



Herencia

Facultad de Ciencias

Diana Isabel Ramírez García

Introducción a Ciencias de la Computación 2021-1

¿Para qué nos sirve la herencia?

Nos sirve para definir una clase C' tal que es muy similar a una clase C existente.

¿Cómo funciona?

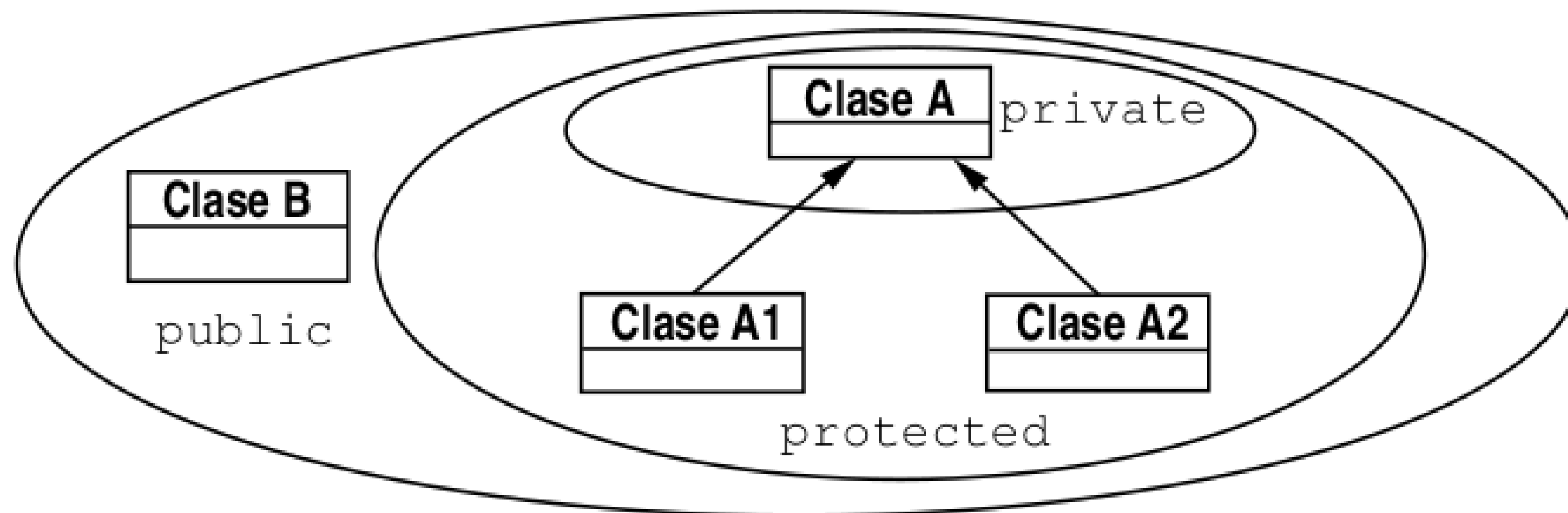
La clase C' tendrá toda la estructura y comportamiento definido en la clase C . Denominamos a C como: **superclase** o **clase padre**. Mientras que C' como: **subclase** o **clase derivada**.

Ventajas:

- Desarrollo de programas de manera incremental
- Evitamos duplicidad de código
- Es posible extender clases sin necesidad de tener el código de ellas.
- Se crean clases de menor tamaño.

Herencia y modificadores

Si deseamos que atributos o métodos de una clase C sean privados excepto para las subclases, es necesario utilizar la palabra reservada **protected** en lugar de private.



Sobrescritura vs sobrecarga

- Los métodos se encuentran en distintas clases (ligadas por herencia)
- La firma es exactamente la misma

- Los métodos se encuentran en la misma clase.
- Únicamente comparten el nombre en su firma

Polimorfismo

- Esta característica nos permite referirnos a distintos métodos a pesar de llevar el mismo nombre. Lo cual resulta de utilidad en una jerarquía de herencia para poder diferenciar los métodos.

Polimorfismo

- Hasta el momento de ejecución se determina a que método se refiere . Por ejemplo, un gato y un perro son animales pero emiten distintos sonidos

```
Gato[] arr = new Gato[3];
```

```
...
```

```
arr[0].hacerSonido();
```

guau

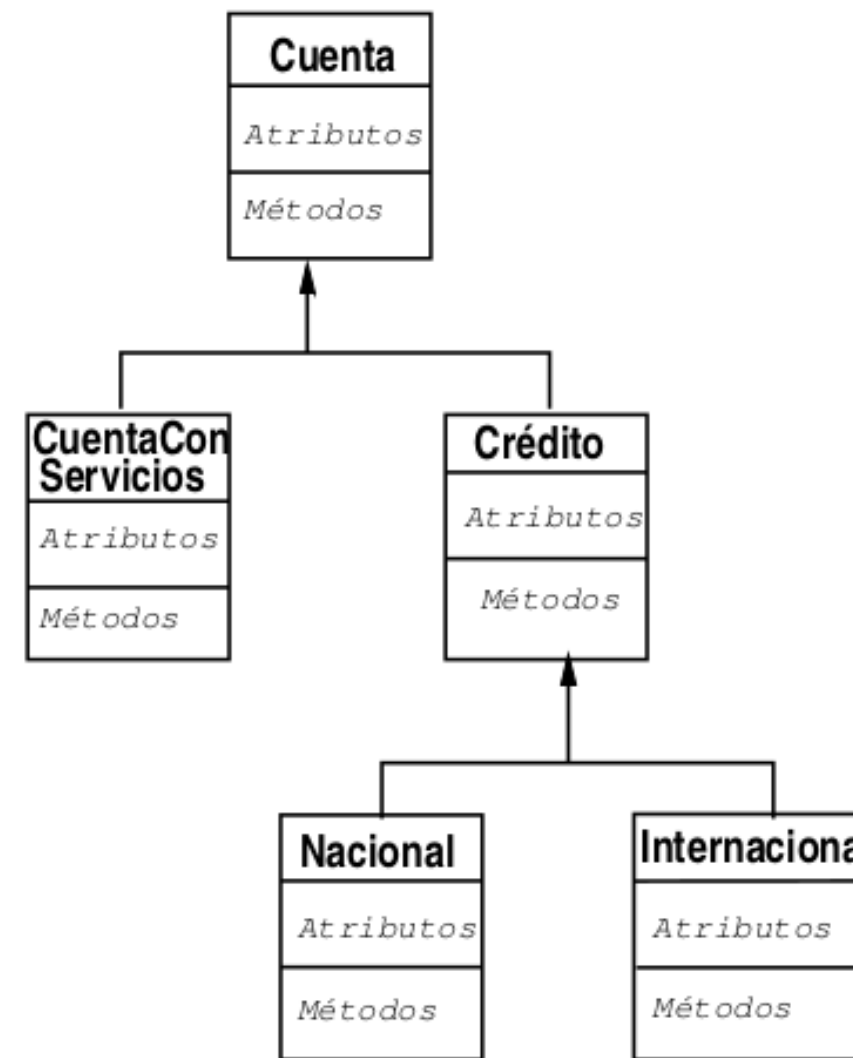


miau

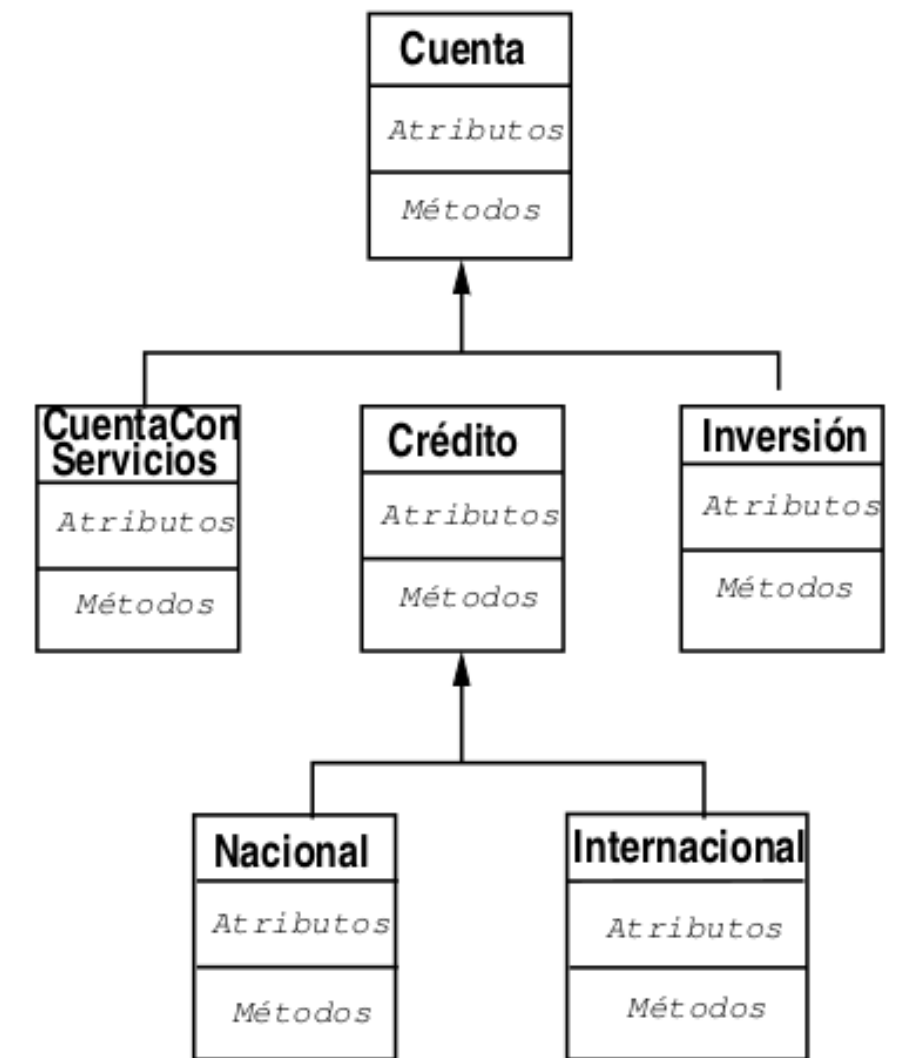


Jerarquía de Clases

- La herencia tiene la característica de ser transitiva, es decir un superclase puede ser una subclase de otra clase y una subclase ser superior a otra.



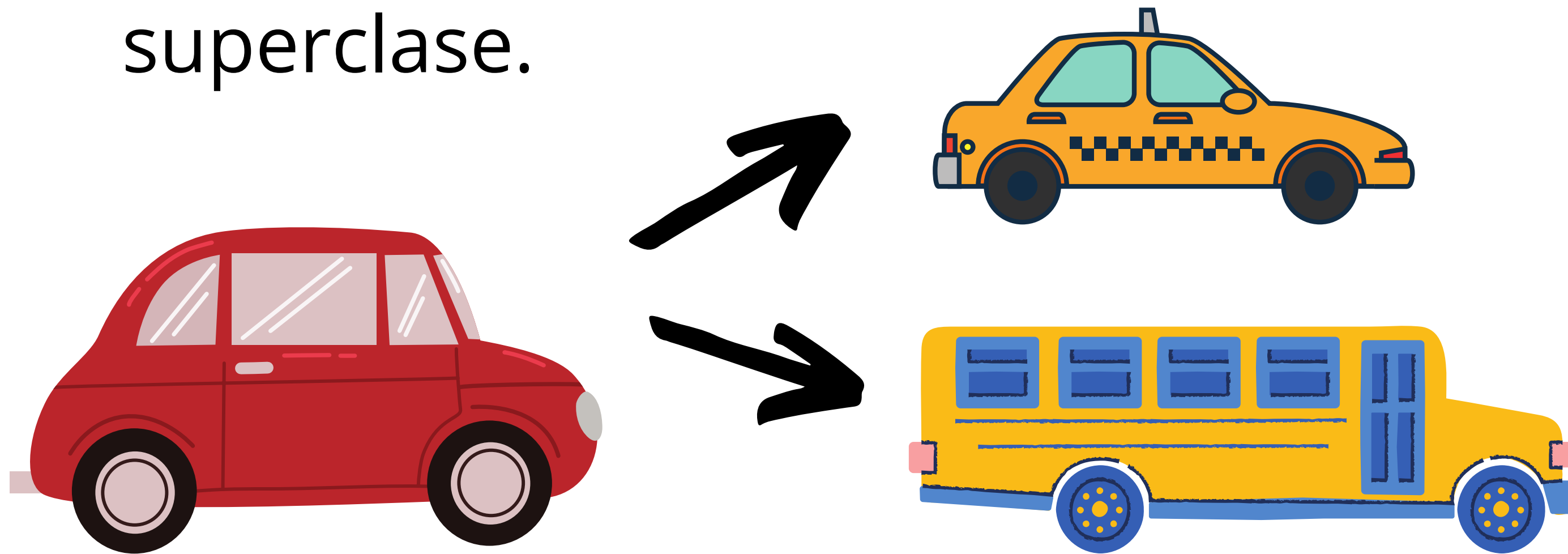
a) Crecimiento a lo largo



b) Crecimiento a lo ancho

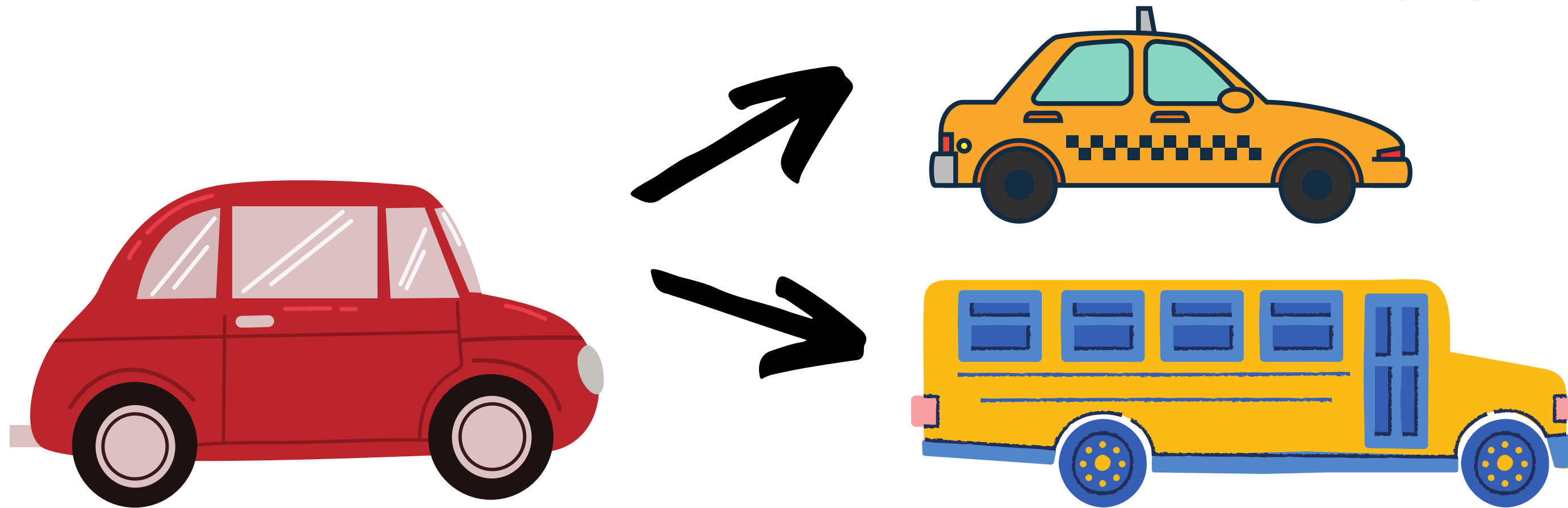
Compatibilidad

- Podemos utilizar un objeto de la subclase en cualquier parte donde se pueda utilizar la superclase.



Compatibilidad

- Todo taxi o autobús es un automóvil :
Automovil coche = new Taxi(...);
Automovil coche2 = new Autobus(...);

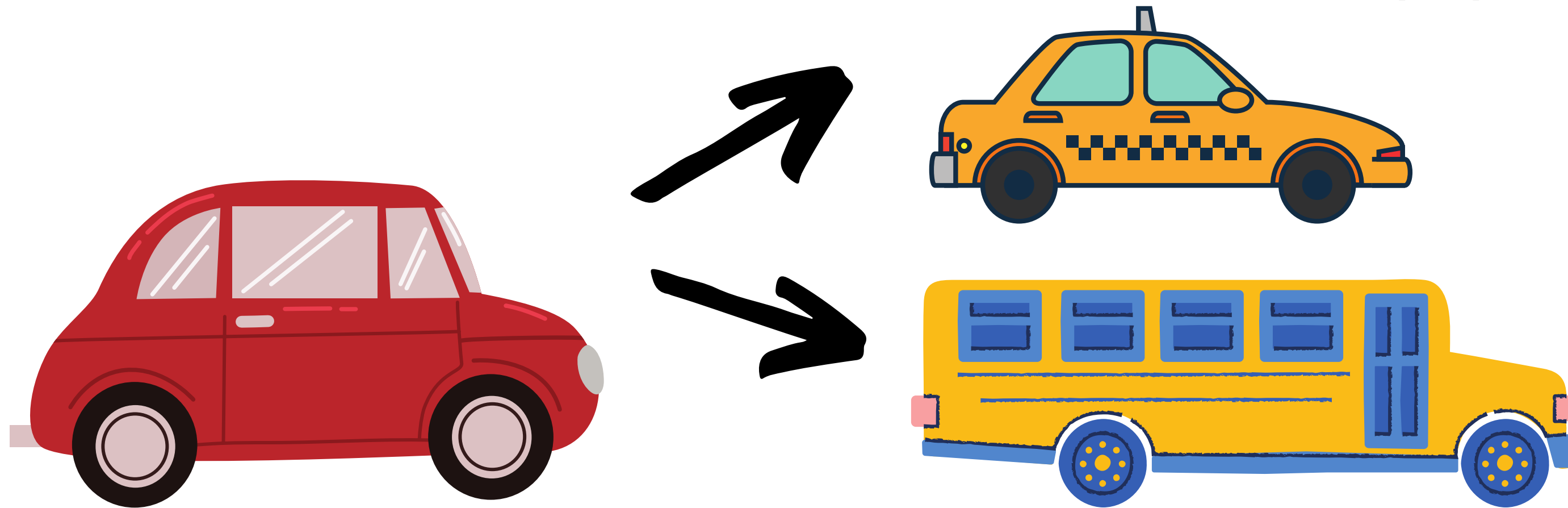


Compatibilidad

- Pero no todo automóvil es un taxi o autobús :

Taxi coche = new Automovil(...);

Autobus coche2 = new Automovil(...);



Especialización vs Generalización

- Cuando tenemos una clase existente y deseamos añadir nuevos elementos en la estructura o comportamiento.

- Cuando contamos con distintas clase similares tal que podemos agruparlas en una clase más general.

Clase Object

- Es una clase de la cual todos los objetos heredan de manera implícita tal que cuenta con los métodos básicos como : equals() y toString().

instanceOf

- Operador para determinar si un objeto es de la misma clase.

Sea "o" un objeto de Object y Automovil una clase.

boolean ejem = o instanceof Automovil;

Bibliografía

- López, A.. (2014). Introducción al desarrollo de programas con Java. México: Las prensas de Ciencias.