

Comparación de instrucciones entre Java y C

# Programación II

Alicia Garrido Alenda

Departamento de Lenguajes y Sistemas Informáticos  
Universidad de Alicante

# Declaración de variables de tipos primitivos

Java	C
<code>int i;</code>	<code>int i;</code>
<code>char a;</code>	<code>char a;</code>
<code>double f;</code>	<code>float f;</code>
<code>boolean b;</code>	no existe

# Declaración de variables de tipos primitivos

- En Java si no se inicializa una variable, se produce un error de compilación:

```
int i=7;  
char a='c';  
double f; f=0.5;  
boolean b; b=false;
```

- En C una variable se puede declarar y no inicializar. El compilador no dará ningún aviso si no se compila con la opción `-Wall`

# Operadores

- Los siguiente operadores funcionan de la misma manera en ambos lenguajes:
  - ▶ relacionales: < > <= >= == !=
  - ▶ aritméticos: + - \* / % ++ --
  - ▶ asignaciones: = += -= \*= /= %=
  - ▶ lógicos: && || !

# Instrucción condicional

- La instrucción condicional tiene la misma sintaxis en ambos lenguajes:

Java	C
<pre>if (condicion) {     instrucciones si se cumple la condición }</pre>	<pre>if (condicion) {     instrucciones si se cumple la condición }</pre>
<pre>if (condicion) {     instrucciones si se cumple la condición } else {     instrucciones si no se cumple la condición }</pre>	<pre>if (condicion) {     instrucciones si se cumple la condición } else {     instrucciones si no se cumple la condición }</pre>

- En Java la `condicion` tiene que ser obligatoriamente de tipo booleano.
- En C la `condicion` puede ser de tipo entero, real, etc... Se considera cierta cuando es distinta de 0.

# Instrucción condicional

- Código para obtener el mínimo de tres números enteros  $m$ ,  $n$  y  $o$ :

Java	C
<pre>int min=Integer.MAX_VALUE; if (n&gt;m) {     if (m&gt;o)         min=o;     else         min=m; }else{     if (n&gt;o)         min=o;     else         min=n; }</pre>	<pre>int min=2147483647; if (n&gt;m) {     if (m&gt;o)         min=o;     else         min=m; }else{     if (n&gt;o)         min=o;     else         min=n; }</pre>

# Instrucción iterativa: `while`

- El bucle `while` tiene la misma sintaxis en ambos lenguajes:

Java	C
<pre>while (condicion) {     instrucciones mientras se cumple la condición }</pre>	<pre>while (condicion) {     instrucciones mientras se cumple la condición }</pre>

- En Java la `condicion` tiene que ser obligatoriamente de tipo `booleano`.
- En C la `condicion` puede ser de tipo entero, real, etc... Se considera cierta cuando es distinta de 0.

# Instrucción iterativa: `while`

- Código para obtener la primera posición que contiene un valor negativo de un array de enteros `v`:

Java	C
<pre>int pos=-1,i=0; while(i&lt;v.length &amp;&amp; pos==-1){     if(v[i]&lt;0)         pos=i;     i++; }</pre>	<pre>int pos=-1,i=0; while(i&lt;TAM_VECTOR &amp;&amp; pos==-1){     if(v[i]&lt;0)         pos=i;     i++; }</pre>



# Instrucción iterativa: `for`

- El bucle `for` tiene la misma sintaxis en ambos lenguajes:

Java	C++
<pre>for(i=ini;i&lt;fin;i++){ instrucciones }</pre>	<pre>for(i=ini;i&lt;fin;i++){ instrucciones }</pre>
<pre>for(i=ini;i&gt;fin;i--){ instrucciones }</pre>	<pre>for(i=ini;i&gt;fin;i--){ instrucciones }</pre>

- En ambos lenguajes la inicialización se puede realizar fuera de la instrucción `for` y la condición para terminar el bucle puede ser compuesta:

```
int i=ini;  
for(;i<fin && v[i]>0;i++){  
instrucciones  
}
```

# Instrucción iterativa: `for`

- Código para obtener el máximo valor de un array de enteros `v`:

Java	C++
<pre>int max=-Integer.MIN_VALUE; int i=0; for(;i&lt;v.length;i++){     if(v[i]&gt;max)         max=v[i]; }</pre>	<pre>int max=-2147483648; int i=0; for(;i&lt;TAM_VECTOR;i++){     if(v[i]&gt;max)         max=v[i]; }</pre>

# Instrucción selección: switch-case

- Tiene la misma sintaxis en ambos lenguajes:

Java	C
<pre>switch(variable) {   case A:...break;   case B:...break;   default:...break; }</pre>	<pre>switch(variable) {   case A:...break;   case B:...break;   default:...break; }</pre>