

TC1030 Programación Orientada a Objetos

Evaluación Práctica

Aplicación de captura de pedidos de Bugsracer Cafe

Objetivo: Capturar un pedido de Bugsracer Cafe en base a una lista de platillos, y desplegar el contenido del pedido.

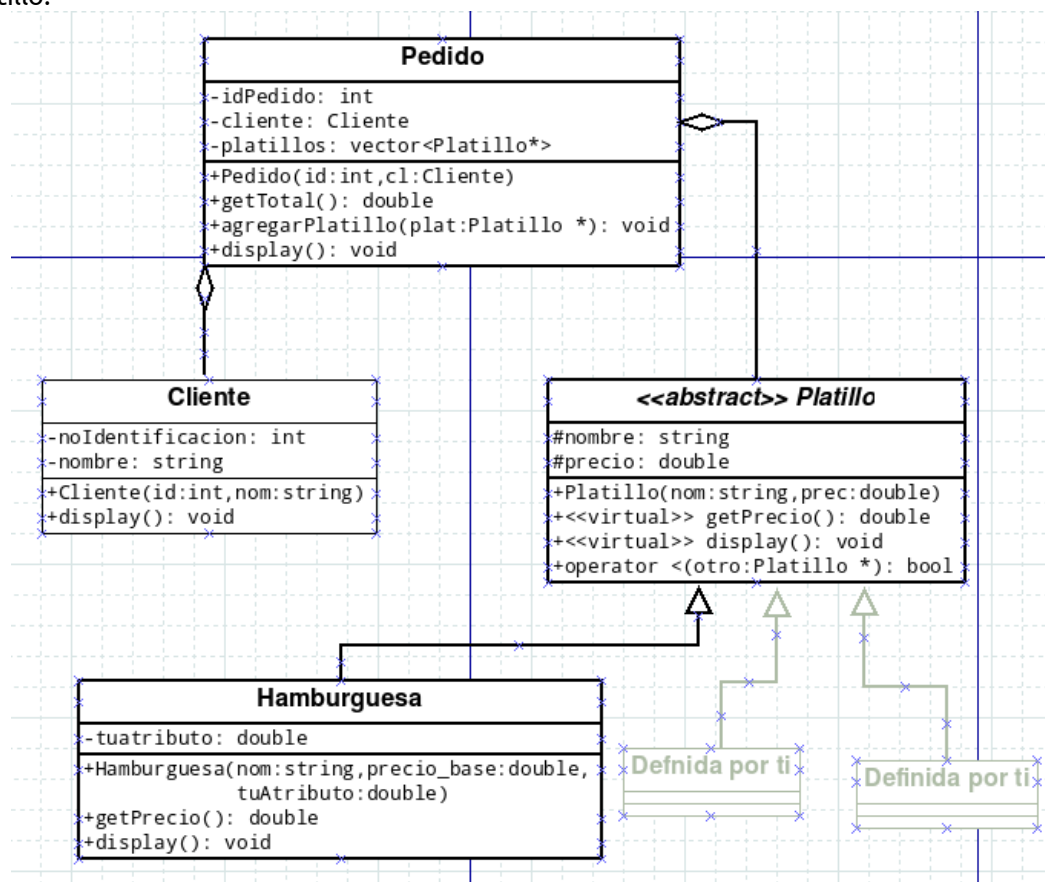
Requerimientos técnicos:

El programa se realizará en C++

Cada clase deberá estar con su archivo .cpp y .h, así como un método main que organize todo



- Debes realizar otras 2 clases de cualquier otro platillo, parecida a la clase Hamburguesa. El atributo propio de la clase derivada te va ayudar a calcular el precio del platillo sea diferente en las tres clases, de tal manera que el método **getPrecio()** sea único en cada clase derivada de Platillo.



Debe de explicar que hace el atributo `tu_atributo` como comentario en la clase

En el método main()

- Crea manualmente 6 o más platillos diferentes, 2 de cada una de las tres clases derivadas de la clase Platillo.
- Crea un cliente
- Crea un pedido
- Agrega uno por uno los platillos al pedido
- Muestra la información del pedido y el costo total del pedido

Esta es una posible salida esperada de su programa:

```
DESPLIEGUE PROPUESTO DEL EXAMEN FINAL
-----
Cliente : El Paisa con id : 101

Información del Pedido número 563
1 : Hamburguesa sencilla y cuesta : 100 pesos
2 : Hamburguesa doble y cuesta : 115 pesos
3 : Hamburguesa triple y cuesta : 130 pesos
4 : Segundo Platillo y cuesta : 120 pesos
5 : Segundo platillo (b) y cuesta : 144 pesos
6 : Tercer Platillo y cuesta : 105 pesos
7 : Tercer platillo (b) y cuesta : 70 pesos

El costo total de tu orden es : 784 pesos
```

Cliente	
Atributos	
idcliente : int	Guardará un ID numérico asignado al cliente
nombre : string	Guardará el nombre del cliente
Métodos	
Cliente()	Constructor de la clase sin parámetros
Cliente(id : int , nom : string)	Constructor de la clase con parámetros, inicializa los atributos
display() : void	Desplegará todos los datos del Cliente
Platillo	
ESTA ES UNA CLASE ABSTRACTA	
Atributos	
Nombre : str	Guardará el nombre del Platillo
precio : double	Guardará el precio base de todos los platillos, en todos los platillos debe ser el mismo
Métodos	
Platillo(nom : string , precio : double)	Constructor de la clase
getPrecio() : double	METODO ABSTRACTO (PURO VIRTUAL)
operator < (other:Platillo &) : bool	Para saber si el precio es menor al precio del "Platillo" recibido en el parámetro.
display() : void	METODO ABSTRACTO (PURO VIRTUAL)
Hamburguesa	
Hereda de Platillo	
Atributos	
tu_atributo : int	Atributo definido por ti que sirva para calcular el precio de la hamburguesa, en este caso es el número de "carne" que tiene, por cada extra aumenta un 15% el precio
Métodos	
Hamburguesa(nom : str , precio : double, mi_atributo : int)	Constructor de la clase
getPrecio() : double	Método para calcular el precio con el atributo precio que se heredó de la clase Platillo más el atributo agregado por ti.
display() : void	Desplegará todos los datos de la Hamburguesa, diciendo que es una Hamburguesa, así como su precio (usando el precio base y tu atributo)
Pedido	
Atributos	
idPedido : int	Guardará un ID numérico asignado al cliente
miCliente : Cliente	Objeto tipo Cliente
platillos : vector <Platillo *>	Vector de platillos que almacenará todos los platillos del Pedido
Métodos	
Pedido(idPed : int, cli : Cliente)	Constructor de la clase, inicializa los atributos de la clase
getTotal() : double	Obtener el precio TOTAL del Pedido
agregarPlatillo (platillo: Platillo *) : void	Método que agrega los platillos al Pedido
display() : void	Desplegar TODOS los platos del Pedido