



RETO 4 – PROGRAMACIÓN BÁSICA

VARIANTE 2

El restaurante italiano “Ristorante speciale per tutti” es un restaurante que tiene sede en Colombia, y sus productos pretenden traer la gastronomía italiana de la forma más fiel posible y al alcance de todos.

Este restaurante ha tenido un crecimiento exponencial en sus pedidos, por lo que ahora está buscando un modo de gestionarlos de un modo más fácil, el gestor que este restaurante necesita debe permitir conocer datos como:

- Ganancias que dejó una lista de pedidos
- El promedio de las ganancias totales
- La desviación estándar las ganancias totales
- Conocer el costo de un pedido en específico

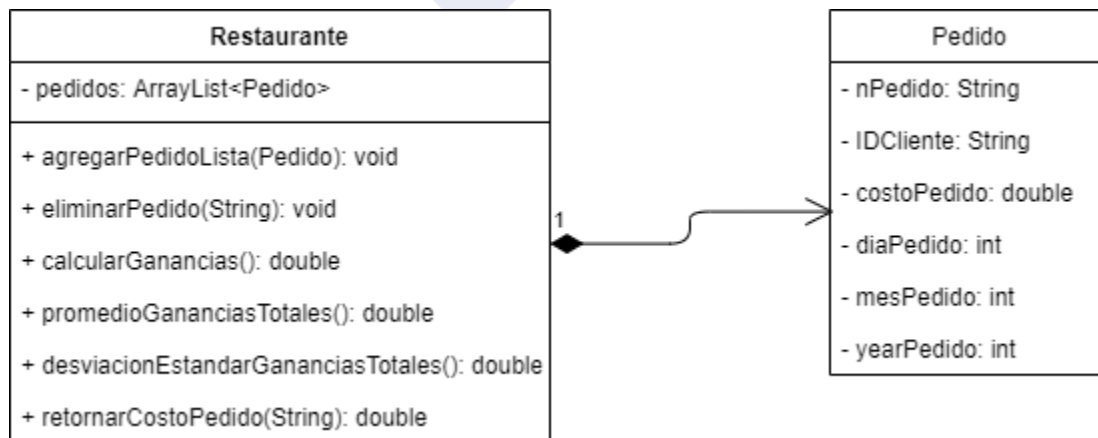
Usted, como tripulante de Misión TIC 2022, ha mostrado un buen desempeño en la realización de los diferentes retos del Ciclo 1 y Ciclo 2 de formación, por lo que ha sido contratado como desarrollador junior, con el motivo de que vaya adquiriendo experiencia en el área de la tecnología.

Usted está encargado de implementar las siguientes clases que ha detallado el área de ingeniería de software de la empresa desarrolladora contratista, para facilitar la codificación de estas clases, el equipo de Ingeniería de software le hace entrega del diagrama de clases (Recuerde que los métodos relacionados a los *getters* y *setters*, así como también el método constructor son obviados en el diagrama de clases, pero deberán ser incluidos en el código, estos métodos deberán ser creados con el estándar camel case).

Hay 1 getter y setter que romperá este estándar:

- Para la variable `nPedido`:
 - `getnPedido()`
 - `setnPedido(String nPedido)`





Además del diagrama, el área de Ingeniería le hace entrega de esta documentación para comprender mejor los elementos del diagrama:

Clase Pedido

Atributos

NOMBRE	TIPO DATO	CONCEPTO	INICIALIZACIÓN
nPedido	String	Guarda el identificador del pedido	En el método constructor
IDCliente	String	Guarda el documento de identidad del cliente	En el método constructor
costoPedido	double	Guarda el costo del pedido	En el método constructor
diaPedido	int	Guarda el día en que se hizo el pedido (Es un número entre 0 y 31)	En el método constructor
mesPedido	int	Guarda el mes en que se hizo el pedido (Es un número entre 1 y 12)	En el método constructor
yearPedido	int	Guarda el año en que se hizo el pedido (Es un número entre 1900 y 3000)	En el método constructor





Clase Restaurante

Atributos

NOMBRE	TIPO DATO	CONCEPTO	INICIALIZACIÓN
pedidos	ArrayList <Pedido>	Guarda la lista de pedidos realizados	Vacío

Métodos

NOMBRE	TIPO RETORNO	PARÁMETROS	CONCEPTO
agregarPedidoLista	void	Pedido p: Es el pedido por agregar a la lista pedidos	Agrega el pedido recibido como parámetro a la lista pedidos si este NO está en la lista (Dos pedidos son diferentes si sus ID son diferentes)
eliminarPedido	void	String nPedido: Número del pedido a eliminar de la lista pedidos	Elimina el pedido con número de pedido igual al recibido por parámetro (nPedido)
calcularGanancias	double	No recibe	Retorna la suma del costo de todos los pedidos
promedioGananciasTotales	double	No recibe	Retorna el promedio del costo de todos los pedidos
desviacionEstandarGananciasTotales	double	No recibe	Retorna la desviación estándar del costo de todos los pedidos





<code>retornarCostoPedido</code> ○	<code>double</code>	String ID: ID del pedido del cual se desea conocer su costo	Retorna el costo del pedido cuyo ID sea igual al ingresado por parámetro
---------------------------------------	---------------------	---	--

PRECISIONES

1. No hay métodos estáticos.
2. Deben existir *getters* y *setters* de todos los atributos de cada clase, estos deben ser escritos en la forma estándar camel case.
3. La fórmula de la desviación estándar a usar es: $\sigma = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \mu)^2}$, donde n es el número de muestras a considerar en el cálculo (Número de pedidos que hay en el ArrayList), x_i es el costo de cada pedido y μ es el promedio del costo del conjunto de pedidos que hay en el ArrayList.
4. El calificador asumirá un error del 1% por aproximaciones que su código pueda hacer en el cálculo del promedio y la desviación estándar.

TAREAS

- En el archivo preconstruído en la plataforma Moodle, codificar las clases especificadas en el diagrama de clases, teniendo en cuenta las precisiones dadas por el equipo de Ingeniería de software.
- Los nombres de los métodos y atributos **DEBEN** ser nombrados tal y como aparecen en el diagrama de clases.
- Usted **NO** debe solicitar datos por teclado, ni programar un método `main`, tampoco use `Java Source Package`, usted está solamente encargado de la construcción de la clase.





NOTA ACLARATORIA

Usted podrá desarrollar la clase requerida en un IDE como NetBeans, y al final copiar y pegar el código en la herramienta VPL, pero **NO** deberá subir archivos, es decir:

Modo incorrecto:

La imagen muestra la interfaz de entrega de una tarea en VPL. En la parte superior, hay una barra de navegación con los logos de 'Misión TIC 2022', 'Ingeniería' y 'UNIVERSIDAD DE ANTIOQUIA'. Debajo, se indica el curso: 'SEM-Programacion basica ciclo 2'. El título de la tarea es 'SEM-Programacion básica ciclo 2'. A la derecha, un texto rojo grande dice 'NO SUBIR NINGÚN ARCHIVO'. En el centro, hay un campo de texto con el texto 'Entrega' y un botón 'Entrega'. A la izquierda del campo de texto, hay un icono de flecha hacia arriba y el texto 'COPIA TU CÓDIGO ACÁ (Como en el Ciclo 1)'. Debajo del campo de texto, hay un botón 'Selecciona un archivo...' y un texto 'Tamaño máximo para archivos nuevos: 5MB'. En la parte inferior, hay un botón 'Enviar' y un botón 'Cancelar'. En la esquina inferior derecha, hay un texto 'VPL'.

Modo correcto:

La imagen muestra la interfaz de entrega de una tarea en VPL. En la parte superior, hay una barra de navegación con los logos de 'Misión TIC 2022', 'Ingeniería' y 'UNIVERSIDAD DE ANTIOQUIA'. Debajo, se indica el curso: 'SEM-Programacion basica ciclo 2'. El título de la tarea es 'SEM-Programacion básica ciclo 2'. En el centro, hay un texto verde grande que dice 'LUGAR CORRECTO'. Debajo del texto, hay un editor de código con el nombre de archivo 'Personaje.java'. El código muestra una clase 'Personaje' con comentarios para insertar atributos, método constructor, métodos (NO LOS GETTER Y SETTERS) y setters y getters. A la derecha del editor, hay un botón 'CLIC AQUÍ PARA ACCEDER AL ENUNCIADO'.

