

PRÁCTICA 5. PROGRAMACIÓN MODULAR

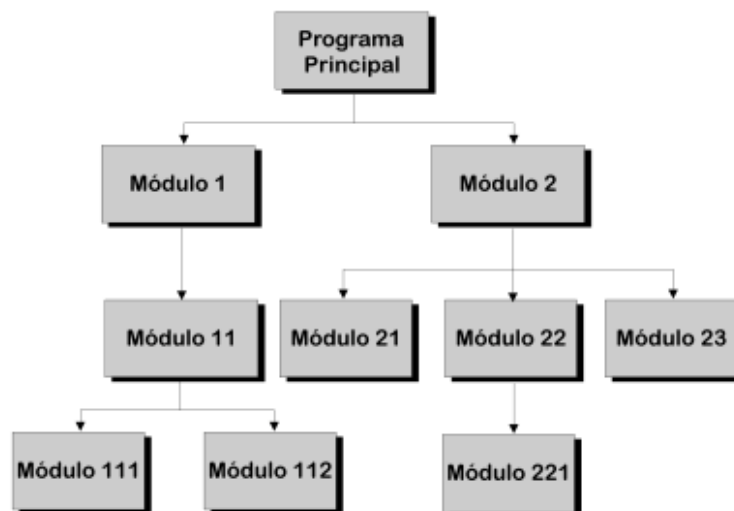
Procedimientos y Funciones.

Paso de Parámetros

OBJETIVOS:

- ✓ Comprender el concepto de módulo
- ✓ Distinguir entre procedimientos y funciones
- ✓ Diferenciar entre paso de parámetros por valor y por referencia
- ✓ Comprender el concepto de retorno de valores en funciones

El diseño descendente resuelve un problema efectuando descomposiciones en otros problemas más sencillos a través de distintos niveles de refinamiento. La programación modular consiste en resolver de forma independiente los subproblemas resultantes de una descomposición. El resultado de dividir reiteradas veces un problema en problemas más pequeños es una estructura jerárquica o en árbol.



Ejercicio Resuelto 1. Realiza un programa en C que permita convertir grados Celsius a Fahrenheit y viceversa. El programa debe mostrar un menú para poder seleccionar qué opción se desea (más la opción terminar). Y preguntar la temperatura que desea convertir.

Cada una de las conversiones se debe realizar por medio de un módulo.

La conversión de grados Celsius a grados Fahrenheit se obtiene multiplicando la temperatura en Celsius por 1,8 y sumando 32.

La conversión de grados Fahrenheit a grados Celsius se obtiene restándole 32 a la temperatura en grados Fahrenheit y dividiéndolo por 1,8.

Ejemplo de ejecución:

Seleccione qué tipo de conversión desea realizar:

1. Convertir grados Celsius a Fahrenheit.
2. Convertir grados Fahrenheit a Celsius.
3. Terminar el programa

Opción:1

Introduzca la temperatura en grados Celsius: 30
La temperatura en grados Fahrenheit es: 86

Solución:

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

//Declaración de módulos
void mostrar_menu();
float Celsius_to_Fahrenheit(float);
float Fahrenheit_to_Celsius(float);

//Programa principal
main(){

    //Declaración de variables
    int opcion;
    float celsius, fahr;

    do{
        mostrar_menu();
        cout<<"Opción: ";
        cin>>opcion;
        cout<<endl;

        switch(opcion){
            case 1: cout<<"Introduzca la temperatura en grados Celsius: ";
                    cin>>celsius;
                    fahr = Celsius_to_Fahrenheit(celsius);
                    cout<<"La temperatura en grados Fahrenheit es: "<<fahr;
                    break;
            case 2: cout<<"Introduzca la temperatura en grados Fahrenheit: ";
                    cin>>fahr;
                    celsius = Fahrenheit_to_Celsius(fahr);
                    cout<<"La temperatura en grados Celsius es: "<<celsius;
                    break;
            case 3: cout<<"Ha seleccionado la opción Salir."<<endl;
                    break;
            default: cout<<"Debe introducir una opción de 1 a 3"<<endl;
        }
    }
```

```

        cout<<endl;
        cin.get();
        cout<<"Pulse Intro para continuar"<<endl;
        cin.get();

    }while(opcion!=3);
}

//Módulo que muestra el menú
void mostrar_menu()
{
    cout<<"Seleccione qué tipo de conversión desea realizar:"<<endl;
    cout<<"1. Convertir grados Celsius a Fahrenheit."<<endl;
    cout<<"2. Convertir grados Fahrenheit a Celsius."<<endl;
    cout<<"3. Terminar el programa."<<endl;
}

//Módulo que convierte de Celsius a Fahrenheit
float Celsius_to_Fahrenheit(float cel)
{
    float fah;

    fah = cel * 1.8 + 32;

    return(fah);
}

//Módulo que convierte de Fahrenheit a Celsius
float Fahrenheit_to_Celsius(float fah)
{
    float cel;

    cel = (fah-32)/1.8;

    return (cel);
}

```

Ejercicio Resuelto 2. Realiza un programa que muestre un menú en el que a partir de un número entero n se elegirá una de las siguientes operaciones:

- Opción 1: calcular de cuántas cifras se compone. (No contar los ceros a la izquierda)
- Opción 2: mostrar la cifra i-ésima de dicho número. La posición i debe pedirse al usuario.
- Opción 3: Salir

Solución:

```

#include <iostream>
using namespace std;

const int OPCION_SALIR = 3;

int CantidadCifras(int); //Declaración de la función
int Cifra_i(int, int);
void menu();
int Leer_Opcion();

int main(void)
{
    int opcion, n, resultado, i;

    do {
        menu();
        opcion = Leer_Opcion();
        if (opcion != OPCION_SALIR) {

```

```

        cout << "Introduzca un número: ";
        cin >> n;
    }
    switch(opcion) {
        case 1:
            resultado=CantidadCifras(n);
            cout <<"El número de cifras de "<<n<<" es "<<resultado<<endl;
            break;
        case 2:
            cout <<"Introduce la i: ";
            cin >>i;
            resultado=Cifra_i(n,i);
            cout << "La cifra "<<i<<" de "<<n<<" es "<<resultado<<endl;
            break;
        case OPCION_SALIR:
            cout << "FIN DEL PROGRAMA"<<endl;
    }
    } while(opcion != OPCION_SALIR);
}

int CantidadCifras(int m)
{
    int contador=1;

    while(m >=10) {
        m=m/10;
        contador ++;
    }
    return(contador);
}

int Cifra_i(int m,int i)
{
    int contador=1,cifra=0,devolver;

    cifra=m%10;
    while(m >=10 && contador<i) {
        m=m/10;
        cifra=m%10;
        contador ++;
    }
    if (contador==i)
        devolver=cifra;
    else
        devolver=-1;
    return(devolver);
}

void menu()
{
    cout << "1. Calcular el número de cifras de un número"<<endl;
    cout << "2. Mostrar la cifra i-ésima de un número"<<endl;
    cout << "3. Mostrar el reverso de un número"<<endl;
    cout << "4. SALIR"<<endl;
}

int Leer_Opcion()
{
    int opcion;
    do {
        cout << "Introduzca la opción: ";
        cin >> opcion;
    } while (opcion < 1 || opcion > 3);
    return(opcion);
}

```

Ejercicio Resuelto 3. **Dados los siguientes algoritmos:**

Algoritmo A

Algoritmo B

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

void numMayor(int , int );

int mayor; //Declaración de variable global

int main(void) {
    int num1, num2;

    mayor = 0;
    cout << "Introduce dos números: ";
    cin >> num1 >> num2;

    numMayor(num1,num2);
    cout << "El mayor es "<< mayor<< endl;
}

void numMayor(int n1, int n2)
{
    if (n1 > n2)
        mayor = n1;
    else
        mayor = n2;
}
```

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

void numMayor(int , int );

int resultado; //Declaración de variable global

int main(void) {
    int num1, num2;

    resultado = 0;
    cout << "Introduce dos números: ";
    cin >> num1 >> num2;

    resultado = num1 + num2 ;
    numMayor(num1,num2);
    cout << "Al sumar num1 y num2 da "<< resultado<< endl;
}

void numMayor(int n1, int n2)
{
    if (n1 > n2)
        resultado = n1;
    else
        resultado = n2;
}
```

a. ¿Qué ocurre? ¿Funcionan correctamente? Realiza una traza de los algoritmos anteriores para comprobar su funcionamiento.

La traza del algoritmo a:

Programa Principal			procedimiento	
num1	num2	mayor	n1	n2
-	-	0		
4	6		4	6
		6		

supuesta la entrada:

llevaría a la ejecución:

Introduce un número : 4
 Introduce un número : 6
 El mayor es 6

En este programa desde el procedimiento numMayor se está modificando una variable global (mayor) sin que se hubiese pasado como parámetro a dicho procedimiento, aunque se produzca un resultado correcto podría haberse producido un efecto lateral. Esto es incorrecto.

La traza del algoritmo b:

Programa Principal			procedimiento	
num1	num2	resultado	n1	n2
-	-	0		
4	6			
		10		
			4	6
		6		

supuesta la entrada:

llevaría a la ejecución:

Introduce un número : 4
 Introduce un número : 6

El resultado de la suma de estos números es 6

En este programa desde el procedimiento numMayor se está modificando una variable global sin que se hubiese pasado como parámetro a dicho procedimiento, como resultado se ha modificado la variable resultado que contenía un valor anteriormente (el de la suma de dos números) y por tanto se visualiza por pantalla un valor incorrecto. A esto se le denomina **efecto lateral**. Esto es incorrecto.

b. ¿Crees que deberíamos modificar los algoritmos anteriores? ¿Por qué? ¿Cómo?

Si, en los dos aunque sólo uno produzca un resultado incorrecto. Deberíamos pasar como **parámetro por referencia** una variable que almacenase el número mayor tras la ejecución del subalgoritmo para evitar efectos laterales.

Algoritmo A (función devuelve valor)

```
#include <iostream>

using namespace std;

int numMayor(int, int);

int main(void) {
    int num1, num2, mayor;

    mayor = 0;
    cout << "Introduce dos números: ";
    cin >> num1 >> num2;

    mayor = numMayor(num1, num2);
    cout << "El mayor es "<< mayor << endl;
}

// La función devuelve un número entero
int numMayor(int n1, int n2)
{
    int max;

    if (n1 > n2)
        max = n1;
    else
        max = n2;

    return(max);
}
```

Algoritmo B (paso por referencia)

```
#include <iostream>

using namespace std;

void numMayor(int, int, int &);

int main(void) {
    int num1, num2, mayor, resultado;

    resultado = 0;
    cout << "Introduce dos números: ";
    cin >> num1 >> num2;

    resultado = num1 + num2;
    numMayor(num1, num2, mayor);
    cout << "Al sumar num1 y num2 da "<< resultado << endl;
}

//El procedimiento tiene un parámetro pasado por
//referencia
void numMayor(int n1, int n2, int &resultado)
{
    if (n1 > n2)
        resultado = n1;
    else
        resultado = n2;
}
```