

# Clases envoltorio

## Programación I

Grado en Ingeniería Informática  
MDR, JCRdP y JDGD

# Clases envoltorio (wrappers)

- ▶ Los objetos de estas clases almacenan un valor de tipo primitivo
- ▶ Este valor no puede ser modificado
- ▶ Existe una clase para cada tipo primitivo
- ▶ Se usan para poder representar valores primitivos donde sólo es posible usar objetos
- ▶ Las clases disponen de métodos auxiliares de conversiones de objetos de una clase envoltorio a diferentes tipos primitivos y a String, y de String a ambos.

# Clases envoltorio (wrappers)

- ▶ Cuando es necesario, el lenguaje crea objetos envoltorio o extrae el valor primitivo que contiene.
- ▶ Cuando el compilador espera un objeto de una clase envoltorio y se le pasa un valor primitivo se crea un envoltorio y se pasa automáticamente
- ▶ Cuando el compilador espera un valor primitivo y se le pasa un objeto de la clase envoltorio se obtiene el valor contenido
- ▶ A este proceso se le denomina autoboxing: boxing (de primitivo a objeto) y unboxing (viceversa)

# Clases envoltorio (wrappers)

Tipo primitivo	Clase envoltorio
boolean	Boolean
char	Character
byte	Byte
short	Short
int	Integer
long	Long
float	Float
double	Double
void	Void

# Ejemplo de comparación de referencias y de objetos

```
public class Equivalencia {  
    public static void main(String args[]) {  
        Integer n1=new Integer(45);  
        Integer n2=new Integer(45);  
        System.out.println(n1 == n2);  
        System.out.println(n1 != n2);  
        System.out.println(n1.equals(n2));  
        n1=n2;  
        System.out.println(n1 == n2);  
        System.out.println(n1.equals(n2));  
    }  
}
```

# Ejemplo de autoboxing

```
public class Equivalencia {  
    public static int f(int i){  
        return i>1? i*f (i-1):1;  
    }  
    public static void main(String args[]) {  
        Integer n=45; // = Integer.valueOf(45)  
        int v = n; // = n.intValue();  
        System.out.println("f("+n+") = "+f(n));  
        //"f("+n.toString()+") = "  
        //+Integer.valueOf(f(n.intValue())).toString()  
    }  
}
```

# Bibliografía

- ▶ **The Numbers Classes**

<http://docs.oracle.com/javase/tutorial/java/data/numberclasses.html>

- ▶ **Autoboxing and Unboxing**

<http://docs.oracle.com/javase/tutorial/java/data/autoboxing.html>