

Un cliente, requiere un sistema modelado e implementado observando los principios de la POO para registrar y administrar las ventas de paquetes turísticos de una agencia de viajes.

Los datos pertinentes a dicho objeto *Paquete Turístico* son:

- **IMPUESTO:** Número real (punto flotante de precisión simple) – Valor único compartido para todas las instancias del objeto. Representa el porcentaje de incremento sobre el precio que se debe cobrar a los efectos de cubrir las imposiciones tributarias. Ejemplo: 10,5%.
- **Número de Paquete:** Número entero del intervalo [0 a 999]. Ejemplo: 583.
- **Detalle:** Cadena de caracteres descriptiva del paquete turístico.
Ejemplo: "Bariloche – 10 días y 8 noches – Bus – Hotel Pucón – Pensión Completa".
- **Precio:** Número real (punto flotante de precisión simple) que representa el valor por pasajero, sin impuestos y para el paquete. Ejemplo: \$8978,50.

Se pide:

A) Código fuente del archivo de clase *CPaquete.cs* conteniendo:

- 1) Adecuada declaración de las variables miembro.
- 2) Un método *setIMPUESTO(porcentaje)*, que recibiendo por valor como argumento, un valor *porcentaje*, número real (punto flotante de precisión simple), establezca adecuadamente valor para la variable miembro de clase *IMPUESTO*.
- 3) Un método constructor *CPaquete (número, descripción)*, que recibiendo por valor como argumentos, un dato *número*, número entero del intervalo [0 a 999] y un valor *descripción*, cadena de caracteres; cree una instancia de la clase *CPaquete* y establezca valor inicial para las variables miembro *numPaquete* y *Detalle*, respectivamente.
- 4) Un método *setPrecio(monto)*, que recibiendo por valor como argumento, un valor *monto*, número real (punto flotante de precisión simple), establezca valor para la variable miembro *Precio*.
- 5) Un método *getPrecio()*, que sin recibir argumentos retorne, como número real (punto flotante de precisión simple), el valor para la variable miembro *Precio*.
- 6) Un método *darMontoTotal()*, que sin argumento alguno, devuelva como número real (punto flotante de precisión simple), el *MontoTotal* a abonar por pasajero para el paquete, producto de aplicarle el porcentaje de incremento de *IMPUESTO* al valor del *Precio* del paquete (Si *IMPUESTO* es 100%, se abonará un *MontoTotal* igual al doble del *Precio*).
- 7) Un método *MostrarDatos()*, que sin argumento alguno, devuelva una cadena de caracteres que concatene los valores presentes en las variables miembro *numPaquete*, *Detalle*, *Precio*, *IMPUESTO* más el valor calculado de *MontoTotal* a abonar por pasajero para el paquete (emplear método *darMontoTotal()*).
- 8) Un método de instancia *esMásBaratoQue(otroPaquete)*, que recibiendo por valor como argumento, una referencia a una instancia de *CPaquete*, devuelva como dato lógico, el valor *True*, cuando la instancia invocante presente un *Precio* menor que la recibida como argumento y, el valor *False*, en cualquier otro caso, de modo que pueda invocarse como en el siguiente ejemplo:

```
...  
    if (paquete1.esMásBaratoQue(paquete2) == true)  
...
```

B) Código fuente del archivo de clase *CEjecutora.cs* conteniendo el método *Main()* de una aplicación de consola, que permita:

- 1) Solicitar al usuario y establecer el valor de *IMPUESTO* único y compartido para todos los *Paquetes Turísticos*.
 - 2) Consultar al usuario y crear con ellos una instancia de la clase *CPaquete* con datos completos.
 - 3) Informar datos completos de la instancia recientemente creada.
- C) Modificar el programa del método *Main()* para que el usuario ingrese una cantidad indeterminada de paquetes y que el sistema devuelva todos los datos del paquete mas económico.