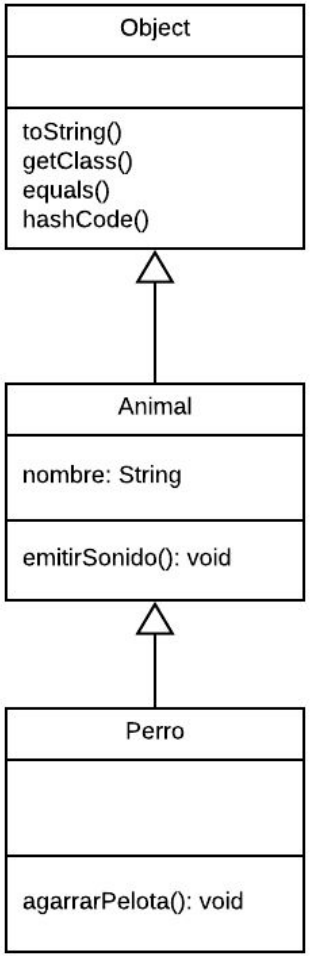
Clase 8 - Polimorfismo & Casting

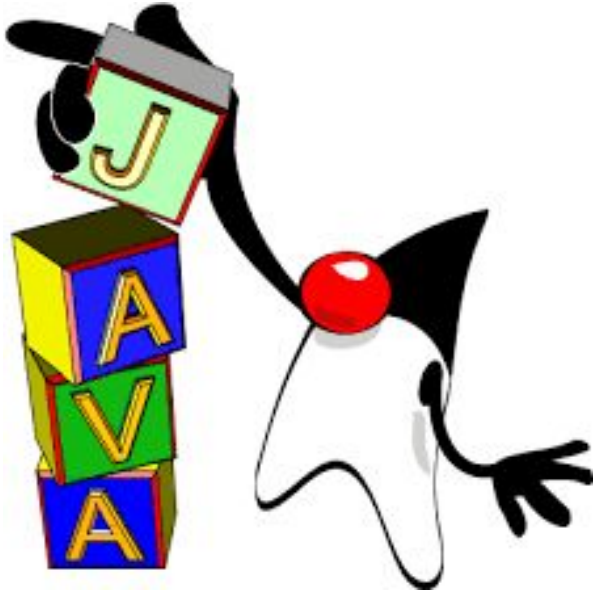
Programación & Laboratorio III

UML - Caso de ejemplo



Agenda

- **Polimorfismo con Object**
- Referencias polimórficas
- Casting



Polimorfismo con Object - Ejemplo

```
public static void main(String [] args) {

     Perro [] perros = new Perro[2];

     Perro perro1 = new Perro(“Ayudante de Santa”);
     Perro perro2 = new Perro(“Procer”);

     perros[0] = perro1;
     perros[1] = perro2;

     for (int i = 0; i < perros.lenght; i++) {
         System.out.println(perros[i].emitirSonido());
     }
}
```

El arreglo puede contener sólo instancias de Perro porque el tipo declarado es Perro

Se crean dos INSTANCIAS de la clase Perro

Se asignan las instancias creadas anteriormente en las posiciones 0 y 1 respectivamente

Accedemos a métodos de la clase Perro porque sabemos que tenemos INSTANCIAS de la clase Perro almacenadas en el arreglo

Polimorfismo con Object - Ejemplo (2)

```
public static void main(String [] args) {

    Object [] objects = new Object[2];

    Object object = new Object();
    Perro perro = new Perro("Ayudante de Santa");

    objects[0] = object;
    objects[1] = perro;

    for (int i = 0; i < objects.lenght; i++) {
        System.out.println(objects[i].emitirSonido());
    }
}
```

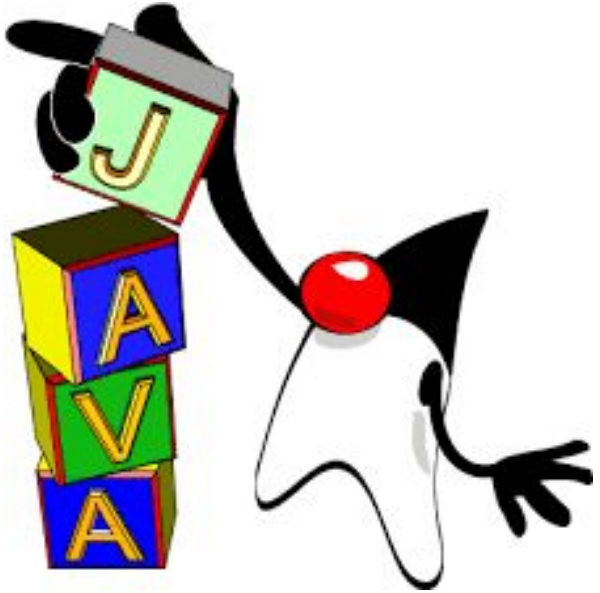
El arreglo puede contener instancias de Object y cualquiera de las subclases en la jerarquía

Se crean dos INSTANCIAS: una de la clase Object y otra de la clase Perro

No compila! Pero cómo sabemos que estamos leyendo una instancia de Perro?

Agenda

- ~~Polimorfismo con Object~~
- **Referencias polimórficas**
- Casting



Referencias polimórficas

- El problema de tener todo polimórficamente como Object es que lo que queremos almacenar tiene a perder su verdadera esencia.

```
Object [] objects = new Object[2];
Object object = new Object();
Perro perro = new Perro(“Ayudante de Santa”);

objects[0] = object;
objects[1] = perro;

Object o = objects[0];
int code = o.hashCode();
o.emitirSonido();
```

Operación válida

Error en tiempo de compilación porque no es un método de la clase Object

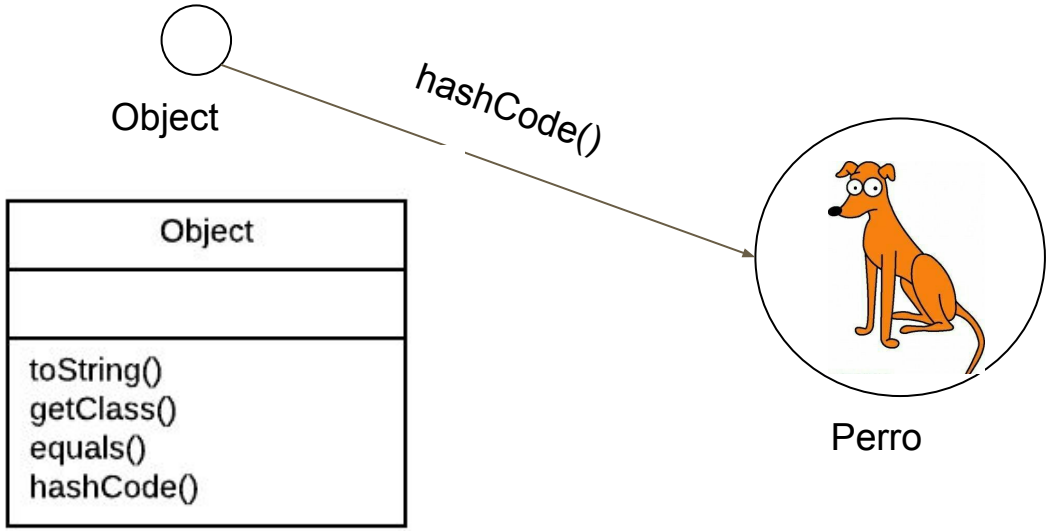
Referencias polimórficas (2)

```
Object [] objects = new Object[2];
Object object = new Object();
Perro perro = new Perro(“Ayudante de Santa”);

objects[0] = object;
objects[1] = perro;

Object o = objects[1];
int code = o.hashCode();
```

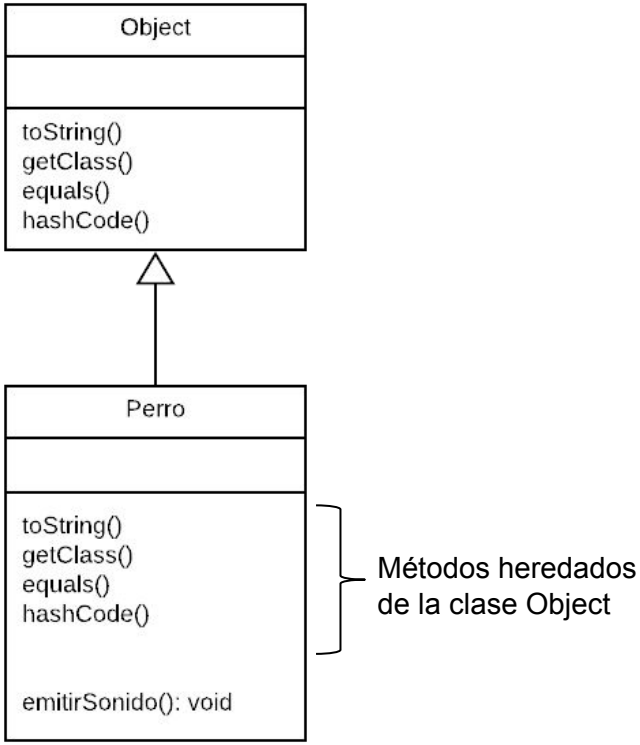
Retorna una referencia de Object de una instancia de Perro

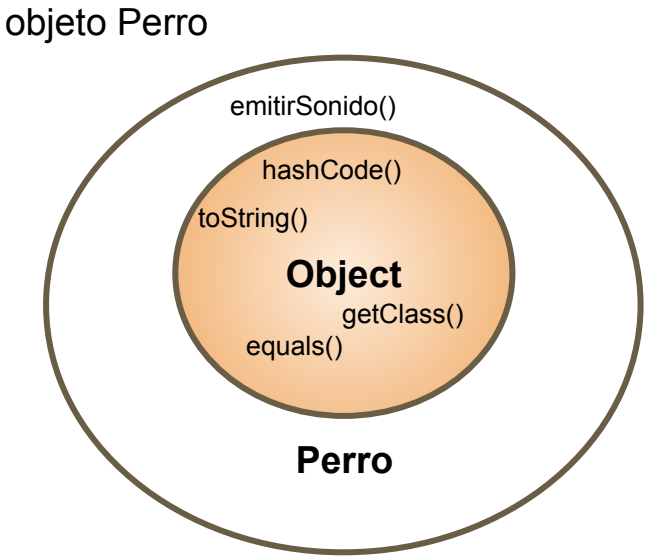
Referencias polimórficas (3)

- El método que se invoca debe estar en la clase del tipo declarado, sin importar el tipo del que realmente es el objeto.
- El compilador verifica el tipo de la referencia, no el tipo del objeto.



Referencias polimórficas (4)

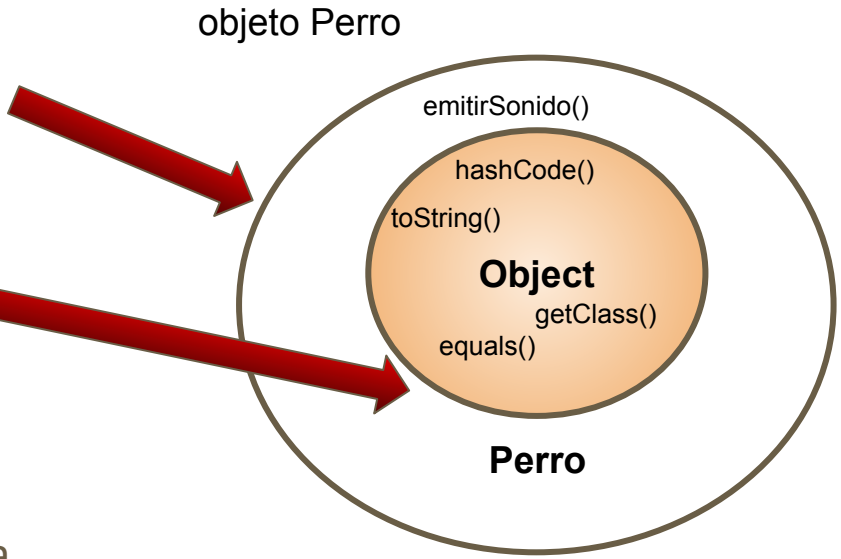




- El objeto Perro contiene la clase **Perro** como parte suya y también la clase **Object**.

Referencias polimórficas (5)

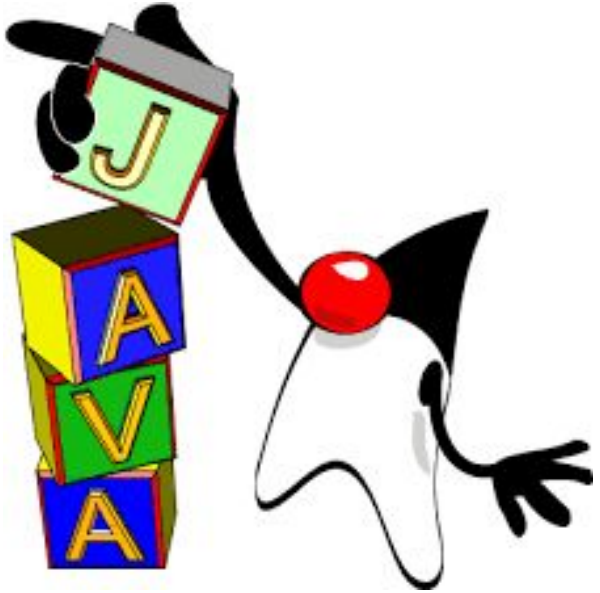
```
Perro p = new Perro(“Ayudante de Santa”);

Object o = p;
```



- La referencia a Object sólo tiene acceso a los métodos de Object.
- La referencia a Perro tiene acceso a los métodos de Perro y de Object.

Agenda

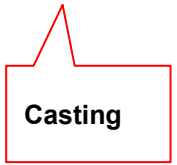
- ~~Polimorfismo con Object~~
- ~~Referencias polimórficas~~
- **Casting**



Casting

- Se denomina proceso de casting a la “conversión” de una referencia de un tipo a otro.

```
Object [] objects = new Object[2];
Perro perro = new Perro(“Ayudante de Santa”);

objects[0] = perro;

Perro p = (Perro) objects[0];
```



Casting (2)

- IMPORTANTE: Debemos estar seguros de que lo que estamos “casteando” pertenece a la clase de conversión.

```
public static void main(String [] args) {

   Object [] objects = new Object[2];
   Perro perro = new Perro(“Ayudante de Santa”);

   objects[0] = perro;

   Gato g = (Gato) objects[0];
}
```

ERROR EN TIEMPO DE EJECUCIÓN: Exception in thread "main" java.lang.ClassCastException: Perro cannot be cast to Gato

Casting (3)

- Para asegurarnos del casting correcto, se puede utilizar el operador instanceof.

```
public static void main(String [] args) {

   Object [] objects = new Object[2];
   Perro perro = new Perro(“Ayudante de Santa”);

   objects[0] = perro;

   if (objects[0] instanceof Gato) {
      Gato g = (Gato) objects[0];
   }
}
```

Bibliografía oficial

- https://docs.oracle.com/javase/specs/jls/se7/html/jls-5.html
- https://docs.oracle.com/javase/7/docs/api/java/lang/ClassCastException.html
- https://docs.oracle.com/cd/E29028_01/wlp.1034/e14255/com/bea/p13n/expression/operator/Instanceof.html
- https://docs.oracle.com/javase/7/docs/api/java/lang/Object.html