

1. IMPLEMENTACIÓN

- a. Implementar un paquete `PaqViviendas` con una clase `Unifamiliar` que contiene lo siguiente:
 - un atributo `double precio`;
 - un atributo `double númerodeplantas`;
 - un único constructor en donde se inicializan los atributos;
 - un método `double impuesto()`; El impuesto se calcula como el 20% del precio.
 - el método `compareTo` de la interfaz `Comparable` de Java, comparando según `impuesto()` y nombre de la clase.
- b. Crea en el mismo paquete una clase abstracta `Vivienda` a partir de la clase `Unifamiliar` y elimina y redefine de esta última clase todo lo que consideres necesario.
- c. Implementar en el mismo paquete las siguientes clases, heredando según se indica. Todas deben tener un único constructor que inicialice todos sus atributos.
 - `Chalet`, que extiende `Unifamiliar`, con un atributo propio `double metrosjardin`.
 - `CasaDeCampo`, que extiende `Vivienda`, con los atributos propios `double metrosparcela` y `Chalet chalet`. El impuesto es el del chalet mas 0.5 euros por metro de parcela.
 - `Piso` que extiende `Vivienda` con un atributo propio `boolean terraza`. El impuesto del piso se calcula multiplicando los metros por 1.5 euros.
- d. Implementar la clase `Inmobiliaria`, que debe tener:
 - un vector de viviendas como atributo privado;
 - un constructor que crea el vector con `n` viviendas donde `n` se pasa como parámetro (pero no se incluye ninguna vivienda);
 - un método para incluir una vivienda (suponemos que cabe);
 - un método para saber cuántas viviendas unifamiliares no chalets hay en la inmobiliaria.
- e. Poner un ejemplo de método `main`, suponiéndolo fuera del paquete `viviendas`, donde
 - se crea una inmobiliaria con 3 viviendas: un chalet, un unifamiliar y una casa de campo (inventa cualquier valor necesario);
 - se muestra cuántos unifamiliares no chalets hay en la inmobiliaria; y
 - se muestra la vivienda con mayor impuesto (implementa `toString()` para cada vivienda).

2. CUESTIONES

- a. Una clase A tiene como único constructor `public A(String n)`, y es extendida por otra clase B. ¿Qué debe contener y dónde, obligatoriamente, cualquier constructor de B?

Respuesta:



- b. Supongamos que la clase `Bedel` extiende a `Empleado`, y que `Empleado` extiende a `Persona`. Por otro lado la clase `Alumno` extiende a `Persona` y contiene el atributo público no heredado `int curso`. Indicar qué sentencias son incorrectas, y si pueden corregirse mediante *casting*:

```
Persona p = new Alumno();  
int c = p.curso;  
Alumno a = p;  
Bedel b = new Bedel();  
Alumno a2 = (Alumno) b;  
int x = new Alumno().curso;
```

Respuesta: indícala junto a cada sentencia.

- c. Supongamos que A es una clase que contiene el método no privado `m()`, y que la clase B extiende a la clase A. Si en B queremos llamar al método `m()`, ¿qué diferencia hay entre hacerlo como `super.m()` o como `this.m()`?

Respuesta:



- d. Poner un ejemplo de declaración del método `equals` *sobreescribiéndolo* y otro *sobrecargándolo* (sin implementarlo) dentro de una determinada clase A. Si hacemos una llamada `x.equals(y)`, con x de clase A, unas veces se ejecutará una versión y otras veces la otra. ¿De qué depende?

Respuesta:



