UNIVERSIDAD SALESIANA DE BOLIVIA Guía de laboratorio Elaborado por Lic. Carmen Mollinedo	PROGRAMACION III Laboratorio 2 Introducción al Java

VARIABLES Y TIPOS DE DATOS

Las variables son las partes importantes de un lenguaje de programación: ellas son las entidades (valores, datos) que actúan y sobre las que se actúa.

Declaración de variable

```
tipoVariable nombre;
```

Ejemplo:

```
int area=3;
char salir='q';
```

TIPOS DE VARIABLES

Todas las variables en el lenguaje Java deben tener un tipo de dato. El tipo de la variable determina los valores que la variable puede contener y las operaciones que se pueden realizar con ella.

Existen dos categorías de datos principales en el lenguaje Java:

a)Tipos primitivos

Tipo	Tamaño/Formato	Descripción
(Números enteros)		
Byte	8-bit complemento a 2	Entero de un Byte
Short	16-bit complemento a 2	Entero corto
Int	32-bit complemento a 2	Entero
Long	64-bit complemento a 2	Entero largo
(Números reales)		
Flota	32-bit IEEE 754	Coma flotante de precisión simple
Double	64-bit IEEE 754	Coma flotante de precisión doble
(otros tipos)		
Char	16-bit caracter	Un solo caracter
boolean	true o false	Un valor booleano (verdadero o falso)

b) tipos referenciados se llaman así porque el valor de una variable de referencia es una referencia (un puntero) En Java tenemos los arrays, las clases y los interfaces como tipos de datos referenciados.

NOMBRES DE VARIABLES

Un programa se refiere al valor de una variable por su nombre. Por convención, en Java, los nombres de las variables empiezan con una letra minúscula (los nombres de las clases empiezan con una letra mayúscula).

Un nombre de variable Java.

1.debe ser un identificador legal de Java comprendido en una serie de caracteres Unicode. Unicode perminte la codificación de 34.168 caracteres.
2. no puede ser el mismo que una palabra clave o el nombre de un valor booleano (true or false)
3. no deben tener el mismo nombre que otras variables cuyas declaraciones aparezcan en el mismo ámbito.

Por convención, los nombres de variables empiezan por un letra minúscula. Si una variable está compuesta de más de una palabra, como 'nombreDato' las palabras se ponen juntas y cada palabra después de la primera

Ejemplo de funciones

```
public static int redondearPromedio(int[] valores){
...
}
```

Ejemplo de clases

```
public class RelojAtomico{
....
}
```

OPERADORES ARITMÉTICOS

Operador	Uso	Descripción
+	op1 + op2	Suma op1 y op2
-	op1 - op2	Resta op2 de op1
*	op1 * op2	Multiplica op1 y op2
/	op1 / op2	Divide op1 por op2
%	op1 % op2	Obtiene el resto de dividir op1 por op2

Nota: El lenguaje Java extiende la definición del operador + para incluir la concatenación de cadenas.

Además, existen dos operadores de atajos aritméticos, ++ que incrementa en uno su operando, y -- que decrementa en uno el valor de su operando.

Operador	Uso	Descripción
++	op ++	Incrementa op en 1; evalúa el valor antes de incrementar
++	++ op	Incrementa op en 1; evalúa el valor después de incrementar
--	op --	Decrementa op en 1; evalúa el valor antes de decrementar

--	-- op	Decrementa op en 1; evalúa el valor después de decrementar

OPERADORES RELACIONALES Y CONDICIONALES

Los valores relacionales comparan dos valores y determinan la relación entre ellos. Por ejemplo, != devuelve true si los dos operandos son distintos.

Operadores relacionales

Operador	Uso	Devuelve true si
>	op1 > op2	op1 es mayor que op2
>=	op1 >= op2	op1 es mayor o igual que op2
<	op1 < op2	op1 es menor que op2
<=	op1 <=op2	op1 es menor o igual que op2
==	op1== op2	op1 y op2 son iguales
!=	op1 != op2	op1 y op2 son distintos

Operadores condicionales.

Operador	Uso	Devuelve true si
&&	op1 && op2	op1 y op2 son verdaderos
||	op1 || op2	uno de los dos es verdadero
!	! op	op es falso

Operadores de Asignación

Puedes utilizar el operador de asignación =, para asignar un valor a otro. anteriores son equivalentes.

Esta tabla lista los operadores de asignación y sus equivalentes.

Operador	Uso	Equivale a
+=	op1 += op2	op1 = op1 + op2
-=	op1 -= op2	op1 = op1 – op2
*=	op1 *= op2	op1 = op1 * op2
/=	op1 /= op2	op1 = op1 / op2
%=	op1 %= op2	op1 = op1 % op2
&=	op1 &= op2	op1 = op1 & op2
|=	op1 |= op2	op1 = op1 | op2
^=	op1 ^= op2	op1 = op1 ^ op2

SENTENCIAS DE CONTROL DE FLUJO EN JAVA

Las sentencias de control de flujo determinan el orden en que se ejecutarán las otras sentencias dentro del programa. El lenguaje Java soporta varias sentencias de control de flujo, incluyendo.

Sentencias	palabras clave
toma de decisiones	if-else, switch-case
Bucles	for, while, do-while
excepciones	try-catch-finally, throw
miscelaneas	break, continue, label:, return

Ejemplo

```
public class HolaMundo {
   public static void main(String[] args) {
   System.out.println("Hola, mi primer programa");
}
}
```

El archivo debe guardarse como HolaMundo.java respetando las mayúsculas y minúsculas, necesitamos compilar el programa para obtener el binario con extensión class

```
Javac HolaMundo.java
Java HolaMundo
```

Ejemplo:

```
public class SumaSimple {
 public static void main(String[] args) {
 int a=1;
 System.out.println("el valor de a="+a);
 a=a+10;
 System.out.println("ahora sumándole 10 es a="+a);
 }
}
```

- **Nota**

El arreglo o vector args se utiliza para recibir todas las órdenes de la JVM, los vectores en Java son objetos que tienen propiedades.

Ejercicio

Implemente un programa en Java que:

a. Sume los dígitos de un número
b. Invierta un número
c. Sume los dígitos pares de un número
d. Inserte el digito 7 en un número
e. Verifique que un número sea capicua
f. Convertir un numero de base 10 a base 2