

Expresiones

Tipos, módulo, inicialización y asignación.

Cátedra 2

Otoño 2005

Las variables y constantes de tipos numéricos pueden ser combinadas mediante operadores aritméticos para formar expresiones. Los operadores aritméticos son suma (+), resta (-), multiplicación (*), división (/) y módulo (%).

Las precedencias son iguales a las del álgebra:

1. expresiones entre paréntesis
2. multiplicación y división
3. suma y resta

En caso de dos o más operadores de igual precedencia, se evalúa de izquierda a derecha. En caso de dudas, siempre se puede usar paréntesis, que son gratis. Por ejemplo,

$$\frac{a-4b}{c}$$

se escribe $(a-4*b)/c$. Es erróneo $a-4*b/c$ ó $(a-4b)/c$.

1 Tipo de una expresión

Para conocer el tipo de una expresión, se puede utilizar la siguiente regla: si todos los operandos son enteros, el tipo de la expresión es entero. Al contrario, si al menos uno es double, la expresión es double

2 El operador módulo (%)

Está relacionado con la división y se denota por %. El módulo entre dos números enteros a y b es el resto de su división. Por ejemplo:

```
7/2 == 3
7%2 == 1
```

ya que $7-7/2*2$ es 1.

¿Qué hace el siguiente extracto de programa?

```
1 con.print ("Ingresa el tiempo transcurrido");
2 int s=con.readInt();
3 con.print ("Eso equivale a ");
4 int minutos=s/60;
5 con.print (minutos+ " minutos y ");
6 int segundos=s%60;
7 con.println (segundos+ "segundos");
```

3 Asignación

El almacenamiento de valores en las variables se lleva a cabo mediante el operador de asignación (=). Una variable puede ser asignada reiteradas veces durante el transcurso de un programa. Por ejemplo,

```
1 int dolar;
2 dolar=580;
3 con.print ("El valor del dólar es "+dolar);
4 dolar=570;
5 con.print ("El nuevo valor del dólar es "+dólar);
```

No solo se puede asignar valores numéricos fijos, también se puede asignar expresiones

```
6 double g=9.87;
7 con.println("Cuanto ha transcurrido desde que soltó
  el objeto?");
8 double t=con.readDouble();
9 double v=g*t;
10 con.println ("La velocidad es de "+v);
```

Una de las cosas que no siempre se puede hacer es asignar una expresión de un tipo a una variable de otro tipo. Por ejemplo, el siguiente código es incorrecto

```
int superficie=3.1416*radio*radio;
```

ya que se está tratando de guardar un número real en una variable que ha sido declarada como entera. Es como tratar de guardar pelotas de tenis en un envase de pelotas de ping-pong.

Nota Una vez que una variable ha sido asignada, su valor no se modifica hasta que se le vuelva a asignar otro valor.

4 Inicialización

Se llama inicialización a la primera asignación de una variable. Una variable no puede ser usada en una expresión hasta que se le ha dado un valor.

```
double pi; // pi es declarada
pi=3.14159; // pi es inicializada
```

Como ya hemos visto, Java permite inicializar una variable en la misma instrucción en la que se le declara:

```
double pi=3.14159;
```

4.1 Asignaciones con la variable asignada en la expresión

Java permite realizar asignaciones donde la variable asignada aparece en la expresión de la derecha:

```
int conejos=5;
conejos=conejos+1;
```

Java evalúa primero la expresión de la derecha y luego asigna ese valor a la variable. En este caso Java calcula cuánto vale conejos+1. Recién cuando sabe que el valor es 6, lo asigna a conejos.

El incremento de una variable, es decir, una asignación del tipo $n=n+1$; se ven tan a menudo en Java que existe una notación abreviada para ella

```
n++;
```

Lo mismo para el decremento, $n--$.

4.2 Asignación de un double a un entero

Para truncar un double y pasarlo a entero, se puede hacer casting:

```
int ipc=(int)12.311; // guarda 12 en ipc
int vol=(int)(ipc+1.6); // guarda 13 en vol
```

5 “Swap” del valor de dos variables

Imagina que tienes dos variables enteras, x e y , que son inicializadas de la siguiente forma:

```
int x=5;
int y=con.readInt();
```

¿Cómo harías para intercambiar el valor de ambas variables? Una solución que puede parecer correcta es

```
x=y;
y=x;
```

Sin embargo, lo que hace realmente Java es dejar en x e y el valor original de y , ya que primero se le asigna el valor de y a x , con esto el valor de x se pierde. Después se le asigna este valor a y , que no se ve modificado, pues x tiene su mismo valor.

Una solución correcta consiste en usar una variable auxiliar. Estas son variables que se ocupan para guardar valores en forma temporal, generalmente para facilitar la realización de un cálculo o tarea compleja. En este caso vamos a usar una variable auxiliar para guardar el valor original de x

```
int auxiliar=x;
x=y;
y=auxiliar;
```

6 Funciones predefinidas

Al igual que las calculadoras científicas, Java trae incorporado una gran gama de funciones, entre las que se encuentran las siguientes: $\text{Math.sqrt}(x)$, $\text{Math.abs}(x)$, $\text{Math.pow}(x,y)$, $\text{Math.exp}(x)$, $\text{Math.cos}(x)$, $\text{Math.sin}(x)$ y $\text{Math.log}(x)$. Por ejemplo, para calcular el área de un círculo cuyo radio es r :

```
double area=3.1416*Math.pow(r,2);
```

7 Problema resueltos

7.1 Máquina reversadora de cifras

Haz un programa que reciba un número de 3 dígitos y muestre el número en con las cifras invertidas. Por ejemplo, si se lee 124 se muestra 421.

7.1.1 Solución:

```
1 con.print("Ingrese un numero: ? ");
2 int numero = con.readInt();
3 // desmenuzamos el numero
4 int dig1= numero / 100;
5 int dig2= (numero % 100) / 10;
6 int dig3= (numero % 10);
7 // lo armamos de nuevo
8 int alReves=dig3*100 + dig2*10 + dig1;
9 // mostramos el resultado
10 con.print("Su número en forma normal es ");
11 con.println(numero);
12 con.println("Su número al revés es " + alReves);
```

AL MARGEN

Problemas propuestos

- Escribe un programa que lea tres números del teclado y los escriba ordenados de menor a mayor, usando Math.min y Math.max .

Problema de ingenio

- Cómo se puede hacer el swap de variables sin usar una variable auxiliar;