MINISTERUL EDUCAȚIEI, CULTURII ȘI CERCETĂRII AL REPUBLICII MOLDOVA

Universitatea Tehnică a Moldovei

Facultatea Calculatoare, Informatică şi Microelectronică Departamentul Inginerie Software şi Automatică

CEBAN VASILE FAF-223

Report

Laboratory work nr. 0

of Software Design Techniques and Mechanisms

Chișinău - 2024

Subject: SOLID Principles

BASIC TASK:

1. Implement 2 SOLID letters in a simple project.

Source Code:

Book Class:

```
public class Book {
    String title;
    String author;
    String publisher;
    int pages;

    public Book(){}

    public Book(String title, String author, String publisher, int pages) {
        this.title = title;
        this.author = author;
        this.publisher = publisher;
        this.pages = pages;
    }
    public String getTitle() {
        return title;
    }
    public void setTitle(String title) {
        this.title = title;
    }
    public String getAuthor() {
        return author;
    }
    public void setAuthor(String author) {
        this.author = author;
    }
    public String getPublisher() {
        return publisher;
    }
    public void setPublisher(String publisher) {
        this.publisher = publisher;
    }
    public int getPages() {
        return pages;
    }
    public void setPages(int pages) {
        this.pages = pages;
    }

    public boolean isAuthor(String author) {
        return this.author.equals(author);
    }

    @Override
    public String toString() {
        return "Name: "+getTitle()+"\nAuthor: "+getAuthor()+"\nPublisher: "+getPublisher()+"\nPages: "+getPages();
    }
}
```

BookPrinter Class:

```
public class BookPrinter {

    public void printBook(Book book){
        System.out.println(book);
    }
}
```

BookWriter Class:

```
package solid_principles.Open_Closed;

import solid_principles.Single_Responsibility.Book;

public class BookWriter extends Book {
    String text;


    public BookWriter() {}


    public BookWriter(String title, String author, String publisher, String
text, int pages) {
        super(title, author, publisher, pages);
        this.text = text;
    }


    public String getText() {
        return text;
    }


    public void setText(String text) {
        this.text = text;
    }


    @Override
    public String toString() {
        return "Name: "+getTitle()+"\nAuthor: "+getAuthor()+"\nPublisher:
"+getPublisher()+"\nDescribe: "+getText()+"\nPages: "+getPages();
    }
}
```

1. Single Responsibility Principle (S):
   - The Book class is responsible for representing a book and its properties.
   - Each method in the Book class has a single, well-defined responsibility.
   - The BookPrinter class has the single responsibility of printing a book.
2. Open-Closed Principle (O):
   - The BookWriter class extends the Book class, adding new functionality without modifying the original Book class.
   - The BookPrinter class is open for extension (e.g., different print formats) but closed for modification.

Conclusion

By following these principles, the code achieves better organization, improved maintainability, and easier extensibility. It separates concerns effectively and allows for future enhancements with minimal risk of breaking existing functionality.

This approach to software design leads to more robust, flexible, and easier-to-maintain code, which are key benefits of adhering to SOLID principles in object-oriented programming.