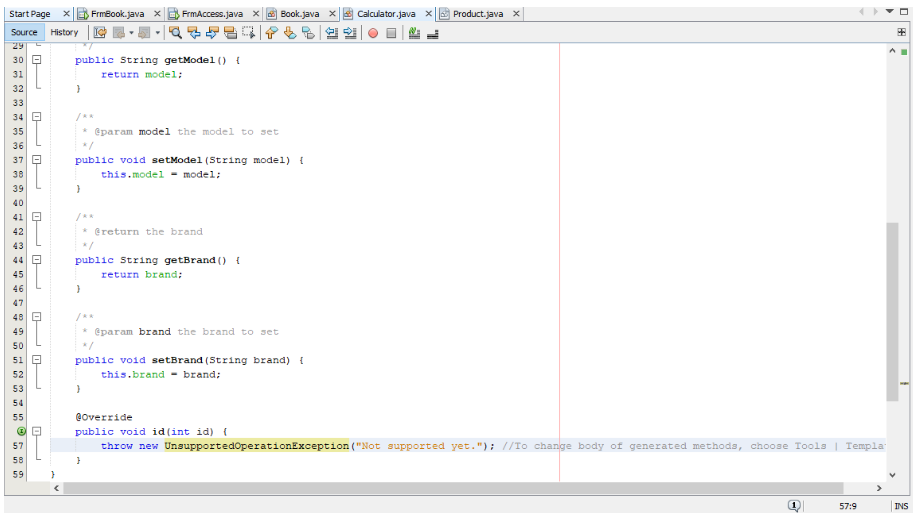
UNIVERSIDAD DE LAS FUERZAS ARMADAS ESPE

Object Oriented Programming

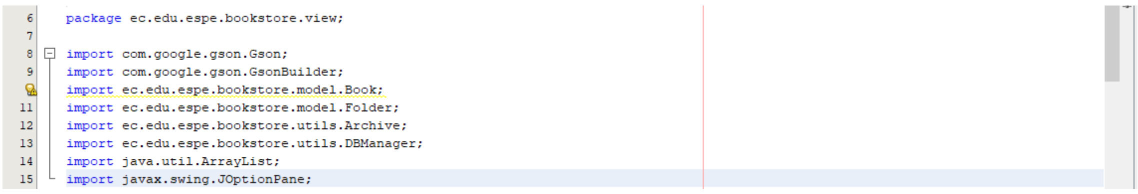
Name: Cajas Karla, Campoverde Carlos,Chablay Esteban, Chamorro Myckel
Team: EMCL.JAVA
NRC: 3682
Evaluated team:

- contains in classes override unnecessary



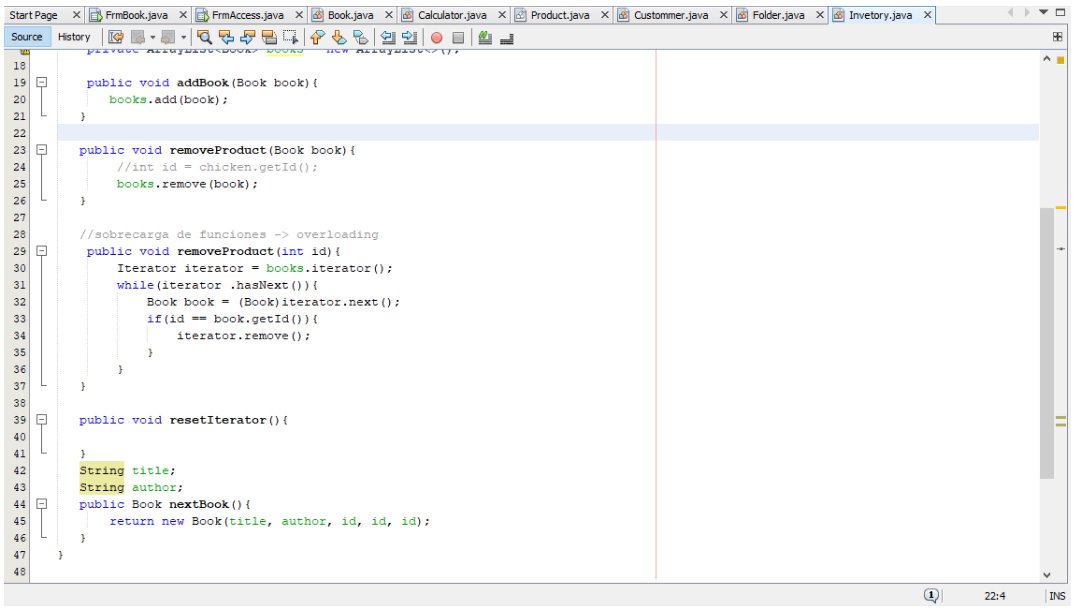
```java
         */
    public String getModel() {
        return model;
    }

    /**
     * @param model the model to set
     */
    public void setModel(String model) {
        this.model = model;
    }

    /**
     * @return the brand
     */
    public String getBrand() {
        return brand;
    }

    /**
     * @param brand the brand to set
     */
    public void setBrand(String brand) {
        this.brand = brand;
    }

    @Override
    public void id(int id) {
        throw new UnsupportedOperationException("Not supported yet."); //To change body of generated methods, choose Tools | Templa
    }
}
```

- use of unnecessary libraries



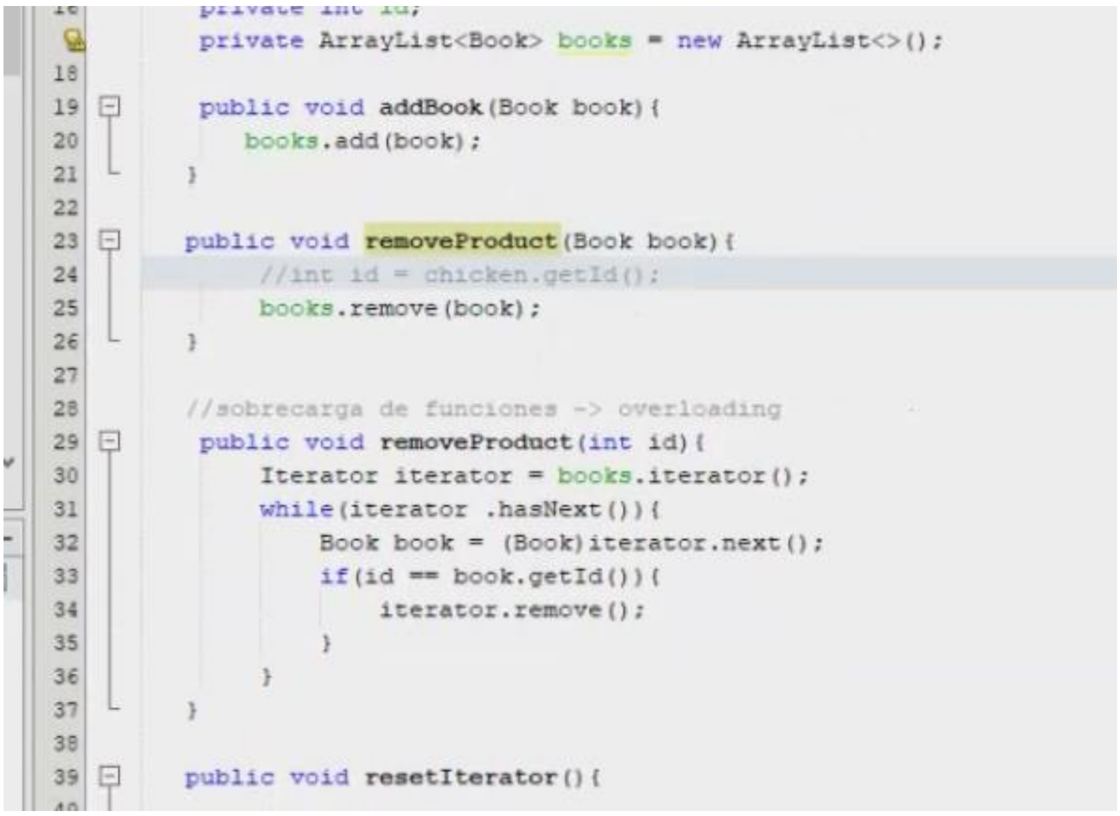
```java
package ec.edu.espe.bookstore.view;

import com.google.gson.Gson;
import com.google.gson.GsonBuilder;
import ec.edu.espe.bookstore.model.Book;
import ec.edu.espe.bookstore.model.Folder;
import ec.edu.espe.bookstore.utils.Archive;
import ec.edu.espe.bookstore.utils.DBManager;
import java.util.ArrayList;
import javax.swing.JOptionPane;
```

- empty method, it is not encapsulated



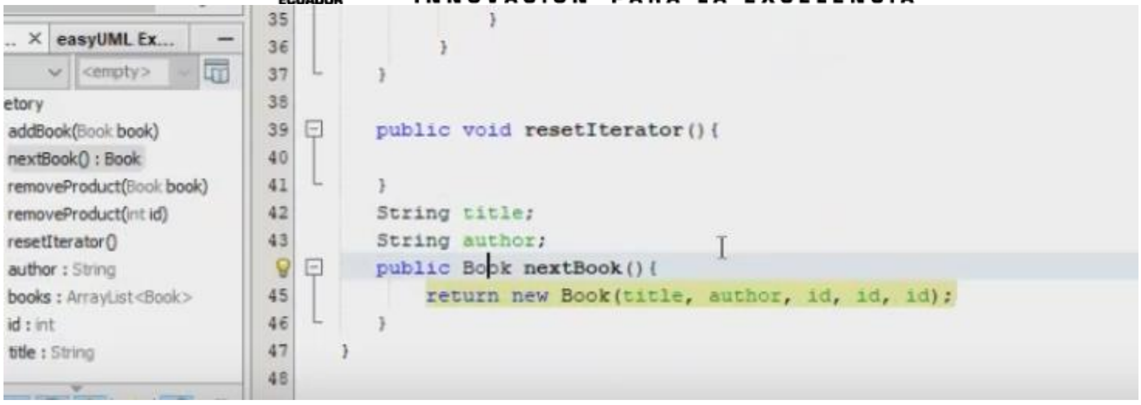
```java
    public void addBook(Book book){
        books.add(book);
    }

    public void removeProduct(Book book){
        //int id = chicken.getId();
        books.remove(book);
    }

    //sobrecarga de funciones -> overloading
     public void removeProduct(int id){
        Iterator iterator = books.iterator();
        while(iterator .hasNext()){
            Book book = (Book)iterator.next();
            if(id == book.getId()){
                iterator.remove();
            }
        }
    }

    public void resetIterator(){

    }
    String title;
    String author;
    public Book nextBook(){
       return new Book(title, author, id, id, id);
    }
}
```

- about unnecessary writing



```java
     private ArrayList<Book> books = new ArrayList<>();

     public void addBook(Book book){
        books.add(book);
    }

    public void removeProduct(Book book){
        //int id = chicken.getId();
        books.remove(book);
    }

    //sobrecarga de funciones -> overloading
     public void removeProduct(int id){
        Iterator iterator = books.iterator();
        while(iterator .hasNext()){
            Book book = (Book)iterator.next();
            if(id == book.getId()){
                iterator.remove();
            }
        }
    }

    public void resetIterator(){
```

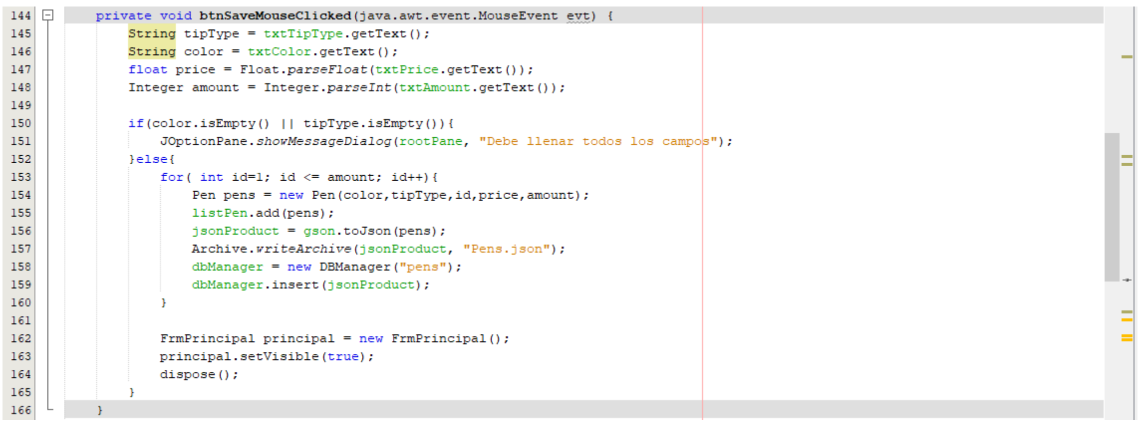
- unnecessary methods



```
etory
addBook(Book book)
nextBook() : Book
removeProduct(Book book)
removeProduct(int id)
resetIterator()
author : String
books : ArrayList<Book>
id : int
title : String
```

```
                }
            }
        }

        public void resetIterator(){

        }
        String title;
        String author;
        public Book nextBook(){
            return new Book(title, author, id, id, id);
        }
    }
```

- repeating code in classes can create a function



```
    private void btnSaveMouseClicked(java.awt.event.MouseEvent evt) {
        String tipType = txtTipType.getText();
        String color = txtColor.getText();
        float price = Float.parseFloat(txtPrice.getText());
        Integer amount = Integer.parseInt(txtAmount.getText());

        if(color.isEmpty() || tipType.isEmpty()){
            JOptionPane.showMessageDialog(rootPane, "Debe llenar todos los campos");
        }else{
            for( int id=1; id <= amount; id++){
                Pen pens = new Pen(color,tipType,id,price,amount);
                listPen.add(pens);
                jsonProduct = gson.toJson(pens);
                Archive.writeArchive(jsonProduct, "Pens.json");
                dbManager = new DBManager("pens");
                dbManager.insert(jsonProduct);
            }

            FrmPrincipal principal = new FrmPrincipal();
            principal.setVisible(true);
            dispose();
        }
    }
```