

# Tipo de objeto y tipo de referencia

# Fundamentos

- En Java se puede asignar una referencia a un objeto de un tipo en una variable del tipo de su superclase

```
Object ob=new String("hello");
```

- El tipo de la referencia (Object) es superclase del tipo del objeto (String)

- Se aplica a varios niveles:

```
class Clase2 extends Clase1{}  
class Clase3 extends Clase2{}
```



```
Clase1 c=new Clase3();
```

# Llamadas a métodos

➤ Con esta referencia se puede llamar a métodos del objeto, pero **SOLO** a aquellos que han sido heredados o sobrescritos:

```
Object ob=new String("hello");
```

```
System.out.println(ob.toString()); //correcto, llamada a método  
                                  //toString() de String
```

```
if(ob.equals("hello")){} //correcto, llamada a equals() de String
```

```
System.out.println(ob.length()); //error de compilación, length() es un  
                                  //método propio de String, no existe en Object
```

# Casting entre tipos objeto

- Para obtener una referencia al tipo original del objeto podemos efectuar un casting:

```
String s=(String)ob;
```

- El compilador permite hacer casting de una referencia de un tipo a cualquier tipo de sus subclases, pero si el objeto no es de ese tipo se producirá una `ClassCastException`:

```
Integer r=(Integer)ob;
```

Esta instrucción compila correctamente, pero al ejecutarla se producirá una excepción `ClassCastException`, pues el tipo de objeto referenciado por `ob` es `String`, no `Integer`