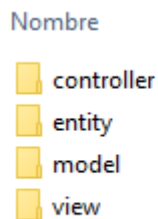


Estructura de Datos y Programación Orientada a Objetos

Semana 11

Crear la siguiente estructura de carpetas en la carpeta Semana_11:



1. Cree la clase **Alumno** con los atributos privados: código, nombre, nota1 y nota2; un constructor que inicialice los atributos privados, los métodos de acceso público set/get y el método calcularPromedio.

Cree la clase **ColeccionAlumnos** con un atributo privado tipo lista e implemente:

- Un constructor que inicialice el atributo tipo lista (vacía).
- Método agregar que reciba un nuevo objeto **Alumno** y lo adicione a la lista.
- Método longitud que retorne la cantidad de objetos **Alumno** registrados en la lista hasta ese momento.
- Método obtener que reciba una posición y retorne el objeto **Alumno** respectivo.
- Método buscar que reciba un código y retorne la posición del objeto **Alumno** que lo contiene. En caso no existe retorne -1.
- Método eliminar que retire un objeto **Alumno** de la lista
- Método modificar que modifique los atributos de un objeto **Alumno** de la lista.

En el formulario:

- Implemente el método listar que muestre todos los objetos **Alumno** de **ColeccionAlumnos**
- Implemente el mantenimiento de alumnos: Listar, Agregar, Editar, Eliminar.

El formulario se titula "Alumno". En la parte superior, hay campos de entrada para "Código", "Nombre", "Nota 1" y "Nota 2". Los campos "Nota 1" y "Nota 2" tienen controles de spin (flechas arriba y abajo) y el valor "0". A la derecha de estos campos hay dos botones: "Grabar" y "Eliminar". Debajo de estos campos, hay una tabla con las siguientes columnas: "Código", "Nombre", "Nota 1", "Nota 2" y "Promedio". La tabla está actualmente vacía. En la parte inferior de la tabla, hay una barra de desplazamiento horizontal.

Propiedades de la tabla:

- `alternatingRowColors`: checked
- `selectionMode`: `SingleSelection`
- `selectionBehavior`: `SelectRows`

<code>alternatingRowColors</code>	<input checked="" type="checkbox"/>
<code>selectionMode</code>	<code>SingleSelection</code>
<code>selectionBehavior</code>	<code>SelectRows</code>

2. Cree la clase **Celular** con los atributos privados: código, marca, modelo y precio; un constructor que inicialice los atributos privados y los métodos de acceso público set/get.

Cree la clase **ColeccionCelulares** con un atributo privado tipo lista e implemente:

- Un constructor que inicialice el atributo tipo lista (vacía).
- Método agregar que reciba un nuevo objeto **Celular** y lo adicione a la lista.
- Método longitud que retorne la cantidad de objetos **Celular** registrados en la lista hasta ese momento.
- Método obtener que reciba una posición y retorne el objeto **Celular** respectivo.
- Método buscar que reciba un código y retorne la posición del objeto **Celular** que lo contiene. En caso no existe retorne -1.
- Método eliminar que retire un objeto **Celular** de la lista
- Método modificar que modifique los atributos de un objeto **Celular** de la lista.

En el formulario:

- Implemente el método listar que muestre todos los objetos **Celular** de **ColeccionCelulares**
- Implemente el mantenimiento de celulares: Listar, Agregar, Editar, Eliminar.

3. Cree la clase **Video** con los atributos privados: código, película, precio, unidades; un constructor que inicialice los atributos privados, los métodos de acceso público set/get y el método calcularImporte (precio * unidades).

Cree la clase **ColeccionVideos** con un atributo privado tipo lista e implemente:

- Un constructor que inicialice el atributo tipo lista (vacía).
- Método agregar que reciba un nuevo objeto **Video** y lo adicione a la lista.
- Método longitud que retorne la cantidad de objetos **Video** registrados en la lista hasta ese momento.
- Método obtener que reciba una posición y retorne el objeto **Video** respectivo.
- Método buscar que reciba un código y retorne la posición del objeto **Video** que lo contiene. En caso no existe retorne -1.
- Método eliminar que retire un objeto **Video** de la lista
- Método editar que modifique los atributos de un objeto **Video** de la lista.

En el formulario:

- Implemente el método listar que muestre todos los objetos **Video** de **ColeccionVideos**
- Implemente el mantenimiento de videos: Listar, Agregar, Editar, Eliminar.