

Datos Elementales y formato de un programa en Java

PROGRAMACION I

Grado en Matemáticas Informática

Introducción a la programación

- **Programa**: Expresión de la solución a un problema usando un lenguaje de programación
 - Una lista de instrucciones, operaciones o frases a ejecutar por el ordenador
- Programa = Datos + Operaciones
- Los datos y las operaciones se guardan en la memoria del ordenador de la misma manera

Datos en un programa

- Los datos representan información que queremos almacenar, manipular, o mostrar
- Pueden ser números, textos, o archivos binarios como imágenes
- Existen datos compuestos (compuesto de varios otros datos)

Variables

- La información unitaria de los programas se almacena en unidades llamadas variables
- Una variable:
 - se identifican por un nombre
 - almacena un valor
 - es de un determinado tipo
- El estado de un programa en un momento dado es el valor de las variables en ese instante

Datos en un programa

- Una variable es un nombre o identificador asociado con un dato
 - Ejemplos: **apellido, telefono, foto**
- Para cada variable, el programa reserva un espacio suficiente en la memoria RAM
- Para recuperar el dato asociado con una variable, la CPU tiene que buscar en el espacio reservado

Datos en un programa

- Una variable puede cambiar el dato asociado con un identificador
 - El valor de las variables cambia en el transcurso de la ejecución un programa

- No obstante, la ubicación reservada en la memoria para una variable nunca cambia durante la existencia de la variable

Identificadores de variables

- Es el nombre que el programador da a la variable
- No pueden coincidir con una palabra reservada del lenguaje de programación
- Distinción entre mayúsculas y minúsculas
- •Secuencia de letras, dígitos, _ (subrayado) y \$. Empezando siempre por una letra
- No se deben poner acentos

Palabras reservadas

- **class** inicio de programa
- **public, static, void** método 'main'
- **byte, short, int, long** enteros
- **float, double** reales
- **char** caracteres
- **boolean** tipo Boolean
- **true, false** literales Boolean

Identificadores de variables

- Java ha acordado:
 - Los identificadores de variables siempre empiezan con una minúscula
 - Si el identificador está compuesto de más que una palabra, cada palabra subsiguiente empieza con mayúscula
unaVariableLarga
 - Los identificadores deben ser significativos y describir de forma no ambigua el dato asociado

Tipos de datos básicos

- Enteros: int, byte, short, long
- Reales: float, double
- Caracteres: char
- Booleanos: boolean – valor true o false
- Bit, Byte, ...

Tipos de datos básicos

Tipo de dato	Representación	Tamaño (Bytes)	Rango de Valores	Valor por defecto	Clase Asociada
byte	Numérico Entero con signo	1	-128 a 127	0	Byte
short	Numérico Entero con signo	2	-32768 a 32767	0	Short
int	Numérico Entero con signo	4	-2147483648 a 2147483647	0	Integer
long	Numérico Entero con signo	8	-9223372036854775808 a 9223372036854775807	0	Long
float	Numérico en Coma flotante de precisión simple Norma IEEE 754	4	$\pm 3.4 \times 10^{-38}$ a $\pm 3.4 \times 10^38$	0.0	Float
double	Numérico en Coma flotante de precisión doble Norma IEEE 754	8	$\pm 1.8 \times 10^{-308}$ a $\pm 1.8 \times 10^308$	0.0	Double
char	Carácter Unicode	2	\u0000 a \uFFFF	\u0000	Character
boolean	Dato lógico	0	true ó false	false	Boolean
void	0	0	0	0	Void

Declaración de variables

- Antes de utilizar una variable, siempre hay que declararla
- Cuando declaramos una variable, indicamos el tipo de su dato asociado y su identificador
- Como resultado, el programa reserva un espacio suficiente para guardar el dato en la memoria RAM
- Todavía no hemos indicado que dato queremos guardar

Declaración de variables

- Declaración de una variable:
`<tipo> <identificador> ;`
- Se puede declarar varias variables a la vez:
`<tipo> <identificador>, <identificador> [...];`
- Ejemplos:
 - **`boolean estaEstudiando;`**
 - **`int numeroDeAlumnos;`**
 - **`int indice1, indice2;`**

Inicialización de variables

- Para asociar un dato con una variable, hay que inicializar la variable
- La inicialización se hace mientras una operación que se llama asignación
- Sólo se puede asignar una variable con un dato del tipo correcto (sino el programa no compila)
- Como resultado, el dato queda guardado en el espacio respectivo

Asignación

- La asignación de una variable sigue el modelo
 $\langle \text{identificador} \rangle = \langle \text{expresión} \rangle ;$
- $\langle \text{expresión} \rangle$ puede ser reemplazado con:
 - un valor básico (literal)
 - otra variable
 - el resultado de una operación más compleja
- El resultado de $\langle \text{expresión} \rangle$ tiene que ser del mismo tipo que la variable

Asignación

- La asignación se puede combinar con la declaración:

`<tipo> <identificador> = <expresión> ;`

- Ejemplos:

- **`boolean estaEstudiando = true;`**
- **`int numeroDeAlumnos;`**
`numeroDeAlumnos = 31;`
- **`int indice1 = 12;`**
- **`int indice2 = indice1;`**

Asignación

- Se puede cambiar el dato asociado con una variable usando la asignación
- Para cambiar el dato asociado, hay que asignar otro valor a la variable
- Ejemplo:
 - **int numeroDeAlumnos;**
 - **int n = 23;**
 - **numeroDeAlumnos = 31;**
 - **numeroDeAlumnos = 26;**

Literales

- Algunos valores básicos se pueden expresar directamente en Java
- Las palabras que describen estos valores se llaman literales
 - Hay varios literales en Java:
 - Enteros: **10**
 - Reales: **18.54**
 - Caracteres: **'d'**
 - Cadenas de caracteres: **"Hola"**
 - Boolean: **true** y **false**

Literales de tipo caracter

- Carácter almacenado en 2 bytes = 16 bits
- En teoría, permite representar $2^{16} = 65,536$ caracteres distintos
- Tabla Unicode recoge más de 30 000 caracteres distintos procedentes de lenguas escritas distintas (Ej: griego, árabe)
- Los primeros $2^8 = 256$ corresponden a los de la tablaASCII

Literales de tipo caracter

- Declaración e inicialización:
 - **char letra = 'A';**
 - **char respuesta = 'Y';**

| ASCII Hex Símbolo |
|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|
| 0 0 NUL | 16 10 DLE | 32 20 (space) | 48 30 0 | 64 40 @ | 80 50 P | 96 60 ` | 112 70 p |
| 1 1 SOH | 17 11 DC1 | 33 21 ! | 49 31 1 | 65 41 A | 81 51 Q | 97 61 a | 113 71 q |
| 2 2 STX | 18 12 DC2 | 34 22 " | 50 32 2 | 66 42 B | 82 52 R | 98 62 b | 114 72 r |
| 3 3 ETX | 19 13 DC3 | 35 23 # | 51 33 3 | 67 43 C | 83 53 S | 99 63 c | 115 73 s |
| 4 4 EOT | 20 14 DC4 | 36 24 \$ | 52 34 4 | 68 44 D | 84 54 T | 100 64 d | 116 74 t |
| 5 5 ENQ | 21 15 NAK | 37 25 % | 53 35 5 | 69 45 E | 85 55 U | 101 65 e | 117 75 u |
| 6 6 ACK | 22 16 SYN | 38 26 & | 54 36 6 | 70 46 F | 86 56 V | 102 66 f | 118 76 v |
| 7 7 BEL | 23 17 ETB | 39 27 ' | 55 37 7 | 71 47 G | 87 57 W | 103 67 g | 119 77 w |
| 8 8 BS | 24 18 CAN | 40 28 (| 56 38 8 | 72 48 H | 88 58 X | 104 68 h | 120 78 x |
| 9 9 TAB | 25 19 EM | 41 29) | 57 39 9 | 73 49 I | 89 59 Y | 105 69 i | 121 79 y |
| 10 A LF | 26 1A SUB | 42 2A * | 58 3A : | 74 4A J | 90 5A Z | 106 6A j | 122 7A z |
| 11 B VT | 27 1B ESC | 43 2B + | 59 3B ; | 75 4B K | 91 5B [| 107 6B k | 123 7B { |
| 12 C FF | 28 1C FS | 44 2C , | 60 3C < | 76 4C L | 92 5C \ | 108 6C l | 124 7C |
| 13 D CR | 29 1D GS | 45 2D - | 61 3D = | 77 4D M | 93 5D] | 109 6D m | 125 7D } |
| 14 E SO | 30 1E RS | 46 2E . | 62 3E > | 78 4E N | 94 5E ^ | 110 6E n | 126 7E ~ |
| 15 F SI | 31 1F US | 47 2F / | 63 3F ? | 79 4F O | 95 5F _ | 111 6F o | 127 7F |

Literales caracteres especiales

- **Códigos Significado**
 - ‘\n’ Nueva línea
 - ‘\r’ Retorno de carro
 - ‘\t’ Tabulación
 - ‘\’ Comilla simple
 - ‘\’\’ Comilla doble
 - ‘\\’ Barra inclinada inversa

Literales de tipo booleano

- Variable con valor lógico
- Acepta dos literales posibles: **true** y **false**
- Ejemplo:
 - **boolean ojosNegros = true;**
- Almacenado en 1 bit de memoria

Literales de cadena de texto

- El tipo String no es básico
- **String** no es una palabra reservada
- Declaración e inicialización:
 - **String saludo = “Hola Jefe”;**
- El literal cadena es un conjunto de caracteres
- Dentro de una cadena se puede utilizar caracteres especiales:
 - **“\“Hola\nJefe\””**

Mi primer programa en Java

- Vamos a escribir nuestro primer programa en Java
- El código fuente siempre se encuentra en un archivo con extensión .java
- En nuestro ejemplo, vamos a guardar el programa en un archivo llamado `MiPrimerEjercicio.java`

Estructura de un programa

```
class MiPrimerEjercicio{  
    // comentario, no es parte del programa  
    public static void main(String[] args)  
    {  
        // aquí se escriben las instrucciones  
    } // fin de 'main'  
} // fin de 'Mi primer ejercicio'
```

Mi primer programa en Java

- Instrucciones útiles:
 - Imprimir por la pantalla una línea de texto

```
System.out.println(TEXTO_A_MOSTRAR);
```

Ejercicio

- Declarar las variables de tipo entero 'valor1' y 'valor2'
- Inicializar valor1 con el literal entero 10
- Inicializar valor2 con el literal entero 20
 - **Entrada:** valor1 es igual a 10 y valor2 a 20
 - **Salida:** valor1 es igual a 20 y valor2 a 10

Otros conceptos necesarios

- Conversión entre tipos de datos
- Operadores aritméticos
- Operadores lógicos

Conversión de tipos

- Para enteros, se puede cambiar el tipo de un dato
- La conversión de tipos se puede hacer solo en este sentido:
- **byte → short → int → long**
- No se pierde información
- **1 byte → 2 bytes → 4 bytes → 8 bytes**

Conversión de tipos

```
// Inicializa una variable con un entero
int goodInt = 79124;
// Convierte el entero al tipo 'long'
long goodLong = goodInt;
// ¡Si se convierte al tipo 'short' se pierde
// información!
short badShort = goodInt;
```

Operadores aritméticos básicos

- + Suma
- – Resta
- * Multiplicación
- / División
- % Resto de una división entre enteros o módulo
- = asignación

Operadores lógicos básicos

- == Igualdad
- && AND
- || OR
- ^ XOR