

Algoritmos y Estructura de Datos – TI

Guía de Problemas - Semana 03

Ciclo: 2017 – II

Clases y Objetos: referencia *this*, modificador, bloque de inicialización *static* sobrecarga de constructores, uso del *this* en sobrecarga y modificador *final*.

Problema_3_1

Diseñe la clase **Factura** en el paquete **semana_03** con los atributos privados: *ruc* (*String*), *empresa* (*String*), *unidades* (*int*) y *precio unitario* (*double*).

Implemente además:

- Una variable privada de clase que cuente la cantidad de objetos tipo Factura creadas (*int*).
- Una variable privada de clase que acumule la suma de los importes facturados (*double*).
- Una constante pública de clase (*String*).
- Un bloque de inicialización *static* para asignarle a la constante el texto "Sunat" e inicializar con cero las variables privadas de clase.
- Un constructor que inicialice a todos los atributos, cuente la cantidad de objetos creados y acumule los importes facturados. Haga uso de la referencia *this*.
- Un constructor con dos parámetros que inicialice sólo los atributos *ruc* y *empresa*, invocando al primer constructor usando la referencia *this*, enviando con el valor 10 las unidades y con 50.0 el precio unitario.
- Un constructor sin parámetros que invoque al segundo constructor usando la referencia *this*, enviando con "1111111111" el *ruc* y con "MN-Global SRL" la empresa.
- Métodos de acceso público *set* para todos los atributos privados. Use de la referencia *this*.
- Métodos de acceso público *get* para todos los atributos privados.
- Métodos públicos de clase *set/get* para las variables privadas de clase.
- Un método que retorne el importe facturado (*unidades * precio unitario*).

En la clase principal, a la pulsación del botón Procesar:

- Declare, cree e inicialice tres objetos de tipo Factura (con datos fijos) haciendo uso de los tres constructores.
- Invoque a un método listado que reciba (como parámetro) la referencia a un objeto de tipo Factura y visualice sus datos completos.
- Muestre el valor de la constante, la cantidad de objetos creados y la suma de los importes facturados.

Problema_3_2

Diseñe la clase **Empleado** en el paquete **semana_03** con los atributos privados: *codigo* (*int*), *nombre* (*String*), *categoria* (*int*) y *número de celular* (*int*).

Implemente además:

- Una variable privada de clase que cuente la cantidad de objetos tipo Empleado creados (*int*).
- Una variable privada de clase que acumule la suma de los sueldos netos (*double*).
- Una constante pública de clase para un factor de descuento (*double*).
- Un bloque de inicialización *static* para asignarle a la constante el valor 0.15 e inicializar con cero las variables privadas de clase.
- Un constructor que inicialice a todos los atributos, cuente la cantidad de objetos creados y acumule los sueldos netos. Haga uso de la referencia *this*.
- Un constructor con dos parámetros que inicialice sólo los atributos *codigo* y *nombre*, invocando al primer constructor usando la referencia *this*, enviando con el valor 2 la categoría y con 999999999 el número de celular.

- Un constructor sin parámetros que invoque al segundo constructor usando la referencia *this*, enviando con 4444 el código y con “Ninguno” el nombre.
- Métodos de acceso público set para todos los atributos privados. Use la referencia *this*.
- Métodos de acceso público get para todos los atributos privados.
- Métodos públicos de clase set/get para las variables privadas de clase.
- Un método que retorne el sueldo bruto del empleado sabiendo que:

categoría	sueldo bruto
0	S/. 7200
1	S/. 6300
2	S/. 5100

- Un método que retorne el descuento (aplicado al sueldo bruto).
- Un método que retorne el sueldo neto (sueldo bruto – descuento).

En la clase principal, a la pulsación del botón Procesar:

- Declare, cree e inicialice tres objetos de tipo Empleado (con datos fijos) haciendo uso de los tres constructores.
- Invoque a un método listado que reciba (como parámetro) la referencia a un objeto de tipo Empleado y visualice sus datos completos.
- Muestre el factor de descuento, la cantidad de objetos creados y la suma de los sueldos netos.

Problema_3_3

Diseñe la clase **Movil** en el paquete **semana_03** con los atributos privados: número de móvil (*int*), nombre del cliente (*String*), segundos consumidos (*int*) y precio por segundo (*double*).
Implemente además:

- Una variable privada de clase que cuente la cantidad de objetos tipo Movil creados (*int*).
- Una variable privada de clase que acumule la suma de todos los importes a pagar (*double*).
- Una constante pública de clase para el factor del IGV (*double*).
- Un bloque de inicialización *static* para asignarle a la constante el valor 0.18 e inicializar con cero las variables privadas de clase.
- Un constructor que inicialice a todos los atributos, cuente la cantidad de objetos creados y acumule los importes a pagar. Haga uso de la referencia *this*.
- Un constructor con dos parámetros que inicialice sólo los atributos número y nombre, invocando al primer constructor usando la referencia *this*, enviando con el valor 75 los segundos consumidos y con 0.28 el precio por segundo.
- Un constructor sin parámetros que invoque al segundo constructor usando la referencia *this*, enviando con 987656789 el número de móvil y con “Juan” el nombre del cliente.
- Métodos de acceso público set para todos los atributos privados. Use de la referencia *this*.
- Métodos de acceso público get para todos los atributos privados.
- Métodos públicos de clase set/get para las variables privadas de clase.
- Un método que retorne el costo por consumo (segundos * precio).
- Un método que retorne el impuesto por IGV (aplicado al costo por consumo).
- Un método que retorne el importe a pagar (costo por consumo + impuesto por IGV).

En la clase principal, a la pulsación del botón Procesar:

- Declare, cree e inicialice tres objetos de tipo Movil (con datos fijos) haciendo uso de los tres constructores.
- Invoque a un método listado que reciba (como parámetro) la referencia a un objeto de tipo Movil y visualice sus datos completos.
- Muestre el valor del IGV, la cantidad de objetos creados y la suma de los importes a pagar.

Propuesto_3_1

Diseñe la clase **Asesor** en el paquete **semana_03** con los atributos privados: nombre del asesor (*String*), dni (*int*), codigo (*int*), y remuneración en dólares (*double*).

Implemente además:

- Una variable privada de clase que cuente la cantidad de objetos tipo Asesor creados (*int*).
- Una variable privada de clase que acumule la suma de todas las remuneraciones (*double*).
- Una constante pública de clase para el nombre de la institución (*String*).
- Un bloque de inicialización *static* para asignarle a la constante el nombre de la institución e inicializar con cero las variables privadas de clase.
- Un constructor que inicialice a todos los atributos, cuente la cantidad de objetos creados y acumule todas las remuneraciones. Haga uso de la referencia *this*.
- Un constructor con dos parámetros que inicialice sólo los atributos nombre y dni, invocando al primer constructor usando la referencia *this*, enviando con el valor 55555 el código y con 2000.0 la remuneración en dólares.
- Un constructor sin parámetros que invoque al segundo constructor usando la referencia *this*, enviando con "NN" el nombre del asesor y con 88888888 el número del dni.
- Métodos de acceso público set para todos los atributos privados. Use de la referencia *this*.
- Métodos de acceso público get para todos los atributos privados.
- Métodos públicos de clase set/get para las variables privadas de clase.

En la clase principal, a la pulsación del botón Procesar:

- Declare, cree e inicialice tres objetos de tipo Asesor (con datos fijos) haciendo uso de los tres constructores.
- Invoque a un método listado que reciba (como parámetro) la referencia a un objeto de tipo Asesor y visualice sus datos completos.
- Muestre el nombre de la institución, la cantidad de objetos creados y la suma de todas las remuneraciones.

Propuesto_3_2

Diseñe la clase **Balon** en el paquete **semana_03** con los atributos privados: marca (*String*), peso en gramos (*double*), presión en libras (*double*), diámetro en centímetros (*double*) y precio (*double*).

Implemente además:

- Una variable privada de clase que cuente la cantidad de objetos tipo Balon creados (*int*).
- Una variable privada de clase que acumule los importes a pagar (*double*).
- Una constante pública de clase para el valor de PI (*double*) y otra para el factor de descuento (*double*).
- Un bloque de inicialización *static* para asignarle a la constante PI el valor 3.1416, a la segunda contante el factor 0.05 e inicializar con cero las variables privadas de clase.
- Un constructor que inicialice a todos los atributos, cuente la cantidad de objetos creados y acumule los importes a pagar. Haga uso de la referencia *this*.
- Un constructor con tres parámetros que inicialice sólo los atributos marca, peso y presión, invocando al primer constructor usando la referencia *this*, enviando con el valor 18.5 el diámetro y con 100.0 el precio.
- Un constructor sin parámetros que invoque al segundo constructor usando la referencia *this*, enviando con "Adidas" la marca, con 1.5 el peso y con 4.8 la presión.
- Métodos de acceso público set para todos los atributos privados. Use la referencia *this*.
- Métodos de acceso público get para todos los atributos privados.
- Métodos públicos de clase set/get para las variables privadas de clase.
- Un método que retorne el radio (diámetro / 2).
- Un método que retorne el volumen del balón ($4 * PI * radio * radio * radio / 3$).
- Un método que retorne el descuento (aplicado al precio).
- Un método que retorne el importe a pagar (precio - descuento).

En la clase principal, a la pulsación del botón Procesar:

- Declare, cree e inicialice tres objetos de tipo Balon (con datos fijos) haciendo uso de los tres constructores.
- Invoque a un método listado que reciba (como parámetro) la referencia a un objeto de tipo Balon y visualice sus datos completos.
- Muestre el valor de PI, el factor de descuento, la cantidad de objetos creados y el importe a pagar acumulado.

Propuesto_3_3

Diseñe la clase **Consultor** en el paquete **semana_03** con los atributos privados: codigo (*int*), nombre (*String*), horas trabajadas (*int*) y tarifa por hora (*double*).

Implemente además:

- Una variable privada de clase que cuente la cantidad de objetos tipo Consultor creados (*int*).
- Una variable privada de clase que acumule los sueldos netos (*double*).
- Dos constantes públicas de clase para factores de descuentos por AFP y EPS (*double*).
- Un bloque de inicialización *static* para asignarle a la constante AFP el factor 0.10, a la constante EPS el factor 0.05 e inicializar con cero las variables privadas de clase.
- Un constructor que inicialice a todos los atributos, cuente la cantidad de objetos creados y acumule los sueldos netos. Haga uso de la referencia *this*.
- Un constructor con tres parámetros que inicialice sólo los atributos código, nombre y horas trabajadas, invocando al primer constructor usando la referencia *this*, enviando con el valor 65.0 la tarifa.
- Un constructor sin parámetros que invoque al segundo constructor usando la referencia *this*, enviando con 333 el código, con "Juan" el nombre y con 30 las horas.
- Métodos de acceso público set para todos los atributos privados. Use la referencia *this*.
- Métodos de acceso público get para todos los atributos privados.
- Métodos públicos de clase set/get para las variables privadas de clase.
- Un método que retorne el sueldoBruto (horas*tarifa).
- Un método que retorne el descuentoAFP (aplicado al sueldo bruto).
- Un método que retorne el descuentoEPS (aplicado al sueldo bruto).
- Un método que retorne el sueldoNeto = (sueldoBruto – descuentoAFP – descuentoEPS).

En la clase principal, a la pulsación del botón Procesar:

- Declare, cree e inicialice tres objetos de tipo Consultor (con datos fijos) haciendo uso de los tres constructores.
- Invoque a un método listado que reciba (como parámetro) la referencia a un objeto de tipo Consultor y visualice sus datos completos.
- Muestre el valor de los factores de descuentos por AFP y EPS, la cantidad de objetos creados y el acumulado de los sueldos netos.

"La mente determina los actos, pero es el impulso lo que hace la acción."

MP