

BLOC B



Cinta Zaragoza Blanch
1R ASIX MP03
Curs 2021-2022

1. Variables. Tipus i utilitat.	3
2. Conversions de tipus de dades.	4
3. Constants. Tipus i utilitat.	6
4. Operadors del llenguatge de programació.	7
Extra	10
Classifica, coneix i utilitza en expressions els operadors del llenguatge.	10
Comprova el funcionament de les conversions de tipus explícites i implícites.	10

1. Variables. Tipus i utilitat.

VARIABLES DE TIPUS PRIMITIUS:

Tipus de dada	Tamany	Descripció	Exemple
Byte	1 byte	Emmagatzema nombres enters des del -128 fins el 127	
Short	2 bytes	Emmagatzema nombres enters des del -32 768 fins al 32 767	
Int	4 bytes	8	
Long	8 bytes	Emmagatzema nombres enters des del -9 223 372 036 854 775 808 fins 9 223 372 036 854 775 807	
Float	4 bytes	Emmagatzema nombres fraccionaris amb decimals amb precisió de 6 7 decimals	
Double	8 bytes	Emmagatzema nombres fraccionaris amb decimals (15 dígitos decimals) 1.7e-308 1.7e+308	
Boolean	1 bit	Pot tenir valor cert o fals. Es sol utilitzar en operacions and i or lògica.	
Char	2 bytes	Emmagatzema una sola lletra o caràcter o valors ASCII	

Taula 7. Tipus de dades primitius del llenguatge Java

Categoria	Tipus	Valors possibles	Grandària en bits	Observacions
Lògics	boolean	Els literals booleans true i false.		
Caràcters	char	Qualsevol literal caràcter, des de \u0000 fins a \uffff		
Enters	byte	$-2^7 \dots 2^7-1$	8	A diferència dels llenguatges C/C++, no hi ha tipus enters sense signe. La implementació sempre és en complement a 2 en qualsevol plataforma.
	short	$-2^{15} \dots 2^{15}-1$	16	
	int	$-2^{31} \dots 2^{31}-1$	32	
	long	$-2^{63} \dots 2^{63}-1$	64	
Reals	float		32	La implementació segueix la norma IEEE 754 i és idèntica en totes les plataformes.
	double		64	

Tipus numèrics enters

Tipus numèrics en punt flotant

Booleans i caràcters

Tipus de dades estructurades/ Classes

Cadenes de caràcters (Strings)

Les cadenes de caràcters es delimiten entre cometes dobles en lloc de senzilles com els caràcters individuals.

Independentment s'indica explícitament que s'ha de crear un nou objecte de tipus string d'això se'n encarrega automàticament el compilador.

Emmagatzemen informació en cadena i podem operar sobre aquesta.

Exemples:

Substituir la cadena per una altra, dividir-la en varies cadenes atenent a un separador en concret...

2. Conversions de tipus de dades.

Dades primitius (simples / elementals)

Són les 4 categories principals:

-Lògics

-Caràcters

-Enters

-Reals

Dades complexos (aquests són construïts a partir de dades de tipus primitius)

Conversions:

Implícit: Es realitza de manera automàtica al assignar un valor de un tipus a un altre compatible

Exemple: byte x=5;

Int y=x;

El valor emmagatzemat en x (byte) es converteix a enter al ser-li assignat ea y (int)

Explícites: es feina del programador especificar el nou tipus al que es transformarà la dada. S'escriu de forma explícita entre parèntesis davant la dada.

Exemple: Byte a=20;

Int x=(int) a;

Al escriure int entre parèntesis es força a canviar la dada de tipus byte a int.

S'ha de tenir compte al realitzar conversions ja que es pot aplicar a tipus no compatible, provocant una pèrdua de dades o errors en l'execució.

Float x = 5.7F

Int y = (int) x;

Es realitzarà la conversió però es perdrà la part decimal del nombre amb punt flotant al guardar-lo en y. Només es guardarà el 5.

Taula de conversió dels tipus de valors

Nuevo tipo	Tipos origen
short	byte
int	byte, short, char
long	byte, short, char, int
float	byte, short, char, int, long
double	byte, short, char, int, long, float

Tabla de conversiones numéricas implícitas en Java

3. Constants. Tipus i utilitat.

Que són les constants?

Un valor que podem assignar o comparar a un tipus de dades determinat, aquest mai podrà variar durant l'execució del programa d'aquí surt el nom.

Per la seva declaració utilitzarem el ENDLINE perquè la dada que li assignem no pugui ser modificada posteriorment. I el tipus de dada que emmagatzemarem.

Variable del sistema que manté un valor immutable al llarg de l'execució del programa.

Quins tipus existeixen?

Tipus primitiu

Quina utilitats els hi podem donar?

Per fer referència a objectes de classes

N'hi ha d'especials?

Els constructors són els mètodes que proporcionen les classes per construir objectes. Tota classe ha de proporcionar, com a mínim, un constructor.

Exemples:

```

package com.clases;

/**
 * Created by ericka.montero .
 */
public class Constantes {

    public static Integer NUMERO_PUERTAS = 4;

    public static final Integer NUMERO_RUEDAS = 4;

    public static final String MARCA_CARRO = "TOYOTA";
}

```

```

public class Constantes
{
    private final String atribConstante;
    //Notar que no ha sido inicializado

    //Constructor
    Constantes(String valor)
    {
        //Aquí se inicializó el atributo
        atribConstante = valor;
    }
}

```

Variables i constants serveixen per emmagatzemar valors de tipus primitius o fer referència

4. Operadors del llenguatge de programació.

Taula dels operadors

Precedência	+						-					
+	.	[]	()									
	++	--										
	!	~	instanceof									
	*	/	%									
	+	-										
	<<	>>	>>>									
	<	>	<=	>=	==	!=						
	&	^										
	&&											
	? ;											
	=	op=	,									

l'operador `op=` fa referència a la combinació de qualsevol dels operadors `*`, `/`, `%`, `+`, `-`, `<<`, `>>`, `>>>`, `&`, `^` i `|` seguits de l'operador `=`.

Quina utilitat tenen?

Una expressió és una combinació determinada d'operadors i operands per obtenir un resultat concret.

Els operadors poden ser variables, constants o mètodes definits en les classes.

Els operadors són símbols de llenguatge que ens permeten fer operacions amb els elements implicats. El tipus dependrà de la seva funcionalitat i el nombre de operands que reben.

****expressió**** combinació de valors, funcions, operadors i altres expressions que fent servir les regles de precedència dels operadors que les formen generen un resultat. (una altra expressió)

Quins tipus hi ha?

Artimètrics (operadors que fan servir un operand / operadors que fan servir 2 operands)

Operador d'assignació:

Els operadors artimètrics es fan servir per assignar valors a les variables, es simbolitza amb un símbol `=` (assignacions disponibles: variable i un valor o entre 2 variables.)

Operadors aritmètics ens permeten fer operacions aritmètiques amb nombres, tant enters com reals.

Operador	Significat	Tipus de dada	Exemple
-	Canvi de signe	Numèric	-3, -4.2
+	Suma	Numèric	3 + 8, 3.4 + 1.5
-	Resta	Numèric	9 - 4, 2.9 - 5.3
*	Multiplicació	Numèric	3 * 4, 3.0 * 4.1
/	Divisió	Numèric	3/4, 4.2/9.3
%	Mòdul	Enter	250%23

Operadors relacionals

També anomenats comparatius, símbols que ens ajuden a expressar relacions entre dues entitats com per exemple més gran que o més petit o igual que...

El resultat serà un booleà.

Operador	Significat	Exemple (cert)	Exemple (fals)
>	Major que	5 > 3	-99 > 13
>=	Major o igual que	5 >= 3.2	6 >= 19
<	Menor que	-3 < -2	4 < 1
<=	Menor o igual que	-3.9 <= 9.8	9.8 <= -5
==	Igual que	3 == 3	3 == 4
!=	Diferent de	2! = 7	2! = 2

Operadors lògics

Són aquells que fem servir per operar amb valors booleans. El resultat sempre serà un booleà.

Tabla 3.2 Operadores lógicos

Operador	Significado
	OR lógico (ó)
&&	AND lógico (y)
!	NOT lógico (no)

Extra

Classifica, coneix i utilitza en expressions els operadors del llenguatge.

```
//Desenar dia
do {
    System.out.println("Introduix un dia del mes (1-31):");
    dia=ent.nextInt();
    if(dia<1 || dia>31) System.out.println("Ha de ser un número enter entre 1 i 31!!");
    else break;
} while (true);
```

En aquest codi podem veure com li indiquem que ens introdueixi un número i després mirem que extingue entre el número 1 o 31.

Comprova el funcionament de les conversions de tipus explícites i implícites.

¿Quieres decir casting? Implícito significa que pasa una instancia de tipo, digamos B, que hereda de un tipo, digamos A como A.

Por ejemplo:

```
Class A;
Class B extends A;

function f(A a) {...};

main() {
    B b = new B;
    f(b); // <-- b will be implicitly upcast to A.
}
```

En realidad, hay otros tipos de fundiciones implícitas, entre primitivas, utilizando constructores predeterminados. Tendrá que ser más específico con su pregunta.

En realidad, hay otros tipos de fundiciones implícitas, entre primitivas, utilizando constructores predeterminados. Tendrá que ser más específico con su pregunta.

implícito con el constructor predeterminado:

```
class A {
    A (B b) { ... };
}

class B {};

main() {
    B b = new B();
    A a = b; // Implicit conversion using the default constructor of A, C++ only.
}
```



3



Ben



25 sept. 2011

Una conversión explícita es cuando usa alguna sintaxis para *decir* el programa para hacer una conversión. Por ejemplo (en Java):

```
int i = 999999999;
byte b = (byte) i; // The type cast causes an explicit conversion
b = i;             // Compilation error!! No implicit conversion here.
```

Una conversión implícita es donde la conversión ocurre sin ninguna sintaxis. Por ejemplo (en Java):

```
int i = 999999999;
float f = i; // An implicit conversion is performed here
```

Cabe señalar que (en Java) las conversiones que involucran tipos primitivos generalmente implican algún cambio de representación, y eso puede resultar en pérdida de precisión o pérdida de información. Por el contrario, las conversiones que involucran tipos de referencia (solo) no cambian la representación fundamental.