Lenguaje de Programación: C++ Funciones

José Luis Alonzo Velázquez

Universidad de Guanajuato

Septiembre 2010

Funciones sobrecargadas

C++, a diferencia del C estándar, permite declarar funciones con el mismo nombre y a esto se conoce como **sobrecarga de funciones**. Las funciones sobrecargadas pueden coincidir en tipo, pero al menos uno de sus parámetros tiene que ser diferente. En todo caso, si usted trata de declarar funciones sobrecargadas que coincidan en tipo y número de parámetros el compilador no se lo permitirá. Para poner un ejemplo vamos a considerar el caso de dos funciones cuyo nombre será divide, ambas regresarán el cociente de dos números, salvo que una de ellas operará sobre números enteros y la otra lo hará sobre números reales (de punto flotante).

Funciones sobrecargadas

C++, a diferencia del C estándar, permite declarar funciones con el mismo nombre y a esto se conoce como **sobrecarga de funciones**. Las funciones sobrecargadas pueden coincidir en tipo, pero al menos uno de sus parámetros tiene que ser diferente. En todo caso, si usted trata de declarar funciones sobrecargadas que coincidan en tipo y número de parámetros el compilador no se lo permitirá. Para poner un ejemplo vamos a considerar el caso de dos funciones cuyo nombre será divide, ambas regresarán el cociente de dos números, salvo que una de ellas operará sobre números enteros y la otra lo hará sobre números reales (de punto flotante).

Observación

Nota: cuando en los programas se hace una llamada a una función sobrecargada, el compilador determina a cual de las funciones invocar en base al tipo y número de parámetros pasados a la función.

Ejemplo

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

using namespace std;
// divide enteros
int divide(int a, int b){
    printf("división entera");
    if(b!=0){
        return a/b;
    }else{
        return 0;
    }
}
```

Ejemplo

```
// divide reales
double divide(double a, double b){
    printf("división entera");
    if(b!=0){
        return a/b;
    }else{
        return 0;
    }
}
// punto de prueba
int main(){
    printf("%d",divide(10, 3));
    printf("%lf",divide(10.0, 3.0));
}
```

Número variable de parámetros

En C,C++ se pueden crear funciones que operen sobre una lista variable de parámetros, es decir, en donde el número de parámetros es indeterminado. En esta sección se mostrará un ejemplo de la manera en que podemos crear funciones para manejar tales asuntos, y para ello haremos uso de tres macros soportadas por C++:

1. **va_list** puntero de argumentos
2. **va_start** inicializar puntero de argumentos
3. **va_end** liberar puntero de argumentos

La sintaxis que usaremos para declarar funciones con lista de parámetros variables es:

```
1) tipo nombrefuncion(...)
2) tipo nombrefuncion(int num, ...)
```

Ejemplo

donde:

1. tipo es el tipo regresado por la función
2. nombrefuncion es el nombre de la función
3. int num es el número de parámetros que la función procesará
4. ... esta notación se emplea para indicar que el número de parámetros es variable

Nota: observe que la primera forma de declaración es realmente variable el número de parámetros a procesar y en estos casos se debe establecer el mecanismo para determinar cuando se ha procesado el último de los argumentos, en el segundo tipo de declaración el número total de parámetros a procesar es igual al valor del parámetro num.

Ejemplo

```
#include <stdio.h>
#include <stdarg.h>
using namespace std;
// Esta función opera sobre una lista variable de números entero
int suma( int num, ... ){
    int total = 0;
    va_list argptr;
    va_start( argptr, num );
    while( num > 0 ){
        total += va_arg( argptr, int );
        num--;
    }
    va_end( argptr );
    return( total );
}
int main(){
    printf("%d",suma(4, 100, 200, 300, 400));
    return 0;
}
```

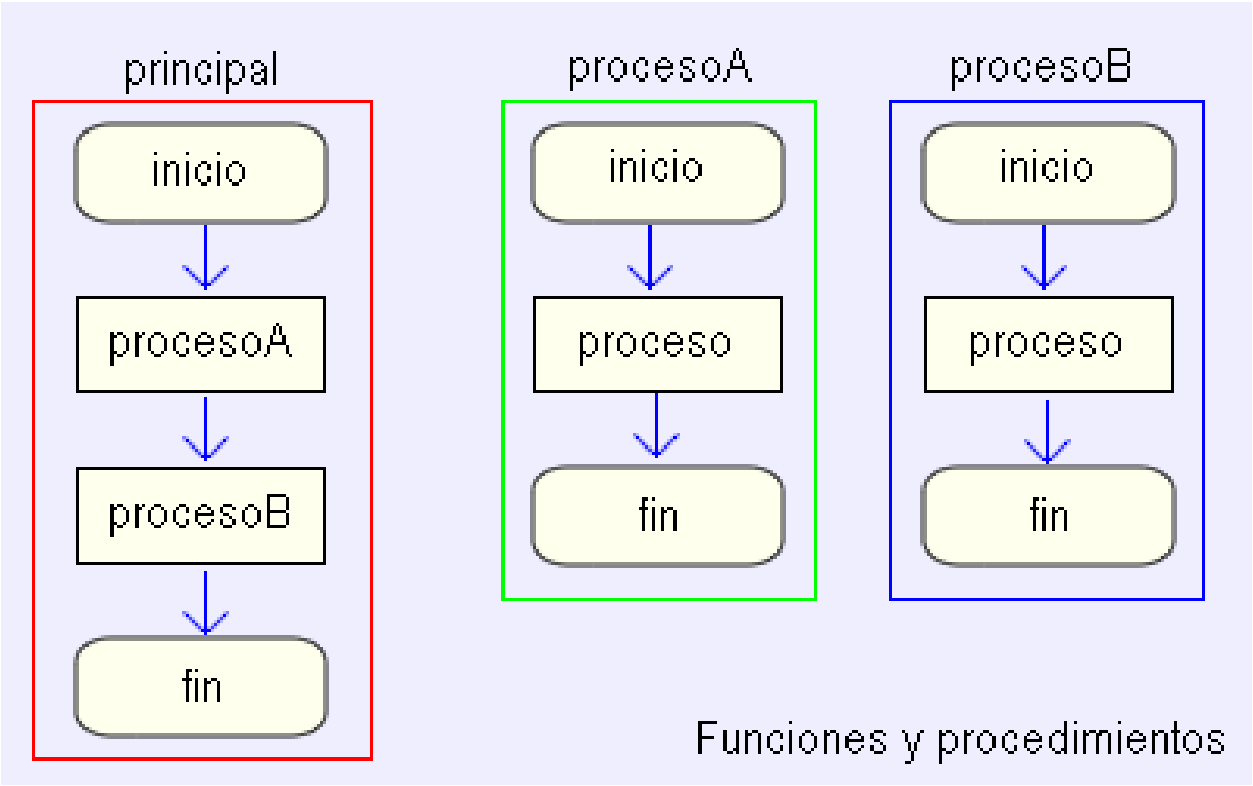

Figura: Repaso de funciones

Problema para clase

Hacer un programa que tenga una función que multiplique 4 números dados por el usuario, y imprima el resultado en pantalla.


Programming Principles and Practice Using C++, Bjarne Stroustrup.


http://www.codeblocks.org


http://www.wxwidgets.org


(O'Reilly) Practical C Programming (3rd Edition)