



Introducción a la Programación en Java

Flujo de control. Condicionales y bucles

- Estructura condicional de flujo
 - Al escribir la resolución de un problema, es normal que tengamos en cuenta **condiciones que influyen sobre la secuencia de pasos** que hay que dar para resolver el problema.
 - Las estructuras de control condicionales o selectivas permiten decidir qué ejecutar y qué no en un programa.

Condicional IF

- Permite elegir si se ejecutan o no un bloque de instrucciones
- Sintaxis
 - `if(condición)`
 código a ejecutar si la condición es verdadera;
else
 código a ejecutar si la condición es falsa;
 - `if(condición){`
 Bloque1;
} else {
 Bloque2;
}

Condicional IF (cont.)

- Java proporciona una **forma abreviar una sentencia if** mediante el operador ?
 - Sintaxis:
variable = condición ? instruccion1 : instruccion2;
 - Ejemplo
user = (nombre != null) ? nombre : "Desconodido";

Condicional switch

- Permite seleccionar entre varias alternativas posibles
- Sintaxis

- ```
switch(expresión){
 case valor 1
 sentencia1;
 break;
 case valor 2
 sentencia2;
 break;
 default:
 //sentencia por defecto;
 break;
}
```

## Condicional switch (cont.)

---

Crear un programa que diga cuál es el área de un rectángulo, triángulo y círculo

1. Para ello el usuario debe elegir la figura a la cual quiere calcularle su área
2. Y además debe solicitarle al usuario los datos necesarios, según el tipo de figura que eligió, para lo cual debe conocer:
  - a. Área del cuadrado:  $\text{base}(\text{elevado a la } 2)$
  - b. Área del triángulo  $\text{base} \times \text{altura} / 2$
  - c. Área del círculo:  $\pi \times \text{radio}(\text{elevado a la } 2)$



# Bucles

---

- **Bucles indeterminados.** No sabemos cuantas veces se van a repetir las líneas de código, las repetirá cuantas veces como le deje la condición, mientras la condición sea verdadera seguirá ejecutando las líneas de código
  - While
  - Do - While
- **Determinados.** Sabemos cuantas veces se va a repetir el código que hay en su interior
  - For
  - For – each

# Bucle While

---

- En el bucle while la evaluación se realiza antes de entrar al ciclo, lo que significa que el bucle puede no llegar ejecutarse.
- Sintaxis
  - while( condición ){  
    linea1  
    linea2  
    linea3  
}



## Bucle While (cont.)

---

- Crea una clase con un método main que pida una entrada de texto por teclado y usando:
  - un bucle **while**
  - el método **length()** y el método **substring(x, y)** de la clase String
- Muestre cada una de las letras que componen el texto introducido.
- Por ejemplo si se introduce "ave" debe mostrar:  
Letra 1: a  
Letra 2: v  
Letra 3: e

# Bucle Do – While

---

- En un bucle do - while, la evaluación se hace después de la primera ejecución del ciclo, lo que significa que el bucle obligatoriamente se ejecuta al menos en una ocasión.
- Sintaxis
  - do{  
    linea1  
    linea2  
    linea3  
} while( condición );

# Bucle For

---

- Antes de la primera iteración realiza la inicialización. Ejecuta el cuerpo del bucle mientras la condición sea cierta. Tras cada iteración ejecuta el incremento.
- Sintaxis
  - ```
for (int i = unNumero; i < otroNumero; i++){  
    linea1  
    linea2  
}
```
 - `i`: variable temporal
 - `unNumero`: número en el que se comienza a contar
 - `i < otroNumero`: condición
 - `++` operador incremento

Bucle For (cont.)

- Crear un programa para calcular el factorial de un número.
- el factorial de ***n*** se define en principio como el producto de todos los números enteros positivos desde 1 (es decir, los números naturales) hasta *n*.
Por ejemplo

$$5! = 1 \times 2 \times 3 \times 4 \times 5 = 120.$$

Bucle For - Each

- Para cada **elemento** del tipo **TipoARecorrer** que se encuentre dentro de la colección **nombreDeLaColección** ejecuta las **instrucciones** que se indican.
- Sintaxis
 - for (TipoARecorrer nombreVariableTemporal :
 nombreDeLaColección) {
 Instrucciones
 }

Mostrar los elementos de una lista



Ejercicios prácticos

- condicional IF
- switch
- while
- for

Hemos terminado por hoy

