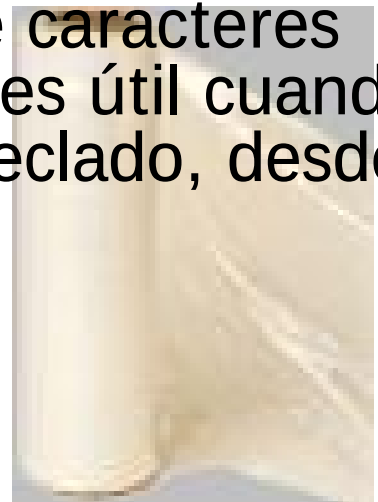

Tema 5: Wrapper classes

Clases “envoltorio”

- Definición

Los tipos de datos primitivos no son objetos pero para que sean tratados como tales se usan las llamadas wrapper class.

- Las wrapper classes implementan objetos “inmutables” ya que una vez convertidos no podrán ser cambiados de nuevo a datos primitivos.
- La ventaja: se puede hacer uso de los métodos existentes para estas clases y para convertir cadenas de caracteres (texto) en números. Esto es útil cuando se leen valores desde el teclado, desde un archivo de texto, etc.



Las wrapper classes

Primitive Data Type	Wrapper Class
boolean	Boolean
byte	Byte
char	Character
short	Short
int	Integer
long	Long
float	Float
double	Double

Métodos y miembros

- Los constructores crean envoltorios a partir de los datos numéricos o de cadenas de caracteres.

- Ejemplo:

```
Integer x=new Integer(34); //Boxing  
Double y= new Double("3.58");  
int z=61;  
Integer w=new Integer(z);
```

- Métodos de instancia para extraer el dato numérico del envoltorio .
xxxValue();

- Ejemplo:

```
int a=x.intValue(); //unboxing  
double b=y.double.Value();
```

Métodos y miembros

- Métodos de clase para crear números a partir de cadenas de caracteres *Xxx.parseXxx(String)*;
- Ejemplo:

```
int i=Integer.parseInt("123");  
double d= Double.parseDouble("34.89");
```
- Métodos de clase para crear envoltorios de números a partir de cadenas de caracteres *Xxx.valueOf(String)*;
- Ejemplo:

```
Integer x=Integer.valueOf("123");  
Double y= Double.valueOf("34.89");
```

```
int x=Integer.valueOf(str).intValue();  
Equiv. int x=Integer.parseInt(str);
```

Métodos y miembros

- Los constructores crean envoltorios a partir de valores lógicos o de cadenas de caracteres.

```
Boolean bo=new Boolean("false");
```

- Si la cadena no es *true* o *false* se toma el valor false, no hay error de compilación.

- Método de instancia para extraer el valor lógico del envoltorio

```
boolean booleanValue();
```

```
boolean bo=(new Boolean("false").booleanValue());
```

- Métodos de clase para crear un envoltorio lógico a partir de cadenas de caracteres *Boolean.valueOf(String)*;

- Ejemplo: `Boolean ob=Boolean.valueOf(str);`

Métodos y miembros

- Constructor único que crea un envoltorio a partir de un *caracter*.

`Character co=new Character('a');`

- Método de instancia para extraer el dato caracter del envoltorio `char charValue()`

`char c =co.charValue();`

- Métodos de clase para comprobar el tipo del caracter:

`boolean isDigit(char)`

`boolean isLetter(char)`

`boolean isLowerCase(char)`

`boolean isUpperCase(char)`

`...`

- Métodos de clase convertir caracteres:

`char toLowerCase(char)`

`char toUpperCase(char)`

`...`

- Ejemplos: `boolean b = Character.isLowerCase('g');`
`char c = Character.toUpperCase('g');`

Boxing y unboxing

- `Integer x=new Integer(34);`
`//Boxing`
- `int a=x.intValue();`
`//unboxing`

Autoboxing y autounboxing

- En el JDK 1.5 las operaciones de inserción y extracción de tipos primitivos incluyen automáticamente las operaciones de boxing y unboxing.

```
int pX= 56;  
Integer wX = pX;    //autoboxing  
int pY= wX;          //autounboxing
```

Adiciones de las wrapper class

- Los wrappers de tipos simples
(*Integer, Long, Double, Char*)
soportan operaciones de bits, como:
 - ✓ highestOneBit,
 - ✓ lowestOneBit,
 - ✓ rotateLeft,
 - ✓ rotateRight,
 - ✓ reverse, etc.