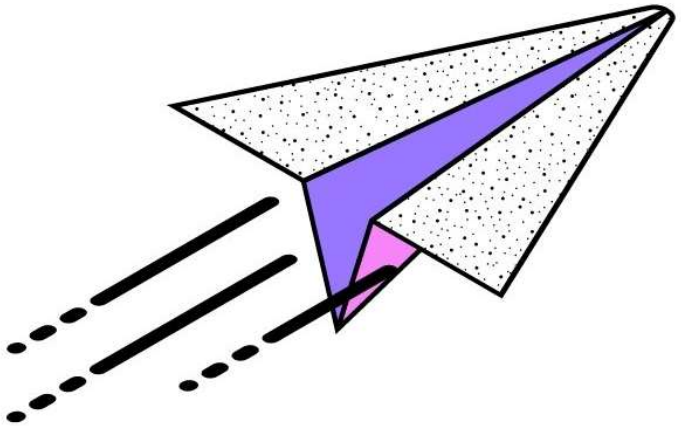
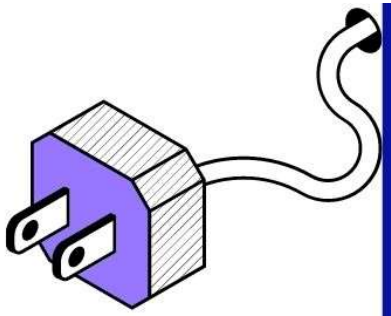
Java 7 SE Fundamentals

Sección 3

Trabajar con variables primitivas



- Declarar y asignar valores a variables.
- Uso de constantes.
- Descripción de tipos de datos primitivos como int, float, double, boolean.
- Utilizar operadores aritméticos para modificar valores.
- Declaración e inicialización de variables de campo.



¿Qué es una variable?

Definición de variable

- Las variables se utilizan para representar valores que pueden cambiar durante la ejecución de un programa.
- Las variables tienen un nombre, que es un identificador para hacer referencia a ellas en nuestros programas.

Reglas para definir el nombre de una variable en Java

- Un identificador es una secuencia de caracteres que consta de letras, dígitos, guiones bajos (_) y signos de dólar ($).
- Un identificador debe comenzar con una letra, un guion bajo (_) o un signo de dólar ($). No puede comenzar con un dígito.
- Un identificador no puede ser una palabra reservada.
- La sintaxis para declarar una variable es **tipo_de_dato** *nombre_variable*;

Notas y tips para definir nombres de variables.

- Dado que Java distingue entre mayúsculas y minúsculas, **area**, **Area** y **AREA** son identificadores diferentes.
- Los identificadores descriptivos hacen que los programas sean fáciles de leer.

Declaración variables

Ejemplo donde se declaran tres variables.

```java
public class Prueba{

    public static void main(String[] args){

    //Ejemplo DECLARACION de variables
    int contador;
    double radio;
    double tasaInteres;

    }
}
```

Asignar valor a una variable

- Una declaración de asignación designa un valor para una variable. Una declaración de asignación se puede utilizar como una expresión en Java.
- Después de declarar una variable, puede asignarle un valor. En Java, el signo igual (**=**) se utiliza como operador de asignación.
- Un identificador no puede ser una palabra reservada.
- La sintaxis para asignar valor a una variable es *nombre_variable = expresión o valor*;

Notas y tips para expresiones de asignación.

- Una expresión representa un cálculo que involucra valores, variables y operadores que, tomándolos juntos, se evalúa como un valor.

Asignar valor a una variable

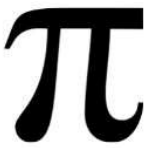
Ejemplo donde se declaran tres variables y se les asigna un valor.

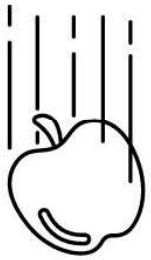
```java
public class Prueba{

    public static void main(String[] args){

    //Ejemplo ASIGNACION a variables
    int contador = 1;
    double radio = 1.0;
    int x = 5 * (3 / 2);
    double area;
    contador = x + 1;
    area = radius * radius * 3.14159;

    }
}
```





Constantes

- Una constante es un identificador que representa un valor permanente.
- El valor de una variable puede cambiar durante la ejecución de un programa, pero una una constante, representa datos permanentes que nunca cambian.
- La sintaxis para asignar valor a una variable es **final datatype** NOMBRE_CONSTANTE = valor;

Notas y tips para expresiones de asignación.

- Una constante debe declararse e inicializarse en la misma instrucción.

Declaración Constante

Ejemplo donde se declaran cuatro constantes y sus valores.

```java
public class Prueba{

    public static void main(String[] args){

        //Ejemplo declaracion de constantes
        final double PI = 3.14159;
        final double E = 2.7182;
        final int VALOR_MINIMO = 1;
        final int VALOR_MAXIMO = 20;

    }
}
```

Tipo de datos primitivos

Diferencia entre tipos primitivos y objetos.

No poseen atributos

Los tipos de datos no poseen atributo alguno, estos solo guardan un valor en alguna dirección en memoria.

No poseen métodos

Los tipos de datos no poseen métodos, de hecho estos solo son utilizados por los métodos en partes del programa.

Son indivisibles, atómicos

Con baja latencia hay menos retrasos; da una reacción genuina en tiempo real.

Tipo de datos numéricos

Entre los tipos numéricos tenemos: **enteros** y de **punto flotante**

byte

-2^7 to 2^7-1 (-128 to 127)

8 bits, con signo

short

16 bits, con signo

-2^{15} to $2^{15}-1$ (-32768 to 32767)

int

-2^{31} to $2^{31}-1$ (-2147483648 to 2147483647)

32 bits, con signo

long

64 bits, con signo

-2^{63} to $2^{63}-1$

(i.e., -9223372036854775808 to 9223372036854775807)

float

32 bits, IEEE 754

Negative range: $-3.4028235E+38$ to $-1.4E-45$

Positive range: $1.4E-45$ to $3.4028235E+38$

double

64 bits, IEEE 754

Negative range: $-1.7976931348623157E+308$ to $-4.9E-324$

Positive range: $4.9E-324$ to $1.7976931348623157E+308$

Tipo de datos no numéricos

Entre los no numéricos, tenemos los de carácter y los booleanos.

boolean

Un tipo de datos booleano declara una variable con el valor **true** (verdadero) o **false** (falso)

char

Un tipo de datos de carácter representa un unico carácter.

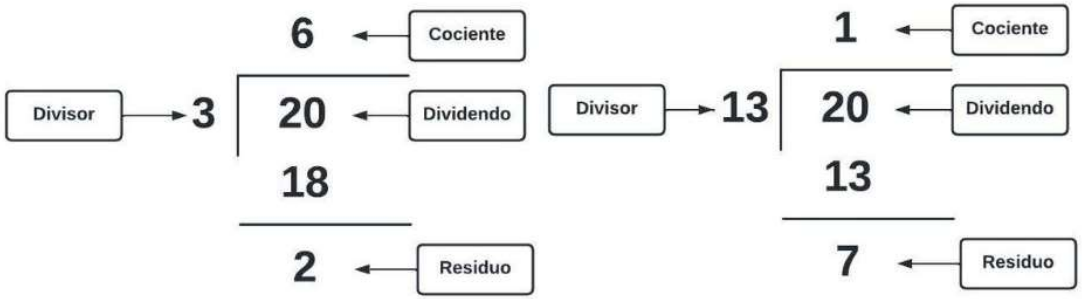
"String"

Una cadena es una secuencia de caracteres. No es primitivo, pero ...

Operadores aritméticos

Los operadores aritmeticos se utilizan para modificar variables.

Simbolo	Nombre	Ejemplo	Resultado
+	Adición	34 + 1	35
-	Sustracción	34.0 - 0.1	33.9
*	Multiplicación	300 * 30	9000
/	División	1.0 / 2.0	0.5
%	residuo o modulo	20 % 3	2

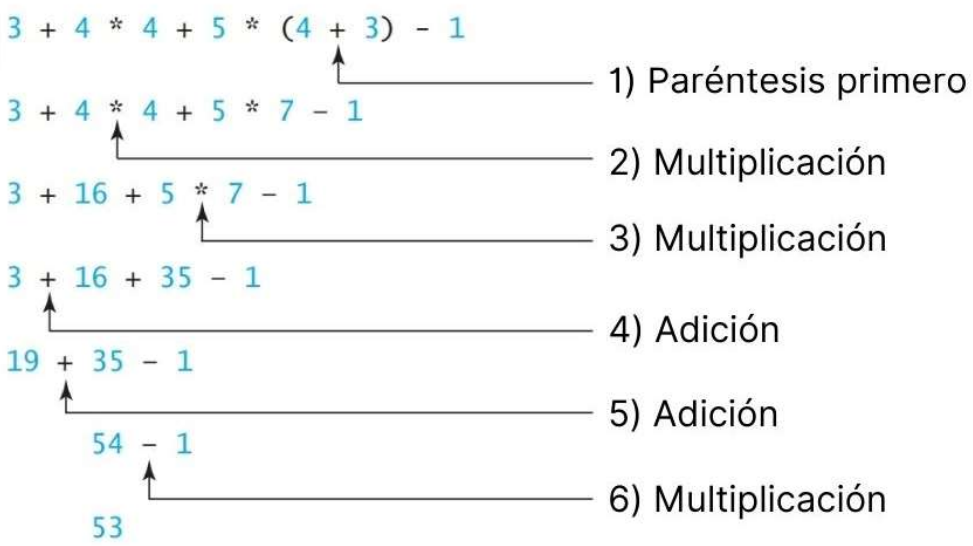


Jerarquía de los operadores aritméticos

Los operadores aritmeticos se utilizan para modificar variables.

Constantes

- La jerarquía funciona tal y como es en la aritmética.
- Primero los paréntesis, pudiendo estos estar anidados.
- Después multiplicación, división y resto, modulo o residuo. Si una expresión contiene varios operadores de multiplicación, división y modulo , se aplican de izquierda a derecha.
- Por ultimo adición y sustracción. Si una expresión contiene varios operadores de suma y resta, se aplican de izquierda a derecha.



Ejercicios practicos

- Dado un numero entero de segundos, calcular a cuantos minutos equivale y cuantos segundos restantes quedan. Ejemplo :
  500 segundos equivalen a 8 minutos y 20 segundos.
- Dado un numero de grados Fahrenheit, convertirlos a Celsius.
- Muestre la hora actual con realizando cálculos y expresiones aritméticas. **

Operadores de asignación aumentados.

Los operadores +, -, *, / y % se pueden combinar con el operador de asignación para formar operadores aumentados.

Operador	Nombre	Ejemplo	Equivalente
+=	Asignación de adición	i += 8	i = i + 8
-=	Asignación de resta	i -= 8	i = i – 8
*=	Asignación de multiplicación	i *= 8	i = i * 8
/=	Asignación de división	i /= 8	i = i / 8
%=	Asignación de modulo	i %= 8	i = i % 8

Operadores de incremento y decremento.

Los operadores de incremento (++) y decremento (– –) son para incrementar y decrementar una variable en 1.

Operador	Nombre	Descripción	Ejemplo (i = 1)
++var	preincremento	Incrementa var en 1 y usa el nuevo valor de var en la declaración	int j = ++i; // j es 2, i es 2
var++	postincremento	Incrementa var en 1, pero use el valor original de var en la declaración	int j = i++; // j es 1, i es 2
- -var	predecremento	Disminuye var en 1 y usa el nuevo valor de var en la declaración	int j = - -i; // j es 0, i es 0
var- -	postdecremento	Disminuye var en 1 y usa el valor original de var en la declaración	int j = i- -; // j es1, i es 0

Variables de campo

Declaración e inicialización de variables de campo

```java
public class Circulo{

    //declaracion e inicializacion
    //de una constante
    final double PI = 3.1416;

    //variables de campo

    //declaracion e inicializacion
    double radio = 1;

    //declaracion
    double area;
    double perimetro;

    void computeArea(){
      area = radio * radio * PI;
    }

    void computePerimetro(){
      perimetro = 2 * radio * PI;
    }

    void setRadio(double newRadio ){
      radio = newRadio;
    }
}
```

Todo lo que este dentro de los parentesis que definen la clase pero fuera de los parentesis de algun metodo , son variables de campo, es decir atributos de la clase.

Las variables definidas dentro de los parentesis de los metodos, no son variables de campo. (*aqui no hay ninguna declaracion*)