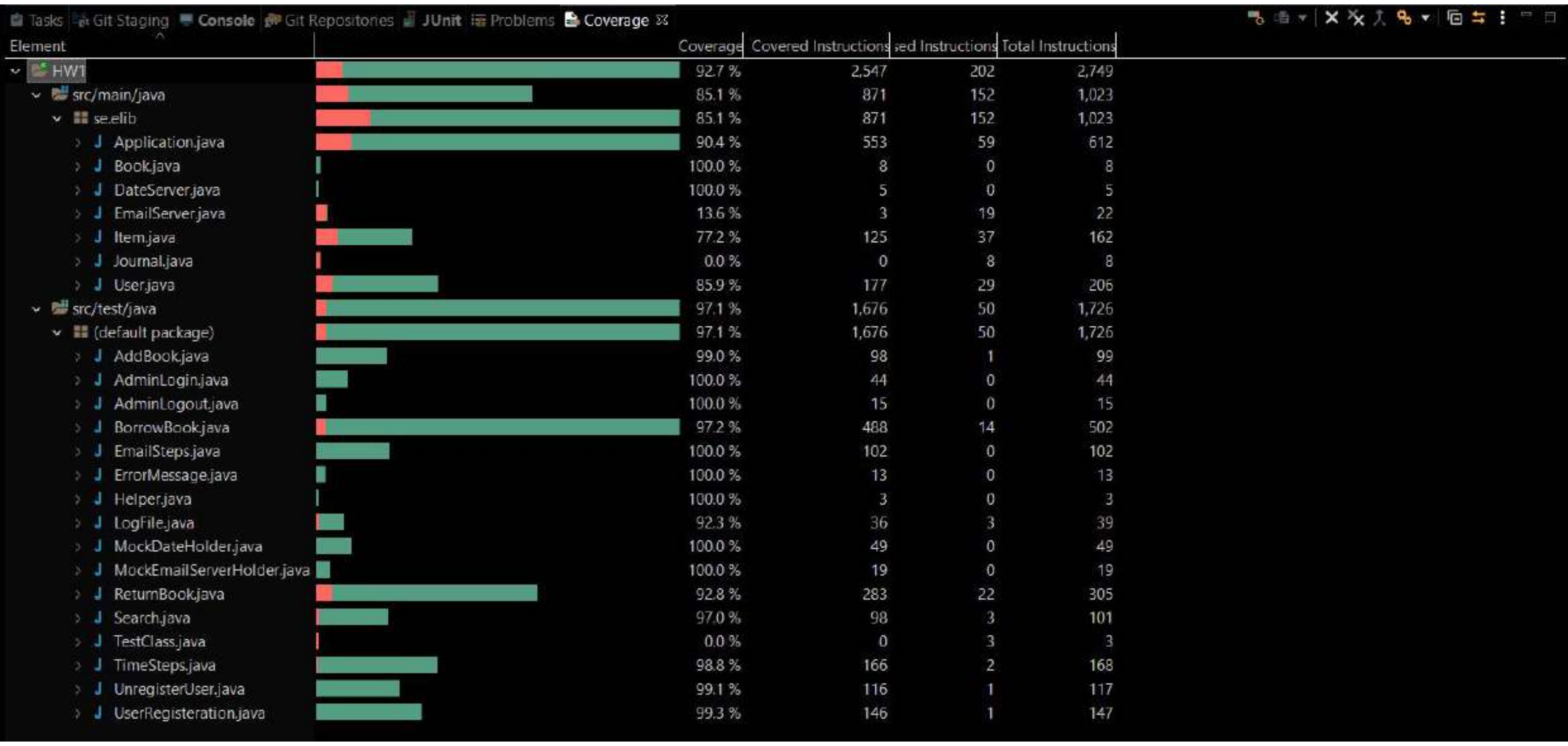
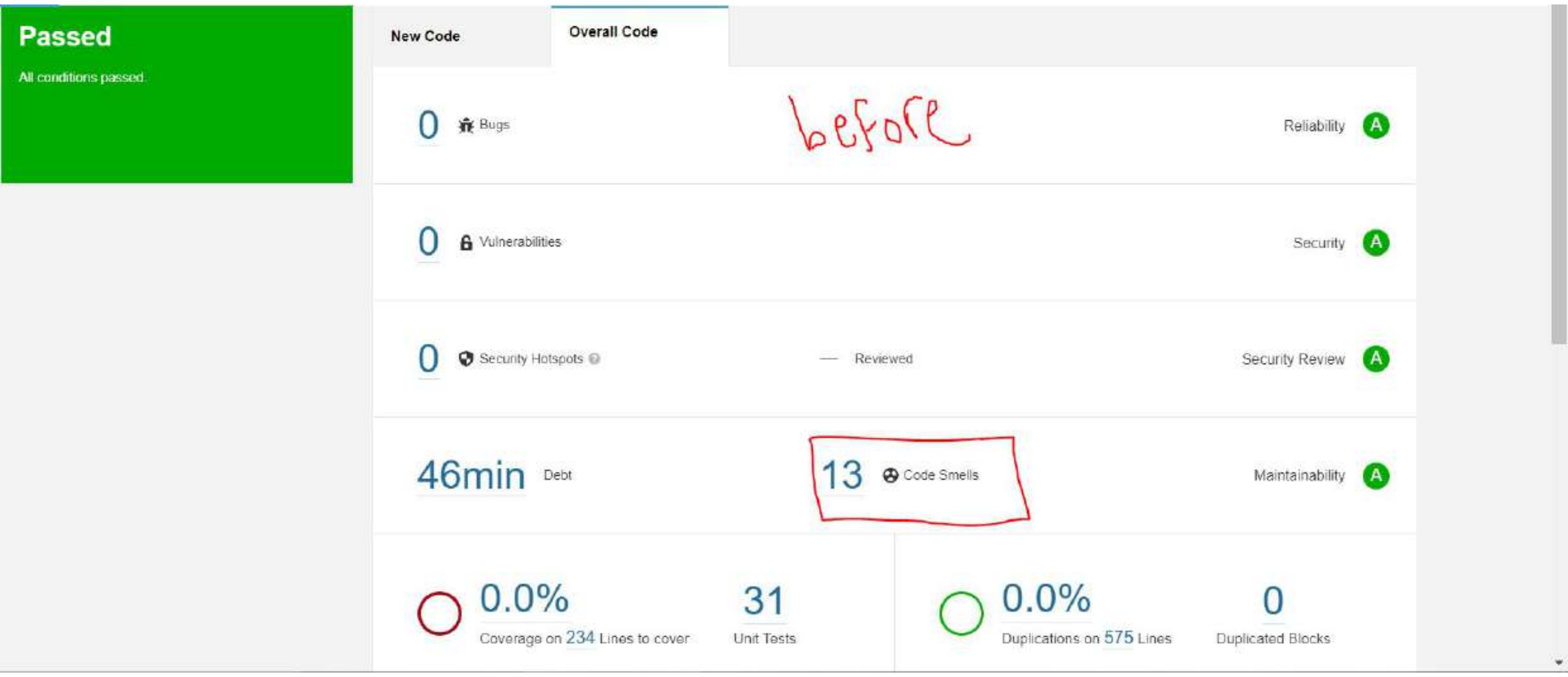


| Element | | Coverage | Covered Instructions | sed Instructions | Total Instructions |
|---|---|---|---|---|---|
| HW1 | | 92.7 % | 2,547 | 202 | 2,749 |



| Element | Coverage | Covered Instructions | sed Instructions | Total Instructions |
|---|---|---|---|---|
| HW1 | 92.7 % | 2,547 | 202 | 2,749 |
| src/main/java | 85.1 % | 871 | 152 | 1,023 |
| se.elib | 85.1 % | 871 | 152 | 1,023 |
| Application.java | 90.4 % | 553 | 59 | 612 |
| Book.java | 100.0 % | 8 | 0 | 8 |
| DateServer.java | 100.0 % | 5 | 0 | 5 |
| EmailServer.java | 13.6 % | 3 | 19 | 22 |
| Item.java | 77.2 % | 125 | 37 | 162 |
| Journal.java | 0.0 % | 0 | 8 | 8 |
| User.java | 85.9 % | 177 | 29 | 206 |
| src/test/java | 97.1 % | 1,676 | 50 | 1,726 |
| (default package) | 97.1 % | 1,676 | 50 | 1,726 |
| AddBook.java | 99.0 % | 98 | 1 | 99 |
| AdminLogin.java | 100.0 % | 44 | 0 | 44 |
| AdminLogout.java | 100.0 % | 15 | 0 | 15 |
| BorrowBook.java | 97.2 % | 488 | 14 | 502 |
| EmailSteps.java | 100.0 % | 102 | 0 | 102 |
| ErrorMessage.java | 100.0 % | 13 | 0 | 13 |
| Helper.java | 100.0 % | 3 | 0 | 3 |
| LogFile.java | 92.3 % | 36 | 3 | 39 |
| MockDateHolder.java | 100.0 % | 49 | 0 | 49 |
| MockEmailServerHolder.java | 100.0 % | 19 | 0 | 19 |
| ReturnBook.java | 92.8 % | 283 | 22 | 305 |
| Search.java | 97.0 % | 98 | 3 | 101 |
| TestClass.java | 0.0 % | 0 | 3 | 3 |
| TimeSteps.java | 98.8 % | 166 | 2 | 168 |
| UnregisterUser.java | 99.1 % | 116 | 1 | 117 |
| UserRegisteration.java | 99.3 % | 146 | 1 | 147 |



**Passed**

All conditions passed.

New Code

Overall Code

0 Bugs

before

Reliability A

0 Vulnerabilities

Security A

0 Security Hotspots

— Reviewed

Security Review A

46min Debt

13 Code Smells

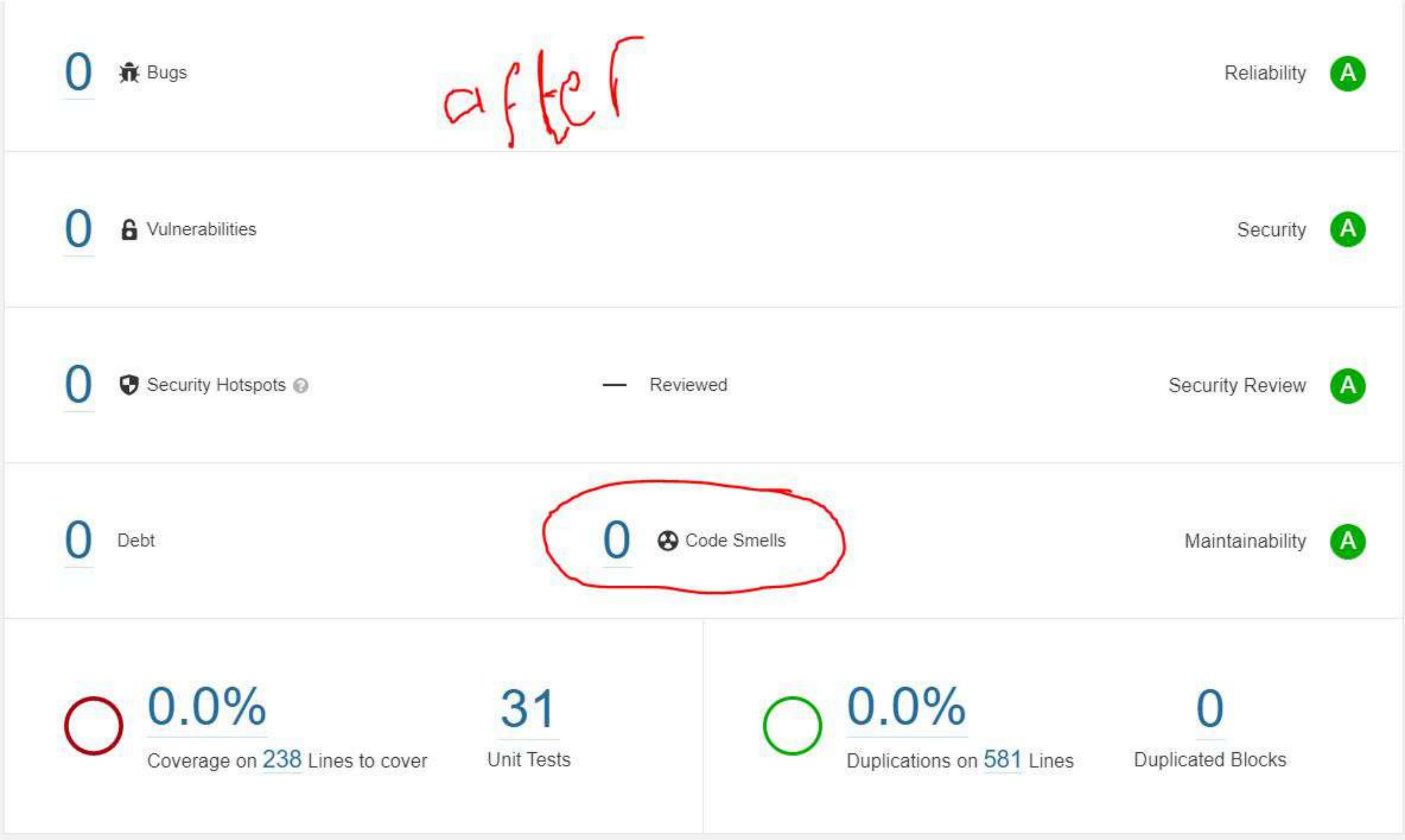
Maintainability A

0.0% Coverage on 234 Lines to cover

31 Unit Tests

0.0% Duplications on 575 Lines

0 Duplicated Blocks



after

0 Bugs Reliability A

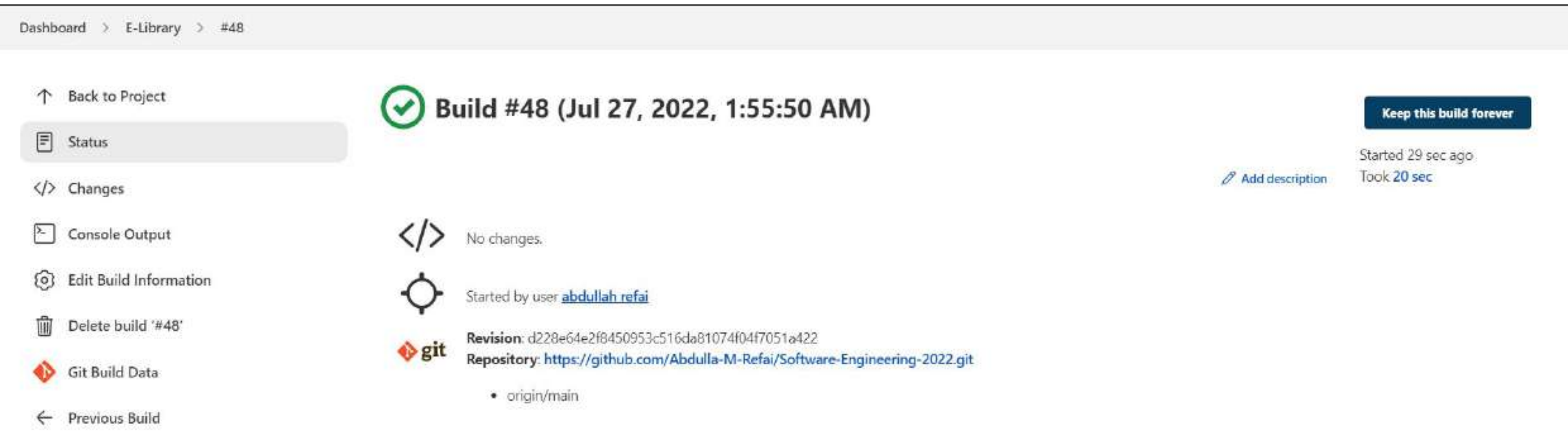
0 Vulnerabilities Security A

0 Security Hotspots — Reviewed Security Review A

0 Debt 0 Code Smells Maintainability A

0.0% Coverage on 238 Lines to cover 31 Unit Tests

0.0% Duplications on 581 Lines 0 Duplicated Blocks



Build #48 (Jul 27, 2022, 1:55:50 AM)

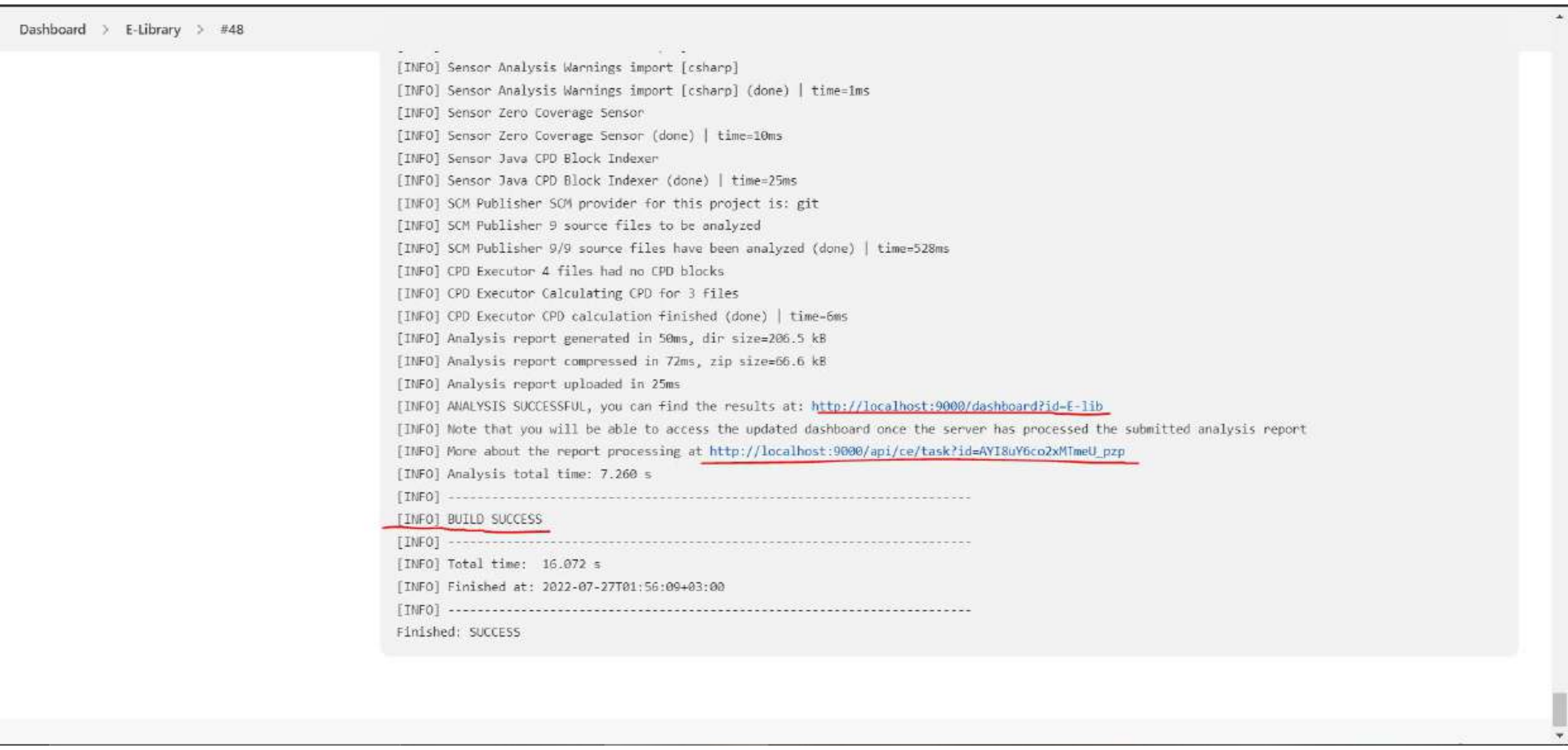
Started 29 sec ago
Took 20 sec

No changes.

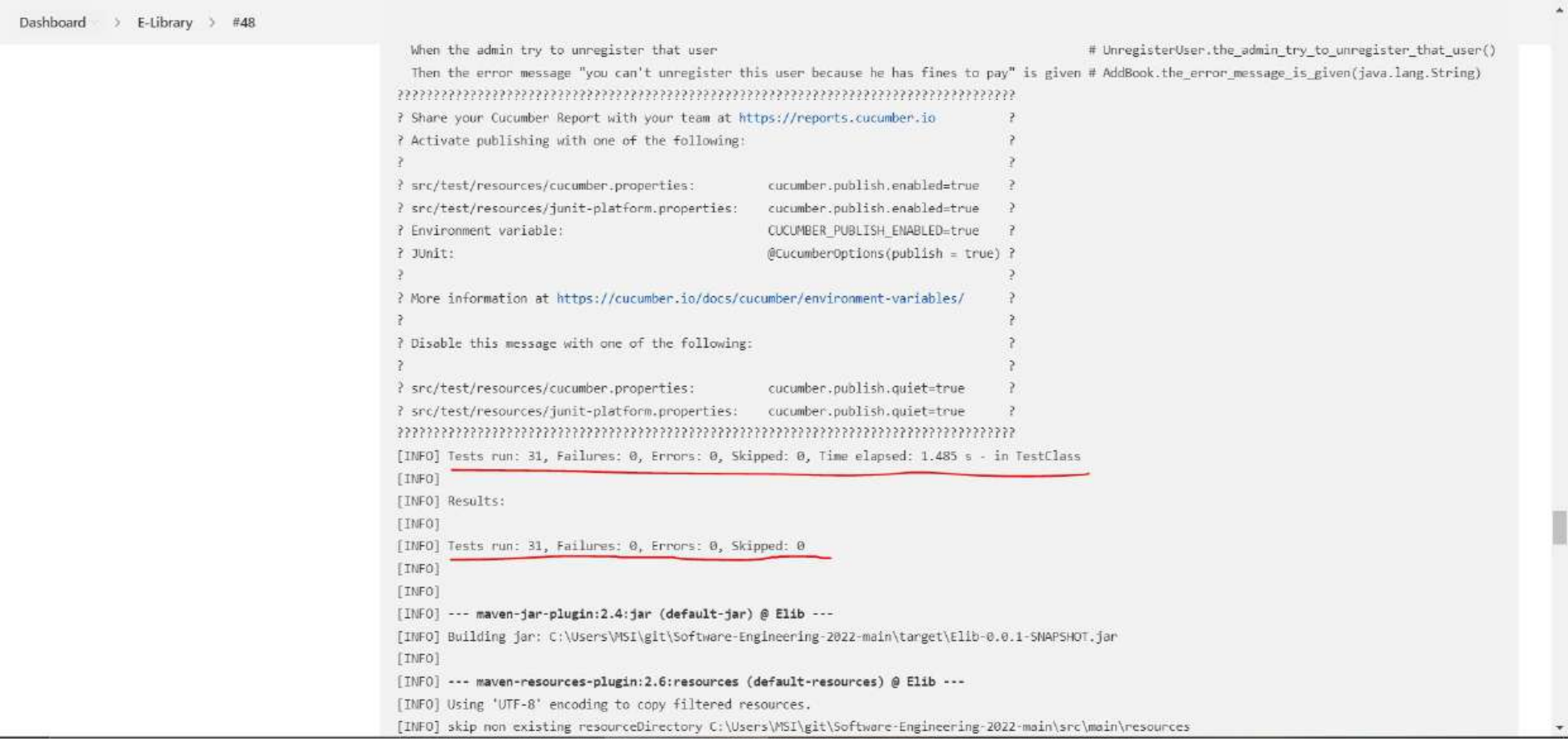
Started by user abdullah refai

**Revision**: d228e64e2f8450953c516da81074f04f7051a422
**Repository**: https://github.com/Abdulla-M-Refai/Software-Engineering-2022.git

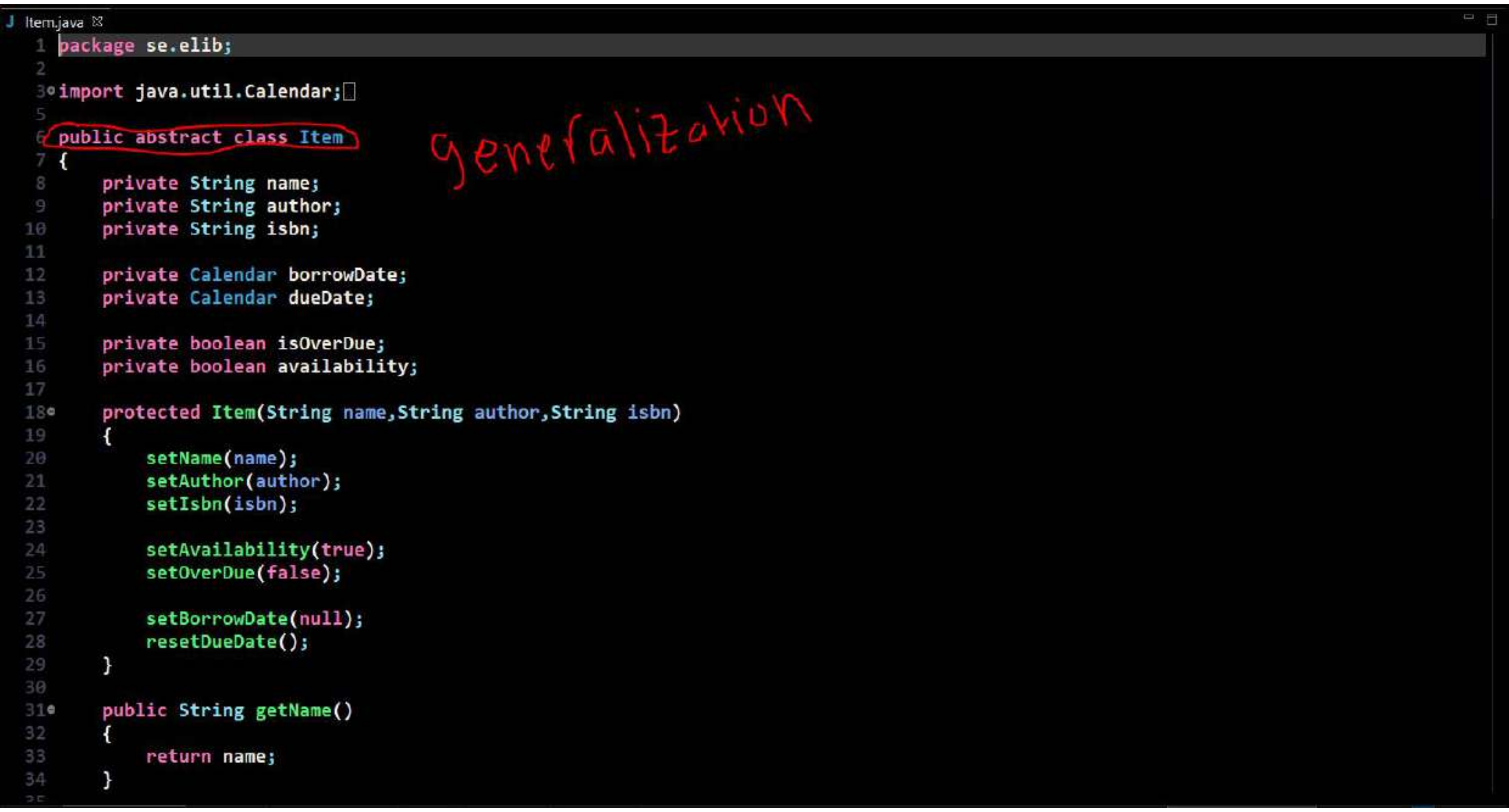
- origin/main



```
[INFO] Sensor Analysis Warnings import [csharp]
[INFO] Sensor Analysis Warnings import [csharp] (done) | time=1ms
[INFO] Sensor Zero Coverage Sensor
[INFO] Sensor Zero Coverage Sensor (done) | time=10ms
[INFO] Sensor Java CPD Block Indexer
[INFO] Sensor Java CPD Block Indexer (done) | time=25ms
[INFO] SCM Publisher SCM provider for this project is: git
[INFO] SCM Publisher 9 source files to be analyzed
[INFO] SCM Publisher 9/9 source files have been analyzed (done) | time=528ms
[INFO] CPD Executor 4 files had no CPD blocks
[INFO] CPD Executor Calculating CPD for 3 files
[INFO] CPD Executor CPD calculation finished (done) | time=6ms
[INFO] Analysis report generated in 50ms, dir size=206.5 kB
[INFO] Analysis report compressed in 72ms, zip size=66.6 kB
[INFO] Analysis report uploaded in 25ms
[INFO] ANALYSIS SUCCESSFUL, you can find the results at: http://localhost:9000/dashboard?id=E-lib
[INFO] Note that you will be able to access the updated dashboard once the server has processed the submitted analysis report
[INFO] More about the report processing at http://localhost:9000/api/ce/task?id=AYI8uY6co2xMTmeU_pzp
[INFO] Analysis total time: 7.260 s
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  16.072 s
[INFO] Finished at: 2022-07-27T01:56:09+03:00
[INFO] ------------------------------------------------------------------------
Finished: SUCCESS
```

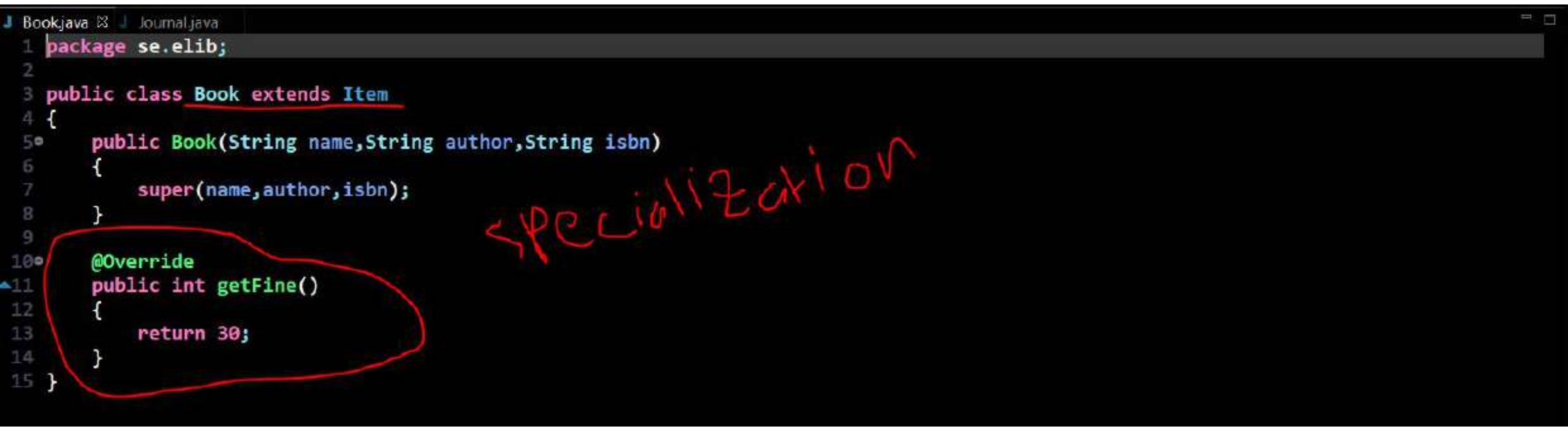


```
  When the admin try to unregister that user                                                # UnregisterUser.the_admin_try_to_unregister_that_user()
  Then the error message "you can't unregister this user because he has fines to pay" is given # AddBook.the_error_message_is_given(java.lang.String)
?????????????????????????????????????????????????????????????????????????????????????
? Share your Cucumber Report with your team at https://reports.cucumber.io          ?
? Activate publishing with one of the following:                                    ?
?                                                                                   ?
? src/test/resources/cucumber.properties:          cucumber.publish.enabled=true    ?
? src/test/resources/junit-platform.properties:    cucumber.publish.enabled=true    ?
? Environment variable:                            CUCUMBER_PUBLISH_ENABLED=true    ?
? JUnit:                                           @CucumberOptions(publish = true) ?
?                                                                                   ?
? More information at https://cucumber.io/docs/cucumber/environment-variables/      ?
?                                                                                   ?
? Disable this message with one of the following:                                   ?
?                                                                                   ?
? src/test/resources/cucumber.properties:          cucumber.publish.quiet=true      ?
? src/test/resources/junit-platform.properties:    cucumber.publish.quiet=true      ?
?????????????????????????????????????????????????????????????????????????????????????
[INFO] Tests run: 31, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 1.485 s - in TestClass
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 31, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO]
[INFO] --- maven-jar-plugin:2.4:jar (default-jar) @ Elib ---
[INFO] Building jar: C:\Users\MSI\git\Software-Engineering-2022-main\target\Elib-0.0.1-SNAPSHOT.jar
[INFO]
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ Elib ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory C:\Users\MSI\git\Software-Engineering-2022-main\src\main\resources
```



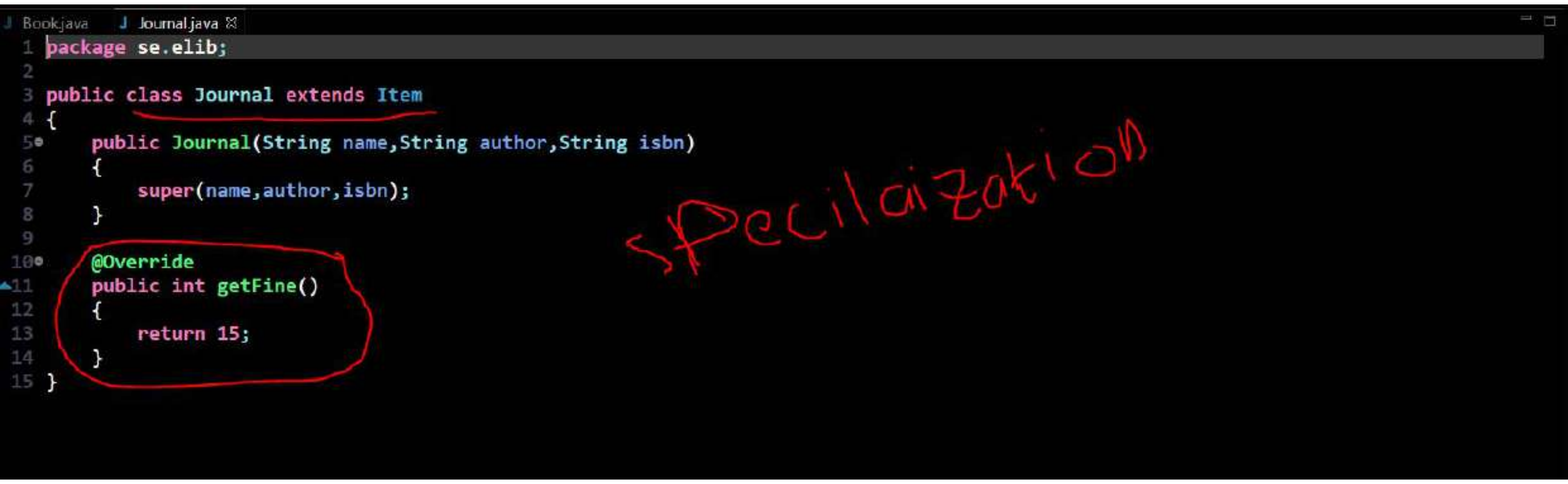
```java
package se.elib;

import java.util.Calendar;

public abstract class Item
{
    private String name;
    private String author;
    private String isbn;

    private Calendar borrowDate;
    private Calendar dueDate;

    private boolean isOverDue;
    private boolean availability;

    protected Item(String name,String author,String isbn)
    {
        setName(name);
        setAuthor(author);
        setIsbn(isbn);

        setAvailability(true);
        setOverDue(false);

        setBorrowDate(null);
        resetDueDate();
    }

    public String getName()
    {
        return name;
    }
```

generalization



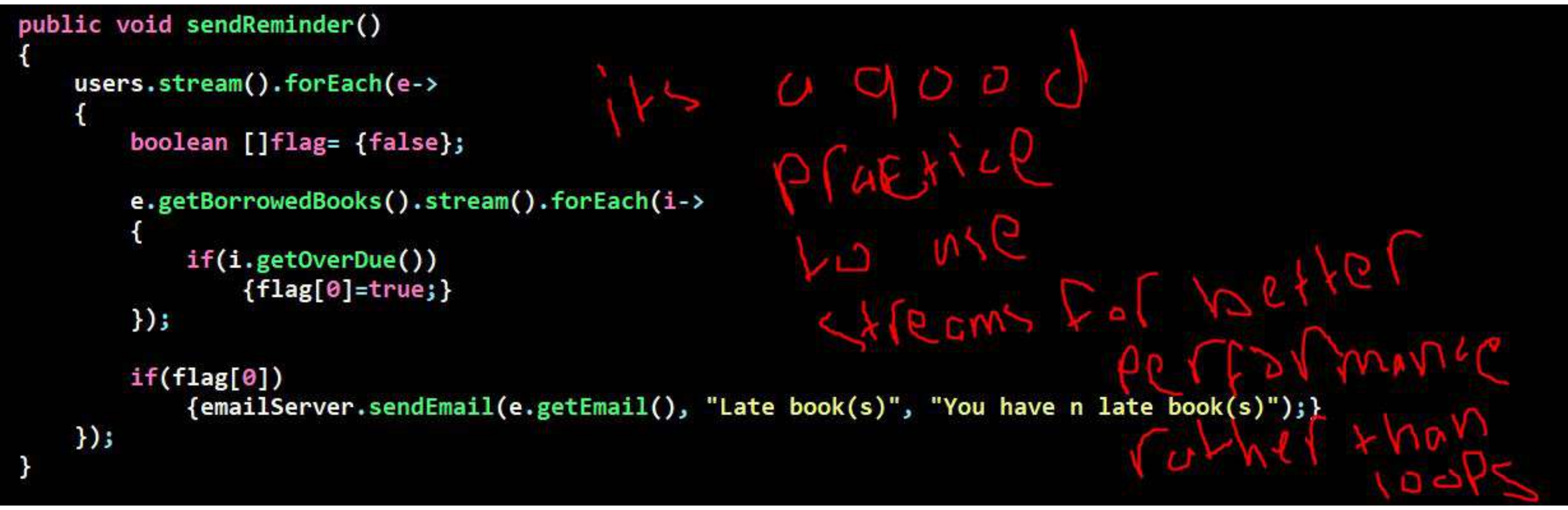
```java
package se.elib;

public class Book extends Item
{
    public Book(String name,String author,String isbn)
    {
        super(name,author,isbn);
    }

    @Override
    public int getFine()
    {
        return 30;
    }
}
```

specialization



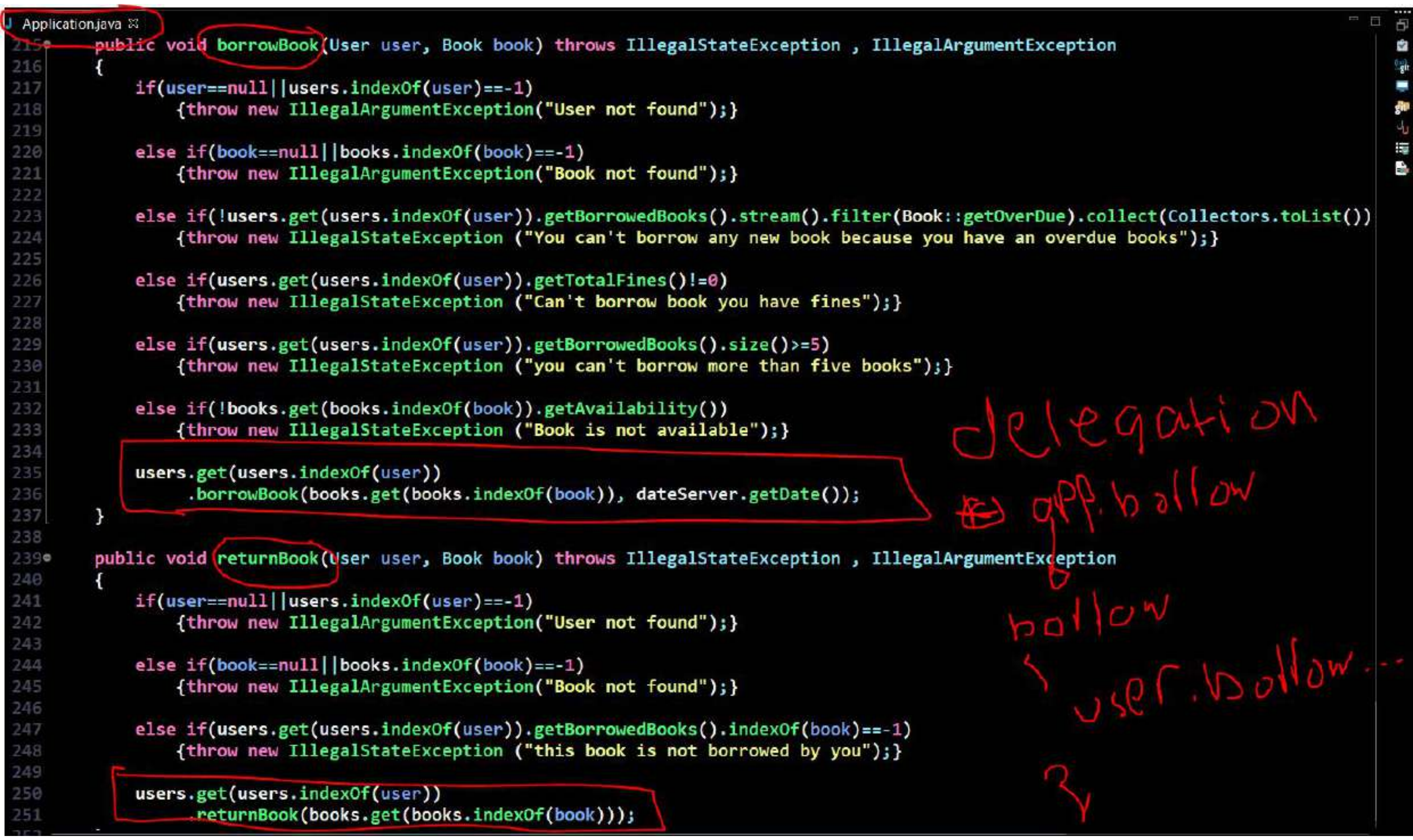
```java
package se.elib;

public class Journal extends Item
{
    public Journal(String name,String author,String isbn)
    {
        super(name,author,isbn);
    }

    @Override
    public int getFine()
    {
        return 15;
    }
}
```

sPecilaization



```
public void sendReminder()
{
    users.stream().forEach(e->
    {
        boolean []flag= {false};

        e.getBorrowedBooks().stream().forEach(i->
        {
            if(i.getOverDue())
                {flag[0]=true;}
        });

        if(flag[0])
            {emailServer.sendEmail(e.getEmail(), "Late book(s)", "You have n late book(s)");}
    });
}
```

its a good practice to use streams for better performance rather than loops



```java
    public void borrowBook(User user, Book book) throws IllegalStateException , IllegalArgumentException
    {
        if(user==null||users.indexOf(user)==-1)
            {throw new IllegalArgumentException("User not found");}

        else if(book==null||books.indexOf(book)==-1)
            {throw new IllegalArgumentException("Book not found");}

        else if(!users.get(users.indexOf(user)).getBorrowedBooks().stream().filter(Book::getOverDue).collect(Collectors.toList())
            {throw new IllegalStateException ("You can't borrow any new book because you have an overdue books");}

        else if(users.get(users.indexOf(user)).getTotalFines()!=0)
            {throw new IllegalStateException ("Can't borrow book you have fines");}

        else if(users.get(users.indexOf(user)).getBorrowedBooks().size()>=5)
            {throw new IllegalStateException ("you can't borrow more than five books");}

        else if(!books.get(books.indexOf(book)).getAvailability())
            {throw new IllegalStateException ("Book is not available");}

        users.get(users.indexOf(user))
            .borrowBook(books.get(books.indexOf(book)), dateServer.getDate());
    }

    public void returnBook(User user, Book book) throws IllegalStateException , IllegalArgumentException
    {
        if(user==null||users.indexOf(user)==-1)
            {throw new IllegalArgumentException("User not found");}

        else if(book==null||books.indexOf(book)==-1)
            {throw new IllegalArgumentException("Book not found");}

        else if(users.get(users.indexOf(user)).getBorrowedBooks().indexOf(book)==-1)
            {throw new IllegalStateException ("this book is not borrowed by you");}

        users.get(users.indexOf(user))
            .returnBook(books.get(books.indexOf(book)));
```

delegation

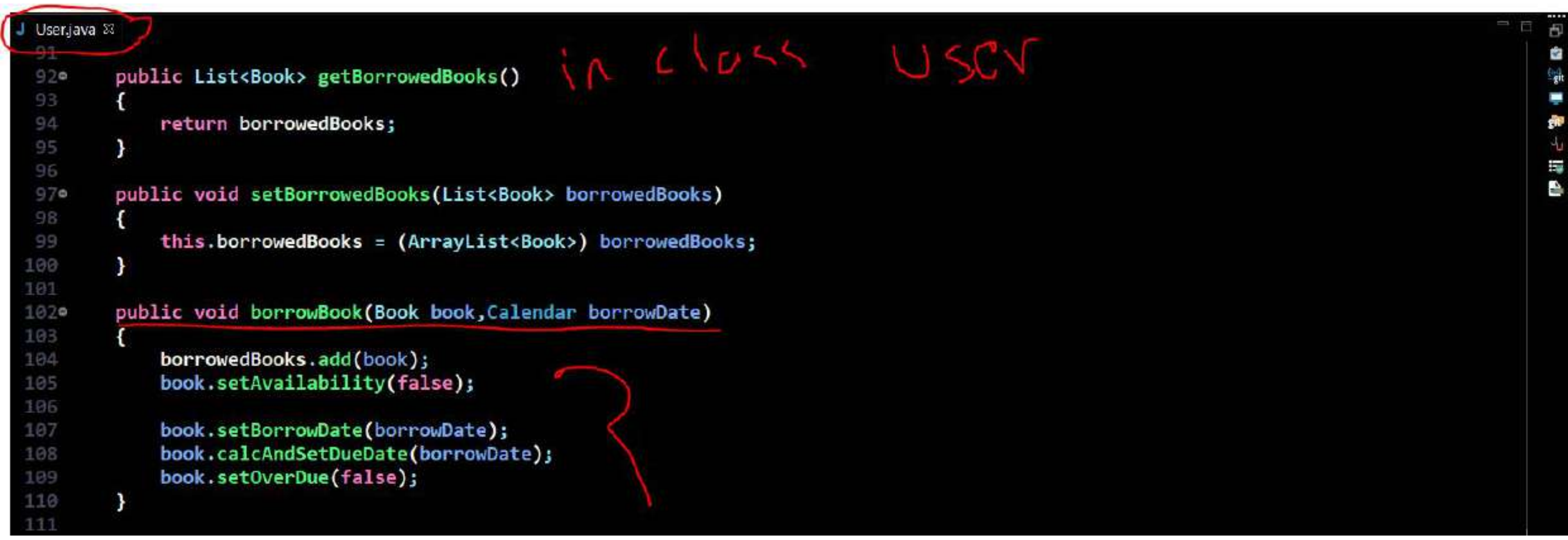
app.bollow

bollow

{

user.bollow...

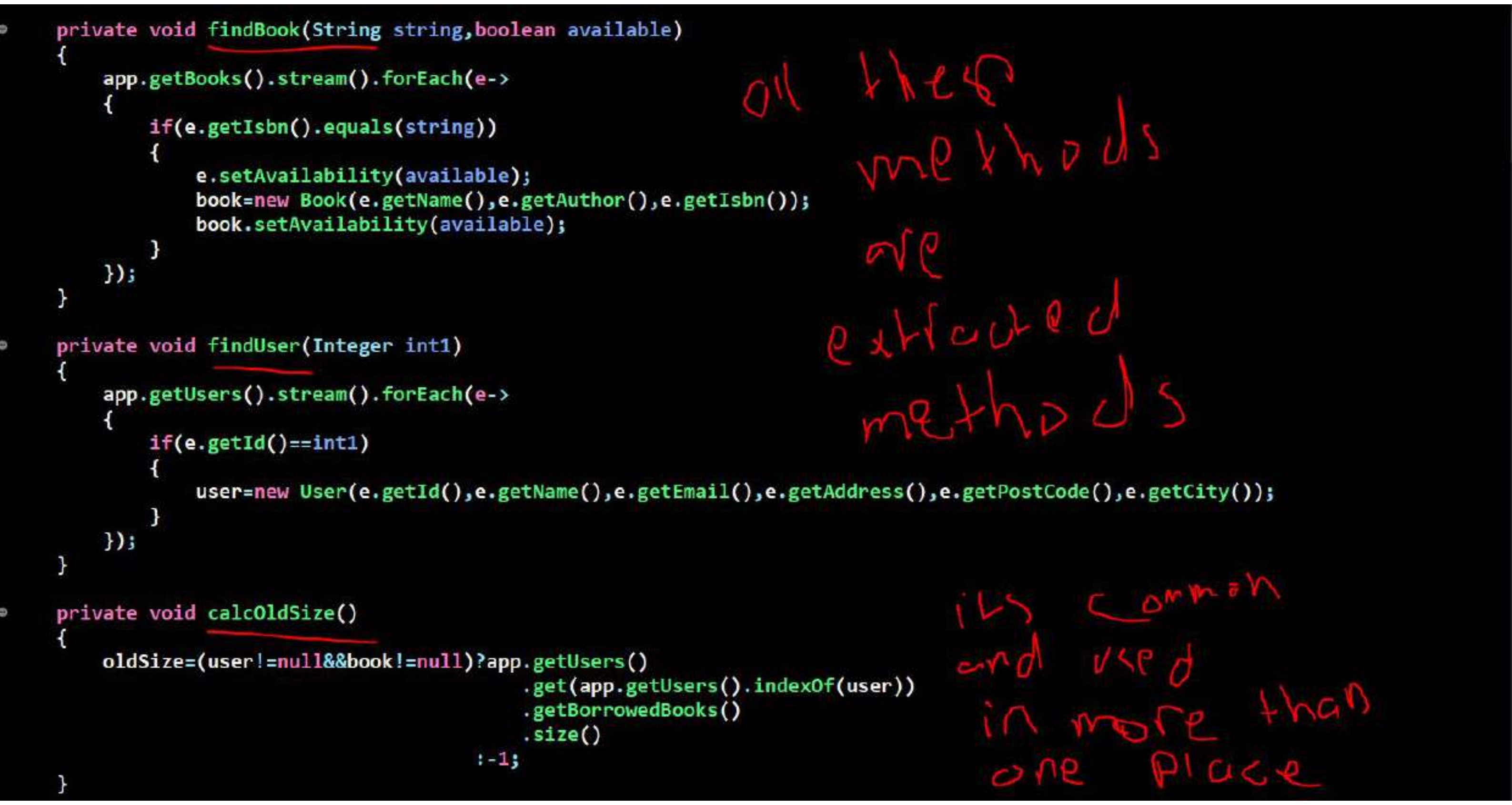
}



in class User

```java
    public List<Book> getBorrowedBooks()
    {
        return borrowedBooks;
    }

    public void setBorrowedBooks(List<Book> borrowedBooks)
    {
        this.borrowedBooks = (ArrayList<Book>) borrowedBooks;
    }

    public void borrowBook(Book book,Calendar borrowDate)
    {
        borrowedBooks.add(book);
        book.setAvailability(false);

        book.setBorrowDate(borrowDate);
        book.calcAndSetDueDate(borrowDate);
        book.setOverDue(false);
    }
```

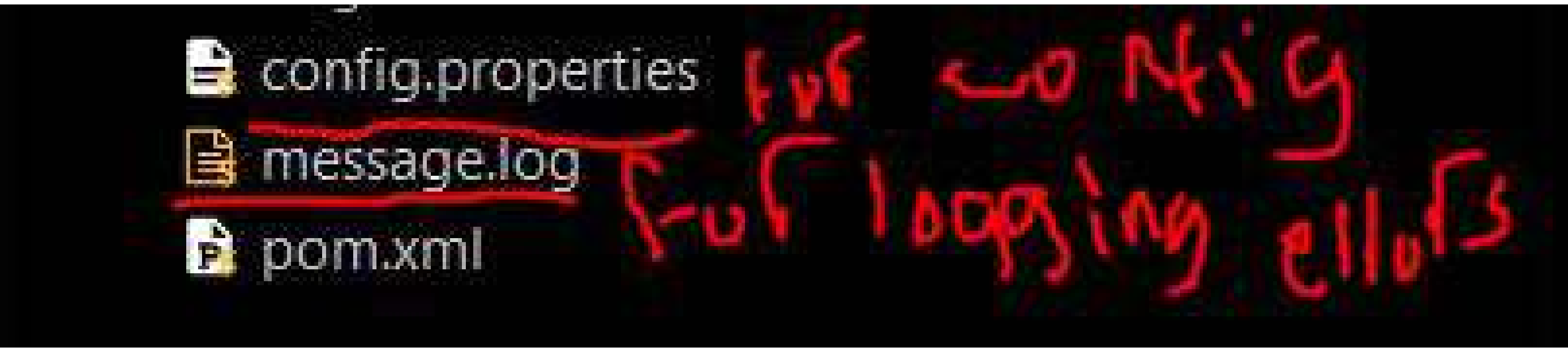
?



```
private void findBook(String string,boolean available)
{
    app.getBooks().stream().forEach(e->
    {
        if(e.getIsbn().equals(string))
        {
            e.setAvailability(available);
            book=new Book(e.getName(),e.getAuthor(),e.getIsbn());
            book.setAvailability(available);
        }
    });
}

private void findUser(Integer int1)
{
    app.getUsers().stream().forEach(e->
    {
        if(e.getId()==int1)
        {
            user=new User(e.getId(),e.getName(),e.getEmail(),e.getAddress(),e.getPostCode(),e.getCity());
        }
    });
}

private void calcOldSize()
{
    oldSize=(user!=null&&book!=null)?app.getUsers()
                                            .get(app.getUsers().indexOf(user))
                                            .getBorrowedBooks()
                                            .size()
                                        :-1;
}
```

all these methods are extracted methods

its common and used in more than one place



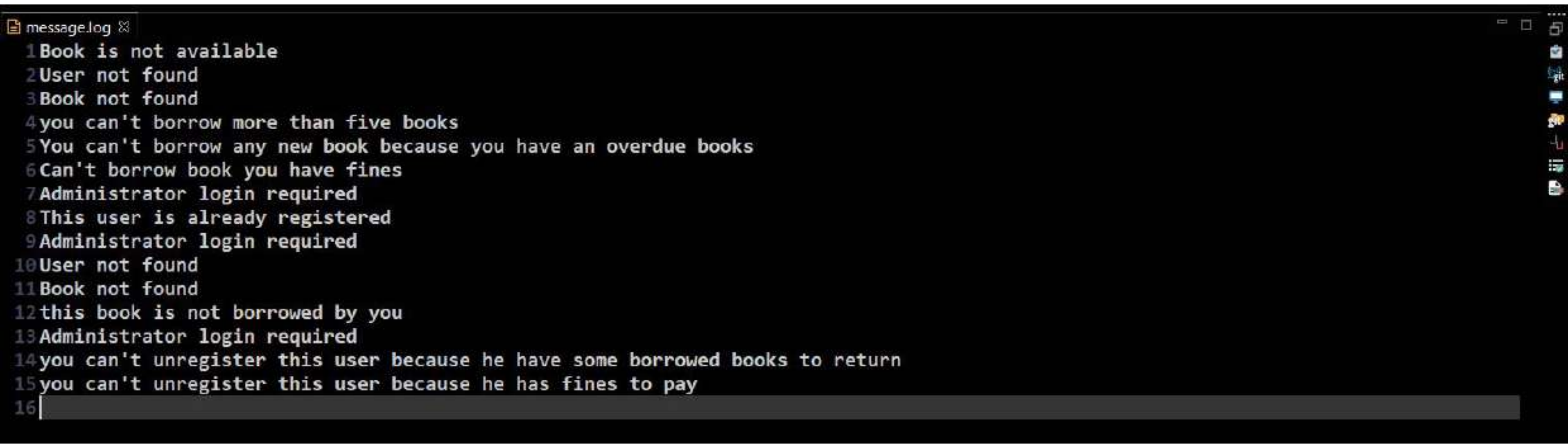
config.properties

message.log

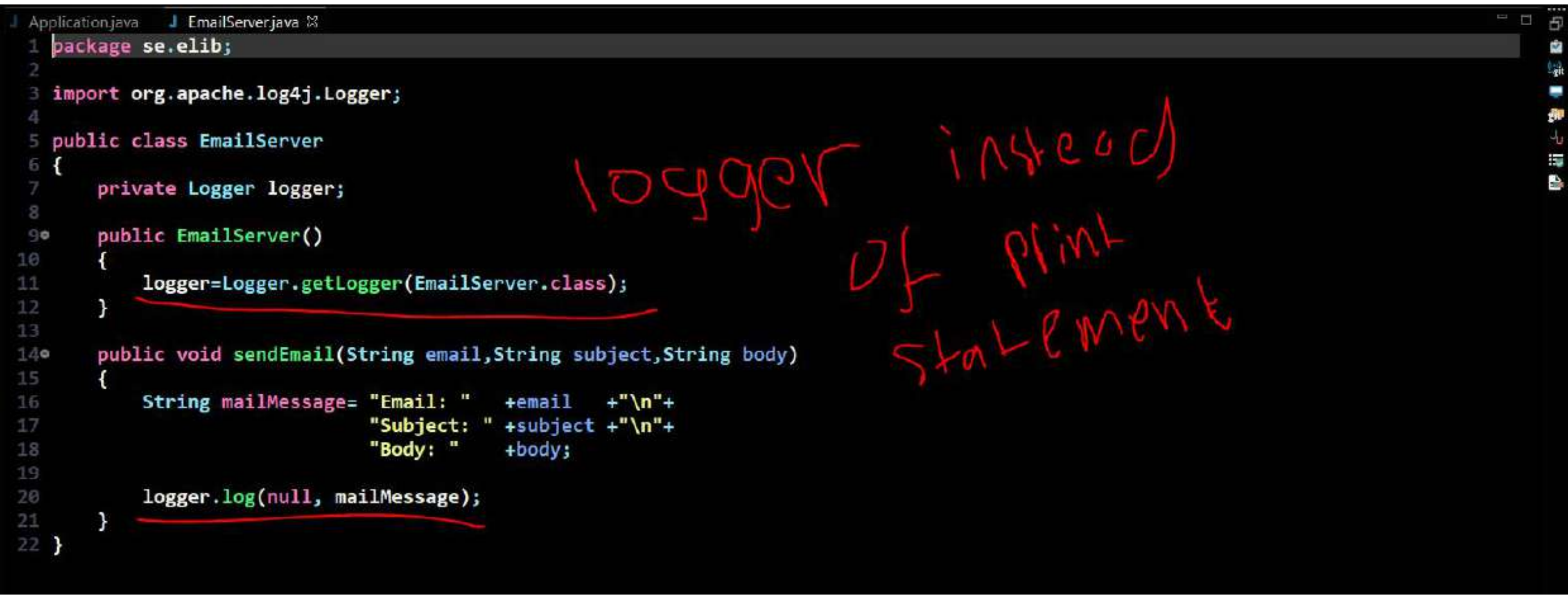
pom.xml

for config

For logging errors

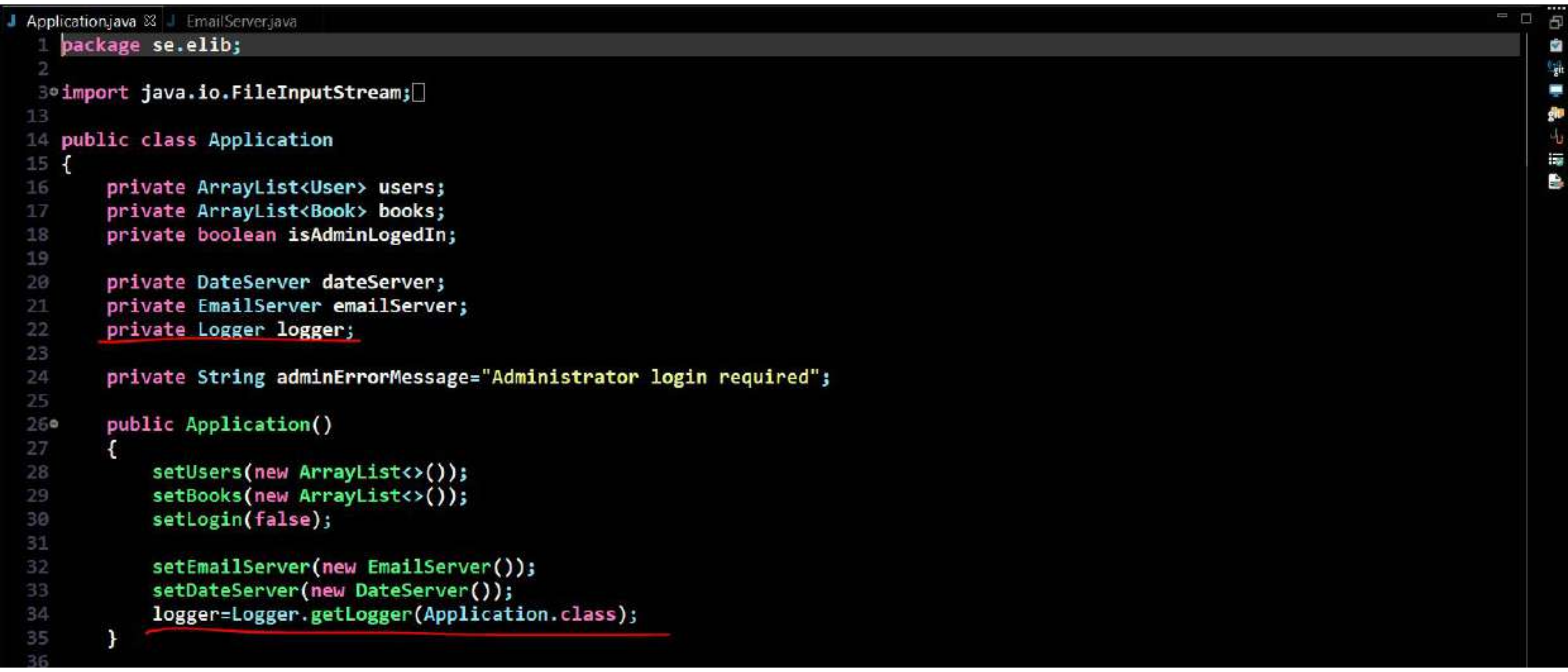


```
Book is not available
User not found
Book not found
you can't borrow more than five books
You can't borrow any new book because you have an overdue books
Can't borrow book you have fines
Administrator login required
This user is already registered
Administrator login required
User not found
Book not found
this book is not borrowed by you
Administrator login required
you can't unregister this user because he have some borrowed books to return
you can't unregister this user because he has fines to pay
```

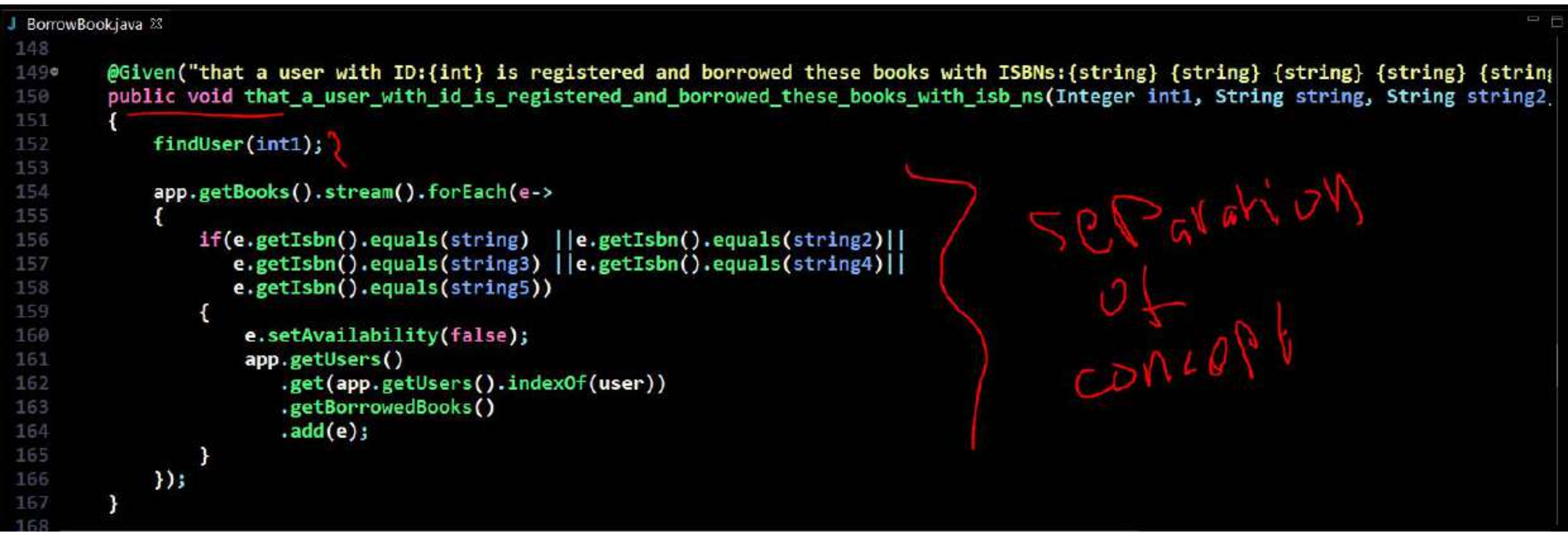


```java
package se.elib;

import org.apache.log4j.Logger;

public class EmailServer
{
    private Logger logger;

    public EmailServer()
    {
        logger=Logger.getLogger(EmailServer.class);
    }

    public void sendEmail(String email,String subject,String body)
    {
        String mailMessage= "Email: "   +email   +"\n"+
                            "Subject: " +subject +"\n"+
                            "Body: "    +body;

        logger.log(null, mailMessage);
    }
}
```

logger instead of print statement

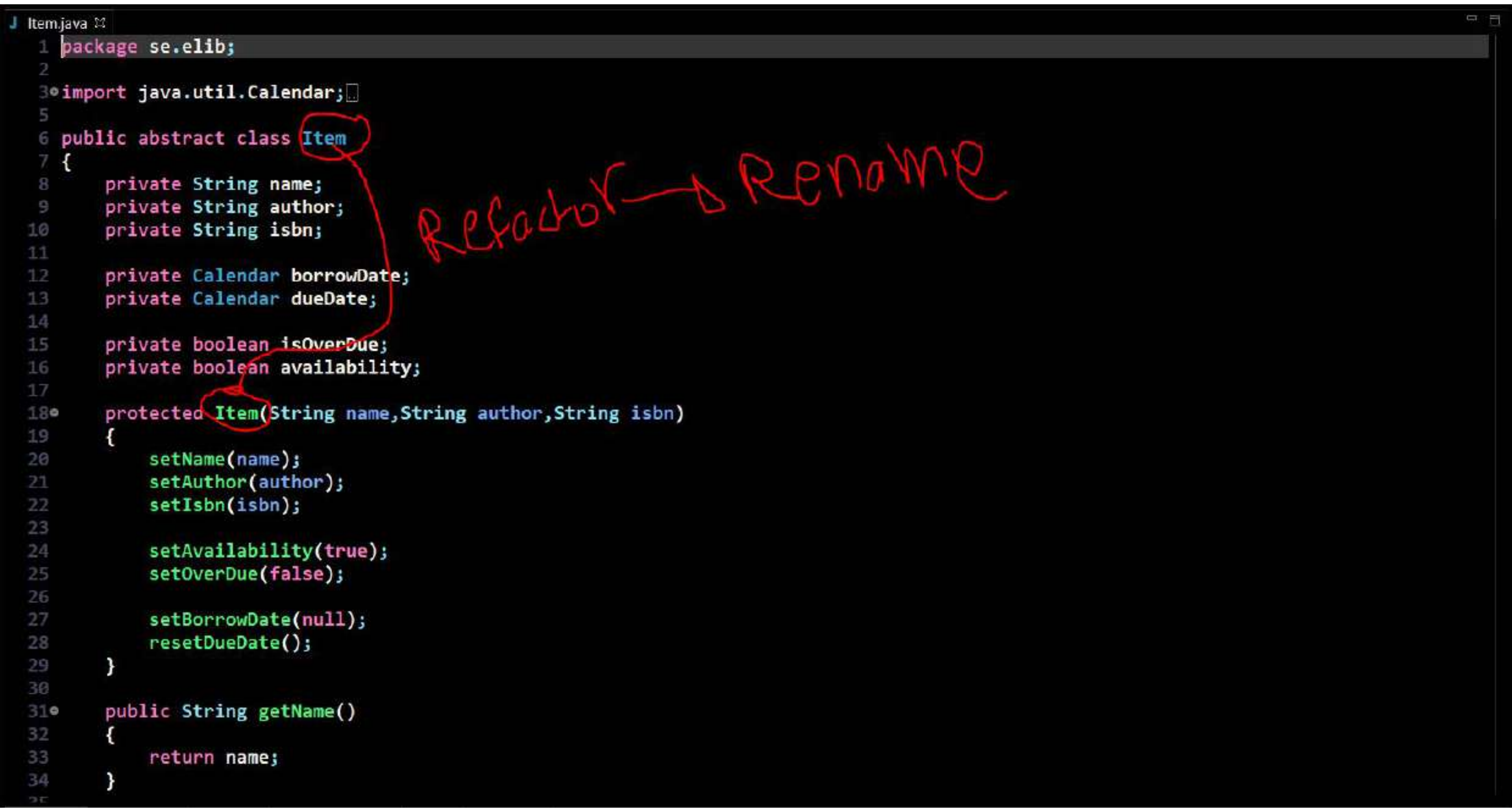


```java
package se.elib;

import java.io.FileInputStream;□

public class Application
{
    private ArrayList<User> users;
    private ArrayList<Book> books;
    private boolean isAdminLogedIn;

    private DateServer dateServer;
    private EmailServer emailServer;
    private Logger logger;

    private String adminErrorMessage="Administrator login required";

    public Application()
    {
        setUsers(new ArrayList<>());
        setBooks(new ArrayList<>());
        setLogin(false);

        setEmailServer(new EmailServer());
        setDateServer(new DateServer());
        logger=Logger.getLogger(Application.class);
    }
```



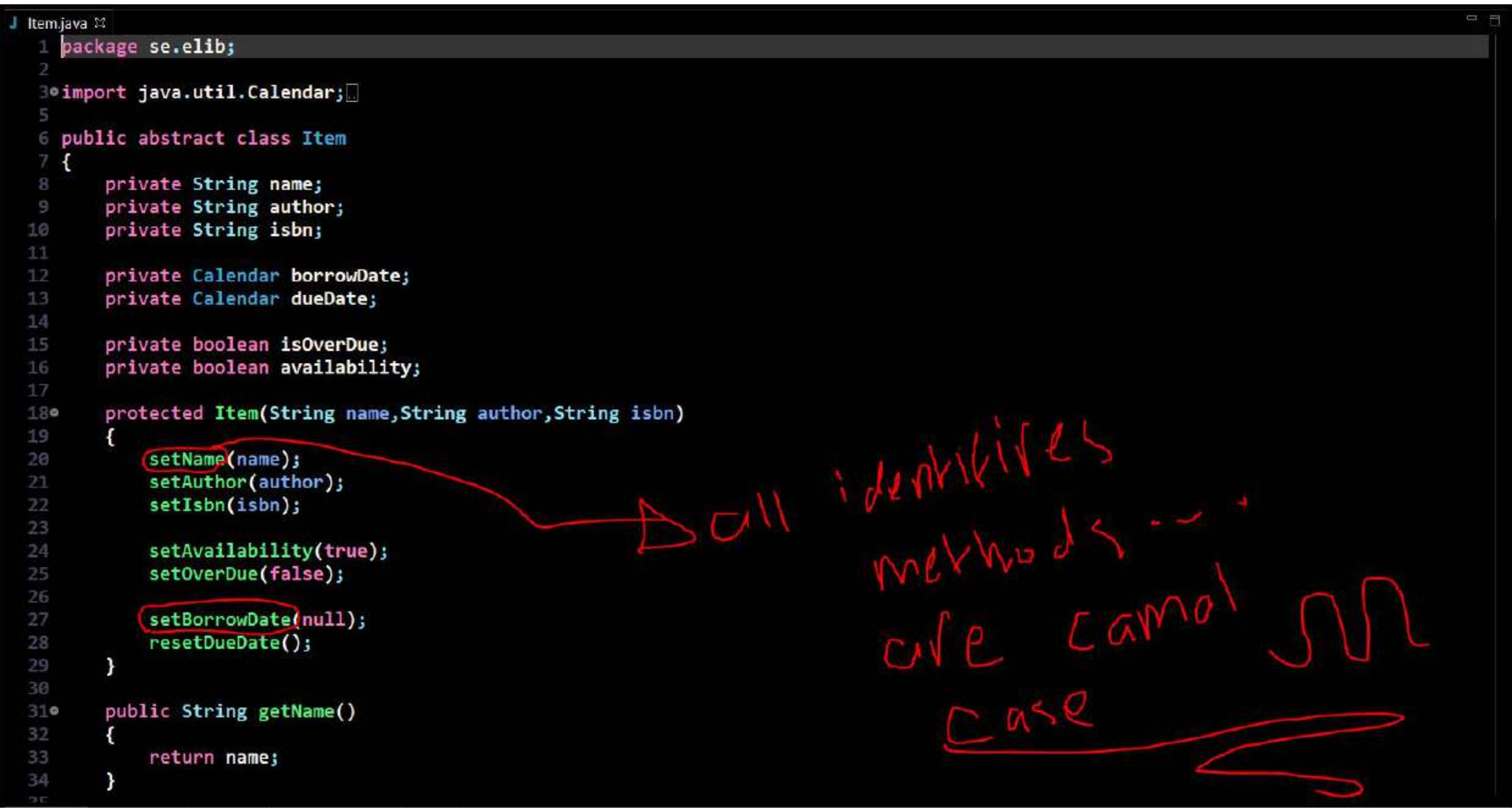
```java
    @Given("that a user with ID:{int} is registered and borrowed these books with ISBNs:{string} {string} {string} {string} {strin
    public void that_a_user_with_id_is_registered_and_borrowed_these_books_with_isb_ns(Integer int1, String string, String string2,
    {
        findUser(int1);

        app.getBooks().stream().forEach(e->
        {
            if(e.getIsbn().equals(string)  ||e.getIsbn().equals(string2)||
               e.getIsbn().equals(string3) ||e.getIsbn().equals(string4)||
               e.getIsbn().equals(string5))
            {
                e.setAvailability(false);
                app.getUsers()
                   .get(app.getUsers().indexOf(user))
                   .getBorrowedBooks()
                   .add(e);
            }
        });
    }
```

separation of concept



```
package se.elib;

import java.util.Calendar;

public abstract class Item
{
    private String name;
    private String author;
    private String isbn;

    private Calendar borrowDate;
    private Calendar dueDate;

    private boolean isOverDue;
    private boolean availability;

    protected Item(String name,String author,String isbn)
    {
        setName(name);
        setAuthor(author);
        setIsbn(isbn);

        setAvailability(true);
        setOverDue(false);

        setBorrowDate(null);
        resetDueDate();
    }

    public String getName()
    {
        return name;
    }
```

Refactor → Rename



```
package se.elib;

import java.util.Calendar;

public abstract class Item
{
    private String name;
    private String author;
    private String isbn;

    private Calendar borrowDate;
    private Calendar dueDate;

    private boolean isOverDue;
    private boolean availability;

    protected Item(String name,String author,String isbn)
    {
        setName(name);
        setAuthor(author);
        setIsbn(isbn);

        setAvailability(true);
        setOverDue(false);

        setBorrowDate(null);
        resetDueDate();
    }

    public String getName()
    {
        return name;
    }
```

all identifires methods ... are camal case