

PROGRAMACIÓN AVANZADA

Dr. Andres Eduardo Gutierrez Rodriguez

CONTENIDO

Módulos

DLLs

Macros

FUNCIÓN

Una función es un bloque (módulo) de código fuente que realiza una **tarea bien definida** y su nombre debe describir lo que hace.

Si no puede pensar en un **nombre** para una función, probablemente la función no debería existir.

Las funciones dividen el código fuente en módulos reutilizables; por lo tanto, descomponen un problema en subproblemas más simples (**Divide y vencerás**).

EJEMPLO

```
int compute_max(int a, int b, int c){  
    int maximum = a;  
    if (b > maximum)  
        maximum = b;  
    if (c > maximum)  
        maximum = c;  
    return maximum;  
}
```

¿Cómo se puede
hacer más eficiente?

VISIBILIDAD

Antes de usar una función, debe declararse (prototipo) y definirse (implementarse).

Todas las variables definidas en una función son locales.

Una buena práctica de programación es declarar prototipos de funciones en archivos .h.

Las funciones estáticas solo son visibles en el archivo donde se implementan.

FUNCIONES REUTILIZABLES A BAJO NIVEL DE ABSTRACCIÓN

Suponga que desea desarrollar un programa para comparar huellas digitales. Para comparar las huellas digitales, necesita una función con dos huellas digitales como entrada y devuelve un valor de similitud.

¿Cómo diseñaría su componente de software? Justifica tu respuesta.



LIBRERÍAS

Una colección de funciones y tipos de datos exportados para otros programas.

- La interfaz de la biblioteca va en un archivo de encabezado (.h)

```
double sum(const double x, const double y);  
  
double diff(const double x, const double y);
```

La implementación va en archivos de implementación (.c):

```
#include "operaciones.h"  
  
double sum(double x, double y){  
  
    return x + y; }  
  
double diff(double x, double y){  
  
    return x - y; }
```

LIBRERÍAS ESTÁTICAS VS DINÁMICAS

Estático: el código ejecutable externo se incluye en su programa en el momento de la compilación.

Una vez que compiló su programa, puede borrar la biblioteca de forma segura

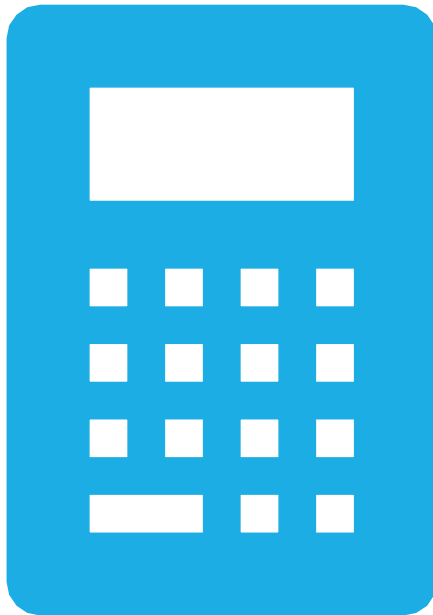
Si un tercero actualiza la biblioteca, debe volver a compilar su programa.

Dinámico: el código ejecutable externo se incluye en su programa en el momento de la ejecución.

No puede borrar la biblioteca porque su programa la utiliza en el momento de la ejecución.

Si un tercero actualiza la biblioteca, su programa funcionará bien siempre que la interfaz de la biblioteca permanezca sin cambios.


EJERCICIO 1



Realice un módulo llamado operaciones que permita calcular la suma y la resta de dos números.

No olvide declarar el archivo de encabezado e implementar en otro archivo la funcionalidad del módulo.

Investigue como crear una librería en su IDE de preferencia.



Cree una librería con el módulo del ejercicio anterior.



De ser posible, genere una librería estática y una dinámica para compararlas.

EJERCICIO 2

MACROS

Una macro es un identificador definido con la directiva del preprocesador `#define`.

```
#define PI 3.1416
```

```
#define CIRCLE_AREA (x) ((PI) * (x) * (x))
```

```
#define ABS_DIFFERENCE (x, y) ((x) > (y)? (x) - (y): (y) - (x))
```

Son patrones de sustitución que el preprocesador expande antes de compilar.

Por convención, se escriben en mayúsculas para distinguir los nombres de variables y funciones.

Una macro puede expandir otras macros.

Las macros no verifican el tipo de variables.

EJERCICIO 3



Complete la librería “operaciones” añadiendo otros operadores matemáticos, por ejemplo: multiplicación, división, potencia y raíz cuadrada.

Puede incluir operaciones sobre figuras geométricas.

Realice además un programa que permita probar el funcionamiento de las funciones que diseñó.

MUCHAS GRACIAS