

Ejercitación de Objetos

Un cliente, requiere un sistema modelado e implementado observando los principios de la POO para registrar y administrar el pago de cuotas de los alumnos de un instituto.

Los datos pertinentes a dicho objeto *Alumno* son:

- *CUOTA*: Número real (punto flotante de precisión simple) – Valor único compartido para todas las instancias del objeto.
- *Legajo*: Número entero del intervalo [0 a 20000]. Identificador unívoco (Clave).
- *Nombre*: Cadena de caracteres que concatena los nombres y apellidos del alumno.
- *Beca*: Número real (punto flotante de precisión simple) del intervalo [0 a 100].

Se pide:

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

- 1) Adecuada declaración de las variables miembro.
- 2) Un método *setNombre(nombApe)*, que, recibiendo por valor como argumento, un valor *nombApe*, cadena de caracteres, establezca adecuadamente valor para la variable miembro *nombre*.
- 3) Un método *setCUOTA(monto)*, que, recibiendo por valor como argumento, un valor *monto*, número real (punto flotante de precisión simple), establezca adecuadamente valor para la variable miembro *CUOTA*.
- 4) Un método constructor *CCuenta(numLegAlumno)*, que recibiendo por valor como argumento, un dato *numLegAlumno*, número entero del intervalo [0 a 20000], establezca valor inicial para la variable miembro *legajo*.
- 5) Un método *setBeca(porcentaje)*, que recibiendo por valor como argumento, un valor *porcentaje*, número real (punto flotante de precisión simple), establezca valor para la variable miembro *beca*.
- 6) Un método *darImporteMensual()*, que sin argumento alguno, devuelva como número real (punto flotante de precisión simple), el *Importe Mensual* a abonar por el alumno, producto de aplicarle a la *CUOTA*, el porcentaje de *Beca* concedido (Si *Beca* es 0%, paga *CUOTA* completa).
- 7) Un método *darDatos()*, que sin argumento alguno, devuelva una cadena de caracteres que concatene los valores presentes en las variables miembro *legajo*, *nombre*, *beca* más el valor calculado de *Importe Mensual* a abonar por el alumno (emplear método anterior).
- 8) Un método *abonaMasImporteQue(otroAlumno)*, que recibiendo por valor como argumento, una referencia a una instancia de *CAumno*, devuelva como dato lógico, el valor *True*, cuando la instancia invocante abone un *Importe Mensual* mayor o igual que la recibida como argumento y, el valor *False*, en cualquier otro caso.

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 *CUOTA* único y compartido para todos los alumnos.
- 2) Solicitar iterativamente y registrar, los datos de un conjunto de extensión desconocida de alumnos; finalizando dicho proceso de carga con el ingreso de un número de *Legajo* de valor igual a 0 (cero), para cuyo caso no deberán solicitarse ni registrarse los datos restantes *Nombre* y *Beca*.
- 3) Informar *Legajo – Nombre – Beca – Importe Mensual* del alumno cuyo *Importe Mensual*, sea el más económico del conjunto ingresado (último de los ingresados en caso de existir más de uno).
- 4) En caso de no haberse ingresado alumno válido alguno, en lugar de lo indicado en el ítem anterior, informar la leyenda “No se ingresaron alumnos.”.

Un cliente, requiere un sistema modelado e implementado observando los principios de la POO para gestionar los viajes de transporte de mercadería de su flota de camiones.

Los datos pertinentes a dicho objeto viaje son:

- Costo por Kilómetro: Número real (flotante de precisión simple)–Valor único compartido para todas las instancias del objeto. Representa el monto a cobrar por cada kilómetro recorrido en los viajes. Ejemplo: \$513,50 el kilómetro.
- Kilometraje Mínimo: Número entero corto sin signo–Valor único compartido para todas las instancias del objeto. Representa el kilometraje mínimo que se cobra para cualquier viaje. Si el viaje realizado es de menor distancia, igualmente se pagará el kilometraje mínimo. Ejemplo: 300 km
- Dominio: Cadena de caracteres que representa la patente del camión. Ejemplo: “MPF153”
- Distancia Recorrida: Número entero corto sin signo, que representa la distancia en kilómetros recorrida en el viaje. Ejemplo: 739 km

Se pide:

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

- 1) Adecuada declaración de las variables miembro.
- 2) Un método setter y un método getter para el costo por kilómetro, de modo tal que este valor se pueda establecer y conocer aún antes de crear cualquier instancia de viaje.
- 3) Un método setter y un método getter para el kilometraje mínimo, de modo tal que este valor se pueda establecer y conocer aún antes de crear cualquier instancia de viaje.
- 4) Un constructor por defecto para el viaje, que asigne inicialmente al dominio, el valor “Sin Establecer”.
- 5) Un método constructor parametrizado, con los argumentos dominio y distancia recorrida.
- 6) Un método setter y un método getter para el dominio.
- 7) Una propiedad de lectura y escritura Kilometraje, que opere con la variable miembro de la distancia recorrida.
- 8) Un método de instancia DarCostoViaje(), que sin argumento alguno, devuelva como número real (punto flotante de precisión simple), el costo a cobrar por el viaje.
Dicho costo, se obtiene de multiplicar la distancia recorrida por el costo por kilómetro, teniendo presente que, si la distancia recorrida es menor al kilometraje mínimo, se cobra por este último valor.
- 9) Un método de instancia DarCostoViaje(peajes), que sobrecargue el anterior e informe el costo calculado del viaje, pero sumándole el monto total de peajes abonado (peajes es flotante de precisión simple).
- 10) Un método DarDatos(), que sin argumento alguno, devuelva una cadena de caracteres que concatene los valores presentes en las variables miembro de dominio, distancia recorrida y costo por kilómetro más el valor calculado de costo de viaje (sin peajes).
- 11) Un método de clase CostoMayor(viaje1, viaje2), que permita comparar los costos de dos instancia de viaje y devuelva la de mayor costo.

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

- 1) Solicitar al usuario, establecer y mostrar por pantalla, el valor de costo por kilómetro y de kilometraje mínimo, valores únicos y compartidos para todos los viajes.
- 2) Solicitar iterativamente y registrar, los datos de un conjunto de extensión desconocida de viajes; finalizando dicho proceso de carga con el ingreso de dominio de valor igual a “Fin”, para cuyo caso no deberán solicitarse ni registrarse los datos restantes.
- 3) Finalizado el proceso de carga, informar dominio, distancia recorrida y costo por kilómetro más el valor calculado de costo de viaje (sin peajes) para el viaje más costoso ingresado.
- 4) En caso de no haberse ingresado viaje válido alguno, en lugar de lo indicado en el ítem anterior, informar la leyenda “No se ingresaron viajes válidos.”.