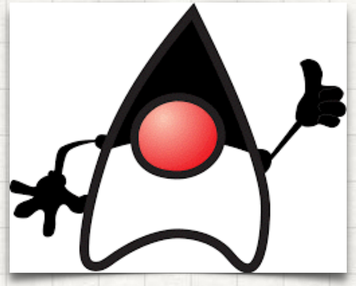
CLONE METHOD

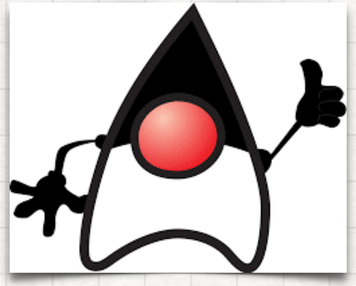


método Object.clone()

```
public class Object

protected Object clone() throws CloneNotSupportedException
```

Crea un nuevo objeto del mismo tipo y copia todos los campos



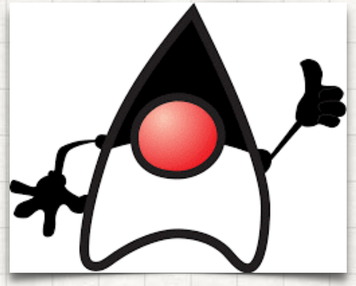
Ampliamos visibilidad

```
public class Test {

  @Override
  public Object clone() throws CloneNotSupportedException {
    return super.clone();
  }
}
```

```
Test test = new Test();
test.clone();
```

```
Exception in thread "main" java.lang.CloneNotSupportedException: clone.Test
  at java.lang.Object.clone(Native Method)
  at clone.Test.clone(Test.java:7)
  at clone.Main.main(Main.java:8)
```

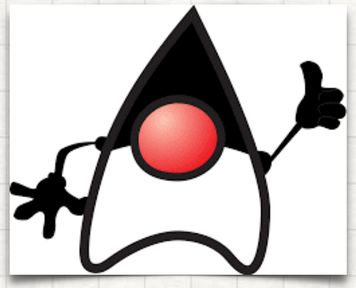


```
public class Test implements Cloneable {

  @Override
  public Object clone() throws CloneNotSupportedException {
    return super.clone();
  }
}
```

```
Test test = new Test();
test.clone();
```

```
try {
    Test copia = (Test) test.clone();
} catch (CloneNotSupportedException e) {
  // tratar error
}
```



Implementación de la clase más cómoda para el cliente:

- No necesita hacer cast del resultado
- No necesita tratar la excepción

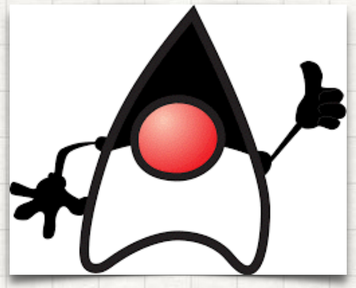
```
public class Test implements Cloneable {

  @Override
  public Test clone() {
    try {
      return (Test) super.clone();
    } catch (CloneNotSupportedException e) {
      // no es posible
      throw new AssertionError();
    }
  }
}
```

Tipo devuelto Test

No exige al cliente tratar la excepcion

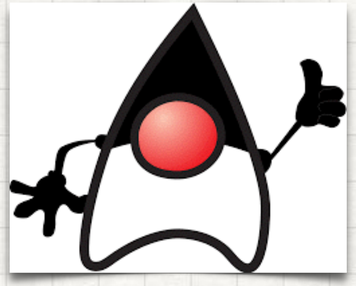
```
Test copia = test.clone();
```



```
private static class Configuracion implements Cloneable {

      ...


  @Override
  protected Configuracion clone()  {
    try {
      return (Configuracion) super.clone();
    } catch (CloneNotSupportedException e) {
      throw new AssertionError();
    }
  }
```



método Object.clone()

PROBLEMAS

- Demasiado extra-linguístico:
  - clone() no es de la interface Cloneable
  - Invocar a super.clone() no es obligatorio, como si sucede con los constructores
- Sigue siendo necesario propagar el clone (deep copy)
- Puede ser problemático no invocar al constructor:
  - No podemos sustituir una variable final

=> Algunos expertos recomiendan usar un "factory method" en lugar de clone

CLONE METHOD