JAVA

TEMA 03:

ESTRUCTURAS DE CONTROL DE FLUJO

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"

"Una manera de hacer Europa". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

ÍNDICE

Oscar Laguna García
Ana M. Arribas Arjona

JAVA

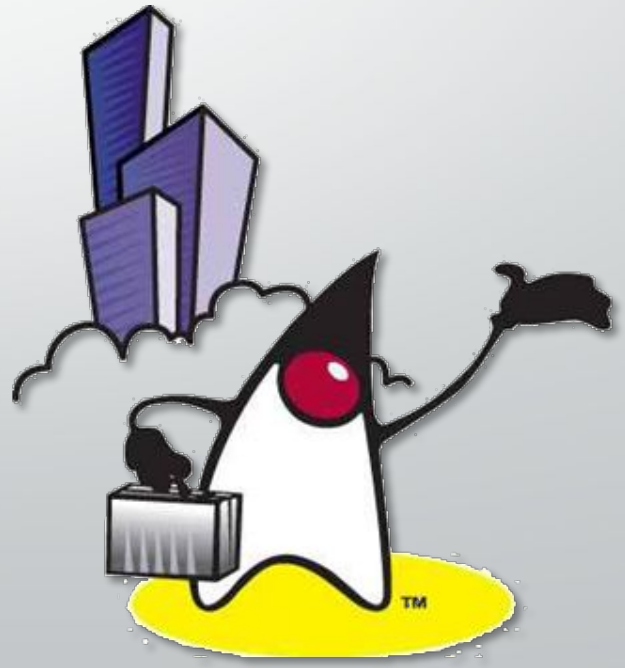
ESTRUCTURAS DE CONTROL DE FLUJO

1. Estructuras condicionales

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"
"Una manera de hacer Europa". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

1.- Estructuras condicionales

- También llamadas estructuras de selección, nos permiten variar el flujo del programa en base a unas determinadas condiciones.
- Pueden ser de tres tipos:
  - Simples: **if**
  - Dobles: **if .... else**
  - Múltiples: **switch**

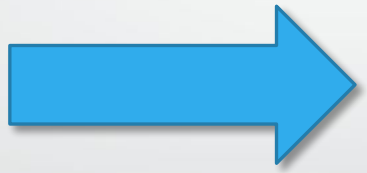


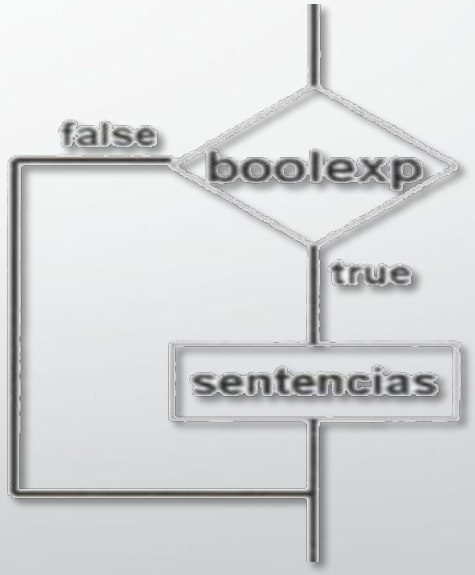
UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"
"Una manera de hacer Europa". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

1.- Estructuras condicionales

- En una **estructura condicional simple**, la sentencia (o sentencias) solo se ejecutarán si se cumple la condición (o condiciones). En caso contrario el programa sigue su curso sin ejecutar la sentencia.
- Sintaxis:

```
if (condicion) {
    sentencia1;
    sentencia2:
        ...
    sentenciaN;
}
```





- Si la expresión **if** tiene una sola sentencia entonces el uso de las llaves sería opcional.

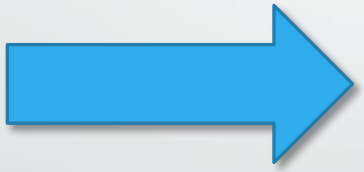
Oscar Laguna García
Ana M. Arribas Arjona

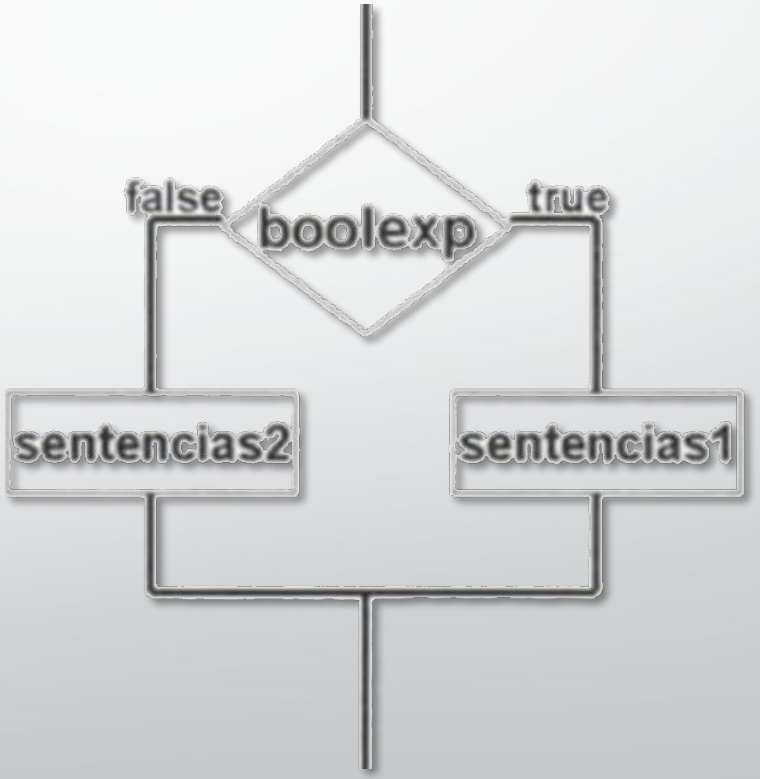
UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"
"Una manera de hacer Europa". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

1.- Estructuras condicionales

- En una **estructura condicional doble**, se ejecutarán unas sentencias u otras en función si se cumple la condición.
- Sintaxis:

```
if (condición) {
        sentencia1;
        sentencia2:
            ...
        sentenciaN;
}
else {
        sentencia1b;
        sentencia2b:
            ...
        sentenciaNb;
}
```





Oscar Laguna García
Ana M. Arribas Arjona

"Una manera de hacer Europa". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

1.- Estructuras condicionales

- Por lo tanto, en una estructura condicional doble, si se cumple la condición (o condiciones) ejecutará las sentencias **1, 2, …, N,** sino ejecutará las sentencias **1b, 2b, …, Nb.** En cualquier caso, el programa continuará a partir de la sentenciaNb.

- Si la expresión if o la expresión else tienen una sola sentencia entonces el uso de las llaves sería opcional en cada caso.

Oscar Laguna García
Ana M. Arribas Arjona

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"
"Una manera de hacer Europa". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

1.- Estructuras condicionales

- Ejemplo:

```
package tema3;
import java.util.Scanner;
/**
 *
 * @author Oscar Laguna García
 */
public class Ejemplo01 {
final static int clave=1234; //Declaro la constante clave con valor 1234
    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // Programa que simula una clave de acceso
        Scanner entrada = new Scanner (System.in );
        /* Creo un objeto llamado entrada para capturar lo que me
        introduzca el usuario por teclado */
        int usuario; //Para almacenar la clave que introduzca el usuario
        System.out.println("Introduce la clave de acceso:");
        usuario = entrada.nextInt(); /* leo la clave que mete el usuario
                                        y la almaceno en usuario*/
        if (usuario == clave) {
                System.out.println ("Acceso permitido");
        }
        else {
                System.out.println ("Acceso denegado");
        }
    }
}
```

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"
"Una manera de hacer Europa". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

1.- Estructuras condicionales

- También existe la posibilidad de anidar varios **if … else.**
- Con este formato el flujo del programa únicamente entra **en una** de las condiciones.
- Cuando una de las condiciones se cumple, se ejecuta la sentencia correspondiente y luego salta hasta el final de la estructura para continuar con el programa.
- Sintaxis:

Oscar Laguna García
Ana M. Arribas Arjona

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"
"Una manera de hacer Europa". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

1.- Estructuras condicionales

```
if (condición_A) {
        sentencias_A;
}
else {
      if (condición_B) {
            sentencias_B;
      }
      else {
            if (condición_C) {
                  sentencias_C;
            }
            else {
                  sentencias_D;
            }
      }
}
```

Oscar Laguna García
Ana M. Arribas Arjona

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"
"Una manera de hacer Europa". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

1.- Estructuras condicionales

```java
import java.util.Scanner;
/**
 * @author Oscar Laguna García
 */
public class Ejemplo02 {
      public static void main(String[] args) {
// Programa que determina tu momento de la vida
        Scanner entrada = new Scanner (System.in );
        /* Creo un objeto llamado entrada para capturar lo que me
        introduzca el usuario por teclado */
        int edad; //Para almacenar la edad del usuario
        System.out.println("Introduzca su edad:");
        edad = entrada.nextInt(); /* leo lo que mete el usuario
                                     y lo almaceno en edad*/
        if (edad < 1) {
            System.out.println ("Lo siento, te has equivocado");
        }
        else {
            if (edad < 3) {
                System.out.println ("Eres un bebé");
            }
            else {
                if (edad < 13) {
                    System.out.println ("Eres un niño");
                }
                else {
                    System.out.println ("Ya eres adulto");
                }
            }
        }
    }
}
```

Oscar Laguna García
Ana M. Arribas Arjona

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"
"Una manera de hacer Europa". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

1.- Estructuras condicionales

- La última estructura condicional es **switch**.
- Se suele utilizar para crear un menú de opciones, de manera que según la opción seleccionada por el usuario, se ejecuten una serie de sentencias u otras.
- Sintaxis:

Oscar Laguna García
Ana M. Arribas Arjona

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"
"Una manera de hacer Europa". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

1.- Estructuras condicionales

```
switch (variable) { // debe ser short, int, byte o char
        case valor1:
                sentencias;
                break; // Para que salga de la estructura
        case valor2:
                sentencias;
                break; //Para que salga de la estructura
        default:
            sentencia por defecto; // se ejecuta cuando la variable no se ajusta a ningún valor
}
```


Oscar Laguna García
Ana M. Arribas Arjona

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"
"Una manera de hacer Europa". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

1.- Estructuras condicionales

- Cada case podrá incluir una o varias sentencias.
- La sentencia break sirve para salir de la estructura switch sin tener que pasar por el resto de case comprobando si se cumple el valor de la **variable**.
- La **variable** evaluada sólo puede ser de tipo short, int, byte o char.
- default sólo se ejecutará en caso de el valor de la **variable** no esté contemplada en ningún case.

Oscar Laguna García
Ana M. Arribas Arjona

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"
"Una manera de hacer Europa". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

1.- Estructuras condicionales

- Ejemplo:

```
import java.util.Scanner;
/**
 * @author Oscar Laguna García
 */
public class Ejemplo03 {
      public static void main(String[] args) {
        // Programa que determina el día de la semana
        Scanner entrada = new Scanner (System.in );
        /* Creo un objeto llamado entrada para capturar lo que me
        introduzca el usuario por teclado */
        int dia; //Para almacenar el numero que introduzca el usuario
        System.out.println("Introduzca el número de día de la semana:");
        dia = entrada.nextInt(); /* leo lo que mete el usuario y lo
                                    almaceno en la variable dia*/
        switch(dia){
            case 1: {
                System.out.println("Has elegido el Lunes");
                break;
            }
            case 2: {
                System.out.println("Has elegido el Martes");
                break;
            }
            case 3: {
                System.out.println("Has elegido el Miercoles");
                break;
            }
        ..................
            default: {
                System.out.println("Te has equivocado. Ese día no existe");
                }
        }
```


Oscar Laguna García
Ana M. Arribas Arjona

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"
"Una manera de hacer Europa". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

EJERCICIOS

- **Ejercicio 01.- (<u>OBLIGATORIO</u>)** Implementa un algoritmo en JAVA que le pida al usuario un número por teclado. Posteriormente el programa le dirá al usuario si el número introducido es positivo o negativo.
- Muestra por pantalla el resultado de la siguiente forma:

Por favor, introduzca un numero: xxx

El número introducido es positivo o negativo

Oscar Laguna García
Ana M. Arribas Arjona

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"
"Una manera de hacer Europa". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

EJERCICIOS

- **Ejercicio 02.- (OPTATIVO)** Realiza un programa en el que le solicites al usuario 2 números y, si el primer número introducido es mayor que 10, se multipliquen, y en caso contrario que se sumen. Muestra al usuario la operación realizada y el resultado.
- Muestra por pantalla el resultado de la siguiente forma:

Por favor, introduzca un numero: xxx

Ahora, introduzca un segundo numero: xxx

La operación que se realizó es suma o producto y el resultado es xxx

Oscar Laguna García
Ana M. Arribas Arjona

EJERCICIOS

- **Ejercicio 03.- (<u>OPTATIVO</u>)** Diseña un programa en JAVA que lea tres números e imprima por pantalla el mayor de ellos.
- Muestra por pantalla el resultado de la siguiente forma:

Por favor, introduzca el primer numero: xxx

Ahora, introduzca un segundo numero: xxx

Por último, introduzca un tercer numero: xxx

El número mayor de los introducidos es el xxx

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"
"Una manera de hacer Europa". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

EJERCICIOS

- **Ejercicio 04.- (<u>OBLIGATORIO</u>)** Escribir un algoritmo en JAVA que pida tres números e imprima por pantalla el menor de ellos.

Oscar Laguna García
Ana M. Arribas Arjona

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"
"Una manera de hacer Europa". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

EJERCICIOS

- **Ejercicio 05.- (<u>OBLIGATORIO</u>)** Implementa un algoritmo en JAVA que le pida al usuario un número por teclado. Posteriormente, el programa le dirá al usuario si el número introducido es par o impar.

EJERCICIOS

- **Ejercicio 06.- (OPTATIVO)** Crea un programa en JAVA en donde el usuario introduzca la nota de un alumno (número entero entre 0 y 10) y se escribirá su calificación según el valor de la nota ingresada:
  - 0 a 4 = Suspenso.
  - 5 a 6 = Bien.
  - 7 a 8 = Notable.
  - 9 a 10 = Sobresaliente.
- ***Nota:*** Se le avisará al usuario de un error en caso de que la nota que nos introduzca no esté entre 0 y 10.

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"
"Una manera de hacer Europa". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

EJERCICIOS

- **Ejercicio 07.- (OBLIGATORIO)** Realiza un programa en JAVA en el que tenga cabida, sin modificar, el siguiente trozo de código:

```
switch (diasemana) {
   case 1:
   case 2:
   case 3:
   case 4:
   case 5:
         laborable=true;
         break;
   case 6:
   case 7:
         laborable=false;
}
```


Oscar Laguna García
Ana M. Arribas Arjona

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"
"Una manera de hacer Europa". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

EJERCICIOS

- **Ejercicio 08.- (OBLIGATORIO):** Realiza un programa que dado un importe en euros nos indique número óptimo de billetes de 50, 20, 10 y 5, así como la cantidad sobrante en monedas de 2 y de 1 euro. En caso de que NO haya billetes/monedas de algún tipo NO se mostrarán.

- Por ejemplo:

Por favor, indique una cantidad de dinero: ***232***

232 *Euros se descomponen en:*

Billetes de 50: ***4***

Billetes de 20: ***1***

Billetes de 10: ***1***

Monedas de 2 euros: ***1***

En el tema anterior: ***232*** *Euros se descomponen en* ***4*** *billetes de 50,* ***1*** *billetes de 20,* ***1*** *billetes de 10,* ***0*** *billetes de 5,* ***1*** *monedas de 2 euros y* ***0*** *monedas de 1 euro.*

Oscar Laguna García
Ana M. Arribas Arjona

EJERCICIOS

- **Ejercicio 09.- (<u>OBLIGATORIO</u>)** Escribe un programa en JAVA en el que el usuario introduzca cuatro números enteros y luego el programa los muestre por pantalla ordenados de forma creciente.(de menor a mayor)
- Muestra por pantalla el resultado de la siguiente forma:

Por favor, introduzca el primer numero: 8

Ahora, introduzca un segundo numero: 5

Introduzca el tercer numero: 9

Por último, introduzca un cuarto numero: 1

El orden de los números introducidos es el 1 - 5 - 8 - 9

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"
"Una manera de hacer Europa". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

EJERCICIOS

- **Pista para el Ejercicio 09:** Una de las formas más utilizada a la hora de ordenar números en cualquier lenguaje de programación es el "Método de la Burbuja". Consiste en ir comparando los números de 2 en 2 e ir ordenándolos entre ellos.
- *Ejemplo: 8591*
  - *Comparo el 8 y el 5. Como el 5 es menor intercambio posiciones: 5891.*
  - *Comparo el 8 y el 9. Como el 8 no es menor, no hago nada.5891*
  - *Comparo el 9 y el 1. Como el 1 es menor intercambio posiciones: 5819.*
  - *Repito el ciclo 2 veces más (ya que tengo 4 números):*
  - *Comparo el 5 y el 8. Como el 8 no es menor, no hago nada.5819*
  - *Comparo el 8 y el 1. Como el 8 es menor……*

JAVA

ESTRUCTURAS DE CONTROL DE FLUJO

2. Estructuras repetitivas

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"
"Una manera de hacer Europa". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

2.- Estructuras repetitivas

- También llamadas estructuras de repetición, bucles o estructuras cíclicas. Nos permiten ejecutar partes del código de forma repetida mientras se cumpla una condición (o varias condiciones unidas por operadores lógico)
- Son de tres tipos:
  - while
  - do while
  - for

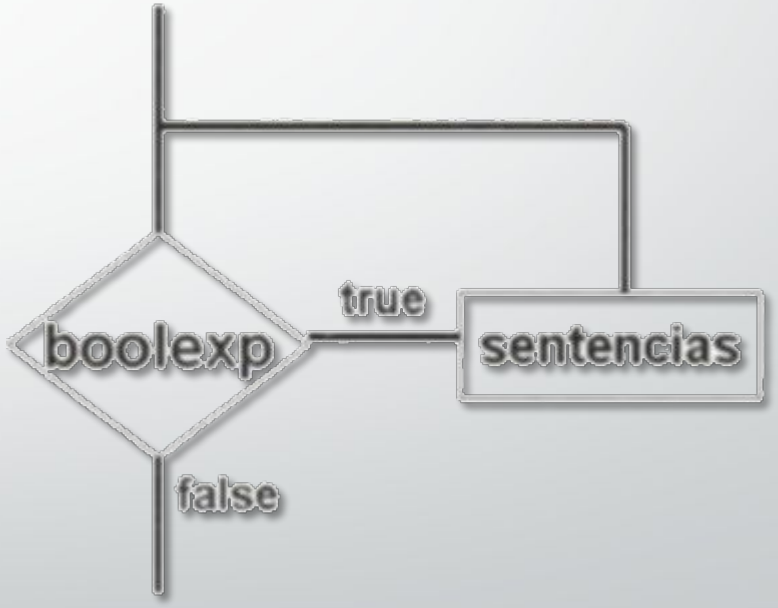
Oscar Laguna García
Ana M. Arribas Arjona

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"
"Una manera de hacer Europa". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

2.- Estructuras repetitivas

- La estructura repetitiva **while** la utilizaremos cuando queramos que un grupo de sentencias se ejecute 0 o más veces en función de una condición.
- Sintaxis:

```
while (condición) {
    sentencia1;
    sentencia2:
    …
    sentenciaN;
}
```

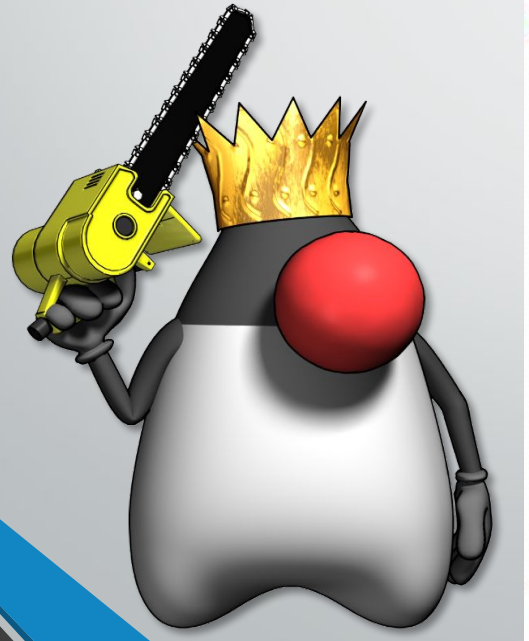


- Con esta sentencia se controla la condición antes de entrar en el bucle. Si ésta no se cumple, el programa no entrará en el bucle.

Oscar Laguna García
Ana M. Arribas Arjona

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"
"Una manera de hacer Europa". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

2.- Estructuras repetitivas



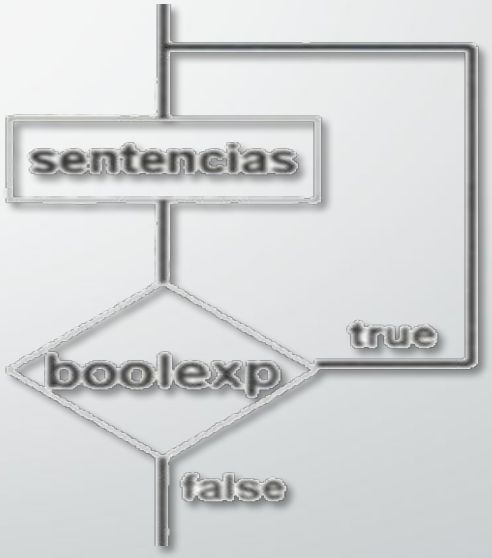
```java
/*
 * Ejemplo de uso de la sentencia while
 */
package ejemplo04;
/**
 * @author Oscar laguna García
 */
public class Ejemplo04 {
    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // Ejemplo de un programa que escribe los números del 1 al 10
        int numero; //Declaro la variable numero de tipo entero
        numero = 1; //Inicializo la varible numero a 1
        while (numero <= 10){ //Mientras que numero sea menor o igual que 10
            System.out.println(numero); //Muestro por pantalla numero
            numero ++; //Equivale a numero=numero+1
        }
    }
}
```

"Una manera de hacer Europa". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

2.- Estructuras repetitivas

- La estructura repetitiva **do ... while** tiene la siguiente sintaxis:

```
do {
    sentencia1;
    sentencia2:
    ...
    sentenciaN;
} while (condicion);
```



- Se controla la condición al final del bucle, con lo cual siempre se ejecutarán las sentencias al menos una vez. Si la condición se cumple el programa vuelve a ejecutar las sentencias del bucle.

Oscar Laguna García
Ana M. Arribas Arjona

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"
"Una manera de hacer Europa". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

2.- Estructuras repetitivas

```java
/*
 * Ejemplo de uso de la sentencia do ... while
 */
package ejemplo05;
import java.util.Scanner;
/**
 * @author Oscar Laguna Garcia
 */
public class Ejemplo05 {
    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // Programa que muestra un menú y del que no se sale hasta pulsar 3
        int opcion; //Para almacenar la opción que introduzca el usuario
        Scanner entrada = new Scanner (System.in );
        /* Creo un objeto llamado entrada para capturar lo que me
        introduzca el usuario por teclado */
        do{
            System.out.println("Pulse 1 para ver HOLA");
            System.out.println("Pulse 1 para ver BUENAS TARDES");
            System.out.println("Pulse 3 para salir del programa");
            opcion = entrada.nextInt();
            switch (opcion){
                case 1:{
                    System.out.println("HOLAAAAA");
                    break;
                }
                 case 2:{
                    System.out.println("BUENOS DIAAAASSSS");
                    break;
                }
                  case 3:{
                    System.out.println("Gracias por utilizar mi programa.");
                    break;
                }
                   default:{
                    System.out.println("Te has equivocado de opción.");
                    break;
                }
            }
        }while (opcion !=3);
    }
}
```

Oscar Laguna García
Ana M. Arribas Arjona

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"
"Una manera de hacer Europa". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

2.- Estructuras repetitivas

- La estructura repetitiva **for** la utilizaremos cuando queramos que un grupo de sentencias se ejecute un número fijo y conocido de veces.

- Sintaxis:

```
for (inicialización; condición; incremento) {
    sentencia1;
    sentencia2:
    …
    sentenciaN;
}
```

- Ejemplo: **for (x=1; x<10; x++)** // **Desde x igual a 1** y **mientras que x sea menor que diez sumamos 1 a x**

Oscar Laguna García
Ana M. Arribas Arjona

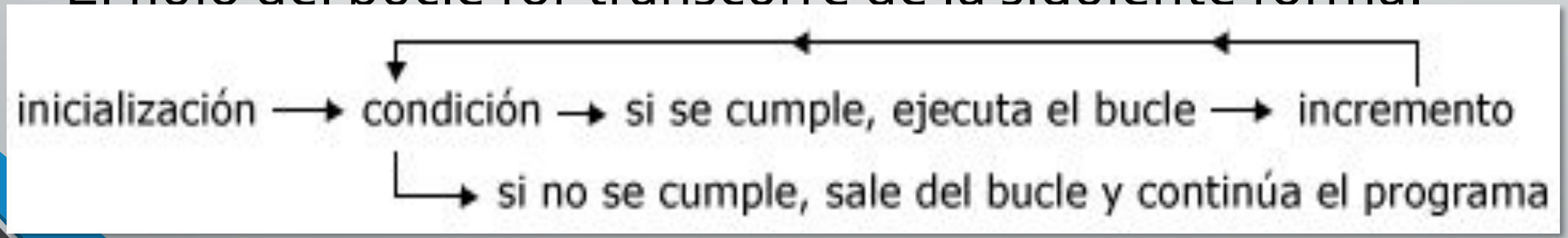
UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"
"Una manera de hacer Europa". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

2.- Estructuras repetitivas

- La inicialización indica una variable (variable de control) que condiciona la repetición el bucle. Si hay más, van separadas por comas.

  Ejemplo: for (a=1, b=100; a!=b, a++, b--) {

- El fluio del bucle for transcurre de la siguiente forma:

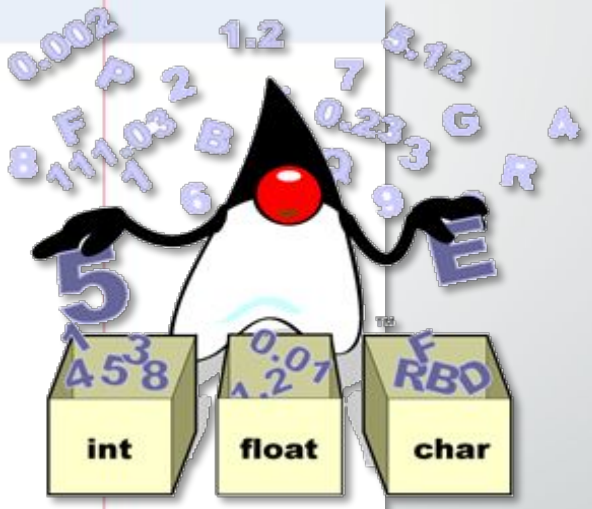


Oscar Laguna García
Ana M. Arribas Arjona

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"
"Una manera de hacer Europa". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

2.- Estructuras repetitivas

```java
/*
 * Ejemplo de uso de la sentencia for.
 */
package ejemplo06;
/**
 * @author Oscar Laguna García
 */
public class Ejemplo06 {
    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // Programa que escribe la tabla de multiplicar del numero 6
        int resultado; // Declaro la variable resultado
        int i; //Declaro la variable i para utilizarla en el bucle for
        for (i=0; i<=10; i++){ //Desde i=0 y mientras i sea menor o igual que 10 sumale 1 a i.
            resultado = 6 * i;
            System.out.println("6 por " + i + " es " +resultado);
        }
    }
}
```



UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"
"Una manera de hacer Europa". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

EJERCICIOS

- **Ejercicio 10.- (OPTATIVO)** Escribe un programa en JAVA que, utilizando bucles, muestre por pantalla el mensaje "Hola" cinco veces.
- *Pista: Como sabes cuantas veces se ejecuta el bucle, deberás utilizar un bucle for.*

EJERCICIOS

- **Ejercicio 11.- (<u>OBLIGATORIO</u>)** Crea un programa en JAVA que, utilizando bucles, muestre por pantalla el mensaje "Hola" seis veces acompañado por un numero que se incrementa cada vez.
- Muestra por pantalla el resultado de la siguiente forma:

- Hola1 – Hola2 – Hola3 – Hola4 – Hola5 – Hola6 -

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"
"Una manera de hacer Europa". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

EJERCICIOS

- **Ejercicio 12.- (<u>OPTATIVO</u>)** Crea un algoritmo en JAVA que, <u>utilizando un bucle ***do…while***</u>, imprima los números pares que existen entre el número 11 y el número 133.

Oscar Laguna García
Ana M. Arribas Arjona

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"
"Una manera de hacer Europa". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

EJERCICIOS

- **Ejercicio 13.- (<u>OBLIGATORIO</u>)** Crea un algoritmo en JAVA que, <u>utilizando un bucle ***while***</u>, imprima los números pares que existen entre el número 11 y el número 133.

Oscar Laguna García
Ana M. Arribas Arjona

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"
"Una manera de hacer Europa". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

EJERCICIOS

- **Ejercicio 14.- (<u>OPTATIVO</u>)** Implementa un algoritmo en JAVA que, utilizando bucles, imprima los 100 primeros números pares.

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"
"Una manera de hacer Europa". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

EJERCICIOS

- **Ejercicio 15.- (OBLIGATORIO)** Escribe un programa en JAVA que, utilizando bucles, imprima la tabla de multiplicar de un número que elija el usuario.
- Ejemplo:

Introduzca un número para calcular su tabla de multiplicar: 8

8 x 0 = 0

8 x 1 = 8

8 x 2 = 16

8 x 3 = 24 …

Oscar Laguna García
Ana M. Arribas Arjona

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"
"Una manera de hacer Europa". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

EJERCICIOS

- **Ejercicio 16.- (<u>OBLIGATORIO</u>)** Crea un programa que imprima los números impares que existen entre los números 20 y el 160. Además, al final, nos dirá cuantos impares ha imprimido en total por pantalla.
- Ejemplo:

Los números impares existentes entre el número 20 y el 160 son:

21 – 23 – 25 – 27 – 29 – 31 - …

La cantidad de números impares impresos han sido: XXX

Oscar Laguna García
Ana M. Arribas Arjona

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"
"Una manera de hacer Europa". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

EJERCICIOS

- **Ejercicio 17.- (OPTATIVO)** Crea un programa que calcule la raíz cuadrada del número que introduzca el usuario. (Utiliza el método Math.sqrt())
- Si el usuario introduce un número negativo, debemos mostrarle un mensaje de error y volver a pedírselo (tantas veces como sea necesario).
- *Pista: Como sabes que al menos se ejecutará el bucle una vez, deberás utilizar un bucle do…while.*

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"
"Una manera de hacer Europa". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

EJERCICIOS

- **Ejercicio 18.- (OBLIGATORIO)** Realiza un programa que le pida una contraseña al usuario. Si la escribe bien le dará la enhorabuena, pero si la escribe mal 3 veces le dará un mensaje de error de acceso.
- *Pista: Como sabes que al menos se ejecutará el bucle una vez, deberás utilizar un bucle do…while.*

JAVA

ESTRUCTURAS DE CONTROL DE FLUJO

3. Estructuras de salto

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"
"Una manera de hacer Europa". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

3.- Estructuras de salto

- Las sentencias de salto son break y continue.
- Sirven para alterar la ejecución normal de un bucle.
- Salvo el uso de break para la estructura switch, las sentencias de salto dificultan la legibilidad de los programas y en algunos ámbitos no está bien visto su uso, por lo que se desaconseja su utilización.

Oscar Laguna García
Ana M. Arribas Arjona

"Una manera de hacer Europa". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

3.- Estructuras de salto

- **break:**
  - Como ya vimos, esta sentencia se utiliza para salir de una sentencia switch, pero también puede ser usada para interrumpir bruscamente la ejecución de un bucle y que salga de él.
- **continue:**
  - Cuando un programa llega a una sentencia continue no ejecuta las líneas de código que hay a continuación, sino que salta a la siguiente iteración del bucle.

Oscar Laguna García
Ana M. Arribas Arjona

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"
"Una manera de hacer Europa". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

3.- Estructuras de salto

- Ejemplo de uso de **break**:

```
package suma;

// Programa de ejemplo del uso de break

public class Suma {

    // el método main empieza la ejecución de la aplicación Java

    public static void main(String args[]) {

        for (int i = 0; i < 10; i++) {
            System.out.println("Este es el número"+i);
            if (i==5){
                break;
            }
        }
    } // fin del método main
} // fin de la clase Suma
```

```
uma.Suma
out - Suma (run)
run:
Este es el número0
Este es el número1
Este es el número2
Este es el número3
Este es el número4
Este es el número5
BUILD SUCCESSFUL (total time: 0 seconds)
```


Oscar Laguna García
Ana M. Arribas Arjona

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"
"Una manera de hacer Europa". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

3.- Estructuras de salto

- Ejemplo de uso de **continue**:


```java
/*
 * Ejemplo de uso de la sentencia continue.
 */
package ejemplo06;
/**
 * @author Oscar Laguna García
 */
public class Ejemplo06 {
    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // Programa que muestra los números del 1 al 10 saltándose el numero 5
        int i=0; //Declaro la variable i para utilizarla en el bucle for
        while (i<10){
            i++;
            if (i==5){
                continue;
            }
            System.out.println(i);
        }
    }
}
```

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"
"Una manera de hacer Europa". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

EJERCICIOS

- **Ejercicio 19.- (<u>OBLIGATORIO</u>)** Modifica el ejemplo de los apuntes en el que se mostraba el uso de la sentencia "break" para que el programa haga lo mismo pero sin utilizar esta sentencia "prohibida".

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"
"Una manera de hacer Europa". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

EJERCICIOS

- **Ejercicio 20.- (<u>OPTATIVO</u>)** Modifica el ejemplo de los apuntes en el que se mostraba el uso de la sentencia "continue" para que el programa haga lo mismo pero sin utilizar esta sentencia "prohibida".

JAVA

ESTRUCTURA Y SINTAXIS DE UN PROGRAMA EN JAVA

4. Control de excepciones

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"
"Una manera de hacer Europa". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

4.- Control de excepciones

- Permite al programador controlar la ejecución del programa evitando que este falle de forma inesperada. Sintaxis:

```
try {
        sentencia a proteger
} catch (excepción_1) {
    control de la excepción 1;
}
...
catch (excepción_n) {
    control de la excepción n;
}
finally {
    control opcional; //trozo de código que
se ejecuta siempre
}
```

Oscar Laguna García
Ana M. Arribas Arjona

4.- Control de excepciones

- Las excepciones más habituales son las siguientes:

Clase de excepción	*Significado*
ArithmeticException	Una condición aritmética excepcional ha ocurrido. Por ejemplo, una división por 0.
ArrayIndexOutOfBoundsException	Una matriz fue accedida con un índice ilegal (fuera de los límites permitidos).
NullPointerException	Se intentó utilizar **null** donde se requería un objeto.
NumberFormatException	Se intento convertir una cadena con un formato inapropiado en un número.

- Las iremos viendo en ejemplos durante el transcurso del año.

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"
"Una manera de hacer Europa". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

4.- Control de excepciones

- Veamos el siguiente trozo de código:

- Al dividir por cero se da una indeterminación, lo que provocará un error en el programa y una finalización anormal del mismo.

```
/*
 * Ejemplo de division indeterminada
 */
package ejemplo09;
/**
 * @author Oscar Laguna García
 */
public class Ejemplo09 {
    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // Programa que intenta hacer una división y muestra un error
        int a, b, c;
        a=10;
        b=0;
        c=a/b;
        System.out.println("Resultado: " + c);
    }
}
```

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"
"Una manera de hacer Europa". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

4.- Control de excepciones

- Vamos a capturar la excepción, que en este caso es del tipo ArithmeticException:

```
/*
 * Ejemplo de control de excepciones
 */
package ejemplo09;
/**
 * @author Oscar Laguna García
 */
public class Ejemplo09 {
    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // Programa que hace una división
        int a, b, c;
        a=10;
        b=0;
        try {
            c=a/b;
        }
        catch(ArithmeticException e){
                    System.out.println("Error: " +e.getMessage());
                    c=0; //Asigno el valor 0 para que prosiga el programa
        }
        System.out.println("Resultado: " + c);
    }
}
```

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"
"Una manera de hacer Europa". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

EJERCICIOS

- **Ejercicio 21.- (OPTATIVO)** Crea un programa que calcule el resultado de dividir los números que introduzca el usuario.
- En caso de que el usuario introduzca un número divisor igual a 0, debemos capturar la excepción y mostrarle un mensaje de error al usuario.

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"
"Una manera de hacer Europa". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

EJERCICIOS

- **Ejercicio 22.- (OBLIGATORIO)** Crea un programa que calcule sume dos números que introduzca el usuario.
- En caso de que el usuario introduzca una letra en vez de un número, debemos capturar la excepción y mostrarle un mensaje de error.

JAVA

ESTRUCTURA Y SINTAXIS DE UN PROGRAMA EN JAVA

5. Ejercicios de consolidación

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"
"Una manera de hacer Europa". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

EJERCICIOS

- **Ejercicio 23.- (<u>OPTATIVO</u>)** Realiza un algoritmo que imprima todos los números existentes entre el número 1 y otro introducido por el usuario.
- Controla que el usuario te meta un número mayor que 1 y, sino, avísale del error y vuélveselo a pedir las veces que hagan falta. (hasta que introduzca un número mayor que 1)

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"
"Una manera de hacer Europa". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

EJERCICIOS

- **Ejercicio 24.- (OBLIGATORIO)** Crea un programa en JAVA que que imprima todos los números múltiplos de 3 que existen entre el número 1 y otro número introducido por el usuario.
- Controla que el usuario te meta un número mayor que o y, sino, avísale del error y vuélveselo al pedir las veces que hagan falta.
- Por último infórmale al usuario del total de números mostrados.

Oscar Laguna García
Ana M. Arribas Arjona

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"
"Una manera de hacer Europa". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

EJERCICIOS

- **Ejercicio 25.- (<u>OPTATIVO</u>)** Escribe un programa en JAVA que te diga la suma total de los números pares existentes entre el número 17 y el número 139.

Oscar Laguna García
Ana M. Arribas Arjona

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"
"Una manera de hacer Europa". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

EJERCICIOS

- **Ejercicio 26.- (<u>OPTATIVO</u>)** Diseña un programa en JAVA que te diga la suma total de los números impares existentes entre el 111 y el 222.

Oscar Laguna García
Ana M. Arribas Arjona

EJERCICIOS

- **Ejercicio 27.- (OBLIGATORIO)** Diseña un programa en JAVA que pida al usuario dos números por teclado. Posteriormente el programa mostrará un menú que le permitirá al usuario:
  - 1.- Sumar los números.
  - 2.- Restar los números.
  - 3.- Multiplicar los números.
  - 4.- Dividir los números.
  - 5.- Salir del programa.
- Nota1: Mientras el usuario no pulse 5, el programa no termina y el menú volverá a aparecer pidiendo nuevamente que le introduzcas una opción.
- Nota 2: Controla el caso de división entre o mediante la captura de excepciones.

EJERCICIOS

- **Ejercicio 28.- (OBLIGATORIO)** Realiza un programa que genere un número aleatorio (*utiliza Math.random()*) entre 1 y 100, que lo muestre por pantalla e indique si es par o impar.
- **Pistas:** El método Math.random() devuelve un double aleatorio entre 0 y el 1, sin incluir el 1 (o sea, desde el 0 y el 0,99999). Ejemplo: double aleatorio = Math.random();
- Si quisiéramos un aleatorio entre 0 y 4 (exclusivo) bastará con multiplicar por 4. Ejemplo: double aleatorio = (Math.random())*4;
- Si quisiéramos un aleatorio entre 1 y 4 (exclusivo) bastará con multiplicar por 4 y sumar 1. Ejemplo: double aleatorio = (Math.random())*4+1;

EJERCICIOS

- **Si queremos obtener un número aleatorio entero, en vez de double, lo podemos hacer en dos pasos:**

I. **Mediante el método Math.floor**, que redondea al entero inferior. Ejemplo de aleatorio entre 0 y 3:

```
double aleatorio = Math.floor((Math.random()*4);
//Otra forma:
double aleatorio=(Math.random()*4);
aleatorio=Math.floor(aleatorio);
```

II. **Luego, hacemos cast**:

```
int aleatorioEntero = (int) aleatorio;
```

EJERCICIOS

- **Ejercicio 29.- (OBLIGATORIO)** Escribe un programa que juegue con el usuario a adivinar un número. El ordenador debe generar un número entero aleatorio entre 1 y 100, y el usuario tiene que intentar adivinarlo.
- Para ello, cada vez que el usuario introduce un valor el ordenador debe decirle al usuario si el número que tiene que adivinar es mayor o menor que el que ha introducido.
- Cuando consiga adivinarlo debe indicárselo e imprimir en pantalla el número de veces que el usuario ha intentado adivinar el número.

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"
"Una manera de hacer Europa". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

EJERCICIOS

- **Ejercicio 30.- (<u>OBLIGATORIO</u>)** Mejora el ejercicio 29, de tal forma que si el usuario introduce algo que no es un número (una letra, por ejemplo) se le avisará del error por pantalla y se contará como un intento.
- Pista: Necesitarás capturar excepciones.

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"
"Una manera de hacer Europa". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

EJERCICIOS

- **Ejercicio 31.- (<u>OBLIGATORIO</u>)** Desarrolla un programa que genere números enteros aleatorios entre 1 y 100 hasta obtener 3 números impares.
- Al final del programa nos mostrará los tres números impares generados y la cantidad de valores aleatorios que han sido necesarios generar hasta obtener los 3 números impares.

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"
"Una manera de hacer Europa". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

EJERCICIOS

- **Ejercicio 32.- (OBLIGATORIO)** Realiza un programa que calcule la edad de una persona, solicitando la fecha actual y la fecha de su nacimiento.
- Ejemplo de ejecución:

*Introduzca el año actual: **XXXX***

*Introduzca el mes actual: **XX***

*Introduzca el día actual: **XX***

*Ahora, introduzca su año de nacimiento: **XXXX***

*Introduzca su mes de nacimiento: **XX***

*Por último introduzca su día de nacimiento: **XX***

*Su edad exacta es de **XX** años, **XX** meses y **XX** días.*

Oscar Laguna García
Ana M. Arribas Arjona

UNION EUROPEA
Fondo Social Europeo
"Una manera de hacer Europa"
"Una manera de hacer Europa". Cofinanciación a cargo del Programa Operativo del FSE 2014-2020 para Extremadura gastos de Ciclos Formativos de Grados Medio y Superior.

EJERCICIOS

- **Ejercicio 34.- (OPTATIVO)** Realizar un programa que calcule el tiempo en horas y minutos que tarda un tren en llegar a su destino.
- Para ello se le pedirán al usuario el día, hora y minuto de salida y el día, hora y minuto de llegada.
- Controla que la hora de llegada sea posible (no se puede retroceder en el tiempo), pero ten en cuenta que, por ejemplo, un tren puede salir a las 6 de la tarde y llegar a las 5 de la tarde del día siguiente.

Oscar Laguna García
Ana M. Arribas Arjona