



INGENIERIA EN COMPUTACION



Materia: Ingeniería en computación

Alumno: Ayon Dimas Andres

Matricula: 366845

Profesor: Pedro Nunes Yepiz

Tema: Ciclos y funciones

INTRODUCCION

Podremos aprender cómo es que se realiza y para qué sirven una función en C#, al igual que como aplicarlas a nuestros programas, ya que estas nos serán de mucha ayuda para poder hacer un programa donde queramos que se ejecute determinada parte del código en C#, así mismo implementaremos lo que son los ciclos, tanto en for, while y también el ciclo do while.

COMPETENCIA

Que el alumno aprenda a utilizar las distintas sentencias durante esta práctica, ya que es importante que sean dominadas en su totalidad. Esto nos facilita a la hora de realizar un programa en C# ya que tenemos más formas de poder realizar un programa, con determinadas sentencias que nos pueden llegar a ayuda a mejorar nuestro programa y que de esta manera sea eficiente y eficaz. Esto nos ayuda a mejorar la eficiencia de espacio, de memoria y de procesador.

FUNDAMENTOS

El código de un programa escrito en C se divide en funciones. Aunque similares a los “métodos” de Java, las funciones no están asignadas ni a una clase ni a un objeto. Una función en C se distingue sólo por su nombre. Dos funciones con igual nombre y con diferente número y tipo de parámetros se considera una definición múltiple, y por tanto un error.

Las funciones suelen encapsular una operación más o menos compleja de la que se deriva un resultado. Para ejecutar esta operación, las funciones pueden precisar la invocación de otras funciones (o incluso de ellas mismas como es el caso de las funciones recursivas).

Las funciones en un programa son entidades que dado un conjunto de datos (los parámetros), se les encarga realizar una tarea muy concreta y se espera hasta obtener el resultado. Lo idóneo es dividir tareas complejas en porciones más simples que se implementan como funciones. La división y agrupación de tareas en funciones es uno de los aspectos más importantes en el diseño de un programa.

TIPO_FUNCIÓN

Puede ser de cualquier tipo de los que conocemos. El valor devuelto por la función será de este tipo. Por defecto, es decir, si no indicamos el tipo, la función devolverá un valor de tipo entero (`int`). Si no queremos que retorne ningún valor deberemos indicar el tipo vacío (`void`).

NOMBRE_FUNCIÓN

Es el nombre que le daremos a la función.

TIPO Y NOMBRE DE ARGUMENTOS

Son los parámetros que recibe la función. Los argumentos de una función no son más que variables locales que reciben un valor. Este valor se lo enviamos al hacer la llamada a la función. Pueden existir funciones que no reciban argumentos.

BLOQUE DE SENTENCIAS

Es el conjunto de sentencias que serán ejecutadas cuando se realice la llamada a la función.

Las funciones pueden ser llamadas desde la función `main` o desde otras funciones. Nunca se debe llamar a la función `main` desde otro lugar del programa. Por último, recalcar que los argumentos de la función y sus variables locales se destruirán al finalizar la ejecución de la misma.

PROCEDIMIENTO

Las funciones en C tienen el siguiente formato:

```
tipo_del_resultado NOMBRE(tipo_param1 param1, tipo_param2 param2, ... )
{
    /* Cuerpo de la función */
}
```

Cuando se invoca una función se asignan valores a sus parámetros y comienza a ejecutar el cuerpo hasta que se llega al final o se encuentra la instrucción `return`. Si la función devuelve un resultado, esta instrucción debe ir seguida del dato a devolver. Por ejemplo:

```
int search(int table[], int size)
{
    int i, j;
    if (size == 0)
    {
        return 0;
    }
    j = 0;
    for (i = 0; i < size; i++)
    {
        j += table[i];
    }
    return j;
}
```

La ejecución de la función comienza en la línea 4. Si el parámetro `size` tiene valor cero, la función termina y devuelve el valor cero. Si no, se ejecuta el bucle y se devuelve el valor de la variable `j`. El tipo de la expresión que acompaña a la instrucción `return` debe coincidir con el tipo del resultado declarado en la línea 1.

La llamada a una función se codifica con su nombre seguido de los valores de los parámetros separados por comas y rodeados por paréntesis. Si la función devuelve un resultado, la llamada se reemplaza por su resultado en la expresión en la que se incluye. Por ejemplo:

```
int addition(int a, int b)
{
    return (a + b);
}

int main()
{
    int c;
    c = c * addition(12, 32);
}
```

}

RESULTADOS Y CONCLUSIONES

De lo que pudimos aprender durante esta práctica, podemos destacar a las funciones ya que es algo sumamente útil y de gran relevancia a la hora de programar en C#. Nos facilita el código a la hora de ejecutarse, ya que nosotros como usuarios podemos decidir que parte del código queremos ejecutar, sin necesidad de que el programa se ejecute completo y haga cosas que en ese momento no se requieran, pero esto no va solo, sino que debe de ir acompañados de los ciclos, ya sea el ciclo For, Ciclo While o el Ciclo Do While, ya que de esta manera podremos realizar un menú en C# como lo hicimos en esta práctica, el cual nos ayudó y nos fue de mucha utilizada para ejecutar cada uno de los programas solicitados por el profesor, donde pasamos de hacer un archivo para cada programa a realizar solamente un archivo para varios programas, esto gracias a las funciones en C#