

Selectivas y condicionales – Primera parte

Programación I

Objetivos del tema

- Comprender el funcionamiento de las sentencias selectivas o condicionales (if, if-else)
- Elaborar programas utilizando sentencias selectivas con múltiples condiciones

Sentencia if

- Las sentencias selectivas o condicionales permiten al programa crear un cambio en el orden de la ejecución.
- La sentencia selectiva **if (expresión lógica){...}** se utiliza para ejecutar una serie de sentencias dentro de {...} si se cumple con la **expresión lógica** o condición.

```
//Si la expresión logica es verdadera se ejecutan las sentencias dentro de  
//las {}
```

```
if (expresión logica) {  
    sentencia_1;  
    ...  
    sentencia_4;  
}
```

```
//las sentencias que siguen de if(...){...}se ejecutan independientemente del  
//resultado de la expresión lógica
```

```
Sentencia 5;
```

```
...
```

```
Sentencia N;
```

Ejemplo 1

```
/*Imprime si un valor es par. Ademas imprime el valor
*/
```

```
public class Clase_4_Ejemplo_1 {
    public static void main (String [] args) {
        int valor = 5;
        if (valor % 2 == 0) {
            System.out.println(valor + " es par");
        }
        System.out.println("El valor es " + valor);
    }
}
```

Sentencia if-else

- La sentencia selectiva **if (expresión lógica){...} else {...}** se utiliza para ejecutar dos posibles opciones de sentencias.

//Si la expresión logica es verdadera se ejecutan las sentencias dentro de las {}

```
if (expresión logica) {
```

```
    sentencia_1;
```

```
    ...
```

```
    sentencia_4;
```

```
}
```

```
else{
```

//Si no se cumple la expresión logica se ejecutan las sentencias dentro del else{...}

```
    sentencia_5;
```

```
    ...
```

```
    sentencia_8;
```

```
}
```

//las sentencias que siguen se ejecutan independientemente del resultado anterior

```
Sentencia 9;
```

```
...
```

```
Sentencia N;
```

Ejemplo 2

```
/*Imprime si un valor es par o impar. Ademas imprime el
*valor
*/
public class Clase_4_Ejemplo_2 {
    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");
        }
        System.out.println("El valor es " + valor);
    }
}
```

Sentencia if-else

- La sentencia selectiva **if (expresión lógica){...} else {...}** puede incorporar más posibles opciones de sentencias selectivas.

//Si la expresión lógica 1 es verdadera se ejecutan las sentencias dentro de las {}

```
if (expresión lógica 1) {
```

```
...
```

```
}
```

```
else if (expresión lógica 2) {
```

//Si la expresión lógica 1 fue falsa y la 2 es verdadera se ejecutan las sentencias dentro de

//las {}

```
...
```

```
}
```

```
else if (expresión lógica N) {
```

//Si la expresiones lógicas anteriores son falsas y la N es verdadera se ejecutan las

//sentencias dentro de las {}

```
...
```

```
}
```

```
else {
```

//Si la expresiones lógicas anteriores son falsas

```
...
```

```
}
```

```
Sentencia 9;
```

```
...
```

```
Sentencia N;
```

Ejemplo 3

```
/*Imprime si un valor es par o es multiplo de 3 o ninguna de las opciones
*anteriores e imprime el valor
*/
public class Clase_4_Ejemplo_3 {
    public static void main (String [] args) {
        int valor = 5;
        if (valor % 2 == 0) {
            System.out.println(valor + " es par");
        }
        else if (valor % 3 == 0) {
            System.out.println(valor + " es multiplo de 3");
        }
        else{
            System.out.println(valor + " no es impar ni multiplo de 3");
        }
        System.out.println("El valor es " + valor);
    }
}
```


Sentencia if-else

- La sentencia selectiva **if (expresión lógica){...} else if {...}...** no requerir la última opción por defecto **else{...}**.

//Si la expresión lógica 1 es verdadera se ejecutan las sentencias dentro de las {}

```
if (expresión lógica 1) {
```

```
    ...
```

```
}
```

```
else if (expresión lógica 2) {
```

//Si la expresión lógica 1 fue falsa y la 2 es verdadera se ejecutan las sentencias dentro de

//las {}

```
    ...
```

```
}
```

```
else if (expresión lógica N) {
```

//Si la expresiones lógicas anteriores son falsas y la N es verdadera se ejecutan las

//sentencias dentro de las {}

```
    ...
```

```
}
```

```
Sentencia 9;
```

```
...
```

```
Sentencia N;
```

Ejemplo 4

```
/*Imprime si un valor es par o es multiplo de 3 e imprime el valor
*/
```

```
public class Clase_4_Ejemplo_4 {
    public static void main (String [] args) {
        int valor = 5;
        if (valor % 2 == 0) {
            System.out.println(valor + " es par");
        }
        else if (valor % 3 == 0) {
            System.out.println(valor + " es multiplo de 3");
        }
        System.out.println("El valor es " + valor);
    }
}
```

Ejemplo 5

```
/*Dado un numero entero ingresado por el usuario hacer: _ imprimir A si el numero es multiplo de 5, _ o imprimir B si el numero es multiplo de 7,
*_ o imprimir C si el numero es multiplo de 2 y de 3, _ o imprimir el numero
*/

import java.io.BufferedReader;
import java.io.InputStreamReader;

public class Clase_4_Ejemplo_5 {

    public static void main(String[] args) {

        int numero;

        try{

            BufferedReader entrada = new BufferedReader(new InputStreamReader(System.in));

            System.out.println("Ingrese un numero entero :");

            numero = Integer.valueOf(entrada.readLine());

            if (numero % 5 == 0) {

                System.out.println("A");

            }

            else if (numero % 7 == 0) {

                System.out.println("B");

            }

            else if ((numero % 2 == 0)&& (numero % 3 == 0)) {

                System.out.println("C");

            }

            else {

                System.out.println("El numero es: "+numero);

            }

        }

        catch (Exception exc){

            System.out.println(exc);

        }

    }

}
```

Ejemplo 6

```
/*Dado un caracter ingresado por el usuario hacer: _ imprimir si es caracter minuscula, _ o imprimir si es caracter mayuscula,
*_ o imprimir no es caracter letra
*/

import java.io.BufferedReader;
import java.io.InputStreamReader;

public class Clase_4_Ejemplo_6 {

    public static void main(String[] args) {

        char character;

        try{

            BufferedReader entrada = new BufferedReader(new InputStreamReader(System.in));

            System.out.println("Ingrese un caracter :");

            character = entrada.readLine().charAt(0);

            if (('a'<=character)&&(character<='z')) {

                System.out.println("Es caracter minuscula");

            }

            else if (('A'<=character)&&(character<='Z')) {

                System.out.println("Es caracter mayuscula");

            }

            else {

                System.out.println("No es caracter letra");

            }

        }

        catch (Exception exc){

            System.out.println(exc);

        }

    }

}
```

Video de explicación

https://drive.google.com/file/d/1ap4ve0emNDGqTPLHkz_KmSrV_Fi6webz/view?usp=sharing

Práctico primera parte

1. Escribir un programa que permita el ingreso de un carácter y realice:
 - _ imprima es carácter dígito si el carácter ingresado es carácter dígito,
 - _ o imprima no es carácter dígito
2. Escribir un programa que permita el ingreso de dos números enteros por teclado y realice:
 - _ imprima A si el primero mayor al segundo,
 - _ o imprima B si ambos son múltiplos de 2,
 - _ o imprima C para ninguna de las opciones anteriores
3. Escribir un programa que ingrese un número entero por teclado y realice:
 - _ imprima es múltiplo de 6 y de 7 si cumple con esa condición,
 - _ o imprima es mayor a 30 y múltiplo de 2 si cumple con esa condición,
 - _ imprima si el cociente de la división de dicho número con 5 es mayor a 10 sin importar las condiciones previas.

Selectivas y condicionales – Segunda parte

Programación I

Objetivos del tema

- Describir el funcionamiento de las sentencias selectivas condicionales (if, if-else y switch) anidadas
- Interpretar el resultado de una secuencia de sentencias condicionales combinadas

Reducción de sentencia if-else

- En la programación de la sentencia múltiple **if (expresión lógica){...}else if(){...}...** suele ocurrir la inclusión de sentencias que aparecen repetidas en los posibles flujos de ejecución.

```
//primer caso
if (expresión lógica){
    A;
    B;
    D;
    E;
}
else {
    A;
    B;
    G;
}
```

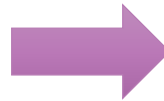
```
//segundo caso
if (expresión lógica){
    C;
    E;
    A;
    B;
}
else {
    F
    A;
    B;
}
```

- En el primer caso A y B siempre se ejecutan primero sin importar la opción que se ejecute. En el segundo caso A y B siempre se ejecutan al final sin importar la opción que se ejecute.

Reducción de sentencia if-else

- En esos casos es posible reducir el código y eliminar la repetición

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



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

```
if (expresión lógica){  
    C;  
    E;  
    A;  
    B;  
}  
else {  
    F  
    A;  
    B;  
}
```



```
if (expresión lógica){  
    C;  
    E;  
}  
else {  
    F  
}  
A;  
B;
```

Ejemplo 7

*/*Repetición en la sentencia if...else... al final
/

```
public class Clase_4_Ejemplo_7 {  
    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);  
        }  
    }  
}
```

*/*Reducción en la sentencia if...else... al final
/

```
public class Clase_4_Ejemplo_7 {  
    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);  
    }  
}
```

Sentencias selectivas anidadas

- Dentro de los bloques {...} de las sentencias selectivas hay sentencias que pueden ser o no otras sentencias selectivas.

```
if (expresión lógica 1) {  
    sentencia_1;  
    if (expresión lógica 2) {  
        sentencia_2;  
        ...  
    }  
    sentencia_4;  
}  
else{  
    sentencia_5;  
    ...  
    sentencia_8;  
}  
Sentencia 9;  
...  
Sentencia N;
```

Ejemplo 8

```
/*Dado un valor, verificar e imprimir si es par o impar. En el caso de que sea
*par verificar e imprimir si es mayor de 4 o no lo es
*/
public class Clase_4_Ejemplo_8 {
    public static void main (String [] args) {
        int valor = 5;
        if (valor % 2 == 0) {
            System.out.println(valor + " es par");
            if (valor > 4) {
                System.out.println(valor + " es mayor que 4");
            }
            else {
                System.out.println(valor + " es menor o igual que 4");
            }
        }
        else{
            System.out.println(valor + " es impar");
        }
    }
}
```

Sentencia switch

- Es una sentencia condicional de selección múltiple: dependiendo del valor de una variable (char o entero) permite ejecutar una o varias sentencias.

```
switch (valor) { //dado valor
    case valor_1: { //si valor es igual a valor_1 ejecuta el bloque entre {...}
        sentencia_1;
        break;
    }
    case valor_2: { //si valor es igual a valor_2 ejecuta el bloque entre {...}
        sentencia_1;
        ...
        sentencia_M;
        break;
    }
    ...
    default: { //si valor no es una opción anterior ejecuta el bloque entre {...}
        sentencia_x;
    }
}
```

Ejemplo 9

```
/*Dado una caracter imprimir vocal a si es una a, vocal e si es una e, vocal i si es una i  
*o indicar que no es ninguna de las opciones anteriores  
*/
```

```
public class Clase_4_Ejemplo_9 {  
    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("Es una letra distinta de a, e, i");  
            }  
        }  
    }  
}
```

Sentencia switch

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

```
switch (valor) {  
    case valor_1: case valor_2: {  
        //si valor es igual a valor_1 o valor_2 ejecuta el bloque entre {...}  
        sentencia_1;  
        break;  
    }  
    case valor_3: { //si valor es igual a valor_3 ejecuta el bloque entre {...}  
        ...  
        sentencia_M;  
        break;  
    }  
    ...  
    default: { //si valor no es una opción anterior ejecuta el bloque entre {...}  
        sentencia_x;  
    }  
}
```

Ejemplo 10

/*Dado un numero entero ingresado por teclado realice: _si el número es 0 imprimir es cero, _si el numero es 1 o *2 imprimir es uno o dos, si es cualquier otro numero imprimir si dicho numero es positivo o negativo.
*/

```
import java.io.BufferedReader;
import java.io.InputStreamReader;
public class Clase_4_Ejemplo_10{
    public static void main (String [] args) {
        int numero;
        try {
            BufferedReader entrada = new BufferedReader(new InputStreamReader(System.in));
            System.out.println ("Ingrese numero entero: ");
            numero = Integer.valueOf(entrada.readLine());
            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");
                    }
                }
            }
        }
        catch (Exception exc ) {
            System.out.println( exc );
        }
    }
}
```

Video de explicación

<https://drive.google.com/file/d/1m5wTB1HsrH1x8Q-chDOHMg70cBIBuQa5/view?usp=sharing>

Práctico

1. Hacer un programa que solicite la carga desde consola de un numero entero y realice:
 - _si el numero es positivo, imprima "grande" si es mayor a 100 o "chico " si es menor. Además deberá imprimir que el valor es positivo.
 - _si no lo es, imprima si el numero es par, o si el numero es múltiplo de 3, o ninguna de las opciones anteriores.
2. Hacer un programa que solicite la carga desde consola de un carácter y realice: imprima si es dígito, o letra minúscula, o es cualquier otro carácter. Si es letra minúscula indicar si es vocal o consonante.
3. 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), para imprimir la cantidad de días deberá solicitar ingresar un número de año (no usar ñ), para determinar si es bisiesto o no con la siguiente sentencia:
 - _si $((\text{año} \% 4 == 0) \ \&\& \ ((\text{año} \% 100 != 0) \ || \ (\text{año} \% 400 == 0)))$ es verdadero, entonces es bisiesto.

Práctico

4. El astrónomo Metón de Atenas descubrió hacia el siglo V a.C que la luna repite su ciclo cada 19 años (el mismo día del mismo mes tendrá la misma fase lunar). Sus aportes permitieron determinar la fase de la luna para una fecha dada. Los pasos para obtener esa información son:
- a. Calcular el número áureo: permite conocer en qué año-tipo del ciclo nos encontramos considerando la repetición de ciclos cada 19 años. El número áureo se calcula de la siguiente forma:
 - Tomamos la cifra del año que nos interesa (por ejemplo 2023) y le sumamos 1 ($2023+1=2024$)
 - A continuación dividimos el resultado por 19 (ciclo de repetición) y nos quedamos con el resto $2024/19 = 106$ resto = 10 (10 es el resto entre 2024 y 19). Al dividir por 19 hemos eliminado los ciclos repetidos. Así el número áureo es 10.
 - b. Calcular la epacta: es la edad de la luna en días el 1 de enero del año elegido, y se calcula de la siguiente forma:
 - Se resta 1 del número áureo, y se multiplica por 11 (en nuestro caso $10-1 = 9$; $9*11 = 99$)
 - Después, se divide por 30 (un mes lunar) y de nuevo nos quedamos con el resto: $99/30 = 3$ resto = 9. Significa que el 1 de enero de 2023 habrán pasado 9 días desde la última luna nueva.

Práctico

- c. Finalmente, para saber la edad lunar en un día cualquiera, (supongamos el 15 de junio de 2023) realizaremos la siguiente operación:
 - 1) A la epacta (9) le sumamos 1 por cada mes a partir de marzo (por ejemplo: de marzo a junio sumamos 4, y si el mes fuera enero de marzo a enero sumamos 11), y luego le sumamos el día del mes requerido (15), es decir $9+4+15=28$.
 - 2) 28 es la edad lunar para esa fecha, y si pasara de 29 habría que reducirlo obteniendo el resto de dividir por 30.
 - 3) Es decir que el 15 de junio de 2023 habrán pasado 28 días desde la última luna nueva, por lo que la luna será menguante casi nueva (a falta de un día aproximadamente).

Se pide: hacer un programa completo en el cual se solicite de teclado el ingreso de un día, un número de mes, y un año; luego calcule la cantidad de días desde la última luna nueva (edad lunar), e informe por pantalla en cual de las 4 fases se corresponde para esa fecha.