

Definición de una clase

Dr. José Lázaro Martínez Rodríguez

Introducción

- Tipos de datos en Java
- Definición de clase
- Relaciones de una clase

Tipos de datos en Java:

- - Tipos primitivos
 - Referencias (instancias de clases que se usan para manipular objetos)

- Los tipos primitivos son los que permiten manipular valores numéricos (con distintos grados de precisión), caracteres y valores booleanos.

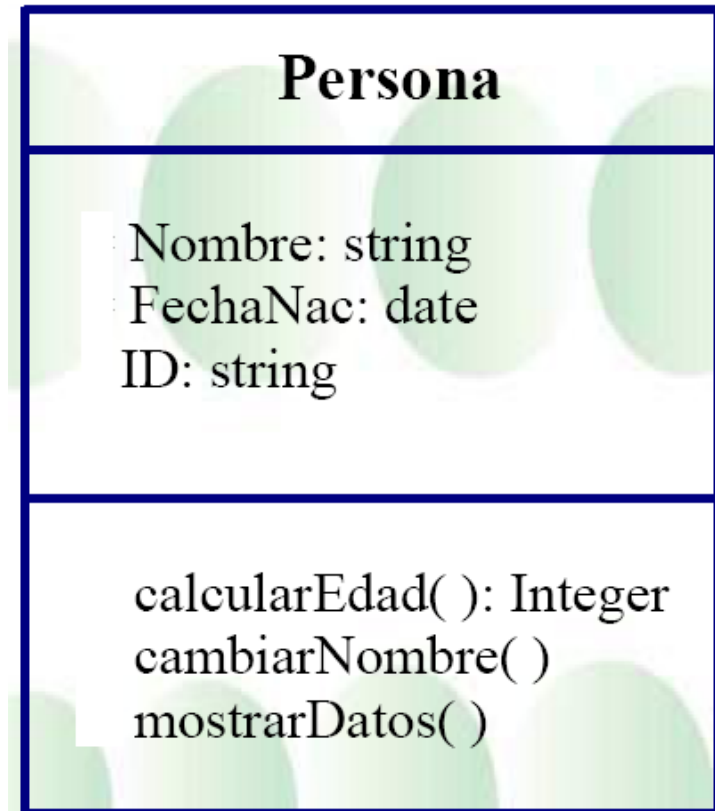
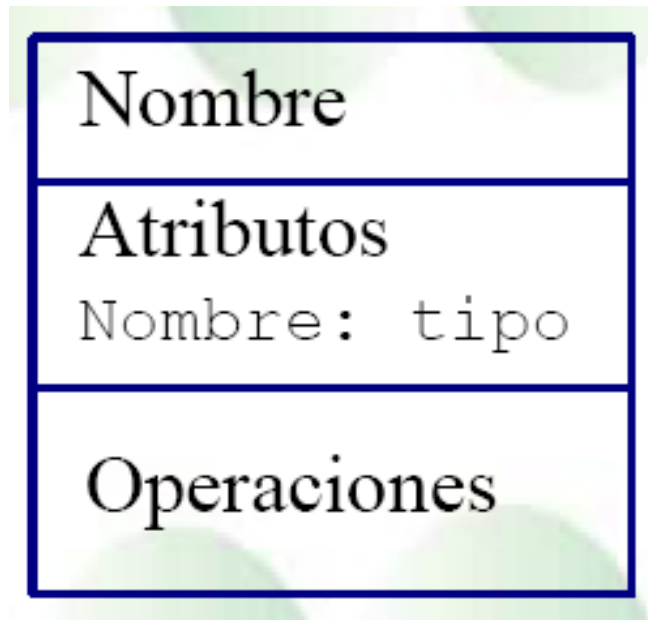
Tipo	
boolean	Puede contener valores true o false
char	Caracteres. Tamaño 16 bits
byte	Enteros. Tamaño 8 bits. Valores entre -128 y 127
short	Enteros. Tamaño 16 bits. Entre -32768 y 32767
int	Enteros. Tamaño 32 bits.
long	Enteros. Tamaño 64 bits.
float	Números de punto flotante. Tamaño 32 bits.
double	Números de punto flotante. Tamaño 64 bits.

Clase

- Es una especificación de estructura, comportamiento y herencia para objetos.
- Una clase representa un conjunto de objetos que comparten estructura y comportamiento común.
- Una clase es un descriptor/constructor de objetos.

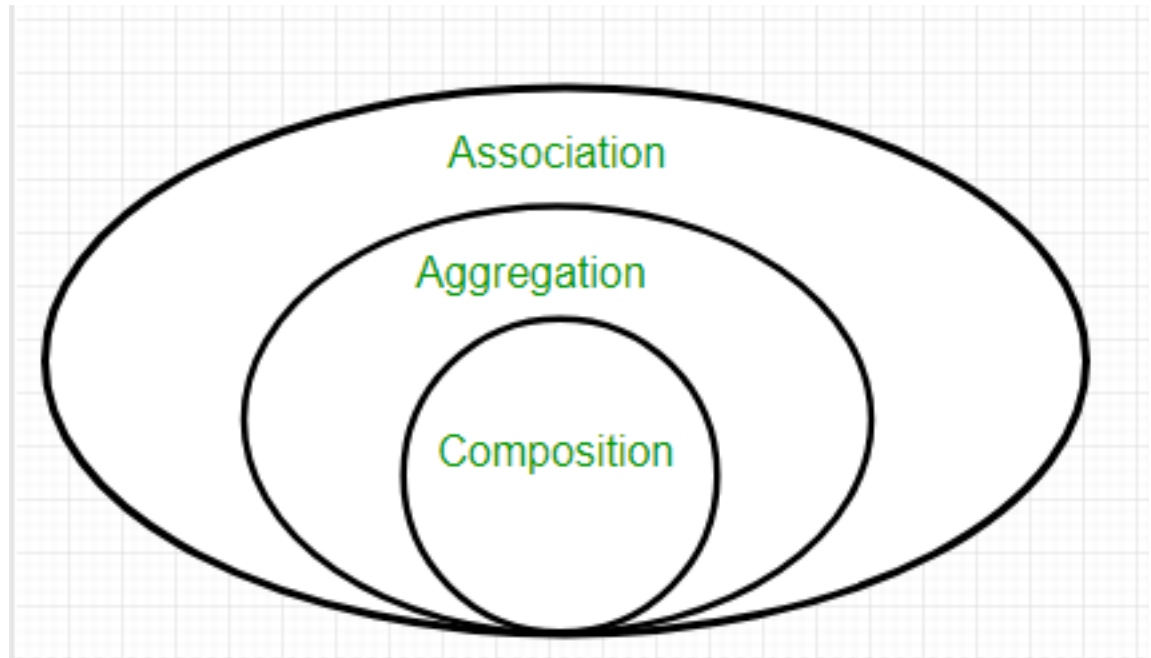
Clase

- ¿Cómo especificar una clase?
- Usando UML (Unified Model Language)



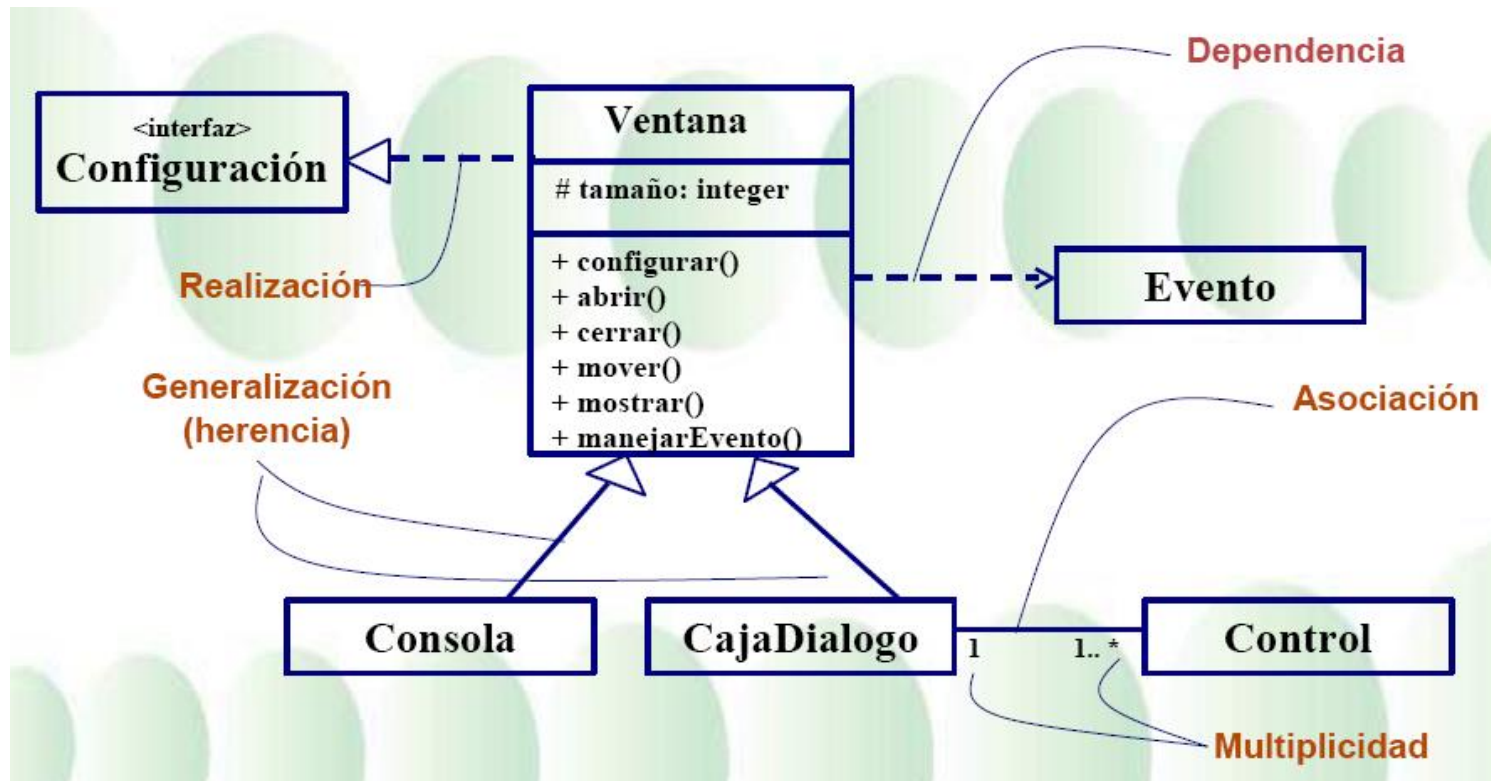
Clases: relaciones

- Dependencia
- Generalización
- Asociación
- Realización
- Composición



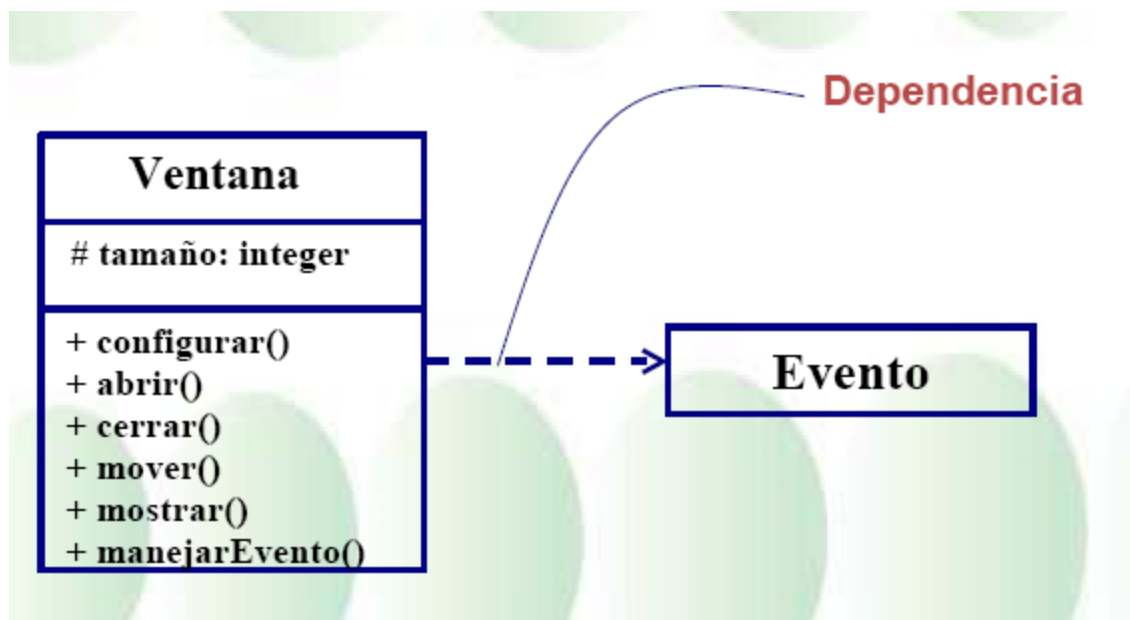
Clases: relaciones

- Diferentes relaciones entre clases:



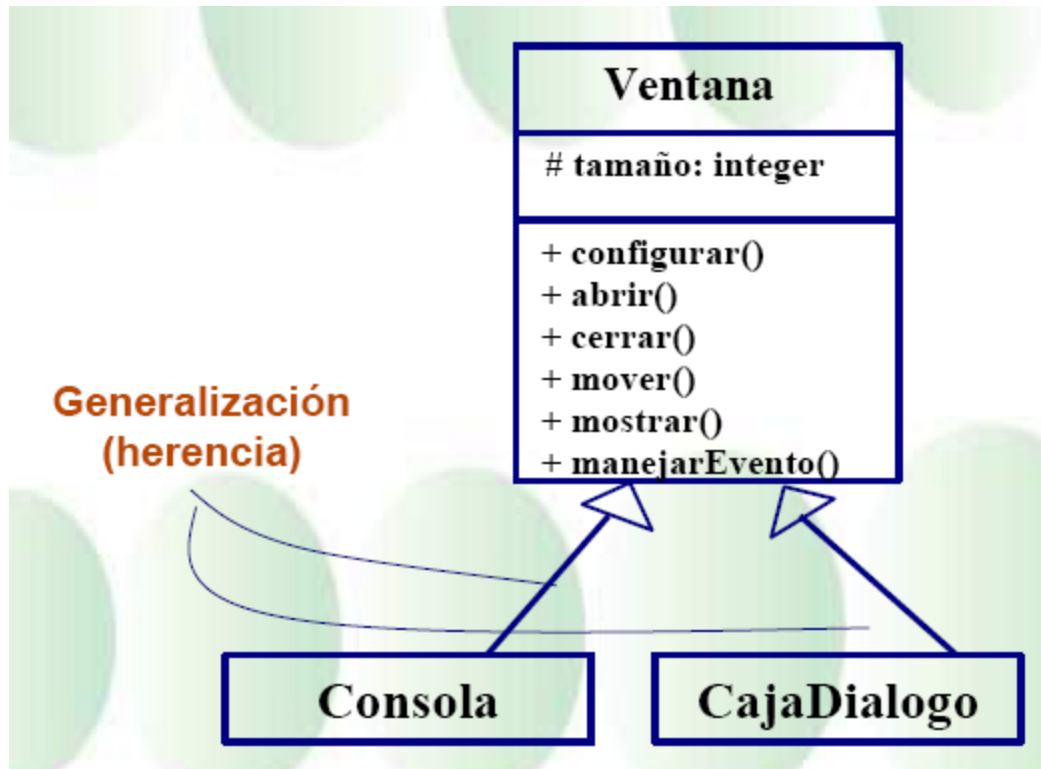
Dependencia

- Es una relación de uso en donde los cambios en la especificación de una clase pueden afectar a otra clase que la usa:



Generalización

- Es una relación entre una clase general (superclase) y otra específica (subclase)

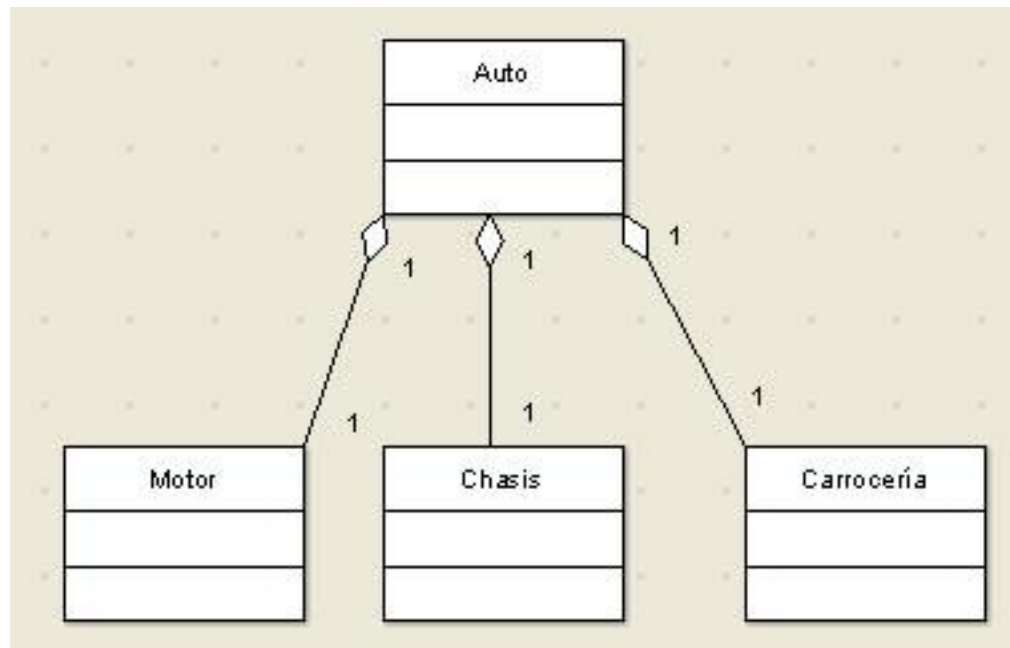


- Asociación

- Es una relación estructural que expresa la conexión recíproca entre objetos de dos clases distintas

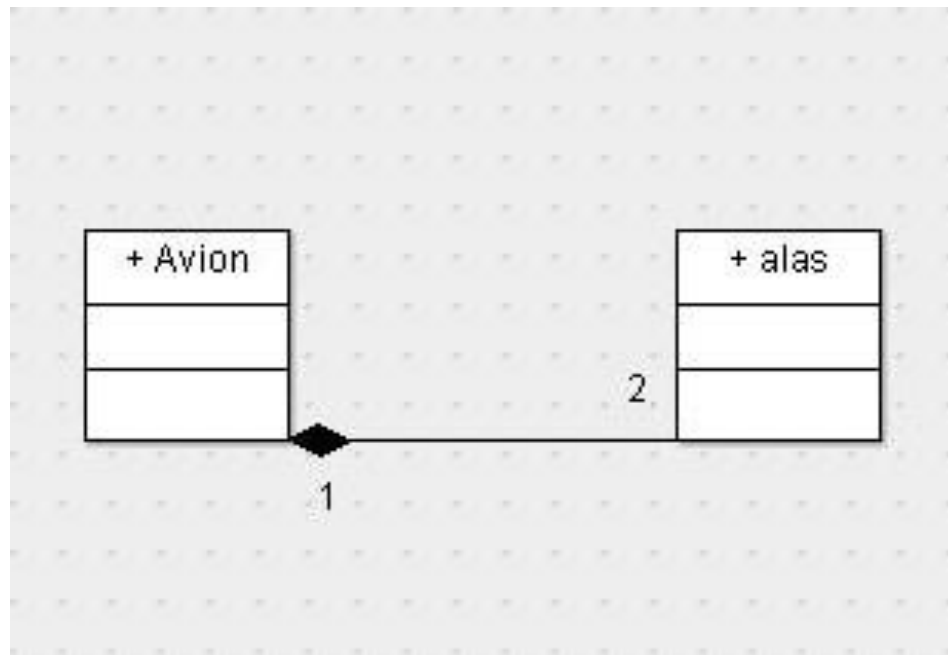
- Regla de asociación: Agregación simple

- Relación todo/partes en donde una clase representa el todo y otra una de sus partes



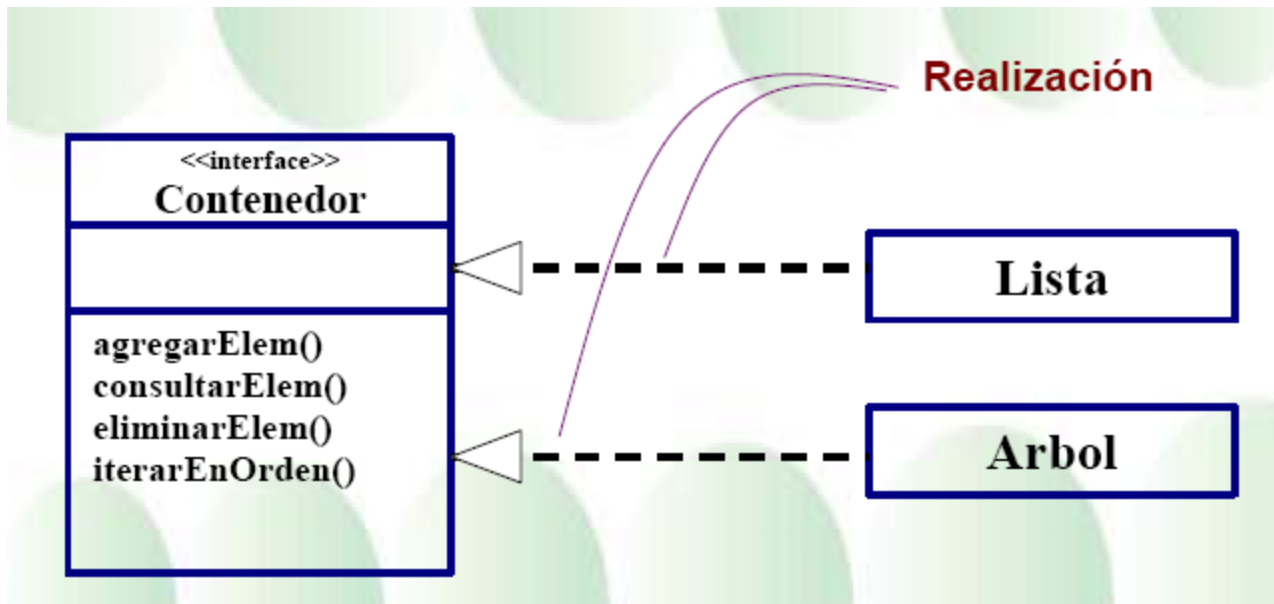
- Regla de asociación: Composición

- Es una forma de agregación, con fuerte pertenencia y tiempo de vida coincide como parte del todo.



- Realización

- Relación semántica donde se establece un contrato entre dos clases.
- Es una combinación entre dependencia y generalización



Miembros de una clase

- Una clase es una agrupación de datos y de código que actúa sobre esos datos, a la que se le da un nombre. Una clase contiene:
 - Datos (se denominan datos miembro). Estos pueden ser de tipos primitivos o referencias.
 - Métodos (se denominan métodos miembro).
- La sintaxis para la declaración de una clase es:

```
modificador class nombre_clase {  
    datos_miembro;  
    métodos_miembros  
}
```

Estructura de una clase

```
class Nombre_Clase {  
    atributos  
    métodos  
    constructor  
    comportamiento  
}
```

Miembros de una clase

- Miembros son tanto atributos como métodos de una clase
- Para entenderlo hay que referirse necesariamente a la diferencia que hay entre lo que sería una **variable de instancia** y una **variable de clase**.
- Variable o miembro de instancia
- pertenece solamente a esa instancia de la clase. Se usan diferentes ubicaciones de memoria para cada instancia/miembro de la clase.

Miembros de una clase

- Variable o miembro de clase o estático
- se define usando el modificador **static**
- están asociados con la clase, en lugar de con cualquier objeto. Significa que cada instancia de la clase comparte una variable de clase, que está en una ubicación fija en la memoria.
- cualquier objeto puede cambiar el valor de una variable de clase, pero las variables de clase también se pueden manipular sin crear una instancia de la clase.

- Ejemplo:

```
public class Punto {
```

```
    int x;
```

```
    int y;
```

```
    void asignarCoordenadas();
```

```
    void pintarCoordenadas();
```

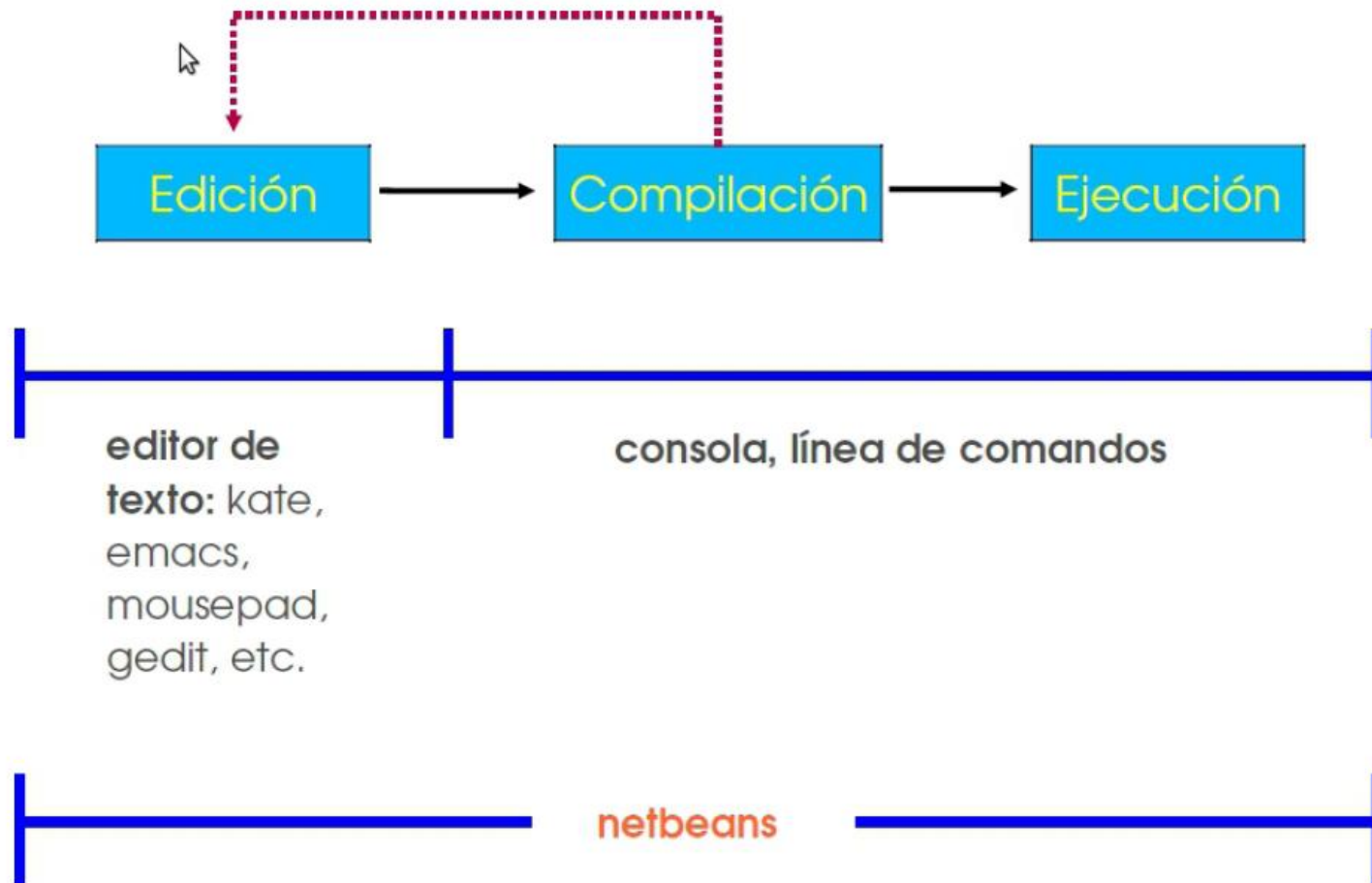
```
    void darCoordenadas();
```

```
}
```

Datos miembro

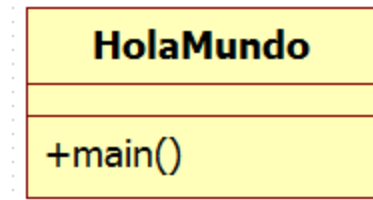
Métodos miembro

Proceso de codificación/implementación



Primer programa en Java

- Clase

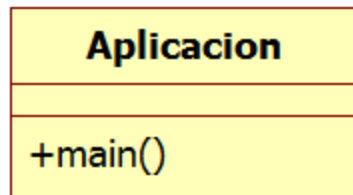


- Código en Java

```
public class HolaMundo {  
  
    // el método main empieza con la ejecución de la aplicación en Java  
    public static void main(String[] args) {  
        //Imprimir una linea de texto  
        System.out.println("Hola mundo");  
    }  
  
}
```

Primer programa en Java

- Clase



- Código en Java

```
public class Aplicacion {  
  
    // el método main empieza con la ejecución de la aplicación en Java  
    public static void main(String[] args) {  
        //Imprimir una línea de texto  
        System.out.println("Hola mundo");  
    }  
} //Fin de la clase Aplicación
```

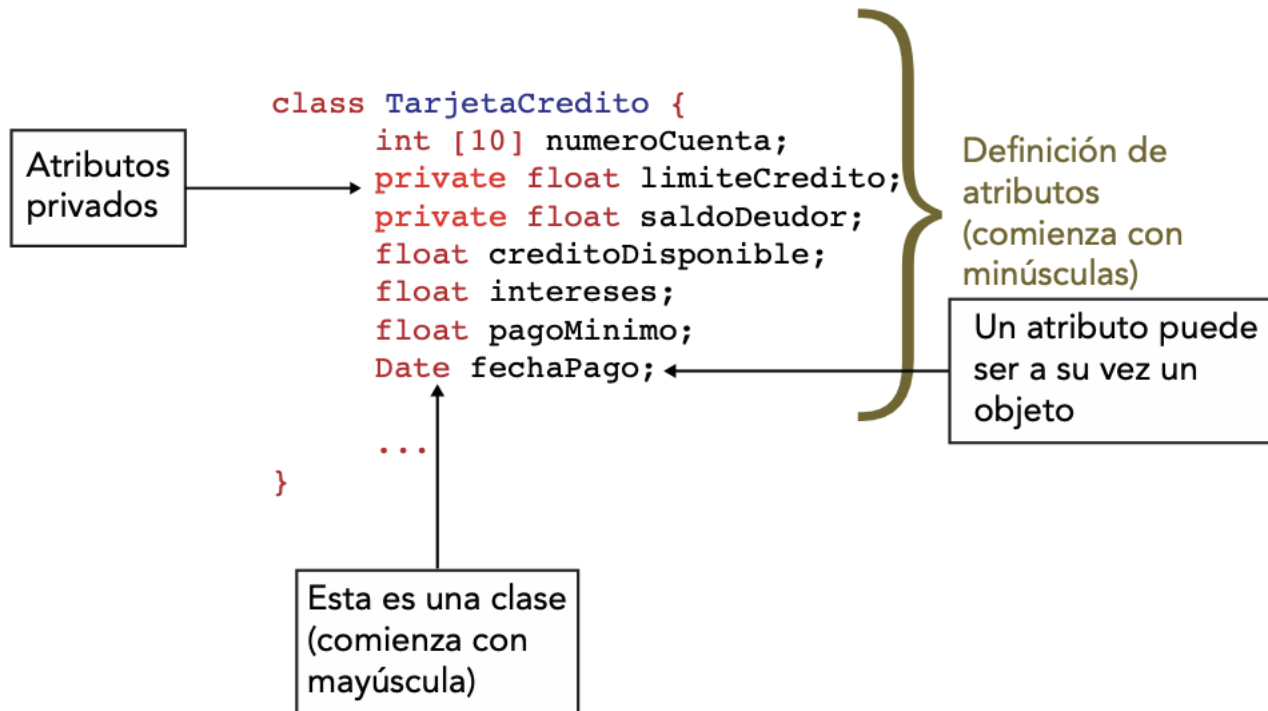
Atributos

- Los atributos definen la estructura de datos de la clase, los cuales, por omisión, son públicos, es decir, accesibles desde otras clases,
 - lo que significa que se modifican desde afuera del objeto.
- Es altamente recomendable declarar todos los atributos con el modificador `private`
 - y solamente cambiarlo a `public` o `protected` cuando sea absolutamente necesario.

```
private int promedio;  
public float saldoCuentaCredito;
```

Ejemplo

- Ejemplo clase TarjetaCredito



Métodos

- Los métodos constituyen el comportamiento de los objetos de la clase.
- La definición de un método es muy similar a la de una función.
- Los métodos públicos son las operaciones que los objetos externos realizan con el objeto en cuestión.
- Los métodos privados son las operaciones internas que no se pueden invocar desde el exterior, pero sí desde otro método dentro de la clase.

Ejemplo

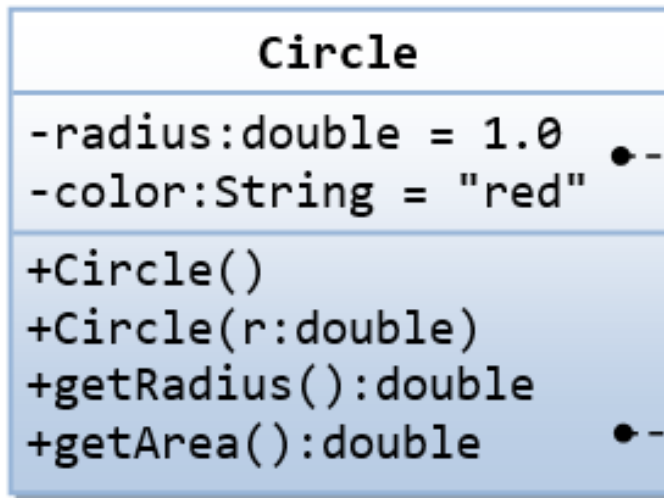
```
public class Ejemplo {  
    public int x;  
    public int a;  
    public void hacerAlgo() {  
        x = 1 + a * 3;  
    }  
    public int suma() {  
        return x + a;  
    }  
    public void imprimir() {  
        System.out.println( "x= " + x + " a= " + a + "\n" );  
    }  
}
```

Ejemplo

```
public class Arbol {  
    public String ojas;  
    public int edad;  
  
    public moverse(String) {  
        ojas -= 10  
    }  
    public int cantidadHojas() {  
        return ojas;  
    }  
    public void mostrarEdad() {  
        system.out.println( "edad " + edad "\n );  
    }  
}
```

Ejercicio

- Realice el código para el siguiente diagrama
- Utilice atributos publicos



±varName:type = default-value

- denotes private access

+ denotes public access

±methodName(parmName:type,...):retunType