



Universidad
Zaragoza

Fundamentos de Informática

Lección 4. Instrucciones de
control



Universidad
Zaragoza

Curso 2010-2011

Pedro Álvarez y José Ángel Bañares

24/09/2010. Dpto. Informática e Ingeniería de Sistemas.



Índice de contenidos

- Introducción a las estructuras de control
- Control secuencial
- Estructuras de selección (control condicional)
- Estructuras de repetición (control iterativo)



Índice de contenidos

- **Introducción a las estructuras de control**
- Control secuencial
- Estructuras de selección (control condicional)
- Estructuras de repetición (control iterativo)



Resolución de un problema

- El programador debe especificar:
 - **qué datos** intervienen en el problema
 - **qué acciones** deben ser ejecutadas
 - **en qué orden** deben ejecutarse estas acciones
- } Control del programa
- [Ejemplo] “Esta mañana debo ir a la Universidad bien preparado”
- Las **estructuras de control** son “bloques de construcción” que permiten al programador definir el **control del programa**



Estructuras de control

- Principio de la **programación estructurada**: *" Todo programa puede ser escrito utilizando tres tipos de estructuras de control"*
 - Estructura de secuencia
 - Estructura de selección (o condicional)
 - Estructura de repetición (o iterativa)
- Los lenguajes de programación definen **instrucciones de control** para implementar estas estructuras de control
 - Sintaxis y semántica muy similar en todos ellos

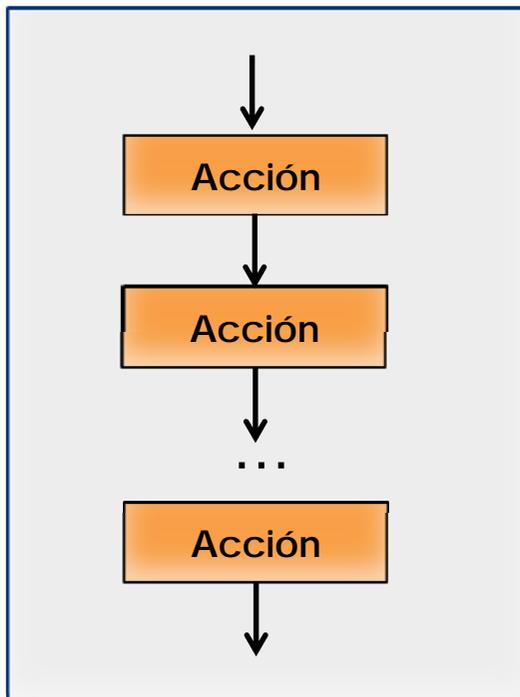


Índice de contenidos

- Introducción a las estructuras de control
- **Control secuencial**
- Estructuras de selección (control condicional)
- Estructuras de repetición (control iterativo)

Control secuencial

- Las acciones son ejecutadas una después de otra, en el orden en el que han sido escritas



Ejemplo en código Java:

```
...  
System.out.print( "Escriba el primer entero: " ); // indicador  
numero1 = entrada.nextInt(); // lee el primer número del usuario  
  
System.out.print( "Escriba el segundo entero: " ); // indicador  
numero2 = entrada.nextInt(); // lee el segundo número del usuario  
  
suma = numero1 + numero2; // suma los números  
  
System.out.printf( "La suma es %d\n", suma ); // muestra la suma  
...
```

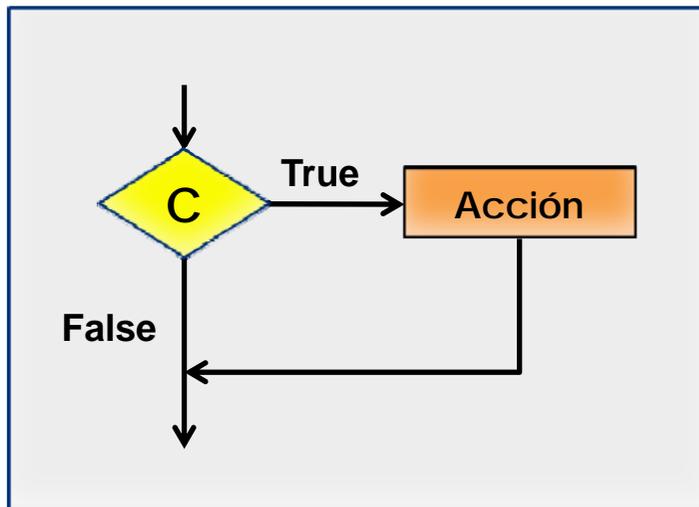


Índice de contenidos

- Introducción a las estructuras de control
- Control secuencial
- **Estructuras de selección (control condicional)**
- Estructuras de repetición (control iterativo)

Selección simple

- La acción (o bloque de acciones) sólo se ejecuta si la condición de control es cierta



Sintaxis Java:

```
if <condición> acción;
```

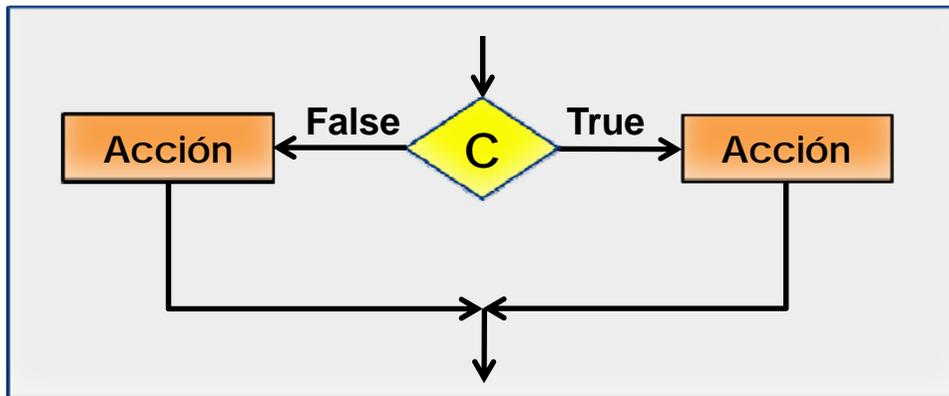
Ejemplos en código Java:

```
if (numero >= 0)
    System.out.println("Es entero positivo");
```

```
if (numero >= 0 && numero < 5)
{
    System.out.println("Es entero positivo");
    System.out.println("Es menor que 5");
}
```

Selección doble

- En función del valor de una condición ejecuta acciones (o bloques de acciones) alternativas



Sintaxis Java:

```
if <condición>
    acción-true;
else
    acción-false;
```

Ejemplos en código Java:

```
if (numero >= 0)
    System.out.println("Es entero positivo");
else
    System.out.println("Es entero negativo");
```

```
if (nota >= 5)
{
    System.out.println("Aprobado");
    System.out.println("Asignatura superada");
}
else
{
    System.out.println("Suspenso");
    System.out.println("Siguiendo convocatoria");
}
```

Selección doble: **if...else** anidados

- Instrucciones **if..else** dentro de instrucciones **if..else** (múltiples alternativas)

Ejemplos en código Java:

```
if (nota >= 9)
    System.out.println("Sobresaliente");
else
    if (nota >= 7)
        System.out.println("Notable");
    else
        if (nota >= 5)
            System.out.println("Aprobado");
        else
            System.out.println("Suspenso");
```



```
if (nota >= 9)
    System.out.println("Sobresaliente");
else if (nota >= 7)
    System.out.println("Notable");
else if (nota >= 5)
    System.out.println("Aprobado");
else
    System.out.println("Suspenso");
```

else siempre asociado al
if más cercano



Selección doble: **if...else** anidados

Ejemplos en código Java:

```
if (x > 5)
    if (y > 5)
        System.out.println("x e y son mayor que 5");
else
    System.out.println("x es menor o igual que 5");
```

Problema del ***else suelto***

Selección doble: **if...else** anidados

Ejemplos en código Java:

```
if (x > 5)
    if (y > 5)
        System.out.println("x e y son mayor que 5");
else
    System.out.println("x es menor o igual que 5");
```



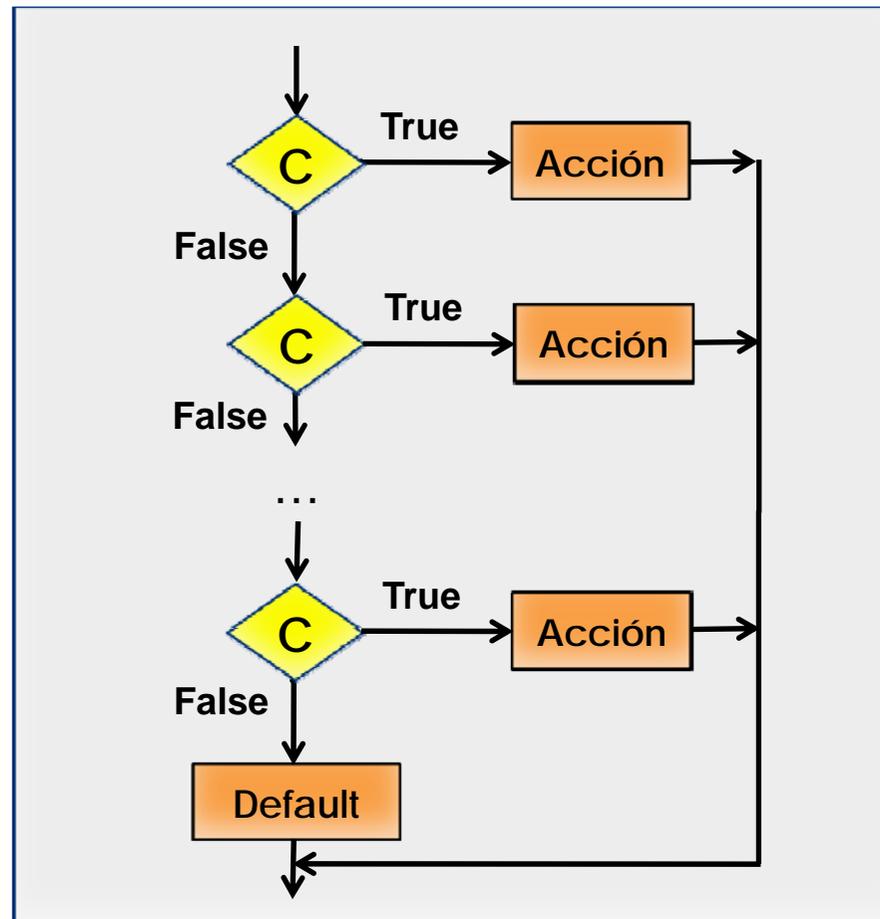
```
if (x > 5)
{
    if (y > 5)
        System.out.println("x e y son mayor que 5");
}
else
    System.out.println("x es menor o igual que 5");
```

Problema del ***else suelto***

Uso de {...}

Selección múltiple

- Ejecuta una acción distinta en base al valor de una variable o expresión entera (múltiples alternativas)





Selección múltiple

- Ejecuta una acción distinta en base al valor de una variable o expresión entera (múltiples alternativas)

Sintaxis Java:

```
switch (variable/expresión)  
{  
    case <valor> : acciones; break;  
    default : acciones;  
};
```

Ejemplos en código Java:

```
switch (dia)  
{  
    case 1: System.out.println("Es lunes");  
        break;  
    case 2: System.out.println("Es martes");  
        break;  
    case 3: System.out.println("Es miércoles");  
        break;  
    case 4: System.out.println("Es jueves");  
        break;  
    case 5: System.out.println("Es viernes");  
        break;  
    default:  
        System.out.println("Fin de semana");  
}
```



Selección múltiple

- **Restricciones** de la instrucción **switch**:
 - **<variable|expresión>** de control debe ser de tipo entero (byte, short, int), carácter o definido por enumeración
 - Cada **<valor>** debe ser una constante (expresión constante o variable constante *-final-*)
 - **Correspondencia de tipo de datos** entre expresión de control y valores
 - No permite definir **rangos de valores** asociados a un caso
 - No puede haber más de **un único caso con el mismo <valor>**
 - Si **no incluyo un break** en un caso, se ejecuta el siguiente caso. Este comportamiento se repite hasta encontrar un break o llegar al final de la estructura

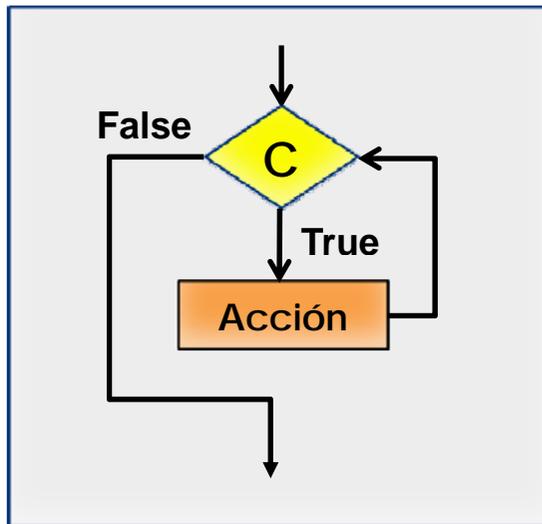


Índice de contenidos

- Introducción a las estructuras de control
- Control secuencial
- Estructuras de selección (control condicional)
- **Estructuras de repetición (control iterativo)**

Repetición pre-test

- Una acción es ejecutada repetidas veces mientras cierta condición sea verdadera



Sintaxis Java:

```
while <condición>  
    acción;
```

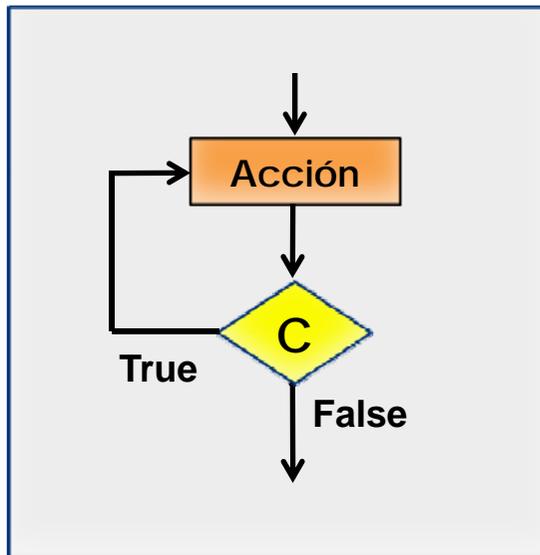
Ejemplos en código Java:

```
int par = 2; // primer número par  
  
while (par <= 100) contador  
{  
    // escribo par y calculo el siguiente  
    System.out.println("%d", par);  
    par = par + 2;  
} //fin while
```

```
numero = entrada.nextInt() // lectura de un entero  
  
while (numero != 0) centinela  
{  
    // escribo par y calculo el siguiente  
    System.out.println("%d", numero);  
    numero = entrada.nextInt()  
} //fin while
```

Repetición post-test

- Similar a la estructura pre-test, pero en este caso la acción se ejecuta antes de evaluar la condición de repetición



Sintaxis Java:

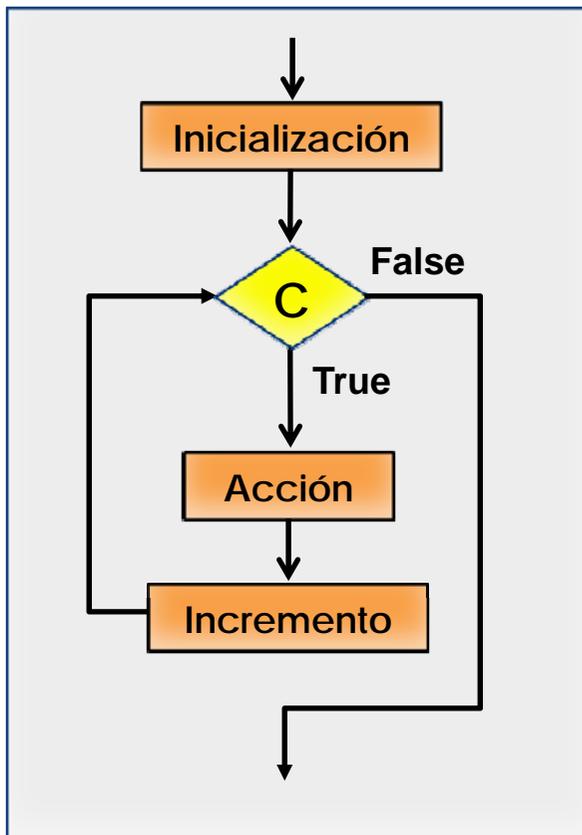
```
do  
    acción;  
while <condición>;
```

Ejemplos en código Java:

```
// Declaración del contador  
int contador = 1; //valor inicial  
  
do  
{  
    // escribo e incremento contador  
    System.out.println("%d", contador);  
    contador = contador + 1;  
}  
while (contador <= 100); //fin do..while
```

Repetición controlada por contador

- La acción se ejecuta un número concreto de veces. El control de la repetición es utilizando un contador



Sintaxis Java:

```
for (<inicialización>; <condición>; <incremento>)  
    acción;
```

Ejemplos en código Java:

```
for (int contador = 2; contador <=20; contador += 2)  
    System.out.println("%d\n", contador);
```

```
for (int mes = 1; mes <=12; mes++)  
{  
    lluvia = entrada.nextInt();  
    System.out.println("%d - %d", mes, lluvia);  
    total += lluvia;  
}
```



¿Cuál es el resultado de ejecución?

```
public class UsoDelFor
{
    public static void main(String[] args)
    {
        int contador; // Contador del FOR

        for (contador = 1; contador <= 10; contador++) // FOR
            System.out.println(contador);

        // ¿Valor final del contador despues del FOR?
        System.out.printf("Valor final del contador: %d", contador);
    }
}
```



¿Cuál es el resultado de ejecución?

```
public class UsoDelFor
{
    public static void main(String[] args)
    {
        // Contador declarado en el FOR
        for (int contador = 1; contador <= 10; contador++)
            System.out.println(contador);

        // ¿Valor final del contador despues del FOR?
        System.out.printf("Valor final del contador: %d", contador);
    }
}
```



¿Cuál es el resultado de ejecución?

```
public class UsoDelFor
{
    public static void main(String[] args)
    {
        // Contador declarado en el FOR
        for (int contador = 1; contador <= 10; contador++)
        {
            System.out.println(contador);
            // Altero el valor de contador dentro del FOR
            contador++;
        }
    }
}
```



Universidad
Zaragoza

