

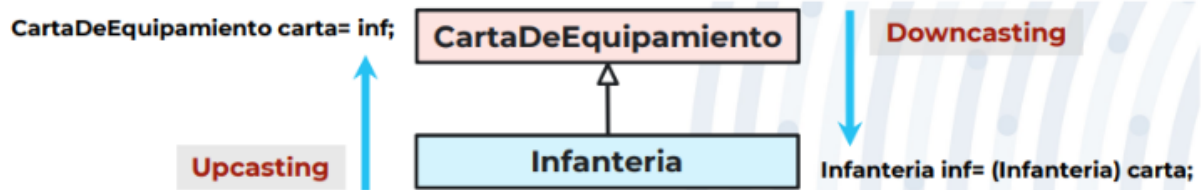
## ¿Cómo se deben definir los atributos de una Clase?

Siempre privados.

## ¿Un método static se puede heredar?

Un método static no se hereda, ya que estará disponible para todas las clases desde el arranque del programa.

## Ejemplos y decir cuando es Upcasting y cuando Downcasting



## Saber la diferencia entre public, private, protected, static, default

Son tipos de acceso, en cuanto a clases, métodos, atributos.

Atributos public, se comparten entre todas las clases

Atributos default, se comparte en clases del mismo paquete

Atributos protected, se comparten en las subclases

Atributos private, solo se usan en la propia clase (encapsulación)

Atributos static, se cargan en el arranque del programa y están disponibles para todas las clases. Se suele utilizar junto a final para definir constantes.

## Características de un método default

En **resumen**, los métodos por defecto solo se pueden definir en interfaces, invocando en su cuerpo cualquier tipo de método. Además, son heredados por clases e interfaces.

## Ventajas de Herencia sobre Composición

La Herencia da un inicio de desarrollo y un diseño del software más rápido y sencillo respectivamente. Además, la adaptación a cambios y la extensibilidad es mucho más fácil de realizar, sin embargo, en jerarquías profundas, la composición cobra más sentido. La diferencia principal entre estas dos, es que la herencia se trata de una semántica del estilo (p.e) en un caso de persona Empleado. En caso de herencia sería que un Empleado es una persona, sin embargo, en caso de composición, un empleado tiene los atributos de una persona.

## Ejemplos y decir cuando Autoboxing y cuando Unboxing

Autoboxing(Envolver) de primitivo a Wrapper

Unboxing(desenvolver) de Wrapper a primitivo

```

public static void main(String[] args) {
    Integer aa = 10;
    Integer bb = 20;
    if(aa > bb)
        System.out.println("aa + bb = " + suma(aa, bb));
}

public static int suma(int a, int b) {
    return a + b;
}
    
```

autoboxing

unboxing

## Clase abstracta que es? Propiedades.

Se trata de una clase no instanciable, que puede tener constructores pero no puede utilizar new. Estas pueden tener métodos y atributos, pero suelen ser abstractos y constantes respectivamente. Facilita la reutilización de código enormemente.

## **Para que sirve el método super?**

Sirve para llamar elementos (atributos, métodos, constructores) de una clase inmediatamente superior (clase base) desde una clase derivada.

## **Como se implementa el método equals?(Ejemplo)**

A continuación, se muestra un ejemplo de una implementación de equals para un criterio de igualdad por referencia de los objetos.

```
public boolean equals(object A){  
    return this == A;  
}
```

## **Comos e definen las constantes en Java**

final static constante

## **Para que sirve static?**

La palabra **static**, hace que se almacene en memoria estática, permitiendo que esté disponible desde el inicio del programa sin necesidad de crear objetos (instanciar su clase).