

## Ejercicios Clases y Objetos

- 1) Cree la clase empleado dentro del proyecto semana2 y declare los siguientes atributos como privados: código (int), nombre (String), horas (int), tarifa (double) e implemente los siguientes métodos:

- a) Un constructor que inicialice a los atributos
- b) Métodos de acceso: set/get para cada atributo
- c) Un método que retorne el sueldo bruto (horas\*tarifa)
- d) Un método que retorne el descuento (11% del sueldo bruto)
- e) Un método que retorne el sueldo neto (sueldo bruto - descuento)

En la **clase principal** (donde esta la GUI), implemente lo siguiente:

- a) Cree un objeto de tipo empleado con datos fijos.
- b) Cree un método listar que imprima todos los atributos del empleado y el sueldo neto.
- c) Incremente la tarifa del empleado en 10% e imprima su nueva tarifa.

- 2) Cree la clase producto dentro del proyecto semana2 y declare los siguientes atributos como privados: código (int), descripción (String), cantidad (int), precio unitario en soles (double) e implemente los siguientes métodos:

- a) Un constructor que inicialice a los atributos
- b) Métodos de acceso: set/get para cada atributo
- c) Un método que retorne el importe a pagar soles (cantidad \* precio unit)
- d) Un método que retorne el importe a pagar dólares. El método recibirá como parámetro el tipo de cambio.

En la **clase principal** (donde está la GUI), implemente lo siguiente:

- a) Cree un objeto de tipo producto cada vez que se pulse el botón procesar. Los datos serán capturados de la GUI. Asuma la existencia de los métodos para la captura de datos.
- b) Cree un método listar que imprima todos los atributos del producto y el importe a pagar en dólares.
- c) Disminuya el precio unitario del producto en 7% e imprima su nuevo precio unitario.

## Ejercicios

Cree la clase Vendedor en el proyecto semana3 y declare los siguiente atributos como **privados**: cod (int), nom (String) y monto vendido (double). La clase debe tener las siguientes **variables estáticas** con acceso **privado**: porcentaje de comisión (debe ser variable estática, porque es el mismo para todos los vendedores), contador de vendedores y el sueldo bruto acumulado de todos los vendedores.

Implemente los siguientes métodos:

- a) Un constructor que inicialice a los atributos
- b) Métodos de acceso: set/get para cada atributo
- c) Un método que retorne el sueldo basico (sueldo fijo de S/ 550.00)
- d) Un método que retorne la comisión en soles (monto vendido \* % de comisión)
- e) Un método que retorne el sueldo bruto (sueldo basico + comisión)

El porcentaje de comisión se debe inicializar con 5% . Para ello, utilice **inicializadores estáticos**. Debe existir la posibilidad de que el usuario pueda modificar el porcentaje de comisión.

En la **clase principal** (donde esta la GUI) implemente lo siguiente:

- a) Cree un objeto de tipo vendedor cada vez que se pulse el **botón Procesar**. Los datos serán capturados de la GUI.
  - b) Cree un método listar que imprima todos los atributos del vendedor, el porcentaje de comisión, el sueldo basico, la comisión en soles y el sueldo bruto del vendedor.
  - c) Imprima la cantidad de vendedores creados.
  - d) Imprima el sueldo bruto acumulado de todos los vendedores.
  - e) Cree un método modificar que permita ingresar el nuevo porcentaje de comisión para los vendedores. Este método debe ser llamado al pulsar el **botón Modificar**.
-

## Ejercicios

- 1) Cree la clase `ArregloSueldos` en el proyecto `semana4` y declare los siguientes atributos como privados:

- Un arreglo de sueldos de tipo `double`
- Una variable índice de tipo `int` que almacene la cantidad de sueldos

Implemente los siguientes métodos:

- Un constructor que cree el arreglo
- Un método que genere 10 sueldos en el rango: 500 .. 3000
- Un método que calcule el sueldo promedio.
- Un método que retorne la cantidad de sueldos mayores a 2000

En la clase **Principal**, debe programar los botones `generar`, `sueldo promedio` y `cantidad de sueldos` de tal forma que, al pulsar estos botones, se llamen a los métodos creados en la clase `ArregloSueldos`.

- 2) Cree la clase `ArregloNumeros` en el proyecto `semana4` y declare los siguientes atributos como privados:

- Un arreglo de números de tipo `int`
- Una variable índice de tipo `int` que almacene la cantidad de números

Implemente los siguientes métodos:

- Un constructor que cree el arreglo
- Un método que genere 15 numeros en el rango: 10 .. 99
- Un método que calcule el número mayor
- Un método que retorne la cantidad de números impares
- Un método que retorne la cantidad de números capicuas

En la clase **Principal**, debe programar los botones:

- `Generar`, que al pulsarlo, se genere los 15 números en el arreglo
  - `Reportar`, que al pulsarlo, imprime el número mayor, la cantidad de números impares y la cantidad de números capicuas.
-

## Ejercicios

- 1) Diseñe un programa que genere un arreglo bidimensional de edades en función de la cantidad de filas y columnas que el usuario ingrese. Las edades deben estar en el rango de 18 a 90 años. Luego, calcule lo siguiente:
  - a) La cantidad de edades generadas
  - b) La suma de todas las edades
  - c) La edad menor
  - d) La edad mayor
  - e) El promedio de edades
  - f) La cantidad de personas que superan los 40 años

El arreglo Bidimensional se debe crear en la clase ArregloEdades dentro del proyecto semana5 y la GUI se debe controlar desde la clase Principal.

- 2) Diseñe un programa que genere una matriz cuadrada ( $n \times n$ ) de números enteros e implemente los siguientes métodos:
  - a) Un método que imprima los elementos y la suma que conforma su diagonal izquierda (que inicia en la esquina superior izquierda y termina en la esquina inferior derecha).
  - b) Un método que imprima los elementos y la suma que conforma su diagonal derecha (que inicia en la esquina superior derecha y termina en la esquina inferior izquierda).

El arreglo Bidimensional se debe crear en la clase MatrizCuadrada dentro del proyecto semana5 y la GUI se debe controlar desde la clase Principal.

---

## Ejercicios

- 1) Asuma la existencia de la clase Empleado que cuenta con los siguientes atributos privados: código (entero), nombre (cadena), horas trabajadas (entero) y tarifa (real). Considere que la clase cuenta con un constructor que inicializa los atributos y los métodos de acceso set y get para cada atributo. Diseñe las opciones de ingresar, modificar y listar del mantenimiento de empleados. Para ello, implemente las clases ArregloEmpleados (clase que maneja el arreglo de objetos) y la clase Principal (clase que controla la GUI).  
Adicionalmente, implemente, en la clase ArregloEmpleados, los siguientes métodos:
    - a) Diseñe un método que aumente la tarifa a los empleados cuyo nombre empiece con la letra 'R'.
    - b) Diseñe un método que retorne el nombre del empleado que tiene la tarifa más alta.
  
  - 2) Asuma la existencia de la clase Vendedor que cuenta con los siguientes atributos privados: código (entero), nombre (cadena), y monto vendido (real). Considere que la clase cuenta con un constructor que inicializa los atributos y los métodos de acceso set y get para cada atributo. Diseñe las opciones de ingresar, consultar y eliminar del mantenimiento de vendedores. Para ello, implemente las clases ArregloVendedores (clase que maneja el arreglo de objetos) y la clase Principal (clase que controla la GUI).  
Adicionalmente, implemente, en la clase ArregloVendedores, los siguientes métodos:
    - a) Diseñe un método que retorne el monto promedio de aquellos vendedores cuyo nombre empiece con 'A'.
    - b) Diseñe un método que retorne el nombre del vendedor que vendió menos.
-

## Ejercicios

- 1) Asuma la existencia de la clase Celular que cuenta con los siguientes atributos privados: código (entero), marca (cadena), modelo (cadena) y precio unitario (real). Considere que la clase cuenta con un constructor que inicializa los atributos y los métodos de acceso set y get para cada atributo. Diseñe las opciones de ingresar, modificar y listar del mantenimiento de celulares. Para ello, implemente las clases ArregloCelulares (clase que maneja un objeto privado de tipo ArrayList) y la clase Principal (clase que controla la GUI).  
Adicionalmente, implemente en la clase ArregloCelulares los siguientes métodos:

- a. Diseñe un método que aumente en 8% el precio unitario a los celulares cuya marca termine con la letra 'a'.
- b. Diseñe un método que retorne los modelos de celular de la marca enviada como parámetro.

- 2) Asuma la existencia de la clase Video que cuenta con los siguientes atributos privados: codVideo (entero), nombre de película (cadena), codGenero (0=comedia, 1=suspenso, 2=terror) y precio unitario (real). Considere que la clase cuenta con un constructor que inicializa los atributos y los métodos de acceso set y get para cada atributo. Diseñe las opciones de ingresar, consultar y eliminar del mantenimiento de videos. Para ello, implemente las clases ArregloVideos (clase que maneja un objeto privado de tipo ArrayList) y la clase Principal (clase que controla la GUI).  
Adicionalmente, implemente en la clase Principal los siguientes métodos:

- a. Diseñe un método que imprima el precio unitario promedio de aquellos videos del género suspenso.
  - b. Diseñe un método que elimine los videos del género ingresado desde la GUI.
-

## Ejercicios

- 1) Diseñe un mantenimiento de celulares que permita ingresar, consultar, modificar, eliminar y listar estudiantes. Para ello, cree las clases Celular, ArregloCelulares (en el proyecto semana10) y Principal. Cree los métodos cargar y grabar en la clase ArregloCelulares. Al cargar el JApplet se deberán leer los datos del archivo celulares.txt; si el archivo no existe, deberá aparecer un mensaje de error.  
Asuma la existencia de la clase Celular que cuenta con los siguientes atributos privados: código (entero), marca (cadena), modelo (cadena) y precio unitario (real). Considere que la clase cuenta con un constructor que inicializa los atributos y los métodos de acceso set y get para cada atributo.
- 2) Diseñe un mantenimiento de videos que permita ingresar, consultar, modificar, eliminar y listar videos. Para ello, cree las clases Video, ArregloVideos (en el proyecto semana10) y Principal. Cree los métodos cargar y grabar en la clase ArregloVideos. Al cargar el JApplet, se deberán leer los datos del archivo videos.txt; si el archivo no existe, deberá aparecer un mensaje de error.  
Asuma la existencia de la clase Video que cuenta con los siguientes atributos privados: codVideo (entero), nombre de película (cadena), codGenero (0=comedia, 1=suspenso, 2=terror) y precio unitario (real). Considere que la clase cuenta con un constructor que inicializa los atributos y los métodos de acceso set y get para cada atributo.

## Ejercicios

- 1) Implemente la clase **Figura** en el proyecto padre con los siguientes miembros:
- Atributos privados: *x*, *y* que representa la ubicación de la figura geométrica
  - Constructor que inicializa los atributos de la clase.
  - Un método *ubicacion()* que retorna la ubicación de la figura geométrica según sus posiciones *x* e *y*.

Luego, implemente dos subclases de **Figura**: **Cuadrado** y **Círculo** en el proyecto hijo.

**Cuadrado** presenta los siguientes miembros:

- Atributo privado: *lado*
- Constructor con parámetros para inicializar los atributos: *x*, *y*, *lado*.
- Método *area()* que retorna el area del cuadrado ( $\text{lado} \times \text{lado}$ )

**Círculo** presenta los siguientes miembros:

- Atributo privado: *radio*
- Constructor con parámetros para inicializar los atributos: *x*, *y*, *radio*
- Método *area()* que retorna el área del círculo ( $\pi \times \text{radio} \times \text{radio}$ )

Por último, implemente el método *Procesar* de la clase **Principal** que contiene el *actionPerformed()* y cree 2 objetos: uno de tipo **Cuadrado** y el otro de tipo **Círculo** e imprima su ubicación y área de cada objeto.

- 2) Diseñe la clase **Trabajador** en el proyecto padre con los siguientes miembros:
- **Atributos protegidos**: *nombre*, *apellido*, *telefono* de tipo *String*
  - Constructor que inicializa los atributos de la clase.
  - Un método *generarCodigo()* que retorna el código formado por el primer carácter del nombre, el último carácter del apellido y el teléfono del trabajador.

Luego, implemente dos **subclases de Trabajador**: **Empleado** y **Consultor** en el proyecto hijo.

**Empleado** presenta los siguientes miembros:

- Atributos privados: *sueldo básico* y *porcentaje de bonificación*
  - Constructor con parámetros para inicializar los atributos: *nombre*, *apellido*, *teléfono*, *sueldo básico* y *porcentaje de bonificación*
  - Un método *boniSoles()* que retorna la bonificación en soles ( $\text{sbas} \times \% \text{ bonificación}$ )
  - Un método *sbruto()* que retorna el sueldo bruto del empleado ( $\text{sbas} + \text{bonificación en soles}$ )
  - Un método *mostrarDatos()* que retorne una cadena conteniendo: *nombre*, *apellido*, *telefono*, *bonificación en soles* y el sueldo bruto
-



**Consultor** presenta los siguientes miembros:

- Atributos privados: horas trabajadas y tarifa horaria
- Constructor con parámetros para inicializar los atributos: nombre, apellido, teléfono, horas y tarifa
- Un método sbruto() que retorna el sueldo bruto del consultor (horas\*tarifa)
- Un método mostrarDatos() que retorne una cadena conteniendo: nombre, apellido, telefono y el sueldo bruto

Por último, implemente el método Procesar de la clase **Principal** que contiene el `actionPerformed()` y cree 2 objetos: uno de tipo Empleado y el otro de tipo Consultor e imprima sus datos invocando al método `mostrarDatos()` y su código generado.

## Ejercicios

1. Implemente la clase abstracta `Animal` en el proyecto padre que contenga el método abstracto `habla()` que se sobreescribirá en las subclases `Perro`, `Gato` y `Pollo` (las subclases se crearán en el proyecto hijo).  
Por último, cree la clase **Principal** que contiene el `actionPerformed()` para crear los objetos `Perro`, `Gato` y `Pollo` e imprima como habla cada `Animal`.
2. Diseñe un programa que, a la pulsación del botón `Procesar`, registre a una persona en la `Reniec`. Para ello, se debe crear un objeto de tipo `persona` con los siguientes atributos: `apellido paterno`, `apellido materno`, `nombres`, `fecha de nacimiento`, `sexo`, `estado civil`; y los siguientes datos del `DNI`: `Nro de dni`, `fecha de inscripción`, `fecha de emisión` y `fecha de caducidad`.