

1. Estructura d'un programa informàtic

1.3 Variables. Tipus i utilitat

Les variables en la programació són un element on es guarda i es recupera dades que s'utilitzen en un programa. El valor guardat pot anar variant durant l'execució del programa. El seu tamany dependrà del tipus de dada que definim a la funció i que el vulguem almacenar.

Les variables estan formades per:

- Tipus de dada:** Tipus de dada que es pot almacenar a la variable.
- **Nom de la variable:** Nom donat a la variable
- Valor:** Valor inicial guardat a la variable.

Al llenguatge de Java hi han 3 tipus de variables: **locals, d'instància i estàtiques o de classe.**

- VARIABLES LOCALS

Són aquelles que només poden accedir-se des del bloc en el qual han estat declarades.

- VARIABLES D'INSTÀNCIA

Són aquelles que es creen quan un objecte es crea i es destrueix quan ho fa l'objecte. A diferència de les locals, podem usar especificadors d'accés.

- VARIABLES ESTÀTIQUES O DE CLASSE

Són similars a les d'instància, la diferència és que aquestes es declaren utilitzant la paraula clau dins d'una classe fora de qualsevol constructor i només podem tenir una còpia d'una variable estàtica per classe.

1.4 Conversions de tipus de dades

En el moment en que s'assigna valor d'un tipus de dada a un altre, els dos tipus pot ser que no siguin compatibles entre ells. Si són compatibles, Java realitzarà la conversió automàticament (**Conversió automàtica de tipus**) i en cas que no ho sigui, haurà de convertir-ho explícitament (**Conversió explícita o Casting**).

Les conversions automàtiques es donen quan dos tipus de dades es converteixen automàticament. Per a que es produeixi automàticament s'han de complir dos requisits: que els tipus siguin compatibles i quan assignem el valor d'un tipus de dades més petites a un tipus de dades més grans.

A Java, els tipus de dades numèrics són compatibles però no admeteixen la conversió automàtica de tipus numèric a tipus char.

byte -> short -> int -> long -> float -> double

Les conversions explícites o càsting són aquelles en què volem assignar un valor de tipus de dades més gran a un tipus de dada més petit. Aquesta conversió és útil on la conversió automàtica no es pot realitzar degut a que els tipus de dades són incompatibles. Per exemple, char i int no són compatibles i s'han de convertir explícitament.

double-> float -> long -> int -> short -> byte

1.5 Constants. Tipus i utilitat

Les constants són una expressió que té un valor fixe, és manté immutable durant tota la vida del programa i es defineixen mitjançant el modificador final. S'utilitzen per a almacenar dades a la memòria que no podran canviar durant l'execució del programa.

- **Constants numèriques:** S'escriu com en el nombre real
- **Constants Data:** Es troba entre signes d'exclamació (! !)
- **Constants Hora:** Es troba entre signes d'interrogació (? ?)
- **Constants cadena:** Es troba entre cometes (" ")

1.6 Operadors del llenguatge de programació

Java proporciona diferents tipus d'operadors que es poden usar segons la necessitat i el classifiquen segons la necessitat.

- **Aritmètics**

S'utilitzen per realitzar operacions aritmètiques simples en tipus de dades primitives.

- * -> Multiplicació
- / -> Divisió
- % -> Mòdul
- + -> Addició
- -> Resta

- **Unaris**

Només necessiten un operant. S'utilitzen per incrementar, disminuir o negar un valor.

- -> Unari menys, per a negar valors.
- + -> Unari més, per a donar valors positius. Només s'usa quan un valor passa de negatiu a positiu.
- ++ -> Operador d'increment, per incrementar el valor en 1. Hi han dos varietats:
 - * Pre Increment: el valor s'incrementa i el calcula.
 - * Post increment: es calcula i s'incrementa el valor.
- -> Operador de decrement: disminueix el valor en 1. hi han dos varietats:
 - * Pre decrement: el valor disminueix i es calcula
 - * Post decrement: Es calcula i disminueix el valor.
- ! -> Operador lògic "no", per a invertir un valor booleano.

- D'assignació

S'utilitza per a assignar un valor a una variable i tenen associació de dreta a esquerra per lo que el valor de la dreta ha de ser declarat abans d'usar-lo o que sigui una constant.

`+=` -> Per a sumar l'esquerra al dret i després assignar-lo a la vairant de l'esquerra.

`-=` -> Per a restar l'esquerra al dret i després assignar-lo a la vairant de l'esquerra.

`*=` -> Per a multiplicar l'esquerra al dret i després assignar-lo a la vairant de l'esquerra.

`/=` -> Per a dividir l'esquerra al dret i després assignar-lo a la vairant de l'esquerra.

`^=` -> Per a potenciar l'esquerra al dret i després assignar-lo a la vairant de l'esquerra.

`%=` -> Per a assignar el mòdul operant esquerre al dret i després assignar-lo a la variant de l'esquerra.

- Relacionals

S'utilitza per a verificar relacions com igualtat, major que i menor que. Retornen el resultat després de la comparació i s'usen a les instruccions de bucle i en sentències condicionals.

`==` -> Retorna si l'esquerra és igual a la dreta.

`!=` -> Retorna si l'esquerra no és igual a la dreta.

`<` -> Retorna si l'esquerra és inferior a la dreta.

`<=` -> Retorna si l'esquerra és inferior o igual a la dreta.

`>` -> Retorna si l'esquerra és major que la dreta.

`>=` -> Retorna si l'esquerra és major o igual que la dreta.

- Lògics

S'utilitzen per a realitzar operacions "lògiques AND" i "lògiques OR". La segona condició no s'avalua si la primera és falsa.

`&&` -> AND lògic: retorna quan les condicions són correctes

`||` -> O lògic: retorna si una condició és correcta.

- Ternaris

Si la condició s'avalua coma correcta, llavors executarà les instruccions després del "?", i si no, executa les instruccions després del ":".

- Bit a bit

S'utilitzen per a la manipulació de bits individuals d'un número i realitzen operacions d'actualització i de consulta de l'arbre indexat binari.

`&`, Operador bit a bit AND: Si els dos bits d'entrada són 1, el resultat és 1. Si no, el resultat és 0 i el resultat binari final es mostra en decimal.

`||`, Operador bit a bit OR: Si un bit d'entrada és 1, el resultat és 1. Si no, el resultat és 0 i el resultat binari final es mostra en decimal.

`^`, Operador bit a bit XOR: Si un dels bits és 0 i l'altre 1, el resultat és 1. Si els dos bits són iguals, el resultat és 0. El resultat binari final es mostra en decimal.

`~`, Operador bit a bit de complement (NOT): Retorna la representació

complementària del valor d'entrada amb tots els bits invertits i el resultat binari final es mostra en decimal.

- Shift

S'utilitzen per a moure els bits d'un número a l'esquerra o dreta, multiplicant o dividint per 2, respectivament. Para multiplicar o dividir un número per 2.

<<, operador de desplaçament a l'esquerra: Mou 2 bits del nombre a l'esquerra i plena de "0" els bits moguts. Efecte similar a la multiplicació d'un nombre amb potència de 2.

>>, operador de desplaçament a la dreta firmat: Mou els bits d'un nombre a la dreta i plena de "0" els bits moguts. El bit de més a l'esquerra depèn del signe del número inicial. Efecte similar a dividir un número amb alguna potència de 2.

>>>, operador de canvi a la dreta sense signe: Mou els bits d'un nombre a la dreta i plena de "0" els bits moguts. El bit de més a l'esquerra s'estableix en 0.

- D'instància

S'utilitza per a verificar el tipus. Es pot usar per a provar si un objecte és una instància de una classe, una subclasse o una interfície.

- Precedència i Associativitat d'operadors

S'utilitzen quan es fan equacions híbrides que involucren més d'un tipus d'operador. Aquestes regles determinen quina part de l'equació considerar primera, ja que poden haver moltes valoracions diferents per a la mateixa equació.

Operador	Aociatividad
++ —	De derecha a izquierda
++ — + - !	De derecha a izquierda
* / %	De derecha a izquierda
+ -	De derecha a izquierda
< <= > >=	De derecha a izquierda
== !=	De derecha a izquierda
&	De derecha a izquierda
^	De derecha a izquierda
	De derecha a izquierda
&&	De derecha a izquierda
	De derecha a izquierda
?:	De derecha a izquierda
= += -= *= /= %=	De derecha a izquierda

2.1 Tipus de dades simples

Representen una única dada simple que pot ser de tipus:

Lista de tipos de datos primitivos del lenguaje Java			
<i>Tipo</i>	<i>Tamaño</i>	<i>Valor mínimo</i>	<i>Valor máximo</i>
byte	8 bits	-128	127
short	16 bits	-32768	32767
int	32 bits	-2147483648	2147483647
long	64 bits	-9223372036854775808	9223372036854775807
float	32 bits	-3.402823e38	3.402823e38
double	64 bits	-1.79769313486232e308	1.79769313486232e308
char	16 bits	'\u0000'	'\uffff'