

Funciones

martes, 9 de noviembre de 2021 07:12 a. m.

Funciones de usuario: Son funciones definidas por el desarrollador de acuerdo a sus necesidades.

Funciones de biblioteca: Son aquellas que se incluyen en librerías como: stdio.h, stdbool.h, stdlib.h, ctype.h, string.h

La utilidad de la modularidad recae en tener programas grandes y complejos con funciones principales cortas y especializadas, las cuales se utilizan como subprocesos/submódulos De tal forma que cada módulo será más sencillo de comprender, siempre y cuando se cumplan las reglas de programación estructurada.

*Las funciones son entidades que dado un conjunto de datos, a los cuales llamaremos "argumentos y parámetros", se les asigna una tarea muy específica en espera de un resultado específico.

Las funciones pueden ser llamadas desde cualquier parte del programa (dentro de la función main, o dentro de otras funciones) y cuantas veces sean necesarias e incluso se pueden ser llamadas a sí mismas (Recursividad).

Si una función no retorna valores, entonces hacemos referencia a que dicha función se trata de un **procedimiento**.

Los nombres de las funciones deben de ser adecuados, para que se pueda fácilmente interpretar el resultado devuelto y se tenga una idea de las operaciones que se realizan internamente.

Sintaxis de declaración y definición de funciones

```
/*
 * Descripción general de las operaciones que realiza la función
 * @param param1: Descripción o utilidad del parámetro
 * @return Descripción del resultado
 */
<tipo de dato de retorno> identificadorFuncion (lista de parámetros (con nombre y tipo de dato)){
    //cuerpo de la función
    //instrucciones

    return valorDeRetorno; //Se utiliza siempre y cuando el <tipo de dato de retorno> sea diferente de "void"
}
```

Sintaxis de declaración de prototipos

```
<tipo de dato de retorno> identificadorFuncion (lista de parámetros (solo el tipo de dato));
```

Para utilizar funciones de manera exitosa, se pueden seguir tres caminos:

1. Declarar y definir el comportamiento de la función antes de la función main.
2. Declarar el prototipo de una función antes del main y definir su comportamiento de dicha función después del main.
3. Declaración de prototipos de funciones dentro de un archivo cabecera <Biblioteca de usuarios -> archivos .h>, luego definiremos el comportamiento de dichas funciones en otro archivo .c, el cual implementará al archivo .h

Ejemplo (1er camino):

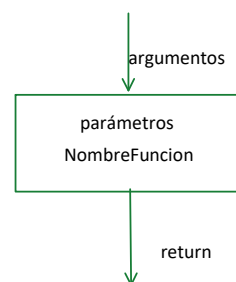
```
#include <stdio.h>
```

```
/*
 * Función que calcula el área de un triángulo.
 * @param base Es el valor que representa a la base del triángulo
 * @param altura Es el valor que representa a la altura del triángulo.
 * @return Devuelve el área calculada con los datos enviados.
 */
float areaTriangulo(int base, int altura){
    //instrucciones
    float area=0.0;
    area=base*altura/2;
    return area;
}
```

```
void main(){
    float resultadoArea=0.0;
    char texto[50]="Se creo la función correctamente";
    puts("Se creo la función correctamente");
}
```

Argumento: Es un dato (Valor) que se envía a una función para ser procesado. Y este valor debe tener el tipo de dato especificado por su parámetro. Estos se envían cuando se llaman a las funciones.

Parámetro: Es el tipo de dato esperado en la función. Se especifican en el momento de definición de una función.



```

    puts(texto);
    resultadoArea=areaTriangulo(30,15);
    printf("Resultado de utilizar la función: area del triangulo=%1.2f.\n",resultadoArea);
}

```

Ejemplo 2

```
#include <stdio.h>
```

```

/*
 * Función que calcula el área de un triángulo.
 * @param base Es el valor que representa a la base del triángulo
 * @param altura Es el valor que representa a la altura del triángulo.
 * @return Devuelve el área calculada con los datos enviados.
 */
float areaTriangulo(const int base, int altura){
    printf("Estos son los valores que recibí: base=%d, altura=%d \n", base, altura);
    //instrucciones
    float area=0.0;
    area=base*altura/2;
    return area;
}

void main(){
    char texto[50]="Se creo la función correctamente";
    puts("Se creo la función correctamente");
    puts(texto);
    printf("Resultado de utilizar la función: area del triangulo=%1.2f.\n",areaTriangulo(2,2));
    int baseUser=0;
    int alturaUser=0;
    puts("Dame la base del triangulo");
    fflush(stdin);
    scanf("%d", &baseUser);

    puts("Dame la altura del triangulo");
    fflush(stdin);
    scanf("%d", &alturaUser);
    float resultadoArea=areaTriangulo(baseUser, alturaUser);
    printf("Resultado de utilizar la función con sus datos: area del triangulo=%1.2f.\n",resultadoArea);
}

```