

Clase 4

# PROGRAMACIÓN 1

# Objetivos del tema

- Describir el funcionamiento de las sentencias selectivas o condicionales (if-else y switch)
- Interpretar el resultado de una secuencia de sentencias condicionales combinadas o no
- Codificar una tarea sencilla utilizando la secuencia y combinación de sentencias condicionales

# Sentencia if-else

- Las sentencias de control de flujo se emplean en los programas para ejecutar sentencias condicionalmente, o cambiar el flujo secuencial de ejecución
- Para un cambio de flujo de ejecución de secuencias

```
if (expresionLogica) {  
    sentencia_1;  
    ...  
    sentencia_N;  
}
```

- Para dos posibles flujos de ejecución

```
if (expresionLogica) {  
    sentencia_1;  
    ...  
    sentencia_N;  
}  
else { //flujo de ejecución en el caso de no cumplirse la expresión  
      //lógica anterior  
    sentencia_1;  
    ...  
    sentencia_M;  
}
```

# Sentencia if-else

- Para más de dos flujos de ejecución

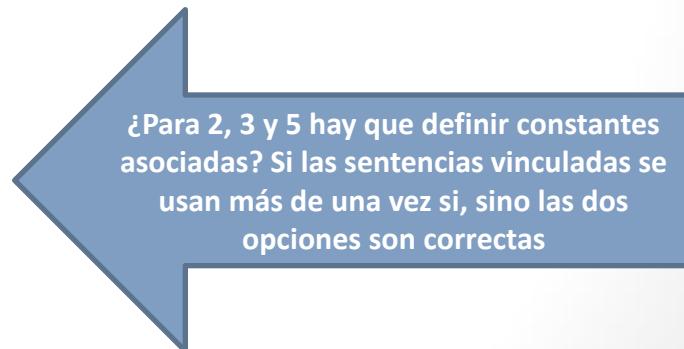
```
if (expresionLogica_1) {  
    sentencia_1;  
    ...  
    sentencia_N;  
}  
else if (expresionLogica_2) //expresión lógica diferente a la 1  
    sentencia_1;  
    ...  
    sentencia_M;  
}  
else if (expresionLogica_3) //expresión lógica diferente a la 1 y 2  
    sentencia_1;  
    ...  
    sentencia_R;  
}  
else {//flujo de ejecución en el caso de no cumplirse 1 o 2 o 3  
    sentencia_1;  
    ...  
    sentencia_P;  
}
```

# Ejemplo

```
//Paridad de un valor
public class Programa {
    public static void main (String [] args) {
        int valor = 5;
        if (valor % 2 == 0) {
            System.out.println(valor + " es par");
        }
        else {
            System.out.println(valor + " es impar");
        }
    }
}
```

# Ejemplo con múltiples casos

```
import java.io.BufferedReader;
import java.io.InputStreamReader;
public class Programa {
    public static void main (String [] args) {
        //DECLARACION DE CONSTANTES Y VARIABLES
        int numero = 0;
        //CARGA DE DATOS DE ENTRADA
        try {
            BufferedReader entrada = new BufferedReader(new InputStreamReader(System.in));
            System.out.println ("Ingrese número: ");
            numero = new Integer(entrada.readLine());
        }
        catch (Exception exc ) {
            System.out.println( exc );
            numero = 0;
        }
        //PROCESAMIENTO Y SALIDA
        if ((numero%2)==0) {
            System.out.println("Es múltiplo de 2");
        }
        else if (((numero%3)==0) && ((numero%5)==0)) {
            System.out.println("Es múltiplo de 3 y de 5");
        }
        else {
            System.out.println("No es múltiplo de 2, o de 3 y 5");
        }
    }
}
```



¿Para 2, 3 y 5 hay que definir constantes asociadas? Si las sentencias vinculadas se usan más de una vez si, sino las dos opciones son correctas

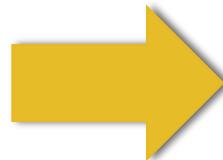
# Ejemplos para reducir sentencias

```
public class Programa {  
    public static void main (String [] args) {  
        int valor = 5;  
        if (valor % 2 == 0) {  
            System.out.println("El valor par es:");  
            System.out.println(valor);  
        }  
        else {  
            System.out.println("El valor impar es:");  
            System.out.println(valor);  
        }  
    }  
}  
  
public class Programa {  
    public static void main (String [] args) {  
        int valor = 5;  
        if (valor % 2 == 0) {  
            System.out.println(valor);  
            System.out.println("El valor par es:");  
        }  
        else {  
            System.out.println(valor);  
            System.out.println("El valor impar es:");  
        }  
    }  
}
```

# Reducción de sentencia if-else

Se repiten sentencias al principio:

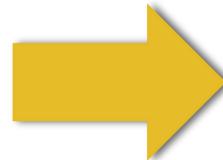
```
if (expresionLogical1){  
    A;  
    B;  
    D;  
    E;  
}  
else {  
    A;  
    B;  
    G;  
}
```



```
A;  
B;  
if (expresionLogical1){  
    D;  
    E;  
}  
else {  
    G;  
}
```

Se repiten sentencias al final:

```
if (expresionLogica2){  
    C;  
    D;  
    E;  
    A;  
    B;  
}  
else {  
    F  
    G;  
    A;  
    B;  
}
```



```
if (expresionLogica2){  
    C;  
    D;  
    E;  
}  
else {  
    F  
    G;  
}  
A;  
B;
```

# Ejemplo

```
public class Programa {  
    public static void main (String [] args) {  
        int valor = 5;  
        if (valor % 2 == 0) {  
            System.out.println("El valor par es:");  
            System.out.println(valor);  
        }  
        else {  
            System.out.println("El valor impar es:");  
            System.out.println(valor);  
        }  
    }  
}
```



```
public class Programa {  
    public static void main (String [] args) {  
        int valor = 5;  
        if (valor % 2 == 0) {  
            System.out.println("El valor par es:");  
        }  
        else {  
            System.out.println("El valor impar es:");  
        }  
        System.out.println(valor);  
    }  
}
```

# Sentencia switch

- Es una sentencia condicional de selección múltiple: dependiendo del valor de una variable o expresión entera permite ejecutar una o varias sentencias de entre muchas

```
switch (expresion) {  
    case valor_1:      sentencia_1; break;  
    case valor_2:  {  sentencia_1;  
                      ...  
                      sentencia_M;  
                      break;  
    }  
    ...  
    case valor_n:      sentencia_N; break;  
    [default: sentencia_x;] //no es obligatoria  
}
```

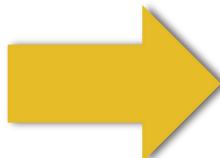
# Ejemplo

```
//Múltiples opciones
public class Programa {
    public static void main (String [] args) {
        char c = 'e';
        switch (c) {
            case 'a': System.out.println("vocal a");
                        break;
            case 'e': System.out.println("vocal e");
                        break;
            case 'i': System.out.println("vocal i");
                        break;
            default: System.out.println("Otra letra");
                        break;
        }
    }
}
```

# Sentencia switch

- Existen algunos casos donde puede haber más de un valor para ejecutar las mismas sentencias:

```
...  
case valor_1: {sentencia_1;  
    ...  
    sentencia_M;  
    break;  
}  
  
case valor_2: {sentencia_1;  
    ...  
    sentencia_M;  
    break;  
}  
...
```



```
...  
case valor_1: case valor_2:  
{ sentencia_1;  
    ...  
    sentencia_M;  
    break;  
}
```

# Ejemplo completo

- Escribir uno programa completo que solicite ingresar un número e indique:
  - si el número es 0 imprimir “es cero”,
  - si el número es 1 o 2 imprimir “es uno o dos”,
  - si es cualquier otro número imprimir si dicho número es positivo o negativo.

# Ejemplo completo

```
import java.io.BufferedReader;
import java.io.InputStreamReader;
public class Programa {
    public static void main (String [] args) {
        //DECLARACION DE CONSTANTES Y VARIABLES
        int numero = 0;
        //CARGA DE DATOS DE ENTRADA
        try {
            BufferedReader entrada = new BufferedReader(new InputStreamReader(System.in));
            System.out.println ("Ingrese número: ");
            numero = new Integer(entrada.readLine());
        }
        catch (Exception exc ) {
            System.out.println( exc );
            numero = 0;
        }
        //PROCESAMIENTO Y SALIDA
        switch (numero) {
            case 0: System.out.println("Es cero"); break;
            case 1: case 2: System.out.println("Es uno o dos"); break;
            default: if (numero > 0) {
                System.out.println("Es positivo");
            }
            else {
                System.out.println("Es negativo");
            }
            break;
        }
    }
}
```

# Práctico

- Escribir un programa completo (declaración de constantes y variable, carga de datos, procesamiento e impresión de resultados) para cada caso o ítem que:
  - solicite un número e indique si es positivo o negativo.
  - solicite un número e imprima “grande” si es mayor a 100 y “chico ” si es menor.
  - solicite un número del 1 al 7 e imprima el día de la semana.
  - solicite una letra e imprima si es vocal o consonante.
  - solicite 3 números e imprima “creciente” si los tres números están en orden creciente, “decreciente” si están en orden decreciente o “error” si no cumplen ningún orden.
- Construir un programa que solicite desde teclado un número de mes y posteriormente notifique por pantalla la cantidad de días de ese mes. En el caso de que ingrese 2 como número de mes (febrero) deberá además solicitar ingresar un número de **año (no usar ñ)**, y dependiendo de si es bisiesto o no con la siguiente sentencia

Si ((anio % 4 == 0) && ((anio % 100 != 0) || (anio % 400 == 0))) es verdadero es bisiesto  
imprimir la cantidad de días correspondiente.