

Práctica 3. Estructuras de control condicionales

Objetivos:

- Revisar el concepto de algoritmo y entender la necesidad del diseño de algoritmos en el estudio y resolución de programas.
- Comprender y diferenciar perfectamente los distintos tipos de instrucciones utilizadas en el diseño de algoritmos.
- Conocer y manejar con habilidad las sentencias de control condicionales.

1. Buenas prácticas de programación

No olvides que antes de pasar a la fase de diseño deberemos analizar exhaustivamente el problema.

Recuerda también que para una buena programación es conveniente utilizar el sangrado del texto. Esta práctica facilita la comprensión del programa realizado. De esta manera, incluiremos una sangría en cualquier sentencia que esté dentro del cuerpo de otra sentencia.

Veamos un ejemplo. ¿Cuál de los códigos que se muestran a continuación es más claro?

```
#include<iostream>
using namespace std;

//Define el número de exámenes parciales
const int NUM_PARCIALES = 5;

main(){
    float nota_parcial;
    float suma;
    int i;
    bool nota_incorrecta;
    float nota_final;

    suma = 0;

    //Introducir la nota de todos los parciales y sumarlas
    //solo cuando el dato introducido es correcto
    for(i=1;i<=NUM_PARCIALES;i++)
```

```

        {
            do{
                cout<<"Dime tu nota del parcial "<<i<<":
";
                cin>>nota_parcial;
                nota_incorrecta = (nota_parcial <0.0 ||
nota_parcial>10.0);

                if(nota_incorrecta)
                    cout<<"La nota introducida es
incorrecta"<<endl;
            }while (nota_incorrecta);
            suma = suma + nota_parcial;
        }

        //Calcula la nota media
        nota_final = suma / NUM_PARCIALES;

        cout<<"Tu nota es: "<<nota_final<<endl;
    }

```

```

#include<iostream>
using namespace std;
const int NUM_PARCIALES = 5;
main(){
    float nota_parcial;
    float suma;
    int i;
    bool nota_incorrecta;
    float nota_final;
    suma = 0;
    for(i=1;i<=NUM_PARCIALES;i++)
    {
        do{
            cout<<"Dime tu nota del parcial "<<i<<": ";
            cin>>nota_parcial;
            nota_incorrecta =(nota_parcial <0.0 || nota_parcial>10.0);
            if(nota_incorrecta)
                cout<<"La nota introducida es incorrecta"<<endl;
            }while (nota_incorrecta);
            suma = suma + nota_parcial;
        }
        nota_final = suma / NUM_PARCIALES;
        cout<<"Tu nota es: "<<nota_final<<endl;
    }
}

```

El primero de los códigos al utilizar comentarios en el programa, espacios para separar las diferentes secuencias de sentencias y la sangría para indentar aquellas sentencias que están dentro del cuerpo de otra sentencia es más claro que el segundo código.

2. Sentencias de selección

Las sentencias de selección nos permiten decidir qué secuencia de sentencias se va a ejecutar a continuación. Dentro de las sentencias de control de selección podemos distinguir tres tipos:

- **Simples:** nos permiten decidir si la secuencia de sentencias se va a ejecutar. Formadas por un único **if**.

```
if(expresión_lógica){  
    secuencia de sentencias;  
}
```

- **Dobles:** permiten seleccionar entre dos secuencias de sentencias. Formadas por un **if-else**.

```
if(expresión_lógica){  
    secuencia de sentencias 1;  
}else{  
    secuencia de sentencias 2;  
}
```

- **Múltiples:** permiten seleccionar entre múltiples secuencias de sentencias. Formadas por un **switch** o **if-else anidados**.

```
if(expresión_lógica 1){  
    secuencia de sentencias 1;  
}else if(expresión_lógica 2){  
    secuencia de sentencias 2;  
}else if(expresión_lógica 3){  
    Secuencia de sentencias 3;  
}
```

```
switch (expresión ) {  
    case valor_1 : secuencia de sentencias 1;  
                    break;  
    case valor_2 : secuencia de sentencias 2;  
                    break;  
    case valor_3 : secuencia de sentencias 3;  
                    break;  
    default      : secuencia de sentencias 4;  
}
```

3. Ejercicio resuelto 1. Enunciado

Implementa un algoritmo en C que tenga como entrada el precio de venta de un piso, y que produzca como salida la comisión del vendedor. Debe tener en cuenta que si el precio de venta es inferior a 100.000€, la comisión será del 4%, si el precio

está entre 100.000€ y 300.000€ la comisión será del 3% y si el precio es superior a 300.000€ la comisión será del 2'5%.

3.1 Solución propuesta 1

En esta propuesta de solución vemos que hay tres condiciones (la segunda de ellas compuesta) que se ejecutan secuencialmente. Independientemente de que este algoritmo produzca el resultado deseado, es mejorable en cuanto a su eficiencia, ya que tal y como está diseñado, las tres condiciones serán evaluadas siempre que ejecutemos el programa, ralentizando por tanto su ejecución.

```
#include<iostream>
using namespace std;
main(){
    long int precio;
    float comision;

    cout<<"Introduce el precio del piso: ";
    cin>>precio;

    if (precio < 100000)
    {
        comision = precio * 0.04;
    }

    if (precio>=100000 && precio < 300000)
    {
        comision = precio * 0.03;
    }

    if (precio > 300000)
    {
        comision = precio * 0.025;
    }

    cout << "La comisión es de "<< comision << "€"<<endl;
}
```

3.2 Solución propuesta 2

En esta solución las dos condiciones son simples. Además, al utilizar la estructura if-else se consigue una mayor eficiencia, ya que si el precio que introducen es menor que 100.000€ sólo se evaluará la primera condición. En cualquier caso, como máximo se evaluarán únicamente dos condiciones.

```
#include<iostream>
using namespace std;

main(){
    long int precio;
```

```

float comision;

cout << "Introduce el precio del piso: ";
cin >> precio;

if (precio <100000)
{
    comision = precio * 0.04;
}else{
    if (precio <300000)
    {
        comision = precio * 0.03;
    }else{
        comision = precio * 0.025;
    }
}

cout << "La comisión es de "<< comision <<"€";
}

```

3.3 Solución propuesta 3

Esta solución es exactamente la misma que la anterior pero se propone para mostrar que los delimitadores de bloques "{" y "}" solo son necesarios cuando dentro del cuerpo del bloque hay más de una instrucción.

```

#include<iostream>
using namespace std;

main(){
    long int precio;
    float comision;

    cout << "Introduce el precio del piso: ";
    cin >> precio;

    if (precio <100000)
        comision = precio * 0.04;
    else
        if (precio <300000)
            comision = precio * 0.03;
        else
            comision = precio * 0.025;

    cout << "La comisión es de "<< comision <<"€";
}

```

3.4 Modificación

Supongamos ahora que nos dicen que la mayoría de los precios de las viviendas son superiores a 300.000€. Evidentemente, esto no afecta a los cálculos que debe hacer el programa y, por tanto, seguirían siendo válidas las soluciones anteriores. Pero, ¿podríamos hacer algo por mejorar la eficiencia?

```
#include<iostream>
using namespace std;

main(){
    long int precio;
    float comision;

    cout << "Introduce el precio del piso: ";
    cin >> precio;

    if (precio >= 300000)
        comision = precio * 0.025;
    else
        if (precio >= 100000)
            comision = precio * 0.03;
        else
            comision = precio * 0.04;

    cout << "La comisión es de " << comision << "€";
}
```

De esta forma, en la mayoría de los casos (cuando el precio sea superior a 300.000€) sólo se evaluará la primera condición. Observad como se han implementado las condiciones contrarias a las de la solución anterior.

4. Ejercicio resuelto 2

Implementa un algoritmo en C que, dada una cantidad en céntimos de euro, escriba la cantidad mínima de monedas que correspondería. Teniendo en cuenta los siguientes tipos de monedas: 1 céntimo, 2 céntimos, 5 céntimos, 10 céntimos, 20 céntimos, 50 céntimos, 1 euro y 2 euros.

4.1 Solución propuesta

```
#include <iostream>
using namespace std;

main(){
    int cantidad = 0,
        mlcent = 0,
        m2cent = 0,
```

```

        m5cent = 0,
        m10cent = 0,
        m20cent = 0,
        m50cent = 0,
        m100cent = 0,
        m200cent = 0;

cout << "Cantidad de céntimos: ";
cin >> cantidad;

if (cantidad >= 200){
    m200cent = cantidad / 200;
    cantidad = cantidad % 200;
}

if (cantidad >= 100){
    m100cent = cantidad / 100;
    cantidad = cantidad % 100;
}

if (cantidad >= 50){
    m50cent = cantidad / 50;
    cantidad = cantidad % 50;
}

if (cantidad >= 20){
    m20cent = cantidad / 20;
    cantidad = cantidad % 20;
}

if (cantidad >= 10){
    m10cent = cantidad / 10;
    cantidad = cantidad % 10;
}

if (cantidad >= 5){
    m5cent = cantidad / 5;
    cantidad = cantidad % 5;
}

if (cantidad >= 2){
    m2cent = cantidad / 2;
    cantidad = cantidad % 2;
}

mlcent = cantidad;

cout << m200cent << " monedas de 2 euros."<<endl;
cout << m100cent << " monedas de 1 euro."<<endl;
cout<<m50cent<<" monedas de 50 céntimos."<<endl;
cout<<m20cent<<" monedas de 20 céntimos."<<endl;
cout<<m10cent<<" monedas de 10 céntimos."<<endl;

```

```

        cout<<m5cent<<" monedas de 5 céntimos."<<endl;
        cout<<m2cent<<" monedas de 2 céntimos."<<endl;
        cout<<m1cent<<" monedas de 1 céntimo."<<endl;

    }

```

Observad como se ha utilizado una secuencia de sentencias condicionales simples, ya que queremos que se evalúen todas las condiciones secuencialmente. Si hubiésemos optado por una estructura if-else anidada (como la utilizada en el ejercicio anterior), el algoritmo no realizaría el proceso correctamente. Esto lo podemos comprobar realizando la traza del ejercicio.

5. Ejercicio resuelto 3

Implementa un algoritmo en C que tenga como entrada dos números enteros y produzca como salida un mensaje indicando cuántos son pares.

La tercera solución propuesta es la más clara de todas, mientras que la primera es la peor.

5.1 Solución propuesta 1

```

#include<iostream>
using namespace std;

main(){
    int a,b;

    cout << "Introduzca el primer número: ";
    cin >> a;

    cout<<"Introduzca el segundo número: ";
    cin>>b;

    if ( (a%2)==0 && (b%2) == 0)
        cout<<"La cantidad de pares es 2";
    else
        if ((a%2) == 0 || (b%2) == 0)
            cout<<"La cantidad de pares es 1";
        else
            cout<<"La cantidad de pares es 0";

}

```

5.2 Solución propuesta 2

```

#include<iostream>
using namespace std;

```



```

main(){
    int a,b;

    cout << "Introduzca el primer número: ";
    cin >> a;

    cout<<"Introduzca el segundo número: ";
    cin>>b;

    if ((a%2) == 0)
        if ((b%2) == 0)
            cout<<" La cantidad de pares es 2.";
        else
            cout<<"La cantidad de pares es 1.";
    else
        if ((b%2) == 0)
            cout<<" La cantidad de pares es 1.";
        else
            cout<<"La cantidad de pares es 0.";

}

```

5.3 Solución propuesta 3

```

#include<iostream>
using namespace std;

main(){
    int a,b;
    int cont = 0;

    cout << "Introduzca el primer número: ";
    cin >> a;

    cout<<"Introduzca el segundo número: ";
    cin>>b;

    if ((a%2) == 0)
        cont++;

    if ((b%2) == 0)
        cont ++;

    cout<<"La cantidad de pares es "<<cont;

}

```

6. Ejercicio propuesto 1

Implementa un algoritmo en C que lea tres números enteros y escriba un mensaje indicando cuál es el mayor de los 3, cuál es el menor de los 3 y cuál es el promedio, intentando que el número de condiciones a evaluar sea el mínimo.

7. Ejercicio propuesto 2

Implementa un algoritmo que lea una fecha (por separado el día, el mes y el año) y muestre un mensaje indicando si la fecha es correcta o no. Hay que tener en cuenta que son bisiestos los años múltiplos de 4, excepto los múltiplos de 100, y que sí son bisiestos los múltiplos de 400.

En el caso de que la fecha sea correcta, deberá mostrar el día siguiente.

8. Ejercicio propuesto 3

Implementa un algoritmo en C que pida un número entre 0 y 9.999 y diga si es capicúa.

9. Ejercicio propuesto 4

Implementa un algoritmo en C que realice las cuatro operaciones básicas de una calculadora (suma, resta, multiplicación y división). El programa debe leer los dos números y la operación. La operación se indicará con un carácter: 's' para la suma, 'r' para la resta, 'm' para la multiplicación y 'd' para la división.

10. Errores de programación comunes

- **No utilizar secuencias de sentencias delimitadas entre llaves.** Prescindir de las llaves cuando se programan secuencias de control puede dar lugar a errores difíciles de detectar y depurar.
- **Incluir un ';' junto a la sentencia de control.** No es correcto escribir un ';' después de la condición de un if. Por ejemplo:

```
if(0 == contador);  
    contador = var + 2;
```

Independientemente del resultado de la evaluación de la condición del if, siempre se ejecuta la sentencia de asignación. ¿Por qué? Pues porque el ';' de detrás indica una sentencia nula en el cuerpo del if, es decir, no hacer nada en dicho cuerpo. La sentencia de asignación estaría fuera del cuerpo del if.

- **Confusiones entre asignación y comparación de igualdad.** Este error es muy frecuente y difícil de detectar. Si se escribe:

```
        if (contador = 0)
```

se está asignando 0 al contador, no se está haciendo la comparación. Esto se evita poniendo la constante primero, es decir:

```
        if (0 = contador)
```

lo que provocaría un error de compilación y detectaría el problema.

ANEXO. Ejemplos de código en Java

Ejercicio resuelto 1. Propuesta 1

```
import java.util.Scanner;

class ejercicio1{

    public static void main(String args[]){
        int precio;
        double comision;
        Scanner sc = new Scanner(System.in);

        comision = 0;

        System.out.println("Introduce el precio del piso: ");
        precio = sc.nextInt();

        if(precio < 100000)
        {
            comision = precio * 0.04;
        }

        if(precio >= 100000 && precio < 300000)
        {
            comision = precio * 0.03;
        }

        if(precio > 300000)
        {
            comision = precio * 0.025;
        }

        System.out.println("La comisión es de "+comision);

    }
}
```

Ejercicio resuelto 2.

```
import java.util.Scanner;

class ejercicio2{
```

```
public static void main(String args[]){
    int cantidad = 0;
    int m1cent = 0;
    int m2cent = 0;
    int m5cent = 0;
    int m10cent = 0;
    int m20cent = 0;
    int m50cent = 0;
    int m100cent = 0;
    int m200cent = 0;

    Scanner sc = new Scanner(System.in);

    System.out.println("Cantidad de céntimos: ");
    cantidad = sc.nextInt();

    if(cantidad >=200)
    {
        m200cent = cantidad / 200;
        cantidad = cantidad % 200;
    }

    if(cantidad >=100)
    {
        m100cent = cantidad / 100;
        cantidad = cantidad % 100;
    }

    if(cantidad >=50)
    {
        m50cent = cantidad / 50;
        cantidad = cantidad % 50;
    }

    if(cantidad >=20)
    {
        m20cent = cantidad / 20;
        cantidad = cantidad % 20;
    }

    if(cantidad >=10)
    {
        m10cent = cantidad / 10;
        cantidad = cantidad % 10;
    }

    if(cantidad >=5)
    {
        m5cent = cantidad / 5;
        cantidad = cantidad % 5;
    }
}
```

```

    if(cantidad >=2)
    {
        m2cent = cantidad / 2;
        cantidad = cantidad % 2;
    }

    m1cent = cantidad;

    System.out.println(m200cent+" monedas de 2 euros");
    System.out.println(m100cent+" monedas de 1 euro");
    System.out.println(m50cent+" monedas de 50 céntimos");
    System.out.println(m20cent+" monedas de 20 céntimos");
    System.out.println(m10cent+" monedas de 10 céntimos");
    System.out.println(m5cent+" monedas de 5 céntimos");
    System.out.println(m2cent+" monedas de 2 céntimos");
    System.out.println(m1cent+" monedas de 1 céntimo");

}
}

```

Ejercicio resuelto 3. Propuesta 1

```

import java.util.Scanner;

class ejercicio3{

    public static void main(String args[]){
        int a,b;
        Scanner sc = new Scanner(System.in);

        System.out.println("Introduce el primer número: ");
        a = sc.nextInt();

        System.out.println("Introduce el segundo número: ");
        b = sc.nextInt();

        if(a%2==0 && b%2==0)
            System.out.println("La cantidad de pares es 2");
        else
            if(a%2==0 || b%2==0)
                System.out.println("La cantidad de pares es 1");
            else
                System.out.println("La cantidad de pares es 0");
    }
}

```