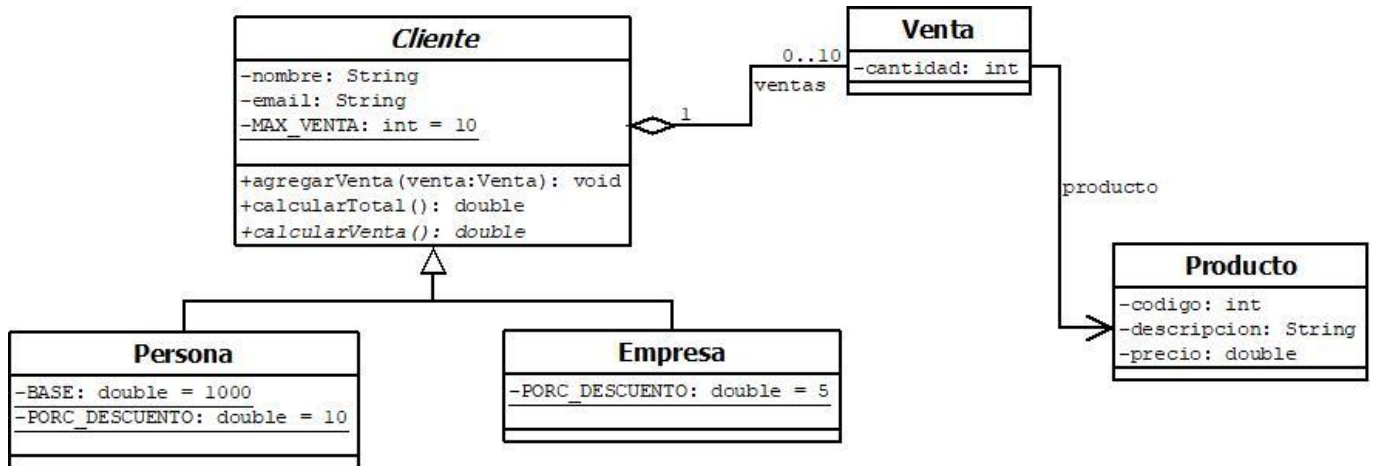


1º RECUPERATORIO DE ALGORITMICA Y ESTRUCTURA DE DATOS II - 2021

1) Desarrollar un sistema para calcular las ventas realizadas a los clientes de determinados productos. Si el cliente es una Empresa se realiza un descuento de 5% sobre el total vendido. Si el cliente es una Persona se realiza un descuento de un 10% sobre el total vendido solo si el mismo supera los 1000 pesos. Los clientes pueden comprar hasta diez productos.

Diagrama de Clases



Crear todas las clases con sus atributos y constructores. El método `calcularTotal()` retorna la sumatoria de todos los productos por la cantidad vendida a un cliente sin realizar ningún tipo de descuento. El método `calcularVenta()` retorna la sumatoria de todos los productos por la cantidad vendida a un cliente realizando el descuento correspondiente de acuerdo al tipo de cliente.

2) Realizar un programa de prueba donde cree un arreglo de clientes y los productos vendidos.

- a) Recorrer el arreglo mostrando el nombre y el total a facturar para cada cliente
- b) Mostrar la sumatoria de todas las ventas realizadas

3) Dada una lista **doblemente enlazada**, agregar el siguiente método y realizar un programa que pruebe los diferentes casos que se pueden presentar (lista vacía, lista con el elemento a buscar repetido, lista donde no está el elemento a buscar, etc.)

```
/*
 * Retorna el índice de la última ocurrencia del elemento especificado en la
 * lista, o -1 si la lista no contiene el elemento
 *
 * Por ejemplo, para una lista que contiene los siguientes elementos:
 *
 * (rojo, azul, blanco, azul, azul, amarillo, verde)
 *
 * si el elemento buscado en la lista es: azul
 *
 * el método retorna el valor: 5
 */
public int lastIndex(E e)
```

4) Dada una lista **doblemente enlazada**, agregar el siguiente método y realizar un programa que pruebe los diferentes casos que se pueden presentar (recibe como argumento una lista vacía, una lista igual a la que se tiene, una lista con elementos distintos, etc.)

```
/*
 * Retiene en la lista todos los elementos que están en "list" y elimina lo que
 * no están en la misma.
 *
 * Considere implementar un método para buscar elementos (search) que se invoca
 * desde retainAll para hacer el código más claro
 *
 * Por ejemplo, para una lista que contiene los siguientes elementos:
 *
 * (rojo, azul, blanco, azul, azul, amarillo, verde)
 *
 * se pide retener los elementos de la siguiente lista:
 *
 * (rojo, azul, amarillo)
 *
 * la lista resultante es:
 *
 * (rojo, azul, azul, azul, amarillo)
 */
public void retainAll(DoublyLinkedList<E> list)
```

IMPORTANTE:

1. Los enunciados no deberían dejar lugar a dudas de los ejercicios a resolver y preguntas a responder. De todas maneras si surge alguna consulta del enunciado enviar la misma al **Foro** que está en la **sección Evaluación**. Las preguntas serán respondidas dentro de los 30 minutos de realizadas.
2. Enviar un WhatsApp al grupo **solo** si tienen algún inconveniente (no pueden entrar al foro, no reciben la respuesta en el tiempo indicado, no pueden subir las soluciones, etc.)
3. Para la parte práctica del parcial, subir solamente los archivos con extensión .java.