

Parcial 1

Carlos José Rueda Sarmiento 1005479438

Enunciado

Implemente un simulador de operaciones básicas de una biblioteca universitaria, diseñando e implementando las clases y conceptos vistos en clase (encapsulamiento, instanciación, atributos, objetos, etc.) que considere pertinentes.

Desarrollo

1. Clases

Se implementaron 4 clases las cuales se denominaron de la siguiente forma

1. Libro: Esta clase se encarga de crear los objetos libros, adicionalmente modifica la clase String para que el método toString formatee la impresión de cada uno de los objetos que pertenezcan a esta clase. Esta clase únicamente contiene un constructor, dos variables (category y bookTitle) las cuales son privadas pues no necesitan ser accedidas por ninguna otra clase.
2. Libros: Esta clase se encarga de crear la lista que va a contener los libros que el usuario vaya registrando. Se importaron dos clases, primero la clase java.util.ArrayList que permite crear un arreglo que contiene una lista, segundo la clase java.util.List la cual nos permite crear la lista de libros. En esta clase encontramos su constructor que asigna la lista al arreglo y dos métodos, primero el método createBook el cual crea un libro y le asigna una categoría con un numero entero y un título y por último lo agrega a la lista de libros. También encontramos un método getLibros el cual nos devuelve un arreglo de los libros existentes. En esta clase la variable libros es instanciada como una lista y es final ya que no será cambiada después de inicializada.
3. Usuario: Esta clase es muy similar a la de libro con la diferencia que solo tiene una variable, pero su estructura es igual.
4. Usuarios: esta clase es muy similar a la de libros con diferentes nombres en las variables, pero su funcionamiento es igual.

2. La clase Parcial1

Dentro de la clase Parcial1 se realizó toda la lógica de la consola. Inicialmente se crean 3 objetos los cuales son las listas de libros y usuarios y el escáner que leerá las entradas del usuario. Se inicializa una variable booleana que determina la salida del programa. Se crea un bucle while para detener el programa cuando la variable "exit" sea true. Ingresamos a un Switch con 5 casos los cuales son opciones para el usuario. Cada uno de los casos ejecuta la función que lo describe con ayuda de los métodos de cada una de las clases. Al final de cada lógica se rompe el switch e inicia nuevamente el bucle.

Descripción del código

Libro

```
package movimiento.parcial1;

/**
 *
 * @author estudiante
 */
public class Libro {

    private int category;
    private String bookTitle;

    public Libro (int category, String bookTitle) {

        this.category = category;
        this.bookTitle = bookTitle;
    }

    @Override
    public String toString() {
        return String.format("Categoria: %d ----- Titulo: %s", category, bookTitle);
    }
}
```

1. Se inicializan las variables
2. Se crea el constructor que asigna las variables
3. Se modifica el método toString de la clase String para la impresión en consola

Usuario

```
package movimiento.parcial1;

/**
 *
 * @author estudiante
 */
public class Usuario {

    private String userName;

    public Usuario (String userName) {

        this.userName = userName;
    }

    @Override
    public String toString() {
        return String.format("Usuario: %s", userName);
    }
}
```

1. Se inicializa la variable
2. Se crea el constructor
3. Se modifica el método toString

Usuarios

```
import java.util.ArrayList;
import java.util.List;

/**
 *
 * @author estudiante
 */
public class Usuarios {

    private final List<Usuario> usuarios;

    public Usuarios() {
        this.usuarios = new ArrayList<>();
    }

    public void createUser(String userName) {
        Usuario usuario = new Usuario(userName);
        usuarios.add(usuario);
    }

    public List<Usuario> getUsuarios() {
        return usuarios;
    }

}
```

1. Se importan las clases necesarias
2. Se inicializa la lista de usuarios
3. Se crea el constructor
4. Se crean los dos métodos que la clase tendrá

Libros

```
package movimiento.parcial1;
import java.util.ArrayList;
import java.util.List;

/**
 *
 * @author estudiante
 */
public class Libros {

    private final List<Libro> libros;

    public Libros() {
        this.libros = new ArrayList<>();
    }

    public void createBook(int category, String bookTitle) {
        Libro libro = new Libro(category, bookTitle);
        libros.add(libro);
    }

    public List<Libro> getLibros() {
        return libros;
    }
}
```

1. Se importan las clases necesarias
2. Se inicializa la lista de libros
3. Se crea el constructor de la clase
4. Se crean los métodos que la clase va a usar

Pruebas de funcionamiento

=== BIENVENIDO A LA BIBLIOTECA DE LA UNAL ===

1. Crear Usuario
2. Mostrar usuarios
3. Registrar un libro
4. Mostrar libros
5. Salir

Seleccionar una opción

```
1
Ingresa el usuario
Carlos
```

Opción seleccionada 1 Ingresar el nombre del usuario

Usuario creado exitosamente.

=== BIENVENIDO A LA BIBLIOTECA DE LA UNAL ===

1. Crear Usuario
2. Mostrar usuarios
3. Registrar un libro
4. Mostrar libros
5. Salir

|

Usuario creado exitosamente. Ingresar nueva opción o 5 para salir.

2

Usuario: Carlos

=== BIENVENIDO A LA BIBLIOTECA DE LA UNAL ===

1. Crear Usuario
2. Mostrar usuarios
3. Registrar un libro
4. Mostrar libros
5. Salir

|

=== BIENVENIDO A LA BIBLIOTECA DE LA UNAL ===

1. Crear Usuario
2. Mostrar usuarios
3. Registrar un libro
4. Mostrar libros
5. Salir

3

Ingrese la categoria

1. Ficcion
2. Ingenieria
3. Medicina

1

Ingrese el titulo

Harry Potter

Libro creado exitosamente.

4

Categoria: 1 ----- Titulo: Harry Potter

=== BIENVENIDO A LA BIBLIOTECA DE LA UNAL ===

1. Crear Usuario
2. Mostrar usuarios
3. Registrar un libro
4. Mostrar libros
5. Salir

|

L 5

BUILD SUCCESS

Total time: 02:47 min

Finished at: 2025-05-08T17:53:35-05:00
