

ENTERPRISE JAVA DEVELOPER

JAVA ORIENTADO A OBJETOS

MIEMBROS DE CLASE

Eric Gustavo Coronel Castillo
gcoronelc.blogspot.com



Temas



- Declaración de variables
- Declaración de métodos
- Encapsulación
- Constructor
- Destructor
- Proyecto ejemplo



DECLARACIÓN DE VARIABLES

Sintaxis:

[modificadorAcceso] tipo nombreVariable [= valor] ;

El **modificadorAcceso** puede ser:

- privado (private)
- paquete
- protegido (protected)
- público (public)

```
public class Factura{  
  
    private int numero = 54687;  
    double importe = 5467.87;  
    protected int vendedor = 528;  
    public String cliente = "Banco de Crédito";  
  
}
```

Factura
- numero : int ~ importe : double # vendedor : int + cliente : String



DECLARACIÓN DE VARIABLES

uno

```
public class ClaseA{  
  
    private int n1;  
    int n2;  
    protected int n3;  
    public int n4;  
  
    public void metodoA(){}  
  
}
```

```
public class ClaseB{  
  
    public void metodoB(){}  
  
}
```

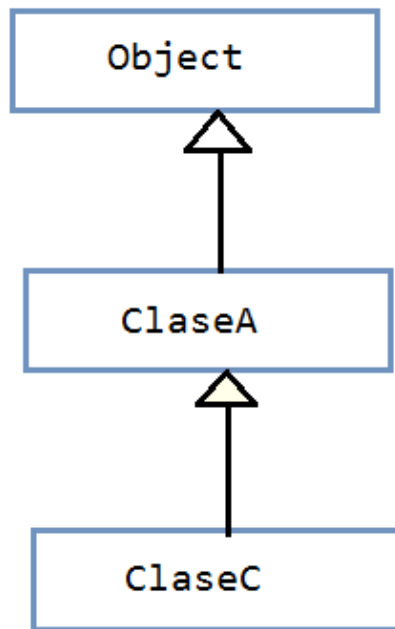
dos

```
public class ClaseC extend ClaseA{  
  
    public void metodoC(){}  
  
}
```

```
public class ClaseD{  
  
    public void metodoD(){}  
  
}
```



DECLARACIÓN DE VARIABLES



Visibilidad de las Variables

METODOS			VARIABLES DE ClaseA			
Paquete	Clase	Metodo	n1	n2	n3	n4
uno	ClaseA	metodoA	SI	SI	SI	SI
uno	ClaseB	metodoB	NO	SI	SI	SI
dos	ClaseC	metodoC	NO	NO	SI	SI
dos	ClaseD	metodoD	NO	NO	NO	SI



DECLARACIÓN DE MÉTODOS

Sintaxis:

```
[modificadorAcceso] tipo nombreMétodo ( [ parámetros ] ) {  
  
    // Implementación  
  
}
```

El modificadorAcceso puede ser:

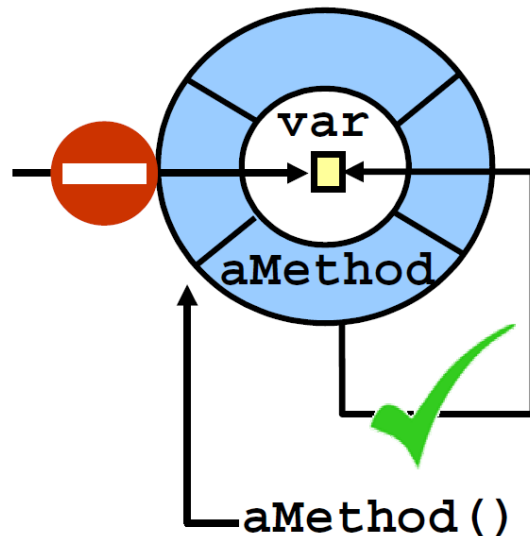
- privado (private)
- paquete
- protegido (protected)
- público (public)



ENCAPSULACIÓN

Características

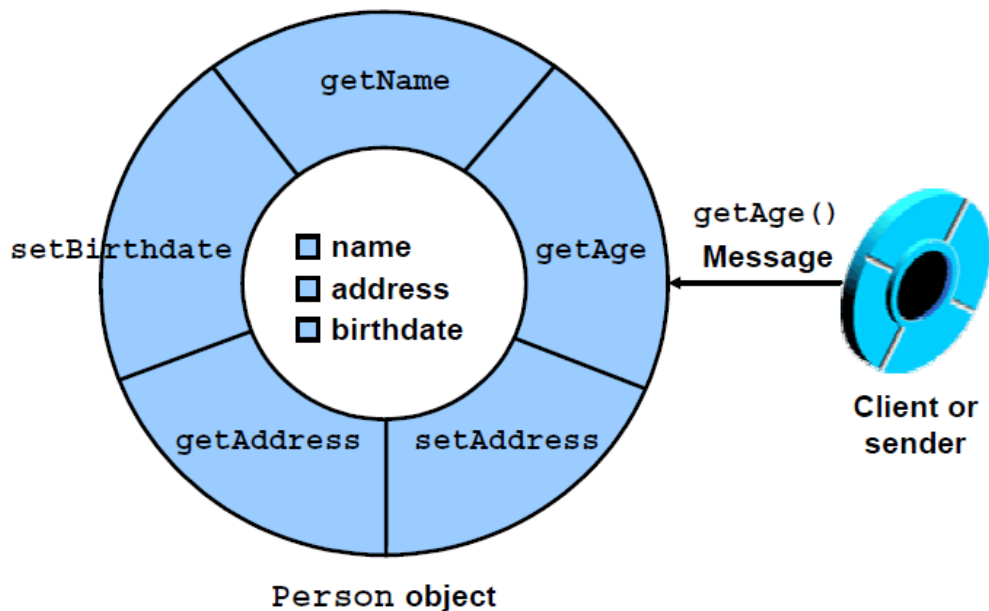
- Las variables de instancia deben ser declaradas como privadas.
- Los métodos de instancia sólo puede acceder a las variables de instancia privadas.





ENCAPSULACIÓN

Implementación





ENCAPSULACIÓN

Implementación

- Variable

```
private tipo variable[ = valor ] ;
```

- Método set

```
public void setVariable( tipo valor ) {  
    this.variable = valor;  
}
```

- Método get

```
public tipo getVariable() {  
    return this.variable;  
}
```

En caso que la propiedad sea de tipo **boolean** se utiliza **isPropiedad** en lugar de **getPropiedad**.



CONSTRUCTOR

Se utiliza para inicializar el objeto.

```
public class NombreClase {  
    public NombreClase() {  
        // Inicialización del objeto  
    }  
}
```



DESTRUCTOR

Se utiliza para liberar los recursos que el objeto está utilizando.

```
public class NombreClase {  
  
    protected void finalize() throws Throwable {  
  
        // Liberar recursos del objeto  
  
    }  
  
}
```



PROYECTOS EJEMPLO

La empresa **Vía Éxitos** Necesita saber cuanto se le debe pagar a sus trabajadores y a cuanto asciende el importe de impuesto a la renta que debe retener.

Los datos son:

- Cantidad diaria de horas trabajadas.
- Cantidad de días trabajados.
- El pago por hora.

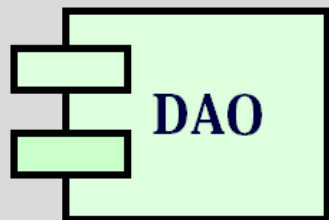
Se sabe que si los ingresos supera los 1500.00 Nuevos Soles se debe retener el 8% del total.



CODIGO FUENTE

EUREKA-CS-ORACLE-JDBC

APLICACIÓN JAVA



JDBC

A light blue rounded rectangular box with a dark blue border, representing the JDBC driver component.

CONEXIÓN

A large, light gray double-headed arrow with a black outline, indicating a bidirectional connection between the application and the database.

Esquema

EUREKA

ORACLE XE 11g

Dirección de descarga: <https://goo.gl/TDgc5R>

ENTERPRISE JAVA DEVELOPER

JAVA ORIENTADO A OBJETOS

Gracias

Eric Gustavo Coronel Castillo
gcoronelc.blogspot.com

