Ejercicio Biblioteca en Java

Dilan Robayo Tavera

Ficha 2848939

Análisis y Desarrollo de Software

Introducción

Este informe tiene como propósito documentar el desarrollo y ejecución de una aplicación básica en Java destinada a la gestión de una biblioteca. El trabajo se enmarca dentro de las actividades prácticas del resultado de aprendizaje, con el objetivo de fortalecer habilidades en la programación orientada a objetos. A lo largo del ejercicio, se aplican conceptos clave como la encapsulación, herencia y modularidad, fundamentales para el desarrollo de sistemas de software escalables y mantenibles.

La aplicación en cuestión permite realizar tareas sencillas, como la entrada y visualización de información sobre libros. Aunque se trata de un sistema básico, es un ejercicio introductorio que sienta las bases para la creación de aplicaciones más complejas en un futuro.

Objetivo del Proyecto

El objetivo principal de este ejercicio es desarrollar un sistema básico que permita gestionar libros dentro de una biblioteca, utilizando Java como lenguaje de programación. A través de la implementación de diversas clases y el uso de la consola para la entrada de datos, el proyecto busca:

- Facilitar el aprendizaje de conceptos fundamentales en programación orientada a objetos (POO).
- Poner en práctica el diseño modular de software mediante la organización de clases y paquetes.
- Familiarizarse con el entorno de desarrollo Visual Studio Code, utilizado para la codificación y ejecución del proyecto.

Descripción del Sistema

El sistema desarrollado se organiza en varios módulos que representan entidades clave dentro de la biblioteca, tales como libros, autores y prestatarios. A continuación, se describe la estructura de clases y su interacción dentro del sistema:

- **Clase General**: Es la clase base de la cual heredan varias otras clases. Contiene los atributos básicos como `código` y `nombre`, y un método de validación de datos. Este enfoque permite la reutilización de código en otras clases que compartan estas características comunes.
- **Clase Libro**: Extiende la clase `General` y agrega características específicas de un libro, como su `edición` y `año de publicación`. Esta clase encapsula la información básica sobre cada libro y permite que sea gestionada de manera individual.
- **Clase Autor**: Al igual que `Libro`, extiende de la clase `General`. Sin embargo, introduce atributos propios de un autor, como la `observación` y el número de `libros publicados`.
- **Clase Prestario**: Extiende `General` y modela a las personas que toman libros prestados de la biblioteca. Agrega atributos como `dirección`, `teléfono` y `RUC`, permitiendo la identificación y contacto del prestatario.

- **Clase CargarLibro**: Es responsable de manejar la interacción con el usuario. A través de esta clase, el sistema solicita al usuario ingresar información relacionada con los libros y la almacena en una instancia de la clase `Libro`. También permite la impresión de los datos ingresados.

Estructura del Proyecto

El proyecto está organizado en un entorno de trabajo dividido en varias clases agrupadas en el paquete `clasesbiblioteca`, con la clase principal `biblioteca.java` localizada en la raíz del proyecto. A continuación, se presenta la estructura jerárquica del proyecto:

```
src/
 ├─ clasesbiblioteca/
 │    ├─ Autor.java
 │    ├─ CargarLibro.java
 │    ├─ Ciudad.java
 │    ├─ General.java
 │    ├─ Libro.java
 │    ├─ Pais.java
 │    └─ Prestario.java
 └─ biblioteca.java
```

Cada una de estas clases juega un papel específico dentro de la lógica del programa. La estructura modular facilita la expansión futura del sistema, permitiendo la adición de nuevas funcionalidades sin necesidad de alterar significativamente el código existente.

Descripción del Código

- **Clase General**: Se define como una clase abstracta que contiene los atributos `código` y `nombre`, junto con un método para validar que estos datos sean correctos. Esto asegura que cualquier clase que herede de `General`

```
public class General {
    private int codigo;
    private String nombre;

    public boolean validarDatos() {
        return codigo > 0 && nombre != null && !nombre.isEmpty();
    }
    // Getters y Setters
}
```

- implemente estos atributos.

Clase Libro Extiende de `General` y añade los atributos `edición` y `año de publicación` . También incluye métodos para obtener el stock de un libro, aunque su implementación es básica en esta versión del sistema.

```
public class Libro extends General {
    private int edicion;
    private int anoPublicacion;

    public int getStock() {
        return 0;  // Implementación básica
    }
    // Getters y Setters
}
```

- **Clase CargarLibro** En esta clase se solicita la entrada de datos a través de la consola. Los datos ingresados se almacenan en una instancia de la clase `Libro` y luego se imprimen.

```
public class CargarLibro {
    private Libro libro;

    public void cargarLibro() {
        Scanner scanner = new Scanner(System.in);
        libro = new Libro();

        System.out.println("Ingrese el código del libro:");
        libro.setCodigo(scanner.nextInt());

        System.out.println("Ingrese el nombre del libro:");
        libro.setNombre(scanner.next());

        System.out.println("Ingrese la edición del libro:");
        libro.setEdicion(scanner.nextInt());

        System.out.println("Ingrese el año de publicación del libro:");
        libro.setAnoPublicacion(scanner.nextInt());

        scanner.close();
    }

    public void imprimirLibro() {
        System.out.println("Datos del libro:");
        System.out.println("Código: " + libro.getCodigo());
        System.out.println("Nombre: " + libro.getNombre());
        System.out.println("Edición: " + libro.getEdicion());
        System.out.println("Año de publicación: " + libro.getAnoPublicacion
());
    }
}
```

Funcionamiento del Sistema

El sistema comienza con la ejecución de la clase `biblioteca.java`, donde se llama a la clase `CargarLibro` para iniciar el flujo de datos. El programa solicita al usuario que ingrese la información necesaria del libro. Una vez que los datos son capturados, se muestran en la consola de manera organizada.

Este sistema de entrada y salida de datos es sencillo, pero su diseño modular permite futuras mejoras, como la integración de bases de datos para la persistencia de la información o la expansión de funcionalidades para gestionar autores y prestatarios.

Resultados Obtenidos

Durante la ejecución del programa, los usuarios son capaces de ingresar información básica de los libros, que luego es procesada y mostrada. A continuación, se presenta un ejemplo del flujo:

```
Bienvenido al sistema de la Biblioteca
Ingrese el código del libro:
101
Ingrese el nombre del libro:
Programación en Java
Ingrese la edición del libro:
3
Ingrese el año de publicación del libro:
2020
Datos del libro:
Código: 101
Nombre: Programación en Java
Edición: 3
Año de publicación: 2020
Gracias por usar el sistema de la Biblioteca
```

Conclusiones

Este ejercicio de programación permitió reforzar conocimientos fundamentales sobre la programación orientada a objetos, como el uso de clases, herencia y encapsulamiento. Asimismo, demostró la importancia de la estructura modular y el diseño de sistemas escalables. Aunque el sistema es básico, se puede expandir fácilmente agregando nuevas funcionalidades como la gestión de usuarios o el registro de préstamos de libros.