

Guía rápida sobre funciones

Esta guía es un resumen de algunos puntos sobre funciones.

Las funciones sirven para dos tareas:

- 1) Escribir el código una vez y utilizarlo muchas
- 2) Realizar programación modular, dividir el problema en subproblemas

Para un ejemplo de 1) ver el fuente del programa aviones.

Para un ejemplo de 2) ver el fuente del programa antiaereo.

Variables

Existen dos tipos de variables, las globales y las locales. Las globales pueden ser usadas en todas las funciones del programa y ser modificadas por ellas, en cambio las locales solo en la función en que fueron declaradas.

Las globales se declaran fuera de las funciones, y están disponibles para las funciones que se escriben inmediatamente después de que se declaran. Una sana costumbre es declararlas al principio del programa antes de cualquier función.

Un ejemplo de variable local la pueden encontrar en el programa antiaéreo:

```
//declaraciones de las variables globales
const int bizq=1; //limites de los bordes
const int bder=80; //están declarados globalmente
const int bsup=1; //para que todas las funciones puedan usarlas
const int binf=24; //de aquí en adelante
```

luego se usan en las funciones:

```
main
caidaBomba
avanceDisparo
```

Entrada y salida de valores

Toda función puede devolver un valor a través de la sentencia **return**, el tipo de valor debe ser declarado en el prototipo de la función al comienzo de la misma, por ejemplo:

```
int colTirAvi(int x,int y,int tx1,int ty1,int tx2,int ty2,int tx3,int ty3);
bool colBombTanq(int x,int y,int bx1,int by1,int bx2,int by2,int bx3,int by3);
void imprBombardero(int x, int y);
```

colTirAvi devuelve un valor **int**.

colBombTanq devuelve un valor **bool**.

imprBombardero no devuelve ningún valor porque se coloca la palabra clave **void**.

Si no se especifica nada el compilador asume que se trata de `int`.

A las funciones se le suministran los valores por los parámetros que se declaran en los paréntesis después del nombre de la función en estos paréntesis se especifican nombres de variables que harán reverencia a los valores pasados a la función:

```
void imprBombardero(int x, int y);
```

En este ejemplo, `x` tendrá el valor del primer parámetro que será un `int` e `y` tendrá el valor del segundo parámetro que también será un `int`.

Estos parámetros pueden ser de dos tipos:

- 1) Por valor
- 2) Por referencia

Por valor

El valor de la función que llama es copiado a la variable que recibe, esto hace que cambio que se haga en la función a dicha variable no modifique la original que llamó.

Por referencia

En este caso las modificaciones que se hagan dentro de la función llamada si cambiarán los valores de la función que llama, para esto en la definición del prototipo se debe agregar un símbolo `&` a las variables que queremos que sean por referencia.

Por ejemplo:

```
void lanzabomba(int x,int y,int &bx1,int &by1,int &bx2,int &by2,int &bx3,int &by3);
```

Los dos primeros parámetros, `int x,int y` , son por valor, el resto, `int &bx1,int &by1,int &bx2,int &by2,int &bx3,int &by3`, por referencia.

Esto es así porque esta función se encarga de buscar entre las variables por referencia un par que estén desocupadas para usarlas para referenciar una nueva bomba. Si encuentra un par libre llena los valores de inicio de caída de la bomba y eso los calcula porque en los parámetros por valor vienen las coordenadas del avión.

Estás últimas son por valor porque no necesitamos cambiar sus valores en el programa principal, pero las demás son por referencia porque necesitamos cambiar sus valores en el programa principal para que se entere que hay un par de variables que antes estaban desocupadas y ahora se están usando.

El uso de parámetros por referencia se puede usar como otra manera de devolver valores desde la función, y es nuestro único recurso cuando hay que devolver más de un valor.

Nota: En los parámetros por valor uno puede llamar la función tanto con variables como con literales (por ejemplo un número):

```
void imprBombardero(int x, int y);
```

Suponiendo que tengo dos variables int a,b;

Puedo llamar a la función de con variables:

```
imprBombardero(a,b);
```

o con literales:

```
imprBombardero(15,10);
```

En cambio los parámetros por referencia solo pueden ser llamados con variables ya que las constantes literales no podrían ser modificadas.

Importante:

En la declaración de los prototipos al principio del programa el prototipo termina en punto y coma.

En la definición de la función no hay que poner punto y coma al prototipo sino la llave abierta y terminar la función con llave cerrada.