

# Programación Orientada a Objetos – Curso 2024-25

## Práctica 6 – Marketplace. Herencia múltiple. Las clases: Client y Seller

Seguimos añadiendo librerías y funcionalidad a nuestro marketplace.

En moodle encontrarás los siguientes ficheros que tendrás que descargar al directorio marketplace/tests:

- client-test.cc
- seller-test.cc

Crea el directorio poo/p6 con una copia del directorio poo/p5.

### **EJERCICIO 1** La clase Client

La clase *Client* representa a una persona que compra productos en el marketplace, por tanto deriva de forma public de la clase *Person*.

Además, un cliente tendrá también la funcionalidad de poder añadir y eliminar productos de su cesta de la compra. Para añadir esta funcionalidad a *Client* también debe derivar de forma public de la clase *Basket*.

Por tanto, el inicio de la declaración de la clase *Client* será de esta forma:

```
class Client: public Person, public Basket{...
```

Además, un cliente tiene además un campo 'premium\_' de tipo entero que indica el tipo de cliente que usaremos en futuras funcionalidades del marketplace.

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

- Constructor que recibe los mismos parámetros que el constructor de la clase *Person* (y se los pasa con un iniciador base) más el parámetro 'premium' con valor por defecto igual a cero que se asignará a 'premium\_'.
- Observador GetPremium().
- Modificador SetPremium().

Hacer un sencillo programa principal para probar la clase *Client* que se llame 'client-main.cc' que declare un objeto de tipo *Client* y muestre algún dato en pantalla.

### **EJERCICIO 2.** La Clase Seller

La clase *Seller* representa a una persona que vende productos en el marketplace, por tanto deriva de forma public de la clase *Person*.

Además, un vendedor tendrá también la funcionalidad de poder añadir y eliminar productos de su cesta, en este caso se trata de la cesta de productos que tiene a la venta. Para añadir esta funcionalidad a *Seller* también debe derivar de forma public de la clase *Basket*.

Por tanto, el inicio de la declaración de la clase *Seller* será de esta forma:

```
class Seller: public Person, public Basket{...
```

Además, un vendedor tiene un campo 'sector\_' de tipo std::string que almacena una cadena que describe el tipo de productos que vende, por ejemplo: electrónica e informática, libros, textil, deportes, etc.

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

- Constructor que recibe los mismos parámetros que sus clases base más el parámetro ‘sector’ con valor por defecto igual a la cadena “empty” que se asignará a ‘sector\_’.
- Observador GetSector().
- Modificador SetSector().

Hacer un sencillo programa principal para probar la clase Seller que se llame ‘seller-main.cc’ que declare un objeto de tipo Seller y muestre algún dato en pantalla.

NOTA: tendrás que modificar los correspondientes ficheros CMakeLists.txt de CMake. Para ello usa como modelo el resto de librerías que ya hemos hecho antes para este proyecto.