



Servicio Nacional de Aprendizaje – SENA

Centro de Diseño y Metrología

InformeBiblioteca

Aprendiz:

Juan Esteban Castro Escamilla

No Ficha 2848530-A

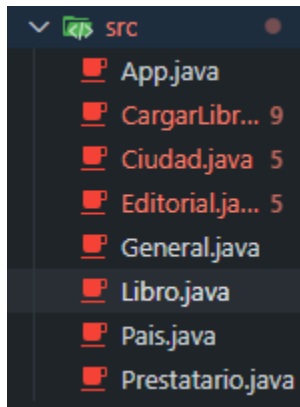
Tabla Contenido

Tabla Contenido	2
Biblioteca Agregación y Composición JAVA	3

Biblioteca Agregación y Composición JAVA

Composicion:

(Biblioteca se compone principalmente de estas clases: CargarLibro, Ciudad, Editorial, General, Libro, Pais, Prestatarios)



Clase CargarLibro:

(Se utiliza la utilidad Scanner para la asignación de los datos)

(Este clase CargarLibro contiene una función para hacer la respectiva carga del mismo y el libro ya cargado y la impresión de este)

(Contamos con la función de CargarLibro la cual compone a un scanner en el cual, dentro de este scanner se guardaran los datos que se le pida por pantalla o terminal al usuario, en este caso ingresar el código, nombre, edición y año de publicación del libro y así mismo retornar dichos valores)

```
import java.util.Scanner;

public class CargarLibro {

    Run | Debug
    public static void main(String[] args) {
        CargarLibro cargador = new CargarLibro();
        Libro libro = cargador.cargarLibro();
        cargador.imprimirLibro(libro);
    }

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

        System.out.println(x:"Ingrese el código del libro:");
        int codigo = scanner.nextInt();
        scanner.nextLine(); // Limpiar el buffer

        System.out.println(x:"Ingrese el nombre del libro:");
        String nombre = scanner.nextLine();

        System.out.println(x:"Ingrese la edición del libro:");
        int edicion = scanner.nextInt();

        System.out.println(x:"Ingrese el año de publicación del libro:");
        int anioPublicacion = scanner.nextInt();

        return new Libro(codigo, nombre, edicion, anioPublicacion);
    }
}
```

(Finalmente concluye con la función imprimirLibro la cual dentro de esta estara el System out es decir aquellos que permiten mostrar por pantalla o terminal los datos que se asignaron previamente es decir, los detalles del libro)

```
public void imprimirLibro(Libro libro) {  
    System.out.println("Detalles 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.getAñoPublicacion());  
    System.out.println("Stock: " + libro.getStock());  
}
```

Clase General:

(Ahora se crea la clase General la cual contendrá alguno datos del prestatario tales como su nombre y un id para identificarlo dentro del sistema)

(Identificamos un constructor para obtener precisamente el id y nombre de los atributos previamente asignados)

(Creamos los respectivos Getter and Setters para cada uno)

```
public class General {  
    private int id;  
    private String nombre;  
    public General() {}  
  
    public General(int id, String nombre) {  
        this.id = id;  
        this.nombre = nombre;  
    }  
  
    public int getId() {  
        return id;  
    }  
  
    public void setId(int id) {  
        this.id = id;  
    }  
  
    public String getNombre() {  
        return nombre;  
    }  
  
    public void setNombre(String nombre) {  
        this.nombre = nombre;  
    }  
}
```

Clase Ciudad:

(Creación de la clase Ciudad la cual hereda los atributos de la CLASS GENERAL así mismo esta clase contendrá un nombre y el respectivo país)

(Contiene un constructor vacío y otro con parámetros los cuales se especifican con el this para obtener precisamente dichos atributos)

(Claramente se crea los Getter And Setter para cada uno de los atributos creados dentro de la clase)

```
public class Ciudad extends General {  
    private String nombre;  
    private Pais pais;  
  
    public Ciudad() {  
          
    }  
  
    public Ciudad(int id, String nombre, Pais pais) {  
        this.nombre = nombre;  
        this.pais = pais;  
    }  
  
    public String getNombre() {  
        return nombre;  
    }  
  
    public void setNombre(String nombre) {  
        this.nombre = nombre;  
    }  
  
    public Pais getPais() {  
        return pais;  
    }  
  
    public void setPais(Pais pais) {  
        this.pais = pais;  
    }  
}
```

Clase Pais:

(Creación respectiva del País la cual tendrá el nombre de dicho país y sus Setter And Getter, realmente aquí no hay tanta explicación ya que no contiene nada extraño nuevo a parte de que hereda con el extends la clase General)

```
public class Pais extends General {  
    private String nombre;  
  
    public Pais() {  
    }  
  
    public Pais(int id, String nombre) {  
        this.nombre = nombre;  
    }  
  
    public String getNombre() {  
        return nombre;  
    }  
  
    public void setNombre(String nombre) {  
        this.nombre = nombre;  
    }  
}
```


Clase Prestatario:

(Creación de la clase Prestatarios es decir dentro de esta clase se contendrá la información de la persona_Usuario que pedirá un libro en específico)

(Hereda los los atributos de general y se crean los respectivos constructores)

```
public class Prestatario extends General {  
    private String nombre;  
    private String apellido;  
    private String direccion;  
    private Ciudad ciudad;  
    private String telefono;  
  
    public Prestatario() {  
    }  
  
    public Prestatario(int id, String nombre, String apellido, String direccion, Ciudad ciudad, String telefono) {  
        this.nombre = nombre;  
        this.apellido = apellido;  
        this.direccion = direccion;  
        this.ciudad = ciudad;  
        this.telefono = telefono;  
    }  
}
```

(Sus respectivos Getter And Setter)

```
// Getters y setters
public String getNombre() {
    return nombre;
}

public void setNombre(String nombre) {
    this.nombre = nombre;
}

public String getApellido() {
    return apellido;
}

public void setApellido(String apellido) {
    this.apellido = apellido;
}

public String getDireccion() {
    return direccion;
}

public void setDireccion(String direccion) {
    this.direccion = direccion;
}

public Ciudad getCiudad() {
    return ciudad;
}

public void setCiudad(Ciudad ciudad) {
    this.ciudad = ciudad;
}

public String getTelefono() {
    return telefono;
}

public void setTelefono(String telefono) {
    this.telefono = telefono;
}
```

Clase Editorial:

(Creamos una clase para la respectiva Editorial con su nombre de la misma y ciudad)

(Asi mismo sus constructores y setter and getter que ya se ha visto previamente)

```
public class Editorial extends General {  
    private String nombre;  
    private Ciudad ciudad;  
  
    public Editorial() {  
    }  
  
    public Editorial(int id, String nombre, Ciudad ciudad) {  
        this.nombre = nombre;  
        this.ciudad = ciudad;  
    }  
  
    public String getNombre() {  
        return nombre;  
    }  
  
    public void setNombre(String nombre) {  
        this.nombre = nombre;  
    }  
  
    public Ciudad getCiudad() {  
        return ciudad;  
    }  
  
    public void setCiudad(Ciudad ciudad) {  
        this.ciudad = ciudad;  
    }  
}
```

Clase Libro:

(Creación de la clase Libro para obtener la edición de este y su año de publicación)

(Creamos los respectivos constructores y Setter and Getters, también agregamos un Get extra para obtener el Stock o cantidad disponible para la prestación de dicho libro)

```
public class Libro {  
    private int edicion;  
    private int añoPublicacion;  
  
    public Libro() {  
  
    }  
    //constructor  
    public Libro(int codigo,int edicion, int añoPublicacion) {  
        this.edicion = edicion;  
        this.añoPublicacion = añoPublicacion;  
    }  
    public int getStock() {  
        return 10;  
    }  
    public int getEdicion(){  
        return edicion;  
    }  
    public void setEdicion(int edicion){  
        this.edicion = edicion;  
    }  
    public int getAñoPublicacion() {  
        return añoPublicacion;  
    }  
    public void setAñopublicacion (int añoPublicacion) {  
        this.añoPublicacion = añoPublicacion;  
    }  
}
```

Agregación Biblioteca:

(Biblioteca a parte de las composiciones principales que esta contiene, también debe tener o conseguir daños extras que ayudarán en el buen funcionamiento del sistema de la biblioteca para esto, se ha creado más clases que agregaran y funcionarán junto a lo que se compone de una biblioteca)

(El libro se puede catalogar como producto, por ende, tendrá sus respectivos atributos como producto, así mismo, este tendrá una factura, comprobante, cliente y fecha para poder guardar dichos datos en el sistema y saber con precisión acerca de la prestación de dicho libro o libros)

Clase Producto:

(Creacion de la clase producto la con contiene el codigo, descripcion y precio de este, así mismo sus respectivos constructores y sus getter and setter)

```
package Clases;

public class Producto {

    private int codigo;
    private String descripcion;
    private float precio;

    public Producto () {

    }

    public int getCodigo () {
        return codigo;
    }

    public void setCodigo (int val) {
        this.codigo = val;
    }

    public String getDescripcion () {
        return descripcion;
    }

    public void setDescripcion (String val) {
        this.descripcion = val;
    }

    public float getPrecio () {
        return precio;
    }

    public void setPrecio (float val) {
        this.precio = val;
    }

    public Producto(int c, String d, float p){
        setCodigo(c);
        setDescripcion(d);
        setPrecio(p);
    }
}
```

Clase Cliente:

(Encontramos una clase para los diferentes clientes que se podrán encontrar en la biblioteca y necesitamos guardar dicha información, para ello, se realizó el atributo código y razón social)

(Contiene de igual manera sus constructores y setter and getter)

```
package Clases;

public class Cliente {
    private int codigo;
    private String razonSocial;

    public Cliente () {
    }

    public int getCodigo () {
        return codigo;
    }

    public void setCodigo (int val){
        this.codigo = val;
    }

    public String getRazonSocial (){
        return razonSocial;
    }

    public void setRazonSocial (String val) {
        this.razonSocial = val;
    }

    public Cliente(int c, String r){
        setCodigo(c);
        setRazonSocial(r);
    }
}
```

Clase Fecha:

(Contiene esta clase Fecha el día, mes y año exacto con todo lo respectivo de una clase)

```
package Clases;

public class Fecha {

    private int dia;
    private int mes;
    private int año;

    public Fecha () {

    }

    public int getAño () {
        return año;
    }

    public void setAño (int val) {
        this.año = val;
    }

    public int getDia (){
        return dia;
    }
    public void setDia (int val) {
        this.dia = val;
    }

    public int getMes (){
        return mes;
    }
    public void setMes (int val){
        this.mes = val;
    }

    public Fecha(int d, int m, int a){
        setDia(d);
        setMes(m);
        setAño(a);
    }
}
```

Clase Comprobante:

(Contenemos los datos del comprobante, en este caso, su tipo, número y fecha de la realización de este)

(A si mismo sus respectivos constructores y set and get)

```
package Clases;

public class Comprobante {

    private char tipo;
    private int numero;
    private Fecha fecha;

    public Comprobante () {
    }

    public Fecha getFecha () {
        return fecha;
    }

    public void setFecha (Fecha val) {
        this.fecha = val;
    }

    public int getNumero () {
        return numero;
    }

    public void setNumero (int val) {
        this.numero = val;
    }

    public char getTipo () {
        return tipo;
    }

    public void setTipo (char var) {
        this.tipo = var;
    }

    public Comprobante(char t, int n, Fecha f){
        setTipo(t);
        setNumero(n);
        setFecha(f);
    }
}
```


Clase Factura:

(Encontramos las utilidades ArrayList e Iterator que se utilizarán a continuación)

(Clase Factura contendrá atributos heredados del Comprobante y así mismo contendrá un Arreglo el cual contendrá el Producto y su nombre será mProducto para su utilización)

(Así mismo tendrá un total y un objeto que se ha sacado de la clase Cliente y se llamará mCliente)

```
package Clases;

import java.util.ArrayList;
import java.util.Iterator;

public class Factura extends Comprobante {

    private ArrayList<Producto> mProducto;
    private float total;
    private Cliente mCliente;
```

(Sus respectivos constructores y getter and setter)

```
public Factura() {  
}  
  
public Cliente getCliente() {  
    return mCliente;  
}  
  
public void setCliente(Cliente val) {  
    this.mCliente = val;  
}  
  
public float getTotal() {  
    return total;  
}  
  
public void setTotal(float val) {  
    this.total = val;  
}  
  
public ArrayList<Producto> getProducto() {  
    return mProducto;  
}  
  
public void setProducto(ArrayList<Producto> val) {  
    this.mProducto = val;  
}  
  
public Factura(char t, int n, Fecha f, Cliente cli) {  
    super(t, n, f);  
    setCliente(cli);  
}
```

```

public ArrayList<Producto> getProducto() {
    return mProducto;
}

public void setProducto(ArrayList<Producto> val) {
    this.mProducto = val;
}

public Factura(char t, int n, Fecha f, Cliente cli) {
    super(t, n, f);
    setCliente(cli);
}

```

(Creamos 3 funciones, una para agregar el producto dentro de arreglo y así mismo calcular el precio total)

(otra para mostrar el producto con la función iterator y así mismo un bucle while que hasta que termine de recorrer cada uno de los productos con su detalle de código, descripción y precio)

(Finalmente la 3 función mostrara cada uno de los datos asignados previamente de las diferentes clases)

```

public void agregarProducto(Producto p) {
    mProducto.add(p);
    setTotal(getTotal() + p.getPrecio());
}

public void mostrarProductos() {
    Iterator<Producto> iter = mProducto.iterator();
    while (iter.hasNext()) {
        Producto p = iter.next();
        System.out.printf(format:"Codigo: %d Descripcion: %s Precio: %5.2f \n",
            p.getCodigo(), p.getDescripcion(), p.getPrecio());
    }
}

public void mostrar() {
    System.out.printf(format:"Tipo: %c Número: %d Fecha: %d/%d/%d\n",
        getTipo(), getNumero(),
        getFecha().getDia(), getFecha().getMes(), getFecha().getAño());
    System.out.printf(format:"Cliente: \n");
    System.out.printf(format:"Codigo: %d Razon Social: %s \n",
        mCliente.getCodigo(), mCliente.getRazonSocial());
    System.out.printf(format:"Productos: \n");
    mostrarProductos();
    System.out.printf(format:"Total: %6.2f \n", getTotal());
}
}

```

(Se hace la respectiva prueba en nuestro main Java donde vamos a crear un objeto de cada unas de las clases creadas y vamos a agregar diferentes valores y así mismo utilizar las funciones para que estos se agreguen y se muestren como debe de ser)

```
import Clases.Fecha;
import Clases.Producto;
import Clases.Cliente;
import Clases.Factura;

public class App {
    Run main | Debug main | Run | Debug
    public static void main(String[] args) throws Exception {
        System.out.println(x:"Hello, World!");
    }

    public class Main {
        Run main | Debug main
        public static void main(String[] args) {
            Fecha hoy = new Fecha(d:20,m:10,a:2011);
            Producto pro1 = new Producto(c:1, d:"Cafe", (float) 8.5);
            Producto pro2 = new Producto(c:2, d:"Media Luna", p:2);
            Cliente cliente = new Cliente(c:1, n:"Juana");
            Factura f1 = new Factura(t:'F', n:1, hoy, cliente);
            f1.agregarProducto(pro1);
            f1.agregarProducto(pro2);
            f1.mostrar();
        }
    }
}
```

