

T1Orientacionaobjetos.pdf



msm_EPS



Análisis y Diseño de Software



2º Grado en Ingeniería Informática



**Escuela Politécnica Superior
Universidad Autónoma de Madrid**

WUOLAH + BBVA

Hazte **cliente de BBVA y...**
ahórrate 6 meses
de suscripción

BOOM

1/6

Este número es indicativo del riesgo del producto, siendo 1/6 indicativo de menor riesgo y 6/6 de mayor riesgo.

BBVA está adherido al Fondo de Garantía de Depósitos de Entidades de Crédito de España. La cantidad máxima garantizada es de 100.000 euros por la totalidad de los depósitos constituidos en BBVA por persona.

Ahora, si te abres una Cuenta Online en BBVA, te reembolsamos una de estas suscripciones durante 6 meses (hasta 9,99€/mes) al pagarla con tu tarjeta Aqua Débito

NETFLIX

Spotify

HBOmax

Disney+

PlayStation Plus

DAZN

Promoción solo para nuevos clientes de BBVA. Válida hasta el 30/06/2023. Estas empresas no colaboran en la promoción.

Abre tu cuenta



Hazte cliente de BBVA y ... ahórrate 6 meses de suscripción

WUOLAH
+ BBVA

NETFLIX

Spotify

HBOmax

Disney+

PlayStation.Plus

DAZN

Ahora, si te abres una Cuenta Online en BBVA, te reembolsamos una de estas suscripciones durante 6 meses (hasta 9,99€/mes) al pagarla con tu tarjeta Aqua Débito

Promoción solo para nuevos clientes de BBVA. Válida hasta el 30/06/2023. Estas empresas no colaboran en la promoción.

1/6

Este número es indicativo del riesgo del producto, siendo 1/6 indicativo de menor riesgo y 6/6 de mayor riesgo.

BBVA está adherido al Fondo de Garantía de Depósitos de Entidades de Crédito de España. La cantidad máxima garantizada es de 100.000 euros por la totalidad de los depósitos constituidos en BBVA por persona.



Orientación a objetos

1. Conceptos de diseño orientado a Objetos

Conceptos fundamentales

1.1. Ejemplo

[Clases](#)

[Atributos](#)

[Objetos](#)

[Métodos](#)

[Ejemplos de comportamiento](#)

[Ventajas](#)

1.2. Programación Estructurada

1.3. Tipos Abstractos de Datos

1.4. Programación orientada a objetos

2. Objetos y Clases

Elementos de la Prog. Orientada a Objetos

2.1. Estructura

2.2. Ciclo de vida de un objeto

2.3. Relaciones con otros objetos

[Asociación](#)

[Contenido](#)

[Agregación](#)

2.4. Navegación

[Navegación bidireccional](#)

[Navegación 1 a muchos](#)

2.5. Composición + navegación

[Unidireccional](#)

[Bidireccional](#)

2.6. Clase abstracta

2.7. Métodos (y atributos) de objeto y clase

3. Encapsulamiento

4. Herencia y Polimorfismo

[Métodos abstractos](#)

[Herencia Múltiple](#)

[Polimorfismo](#)

[Ligadura Dinámica](#)

[5. Resumen y Conclusiones](#)

[Malas Soluciones en Diseños](#)

[Asociaciones vs. atributos identificadores](#)


[Importancia del diseño](#)

1. Conceptos de diseño orientado a Objetos

Considera las aplicaciones como un conjunto de objetos que interaccionan. → Cada objeto contienen datos y métodos.

Conceptos fundamentales

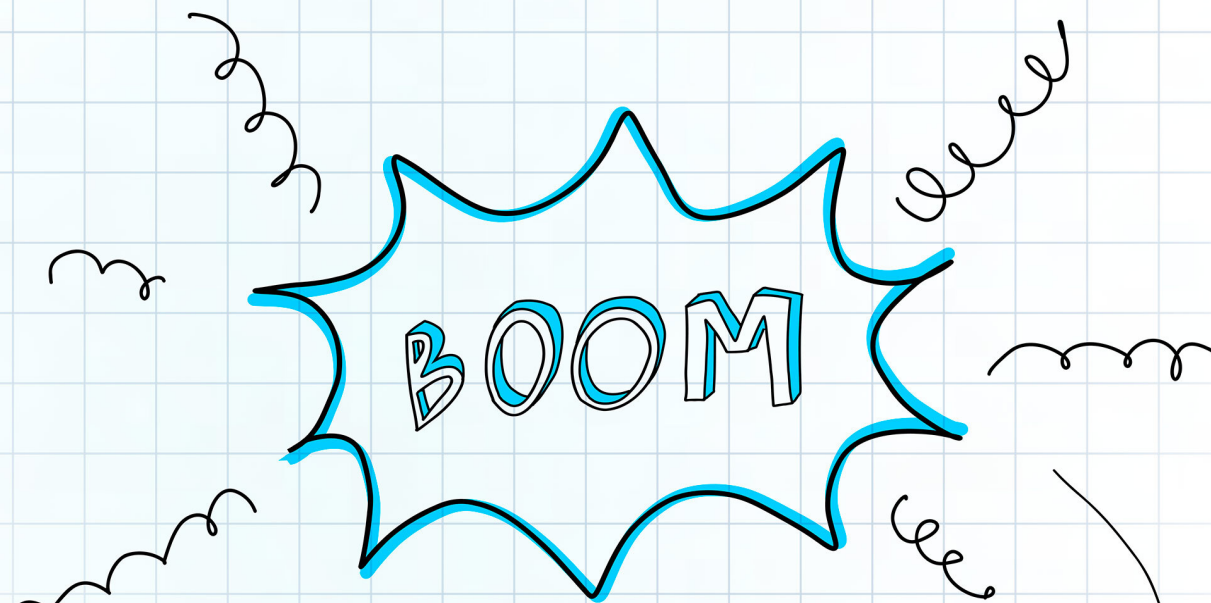
- **Clase** → “plantilla” describe los datos y el comportamiento de un conjunto de objetos
- **Objeto** → instancia en tiempo de ejecución de 1 clase
- **Encapsulación** → Ocultación de info.
- **Polimorfismo de tipos** → Herencia de tipos + Poder usar de manera segura un objeto especialización en lugar del objeto más general.

 UML → Lenguaje diseño

1.1. Ejemplo

Aplicación para la gestión de empleados y clientes de una empresa. De las personas almacenamos su nombre y fecha de nacimiento. De los clientes, el nombre de su empresa y teléfono. De todas las personas queremos mostrar sus datos personales. Los empleados tienen un sueldo bruto y un departamento. Queremos calcular su sueldo neto. Hay trabajadores que son directivos y estos tienen una categoría.

Clases

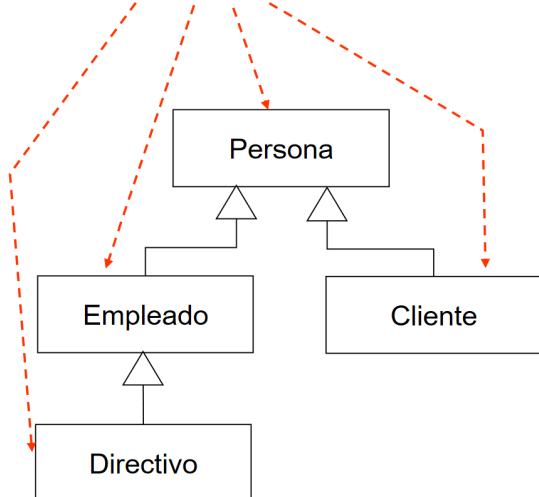


Hazte cliente de BBVA y ... **ahórrate 6 meses** **de suscripción**

Ahora, si te abres una Cuenta Online en BBVA, te reembolsamos una de estas suscripciones durante 6 meses (hasta 9,99€/mes) al pagarla con tu tarjeta Aqua Débito

NETFLIX**HBOmax**[Abre tu cuenta](#)

Clases

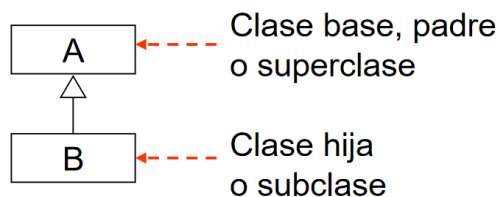


Se agrupan en clases con datos y comportamientos comunes.

- Representación UML → cajas
- Deberían ser sustantivo singular

Tenemos personas, las personas pueden ser empleados o clientes y a la vez los empleados pueden ser directivos.

- Las flechas significan “es un” → Indican la **herencia**. Ejemplo: “Un empleado es una persona”



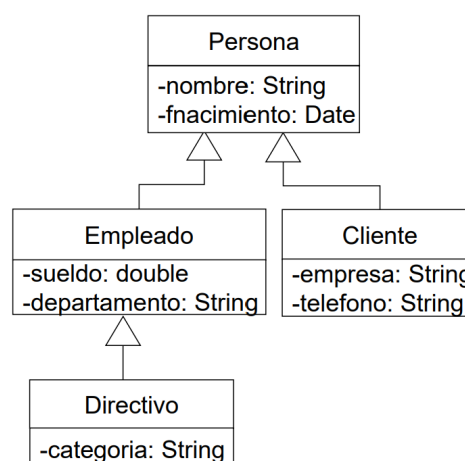
Atributos

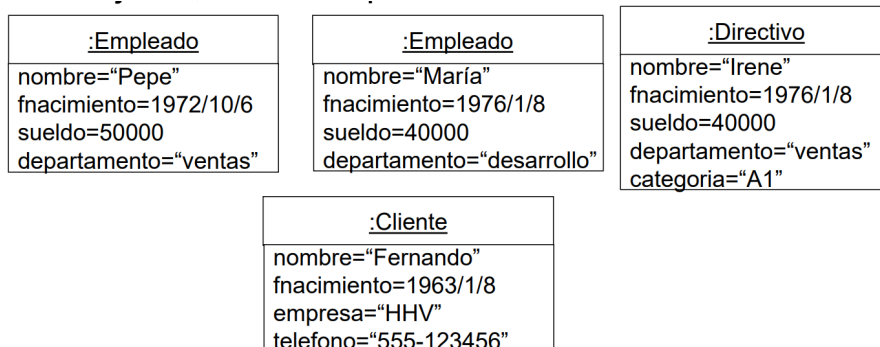
- Son datos comunes de las clases.
- Los datos de la clase padre se heredan por la clase hija.
- Las clases hijas añaden datos adicionales → según vamos heredando añadimos nueva información.

Objetos

- Metemos los datos de los atributos en objetos → Instancias de clases en tiempo de ejecución.
- Los atributos van a necesitar un valor

Ejemplos de objetos:





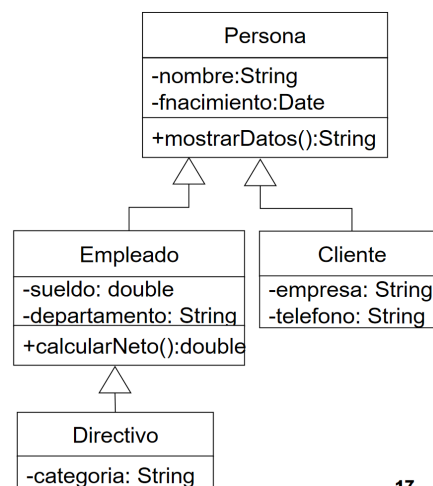
Métodos

Son comportamiento común de los objetos.

- Los métodos de la clase padre se heredan por la clase hija.
- La clase hija puede añadir métodos adicionales.


i Visibilidades en UML:

- El **+** es **público**
- El **-** es **privado**



17

Ejemplos de comportamiento

- `mostrarDatos()` → al estar diseñada en persona, solo muestra los datos de persona
→  no muestra el sueldo del empleado o la empresa del cliente

Especialización de comportamiento

- Especialización de métodos → Acciones adicionales
- Modifican el comportamiento de los métodos de la clase padre.

Solución al problema:

Hazte cliente de BBVA y ... ahórrate 6 meses de suscripción

WUOLAH
+ BBVA

NETFLIX

Spotify

HBOmax

Disney+

PlayStation.Plus

DAZN

Ahora, si te abres una Cuenta Online en BBVA, te reembolsamos una de estas suscripciones durante 6 meses (hasta 9,99€/mes) al pagarla con tu tarjeta Aqua Débito

Promoción solo para nuevos clientes de BBVA. Válida hasta el 30/06/2023. Estas empresas no colaboran en la promoción.

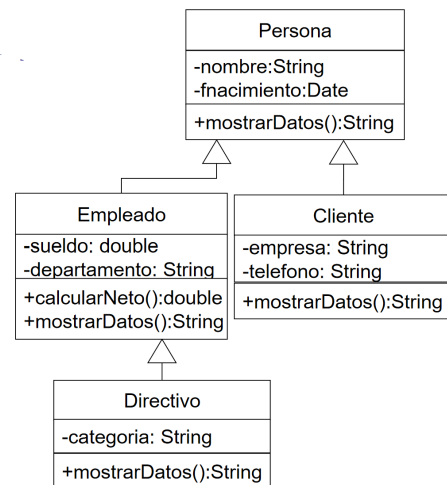
1/6

Este número es indicativo del riesgo del producto, siendo 1/6 indicativo de menor riesgo y 6/6 de mayor riesgo.

BBVA está adherido al Fondo de Garantía de Depósitos de Entidades de Crédito de España. La cantidad máxima garantizada es de 100.000 euros por la totalidad de los depósitos constituidos en BBVA por persona.

Ahora, `mostrarDatos()` muestra además de los datos de la persona, los datos del empleado (si es empleado), los datos del directivo (si es directivo) y los datos del cliente (si es cliente).

- Pueden llamar al método heredado original.



Lo importante de esta asignatura es tener un **buen diseño**

Ventajas

- Extensibilidad de los diseños
 - Mediante herencia → Añadir nuevas clases, extender el comportamiento de métodos.
 - Mediante encapsulamiento → el usuario de una clase no tiene acceso a detalles innecesarios.
- Potencia la reutilización.

1.2. Programación Estructurada

Teorema de la programación estructurada → “secuencia, selección, e iteración—son suficientes para expresar cualquier función computable”.

- Diseño de arriba a abajo
- Independencia de los módulos

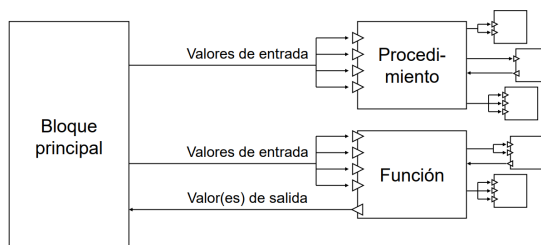
- Separación de algoritmos y estructuras de datos.

Estructura de un módulo:

- Datos de entrada
- Datos de salida
- Descripción funcionalidad

Ventajas:

-



1.3. Tipos Abstractos de Datos

Consiste en:

- Estructura de datos → almacena información.
- Funcionalidad → operaciones que se realizan sobre el tipo de datos.

Ventajas:

- Fácilmente extensibles
- Encapsulamiento
- Utilización del tipo de datos independientemente de su programación interna
- Mayor modularidad

1.4. Programación orientada a objetos

Programación orientada a objetos =

- Soporte sintáctico para los TADs
- Prestaciones asociadas a las jerarquías de clases
- Cambio de perspectiva

2. Objetos y Clases

Elementos de la Prog. Orientada a Objetos

- **Objetos** → atributos + métodos
- **Métodos** → operaciones sobre objetos
- **Clases** → categorías de objetos con propiedades y operaciones comunes
- **Herencia** → jerarquía de clases
- **Relaciones** → Relaciones entre objetos → objetos compuestos

2.1. Estructura

- **Clase** → Plantilla para crear objetos
- **Objeto** → Variable con tipo = una clase → Da valores a los atributos + reacciona a los métodos

2.2 Ciclo de vida de un objeto

1. Creación → Reserva de memoria + inicialización atributos
 - a. `Empleado x = new Empleado(...);`
2. Manipulación
 - a. Acceso a los atributos → `x.nombre`
 - b. Invocación de métodos → `x.calcularNeto();`
3. Destrucción → Liberar memoria + destruir partes internas + eliminar referencias al objeto destruido
 - a. Está implícita

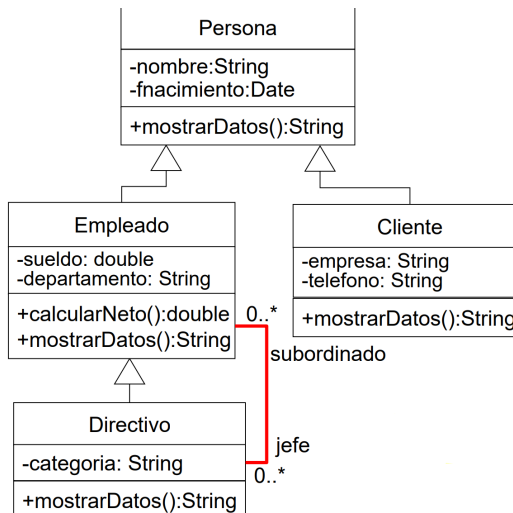
2.3. Relaciones con otros objetos

Asociación

Expresa una relación entre los conceptos

- **Roles** (nombre) → Un *Directivo* tiene *Empleados subordinados* & un *Empleado* tiene *jefes Directivos*.

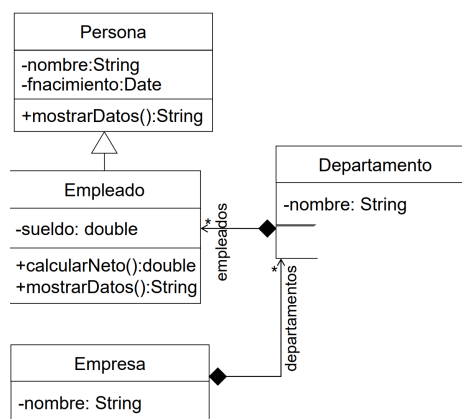
- Además, los *Directivos* son *Empleados* \Rightarrow Pueden tener jefes
- **Cardinalidad** (número) \rightarrow Un *Directivo* tiene 0 o más *Empleados* subordinados & un *Empleado* tiene 0 o más jefes *Directivos*.
 - Formato: `inicio..final`
 - Si no se pone nada $\equiv 1$ \oplus `x..*`
 $\equiv [x, \infty]$ \oplus `*` $\equiv [0, \infty]$



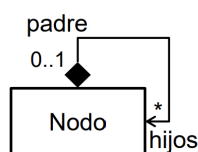
Contenido

Define una relación de **agregación fuerte** = “Está formado por”

- Borrar el contenedor borra el contenido \rightarrow si se elimina el objeto grande, se borran todos los objetos que están contenidos en él.
- **⚠ Un objeto no puede tener 2 contenedores & no se permiten ciclos**
 \rightarrow No se permite que un empleado este en varios departamentos
- Si fuese relación departamento-persona en vez de departamento-empleado se eliminan las personas y los empleados.



Composición de Árbol:



Hazte cliente de BBVA y ... ahórrate 6 meses de suscripción

WUOLAH
+ BBVA

NETFLIX

Spotify

HBOmax

Disney+

PlayStation.Plus

DAZN

Ahora, si te abres una Cuenta Online en BBVA, te reembolsamos una de estas suscripciones durante 6 meses (hasta 9,99€/mes) al pagarla con tu tarjeta Aqua Débito

Promoción solo para nuevos clientes de BBVA. Válida hasta el 30/06/2023. Estas empresas no colaboran en la promoción.

1/6

Este número es indicativo del riesgo del producto, siendo 1/6 indicativo de menor riesgo y 6/6 de mayor riesgo.

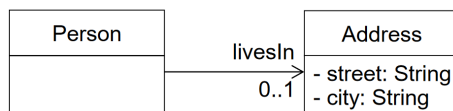
BBVA está adherido al Fondo de Garantía de Depósitos de Entidades de Crédito de España. La cantidad máxima garantizada es de 100.000 euros por la totalidad de los depósitos constituidos en BBVA por persona.

Agregación

Define una asociación entre un objeto contenedor y sus objetos contenidos = contención débil → si se borra el contenedor no se borra el contenido

- Rombo hueco
- Si eliminamos el departamento no eliminamos empleados

2.4. Navegación



```
class Address {
    private String street;
    private String city;
}
class Person {
    private Address livesIn;
}
```

Indica si desde una clase podemos navegar a otra mediante la asociación.

- Se muestra con una flecha

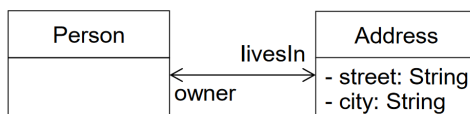
Si no se especifica la cardinalidad → cardinalidad = 1 → **livesIn** de *Person* no puede ser *null*.

Pero con **0..1**, *livesIn* puede ser *null*, puede existir o no.

Navegación bidireccional

Navegación entre ambas clases (flechas en ambos sentidos ≡ omitir las flechas)

Ejemplo:

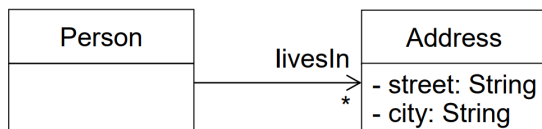


```
class Address {
    private String street;
    private String city;
    private Person owner;
}
class Person {
    private Address livesIn;
}
```

Navegación 1 a muchos

Caso: Una persona que puede tener

varias direcciones. → Desde un objeto instancia de Person se pueda acceder a varios objetos Address.



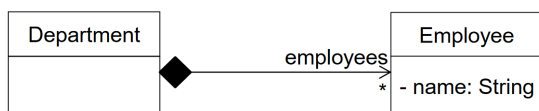
```

class Address {
    private String street;
    private String city;
}
class Person {
    private Address[] livesIn;
}
  
```

2.5. Composición + navegación

Unidireccional

- El contenido no pertenece a más de un contenedor
- Al borrar el contenedor se borra su contenido



Si se borra el departamento recorremos todos los empleados y se eliminan los empleados

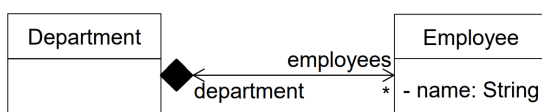
El employee solo puede estar en un departamento y no en más de uno (aunque si el rombo fuese transparente si que se puede)

```

class Employee {
    private String name;
}
class Department {
    private Employee[] employees;
}
  
```

Bidireccional


→ Una colección en un extremo + una referencia en el otro



```

class Employee {
    private String name;
    private Department department;
}
class Department {
    private Employee[] employees;
}
  
```

2.6. Clase abstracta

No se puede instanciar →  `ElementoFabrica ef = new ElementoFabrica()`

- Representación: en cursiva
- Sirve para determinar elementos en común entre dos clases hijas

2.7. Métodos (y atributos) de objeto y clase

Atributos y métodos de objeto:

- Cada objeto guarda su propio valor de atributos & para ejecutar un método necesitamos el objeto

Atributos y métodos de clase:

- Todos los objetos comparten el valor del atributo → Ejemplo: cambiar valor de PI
- El método se ejecuta en el contexto de la clase → no hay porque instanciar un objeto de esa clase para poder invocarlo → No hace falta crear un elemento math para aplicar la función sqrt.

3. Encapsulamiento

Controlar el acceso a los atributos y métodos de una clase desde el exterior.

- Elemento **privados** (-) → no son accesibles ni visibles desde el exterior
 - un método privado no se puede invocar desde un objeto distinto
- Elementos **públicos** (+) → accesibles desde el exterior
 - un método público se puede invocar desde un objeto distinto.
- Elementos **protegidos** (#) → accesibles desde la clase y subclases

 Solo hay que exponer la interfaz relevante al resto del sistema.

Todos los atributos de una clase se declaran como privados y se declaran métodos get (acceso) y set (modificación) de los atributos necesarios. (si un valor está derivado de otro, no se almacena el valor y se calcula en el momento con el get).

Código de ejemplo → [enlace](#)

4. Herencia y Polimorfismo

Herencia → Todo lo definido en la clase padre van a estar accesibles en las clases hijas

Subtipado → Podemos usar una variable tipo persona para pasar de empleado a persona.

Luego, si se realiza `x.mostrarDatos()` se ejecuta la función del objeto al que se hace referencia, es decir al de empleado o directivo. → **Ligadura dinámica**

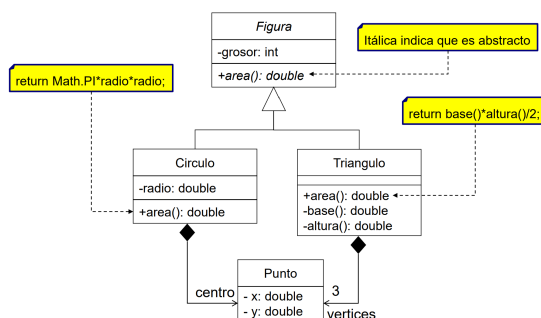
```
Persona x;  
Empleado y = new Empleado();  
Directivo z = new Directivo();  
x = y; // un Empleado es Persona  
x = z; // un Directivo es Persona
```

- Sobreescritura de métodos → `mostrarDatos()`

Métodos abstractos

Métodos sin código → Definidos en clases abstractas & deben sobrescribirse obligatoriamente en las clases que la heredan. → Obligas a que las de abajo implementan un método determinado.

- Definen una interfaz común para todas las subclases de la clase padre



Herencia Múltiple

- Una clase con varios padres → se heredan los atributos y métodos de todos los padres.

Polimorfismo

= “muchas formas” → Presentar la misma interfaz para distintas formas. Tipos:

- **Subtipado** → herencia + sobreescritura de métodos

Hazte cliente de BBVA y ... ahórrate 6 meses de suscripción

WUOLAH
+ BBVA

NETFLIX

Spotify

HBOmax

Disney+

PlayStation.Plus

DAZN

Ahora, si te abres una Cuenta Online en BBVA, te reembolsamos una de estas suscripciones durante 6 meses (hasta 9,99€/mes) al pagarla con tu tarjeta Aqua Débito

Promoción solo para nuevos clientes de BBVA. Válida hasta el 30/06/2023. Estas empresas no colaboran en la promoción.

1/6

Este número es indicativo del riesgo del producto, siendo 1/6 indicativo de menor riesgo y 6/6 de mayor riesgo.

BBVA está adherido al Fondo de Garantía de Depósitos de Entidades de Crédito de España. La cantidad máxima garantizada es de 100.000 euros por la totalidad de los depósitos constituidos en BBVA por persona.

- **Ad-hoc** → sobrecarga de funciones
 - Mismo nombre de método con distintos argumentos
- **Paramétrico** → genericidad

Ligadura Dinámica

```
Persona x;  
Empleado y = new Empleado();  
x = y;  
x.mostrarDatos ( ) // (1)?  
y.mostrarDatos ( )
```

¿Qué método se ejecuta?

- Ligadura estática (C++) → el de *Persona*
- **Ligadura dinámica** (Java) → el de *Empleado*

i hasta el tiempo de ejecución el compilador no sabe que método se ejecuta

5. Resumen y Conclusiones

- Conceptos: clases, objetos, encapsulación, polimorfismo y herencia
- Ventajas: Extensibilidad, reutilización

Parte II: Malas Soluciones en Diseños

Malas Soluciones en Diseños

- Se repite información (atributos)
- Se repite código

Diseño tipoPersona → una única clase donde estan todos los campos + un campo adicional que indica el tipo de persona

- **X** Los objetos contienen atributos innecesarios
- **X** El código se hace innecesariamente complicado → mucha complejidad ciclomática (muchos caminos de código (muchos if))

- **✗** Para añadir un nuevo tipo de Empleado hay que modificar mucho código y añadir muchos atributos a persona

Solución óptima:

Asociaciones vs. atributos identificadores

El de la derecha está mal. Para acceder a TarjetaCredito se pasa por banco.

2. El de la izquierda está mal

Importancia del diseño

- Hay muchos diseños