

Ejercicios POO Ivl 2

Ejercicio 1: Pizzería

Una pizzería de la ciudad ofrece a sus clientes una amplia variedad de pizzas de fabricación propia, de varios tamaños (8, 10 y 12 porciones).

Los clientes tienen a disposición un menú que describe para cada una de las variedades, el nombre, los ingredientes y el precio según el tamaño y el tipo (a la piedra, a la parrilla, de molde) de la pizza. Los clientes realizan sus pedidos en el mostrador. El pedido debe contener el nombre del Cliente, para llamarlo cuando su pedido está listo; la cantidad de pizzas, el tamaño, la variedad, la fecha del pedido, la hora en la que el pedido debe entregarse y la demora estimada informada al cliente. El pedido va a la cocina y cuando está preparado se informa al que lo tomó para que se genere la factura correspondiente y se le entregue el pedido al cliente.

El dueño de la pizzería ha manifestado la necesidad de acceder al menos a la siguiente información:

- Variedades y tipos de pizzas más pedidas por los clientes.
- Ingresos (recaudaciones) por períodos de tiempo.
- Pedidos (cantidad y monto) por períodos de tiempo.

Ejercicio 2: Supermercado

Un supermercado maneja el catálogo de los productos que vende. De cada producto se conoce su nombre, precio, y si el mismo es parte del programa Precios Cuidados o no. Por defecto, el producto no es parte del programa, a menos que se especifique lo contrario.

Para ayudar a los clientes, el supermercado quiere realizar descuentos en productos de Primera Necesidad. Es por eso que al calcular el precio de un producto de Primera Necesidad, se aplica un descuento del 10%. Es decir:

$$\text{precioProductoPrimeraNecesidad} = \text{precioBaseDelProducto} * 0.9$$

El supermercado, del cual se conoce el nombre y la dirección, desea conocer la cantidad total de productos que comercializa y la suma total de los precios de los mismos.

Suponga ahora que el descuento a aplicar en cada producto de primera necesidad puede ser distinto y se debe poder definir al momento de crear el mismo. Por ejemplo, el arroz puede ser un producto de primera necesidad con un descuento del 8%, mientras que leche podría ser otro producto de primera necesidad con un descuento del 11%. Esto es sólo un ejemplo. El descuento a aplicar en cada producto de primera necesidad debe ser configurable al momento de crearlo.

- Implementar un nuevo programa basado en el anterior que incorpore este nuevo requerimiento.

Ejercicio 3: Barajas

Vamos a crear una baraja de cartas españolas orientado a objetos.

Una carta tiene un número entre 1 y 12 (el 8 y el 9 no los incluimos) y un palo (espadas, bastos, oros y copas)

La baraja estará compuesta por un conjunto de cartas, 40 exactamente.

Las operaciones que podrá realizar la baraja son:

- **barajar**: cambia de posición todas las cartas aleatoriamente
- **siguienteCarta**: devuelve la siguiente carta que está en la baraja, cuando no haya más o se haya llegado al final, se indica al usuario que no hay más cartas.
- **cartasDisponibles**: indica el número de cartas que aún puede repartir
- **darCartas**: dado un número de cartas que nos pidan, le devolveremos ese número de cartas (piensa que puedes devolver). En caso de que haya menos cartas que las pedidas, no devolveremos nada pero debemos indicárselo al usuario.
- **cartasMonton**: mostramos aquellas cartas que ya han salido, si no ha salido ninguna indicárselo al usuario
- **mostrarBaraja**: muestra todas las cartas hasta el final. Es decir, si se saca una carta y luego se llama al método, este no mostrara esa primera carta.

Ejercicio 4: Cuenta Joven

Vamos a definir ahora una "Cuenta Joven", para ello vamos a crear una nueva clase **CuentaJoven** que deriva de lo que definas al resolver el Ejercicio 6: Cuentas Electrónicas de los ejercicios Ivl 1.

Cuando se crea esta nueva clase, además del titular y la cantidad se debe guardar una bonificación que estará expresada en tanto por ciento. Construye los siguientes métodos para la clase:

- En esta ocasión los titulares de este tipo de cuenta tienen que ser mayor de edad, por lo tanto hay que crear un método **esTitularValido()** que devuelve verdadero si el titular es mayor de edad pero menor de 25 años y falso en caso contrario.
- Además la retirada de dinero sólo se podrá hacer si el titular es válido.
- El método **mostrar()** debe devolver el mensaje de "Cuenta Joven" y la bonificación de la cuenta.

Piensa los métodos heredados de la clase madre que hay que reescribir.

Ejercicio 5: Prestamos

Se requiere un programa para registro de préstamos en una cooperativa.

Los datos que se manejan para el préstamo son los siguientes:

- **Número de Préstamo,**
- **Solicitante del préstamo. De quien se requiere únicamente: DNI, Primer Nombre, Primer y Segundo Apellido, teléfono de casa y teléfono móvil.**
- **Valor del préstamo**
- **Fechas de pago de las cuotas (un máximo de 6 fechas, se asume que el plazo máximo de pago son 6 meses).**
- **Fecha de autorización del préstamo.**
- **Fecha tentativa de entrega del préstamo.**

Las reglas que debe respetar este proyecto son las siguientes:

- **El número de préstamo siempre deberá ser un valor mayor que cero.**
- **El valor del préstamo siempre debe ser mayor a cero.**
- **Se debe solicitar los datos de quien toma el préstamo.**
- **La fecha tentativa de entrega del préstamo será siete días después de la fecha de autorización del préstamo.**
- **Las fechas de pago del préstamo se calculan, sumando 30 días a cada una a partir de la fecha de entrega del préstamo.**
- **Los préstamos solo se pueden autorizar en los primeros 20 días del mes. Esta es una política que nunca va a cambiar.**

Además debes tener en cuenta que:

- **Existe una fecha máxima para la autorización de los préstamos.**
- **Existe un valor máximo a prestar. La sumatoria de los préstamos que se ingresen no debe exceder este valor.**
- **Debe permitir la carga de tantos préstamos como desee ingresar el usuario, a menos que se haya llegado al valor máximo a prestar.**
- **Debe imprimir los datos completos del préstamo, incluyendo la fecha de entrega y las fechas de pago de las cuotas.**