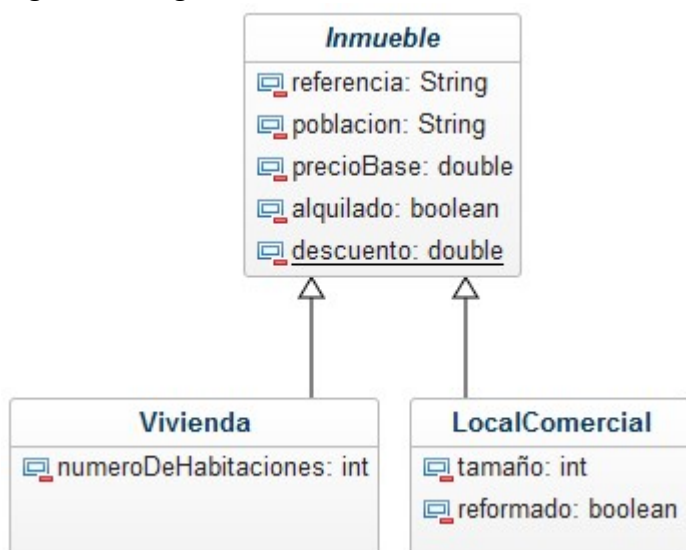


PROGRAMACIÓN JAVA EJERCICIO HERENCIA Y POLIMORFISMO 7.

Este ejercicio es una variación del ejercicio Básico Herencia - 6. Se utilizan las mismas clases con las modificaciones que se indican en el enunciado.

Dado el siguiente diagrama de clases:



Crea un proyecto que simule algunos aspectos básicos de la gestión de una pequeña empresa dedicada al alquiler de inmuebles. Mediante un menú se darán de alta inmuebles, se realizarán alquileres y devoluciones, y se mostrarán los datos según distintos criterios de ordenación.

Como estructura de datos se utilizará un *ArrayList* de tipo *Inmueble* donde se almacenarán los inmuebles (viviendas y locales comerciales) que tiene la empresa para alquilar.

Clase Inmueble

Clase abstracta.

Contiene los atributos:

referencia : String que identifica al inmueble.

población : String que contiene la población donde se ubica el inmueble.

precioBase : precio inicial de alquiler del inmueble.

alquilado : boolean. Si el local está alquilado su valor es *true* y *false* en caso contrario. Cuando se crea un Inmueble su valor será *false*.

descuento : Atributo **static** de tipo double. Contiene el % de descuento que se le aplica al inmueble. Su valor inicial será 0. Mediante la opción de menú correspondiente se puede modificar su valor.

La clase contiene el método abstracto *calcularPrecio()*:

```
public abstract double calcularPrecio();
```

Escribe en la clase todos los constructores y métodos que consideres adecuados.

Clase Vivienda

Contiene el atributo *numeroDeHabitaciones* de tipo int.

Escribe en la clase todos los constructores y métodos que consideres adecuados.

Esta clase deberá implementar el método *calcularPrecio* heredado de *Inmueble*.

El precio de alquiler de la vivienda se calcula de la siguiente forma:

Si la vivienda tiene más de 3 habitaciones se suman 200€ al precio base.
Al precio obtenido se le aplica el descuento que indique la variable static *descuento*.
El método devuelve el precio final de alquiler de la vivienda.

Clase LocalComercial

Contiene los atributos:

tamaño : indica el tamaño en metros cuadrados del local.

reformado : boolean. Si el local está reformado su valor es *true* y *false* en caso contrario.

Escribe en la clase todos los constructores y métodos que consideres adecuados.

Esta clase deberá implementar el método *calcularPrecio* heredado de Inmueble.

El precio de alquiler de un local comercial se calcula de la siguiente forma:

Si el local está reformado se suman 100€ al precio base.

Si el tamaño del local es superior a 200 m² se suman 100€ al precio base.

Al importe obtenido se le aplica el descuento que indique la variable static *descuento*.

El método devuelve el precio final de alquiler del local.

Clase Menu

El programa utilizará una clase Menu. El menú tendrá las siguientes opciones:

```
1-> Nuevo inmueble      2-> Modificar descuento
3-> Alquilar            4-> Devolver
5-> Mostrar inmuebles   0-> FIN
Introduzca opción:
```

Clase Principal

En la clase principal se mostrará el menú y se ejecutará la opción correspondiente.

El programa finaliza cuando se introduzca la opción 0.

Métodos de la clase principal correspondientes a cada opción de menú:

Opción 1: se ejecutará el método *nuevoInmueble()*

En este método se da de alta un nuevo inmueble, vivienda o local comercial. Se introducen los datos del inmueble por teclado y se añade al ArrayList de inmuebles.

Debemos tener en cuenta que la referencia de un inmueble debe tener una longitud mínima de 4 caracteres y máxima de 12 caracteres y estará formada únicamente por letras y números.

Además la referencia es un valor que no se puede repetir.

Si en el ArrayList ya hay un inmueble con esa referencia o la longitud no es correcta se muestra un mensaje indicándolo y se vuelve al menú principal.

El valor del atributo *alquilado* no se pide por teclado. Inicialmente su valor es *false*.

Opción 2: se ejecutará el método *modificarDescuento()*

Este método es el encargado de modificar el valor del atributo static porcentaje de descuento.

Se muestra el valor actual y se pide que se introduzca el nuevo valor.

Opción 3: se ejecutará el método *alquilarInmueble()*

Se elige un inmueble al azar de entre todos los existentes y se alquila. Alquilar un inmueble supone cambiar a *true* el valor del atributo *alquilado*. Si el inmueble elegido ya estaba alquilado se vuelve a elegir otro.

El método mostrará por pantalla inmueble que se ha alquilado.

Si todos los inmuebles están alquilados es ese momento se muestra un mensaje indicándolo y se vuelve al menú principal.

Opción 4: se ejecutará el método *devolverInmueble()*

Se elige un inmueble al azar de entre todos los existentes y se devuelve. Devolver un inmueble supone cambiar a *false* el valor del atributo *alquilado*. Si el inmueble elegido no estaba alquilado se vuelve a elegir otro.

El método mostrará por pantalla inmueble que se ha devuelto.

Si no hay inmuebles alquilados es ese momento se muestra un mensaje indicándolo y se vuelve al menú principal.

Opción 5: se ejecutará el método *mostrarInmuebles()*

Se muestran todos los inmuebles según distintos criterios.

Los inmuebles se muestran en el siguiente orden:

1º Se muestran todos los inmuebles ordenados alfabéticamente por población.

2º Se muestran todas las viviendas que no están alquiladas ordenados por precio final **de menor a mayor**.

3º Se muestran todos los locales que no están alquilados ordenados por precio final **de mayor a menor**.

Si no hay inmuebles para mostrar en cualquiera de los puntos anteriores se mostrará el mensaje correspondiente.

Los inmuebles se mostrarán con este formato:

VIVIENDA AB123 ALBATERA PRECIO BASE: 500,00€ HABITACIONES: 4 DISPONIBLE PRECIO ALQUILER: 630,00€

LOCAL QWE94 ELCHE PRECIO BASE: 1900,00€ SUPERFICIE: 150m2 NO REFORMADO ALQUILADO PRECIO ALQUILER: 1710,00€

En el ejemplo el porcentaje de descuento aplicado al precio ha sido del 10%.

Además de estos métodos puedes escribir en la clase principal todos los métodos que consideres necesarios.