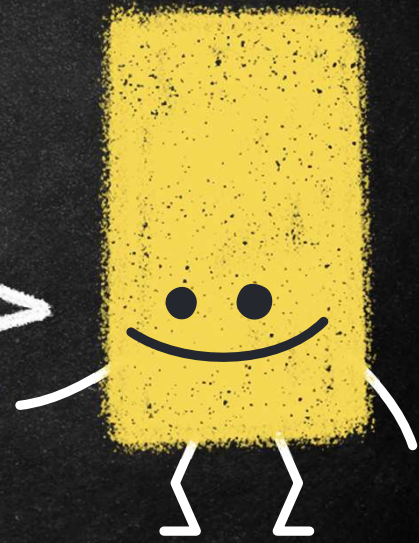


MI PRIMER PROGRAMA



ROADMAP

Preparar entorno

1

Introducción de variables

3

Operadores
Aritméticos, relacionales
y lógicos

5

Hola Mundo

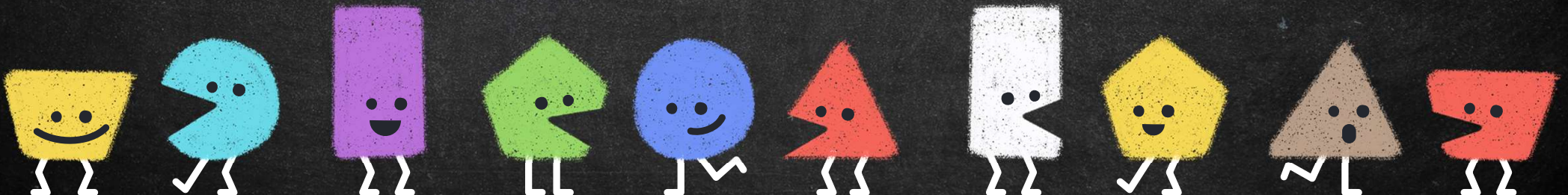
2

Comentarios de código

4

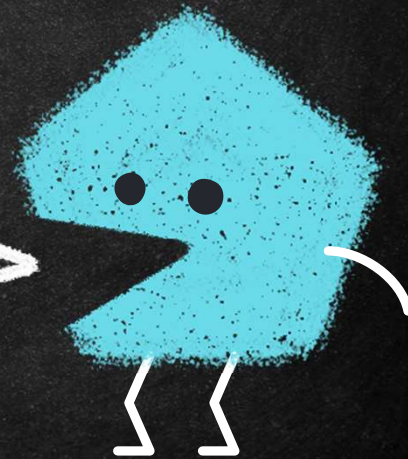
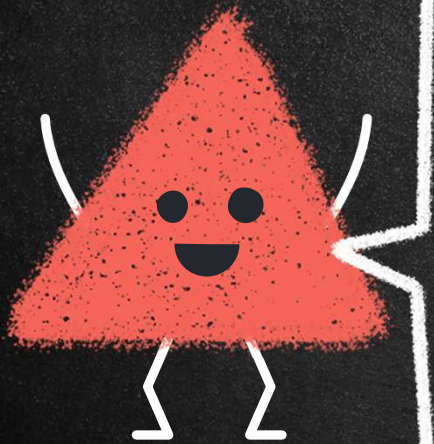
Conversiones de tipo

6



CONVERSIONES DE TIPO

CASTING



CASTING

Operación especial que nos permite realizar una **conversión entre diferentes tipos** de variables.

Un casting **se indica entre paréntesis** con el tipo de dato al que queremos pasar.

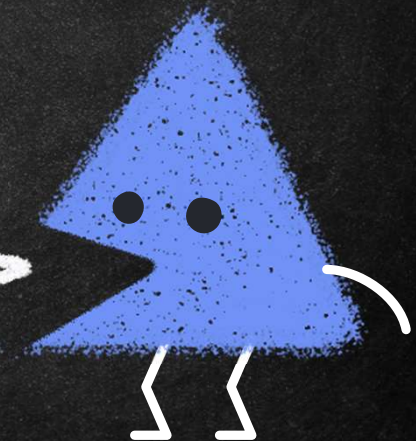
```
int a = 2;  
int b = 3;  
int result = (int)Math.pow(a, b);
```

OVERVIEW MODULE PACKAGE **CLASS** USE TREE PR

SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL:

pow

```
public static double pow(double a,  
                        double b)
```



CASTING IMPLÍCITO

IMPLÍCITO (Widening Casting): Cuando el tipo de dato destino es mayor (en memoria) que el tipo origen, por tanto, no se pierde información.

Para los números, la secuencia de casting implícito es:
byte -> short -> char -> int -> long -> float -> double

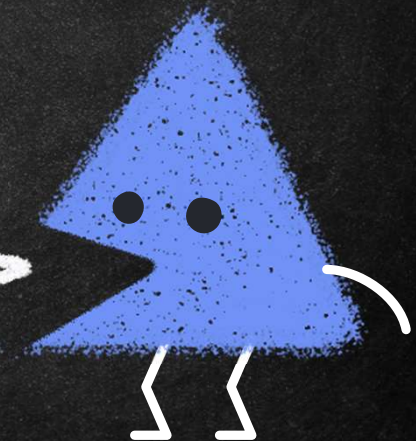
```
int a = 65;
double b = 4.35;
long c = 124578369L;
long d = 1234567890123456789L;
b = a + b;
System.out.println(b);
// int passa a double, se suma al double i es guarda el double

b = c;
System.out.println(b);
// long passa a double

b = d;
System.out.println(b);
// long passa a double i es perden dígitos significatius

a = 'E';
System.out.println(a);
// s'obté el codi de literal 'E' i es converteix a int
```

```
C:\JavaBea\UD2.Ejercicios>java CastingImplicito
69.35
1.24578369E8
1.23456789012345677E18
69
```



CASTING EXPLÍCITO

EXPLÍCITO (Narrowing Casting): Cuando el tipo de dato destino es más pequeño (en memoria) que el tipo origen. En este caso se especifica el casting al tipo que se desea obtener.

Los castings que no es posible realizar dan error de compilación.

```
double d = 65 * 0.33;
System.out.println(d);
// 65 passa a double (implícit) i es realitza el producte

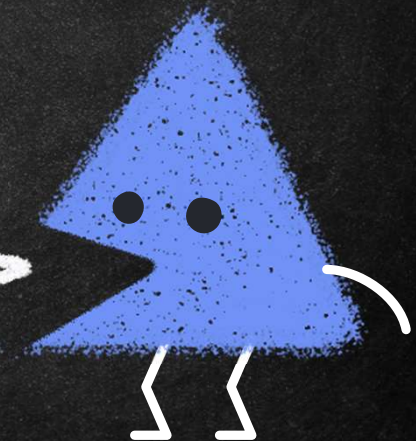
int e = 65 * (int) 0.33;
System.out.println(e);
// el casting afecta al 0.33 i el transforma a int 0

short s = (short) 65 * (short) 3.33;
System.out.println(s);
// el casting afecta al 65 i al 3.33 (el transforma a 3)

s = (short) (65 * 3.33);
System.out.println(s);
// el 65 passa a double (implícit) i el càsting s'aplica al resultat

byte b = (byte) (65 * 0.33);
System.out.println(b);
```

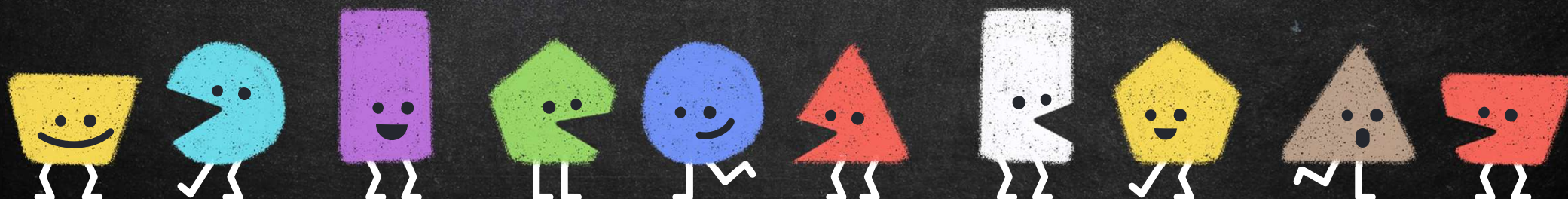
```
C:\JavaBea\UD2.Ejercicios>java CastingImplicitoExplicito
21.45
0
195
216
21
```



Convertir desde	Convertir a...							
	boolean	byte	short	char	int	long	float	double
boolean		no	no	no	no	no	no	no
byte	no		si	cast	si	si	si	si
short	no	cast		cast	si	si	si	si
char	no	cast	cast		si	si	si	si
int	no	cast	cast	cast		si	si*	si*
long	no	cast	cast	cast	cast		si*	si*
float	no	cast	cast	cast	cast	cast		si
double	no	cast	cast	cast	cast	cast	cast	

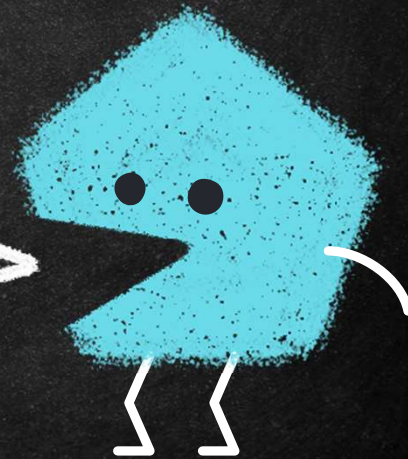
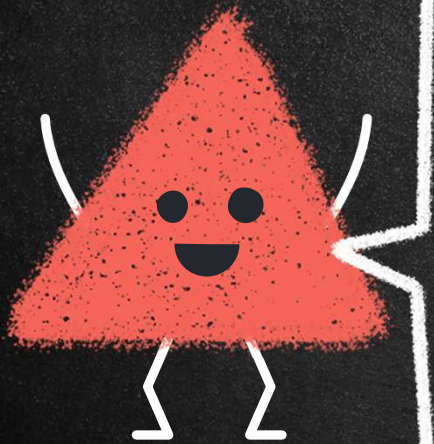
Donde:

- no: indica que no hay posibilidad de conversión.
- si: indica que el casting es implícito.
- si*: indica que el casting es implícito pero se puede producir pérdida de precisión.
- cast: indica que hay que hacer casting explícito.



CONVERSIONES DE TIPO

PARSE



PARSE

El parse no es simplemente el cambio del "tipo", sino una **adaptación y análisis sintáctico** al nuevo tipo que desea.

Para utilizarlo es necesario hacer uso de las **clases envoltura** y de sus métodos.

```
String texto ="1950";  
int numero = Integer.parseInt(texto);  
System.out.println("El String convertido en int es " + numero);
```



WRAPPERS – CLASE ENVOLTURA

Tipo	Clase de envoltura	Método
byte	Byte	Byte.parseByte(aString)
short	Short	Short.parseShort(aString)
int	Integer	Integer.parseInt(aString)
long	Long	Long.parseLong(aString)
float	Float	Float.parseFloat(aString)
double	Double	Double.parseDouble(aString)
boolean	Boolean	Boolean.parseBoolean(aString)

SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR

parseInt

```
public static int parseInt(String s)  
    throws NumberFormatException
```



EJERCICIOS PARSE

```
stringToInt();           //String = "2"  
intToString();           //int numero = 1985  
charToString();          //char letra = 'P';  
stringToBoolean();       //String texto = "true";  
stringToDouble();        //String texto = "2.5";
```



Fin

