

Programación Orientada a Objetos – Curso 2024-25

Práctica 4 – Herencia. Clases Product, Computer y Tv.

Cuando hayas terminado por completo la práctica 3, haz una copia del directorio poo/p3 en poo/p4. En moodle encontrarás los siguientes ficheros que tendrás que descargar y añadir al directorio marketplace/tests:

- product-text.cc
- tv-test.cc
- computer-test.cc
- CMakeLists.txt

Este fichero CMakeLists.txt es para los test, por tanto, pon # delante de las líneas de los tests de product, tv y computer (deja los tests de person tal y como están) y las irás descomentando conforme vayas terminando ejercicios.

Recuerda ejecutar en el directorio 'build' la orden:

```
$ cmake .
```

cada vez que cambies algún fichero CMakeLists.txt y después ya solo será necesario ejecutar:

```
$ make
```

EJERCICIO 1. La clase *Product*

La clase Product irá en el directorio src/product de marketplace. Copia el fichero CMakeLists.txt de src/person a src/product y realiza sobre él los cambios pertinentes.

La clase Product representa un producto a la venta en el marketplace. Tiene los siguientes atributos:

- id: string con el id del producto
- name: string con el nombre
- price: float con el precio en euros
- maker: string con el nombre del fabricante o la marca del producto
- seller: string con el id del vendedor del producto

Utiliza los ficheros product.h y product.cc para codificar esta clase y los métodos:

- Constructor que recibe como parámetros todos los atributos anteriores en ese orden, siendo id el único obligatorio. El resto de parámetros tendrán como valor por defecto "empty", los que son de tipo string, y 0.0 el parámetro de tipo float.
- Observadores (Get) y modificadores (Set) para todos sus atributos.

Hacer un sencillo programa principal para probar la clase Product que se llame 'product-main.cc' que declare un objeto de tipo Product y muestre sus datos en pantalla.

EJERCICIO 2. La clase *Tv*

La clase Tv irá en el directorio src/tv de marketplace. Copia el fichero CMakeLists.txt de src/person a src/tv y realiza sobre él los cambios pertinentes.

La clase Tv irá en el directorio src/tv, y representa una televisión que es un producto más a la venta en el marketplace. Por tanto, hereda de la clase Product y le añade un atributo 'inch_' de tipo float con las pulgadas del televisor.

Utiliza los ficheros tv.h y tv.cc para codificar esta clase y los métodos:

- Constructor que recibe como parámetros todos los atributos en el mismo orden que en la clase Product, siendo id el único obligatorio y siendo las pulgadas el último. El resto de

parámetros tendrán como valor por defecto los mismos que en la clase *Person* y el parámetro con las pulgadas tendrá como valor por defecto 0.0.

- Método *GetInch()*: observador .
- Método *SetInch()*: recibe un parámetro con el nuevo valor de las pulgadas del televisor. Si el valor recibido es mayor que 0, se modifica y devuelve true. En caso contrario, no se modifica el dato y devuelve false.

Hacer un sencillo programa principal para probar la clase *Tv* que se llame ‘tv-main.cc’ que declare un objeto de tipo *Tv* y muestre sus datos en pantalla.

Para hacer el fichero *CMakeLists.txt* de *Tv*, copia por ejemplo el de *Product*. Ten en cuenta que como la clase *Tv* utiliza la clase *Product*, tendrás que añadir la siguiente línea después de *add_library(tv tv.cc tv.h)*:

```
target_link_libraries(tv PUBLIC product)
```

En lo sucesivo, ten en cuenta estas dependencias de cada nuevo módulo.

EJERCICIO 3. La clase *Computer*

La clase *Computer* irá en el directorio *src/computer*, y representa un ordenador a la venta como producto en el marketplace y, por tanto, hereda de la clase *Product* y le añade un atributo ‘type_’ de tipo ‘*ComputerType*’ que será un enumerado (enum class) que habrás definido previamente a la declaración de la clase con los siguientes 5 valores:

1. Desktop
2. Laptop
3. Server
4. Tablet
5. Gaming

Utiliza los ficheros *computer.h* y *computer.cc* para codificar el dato enumerado, la clase *Computer* y los métodos de la clase *Computer* siguientes:

- Constructor que recibe como parámetros todos los atributos de *Product* y el tipo de ordenador en el siguiente orden: id, type, name, price, maker, seller. Son obligatorios id y type. Los parámetros tendrán como valor por defecto “empty” los que son string y 0.0 el dato de tipo float.
- Observadores (Get) y modificadores (Set) para todos sus atributos.

Hacer un sencillo programa principal para probar la clase que se llame ‘computer-main.cc’ que declare un objeto de tipo *Computer* y muestre sus datos en pantalla.

Para hacer el fichero *CMakeLists.txt* de *Computer*, copia por ejemplo el de *TV* y realiza los cambios pertinentes.

EJERCICIO 4. *Testing*

1. Compila y ejecuta todos los programas principales que has creado
2. Ejecuta los tests de *Product*, *Tv* y *Computer* hasta que todos pasen correctamente. Puedes comprobar que ejecutando simplemente ‘ctest’ en el directorio build se ejecutan todos los tests a la vez.