

Creación y uso de una librería estática (.h)

Escrito por adrianvaca

Domingo, 20 de Marzo de 2011 19:03 -

Las librerías son un conjunto de funciones que usamos frecuentemente y que por lo general están agrupadas de acuerdo a su utilidad, por ejemplo la iostream tiene funciones para controlar la entrada y salida de datos.

En el presente artículo crearemos una librería estática (.h) y luego haremos un programa en el cual usemos esta librería, para ello vamos a usar el compilador

Dev-C++ 4.9.9.2

, si no lo tienes descárgalo primero de la sección

Compiladores

de este sitio.

Empecemos, como siempre paso a paso:

Creación de la librería

1. Primero que nada crearemos una carpeta de nombre libUtilidades en C:\Dev-Cppbin en la cual iremos guardando todos los archivos del proyecto. Abrimos el Dev-C++ y vamos a Archivo

->

Nuevo

->

Proyecto

, seleccionamos

Static Library

y el nombre del proyecto será

libUtilidades

tal como se muestra:



2. Ahora agregamos un nuevo fichero, para esto vamos a Proyecto -> Nuevo Código Fuente, escribimos el siguiente código ahí y guardamos el archivo como

Creación y uso de una librería estática (.h)

Escrito por adrianvaca

Domingo, 20 de Marzo de 2011 19:03 -

utilidades.h

:

```
codeDivStart()/* utilidades.h */  
  
int suma(int a, int b);  
int es_palindroma(char *cadena);
```

Como podemos ver aquí definimos únicamente el prototipo o molde de las funciones a usar. La primera devolverá el resultado de la suma de 2 enteros y la segunda función devolverá 1 si la cadena ingresada como parámetro es palindroma ó 0 si no lo es.

Si, anteriormente dije que una librería debe tener funciones agrupadas de acuerdo a su utilidad. En este caso estas funciones no tienen nada que ver una con la otra, pero sirve de ejemplo para comprender ciertas cosas.

Otra cosa, en este fichero .h únicamente colocamos los prototipos de funciones que queramos hacer públicas, es decir que queramos que los usuarios vean y por lo tanto puedan usar.

3. Ahora vamos a crear la implementación de las funciones anteriores, para esto vamos a Proyecto -> Nuevo Código Fuente, escribimos el siguiente código ahí y guardamos el archivo como utilidades.cpp:

```
codeDivStart()/* utilidades.cpp */  
  
#include <string>  
  
int suma(int a, int b)  
{  
    return a + b;  
}  
  
int longitud(char *cadena)  
{  
    int len = 0;  
    for(int i=0; ; i++)  
    {  
        if(cadena[ i ] == '\0')  
            break;  
  
        len ++;  
    }  
    return len;
```


Creación y uso de una librería estática (.h)

Escrito por adrianvaca

Domingo, 20 de Marzo de 2011 19:03 -

necesarios los 2 ficheros anteriores

Uso de la librería

1. En primer lugar copiamos el archivo utilidades.h al directorio include del compilador y el archivo libUtilidades.a al directorio li

2. Una vez hecha la librería, crearemos ahora un programa de prueba. Primero que nada crearemos una carpeta de nombre pruebaUtilidades en C:Dev-Cppbin en la cual iremos guardando todos los archivos de este proyecto de prueba. Abrimos el Dev-C++ y vamos a Archivo

->

Nuevo

->

Proyecto

, seleccionamos

Console Application

y el nombre del proyecto será

pruebaUtilidades

tal como se muestra:



Reemplazamos el código inicial por el siguiente:

```
codeDivStart()#include <stdlib>
```

```
#include <iostream>
```

```
#include <utilidades.h>
```

Creación y uso de una librería estática (.h)

Escrito por adrianvaca

Domingo, 20 de Marzo de 2011 19:03 -

```
using namespace std;
```

```
int main(int argc, char *argv[])
```

```
{
```

```
    int a, b, resultado;
```

```
    char cadena[50];
```

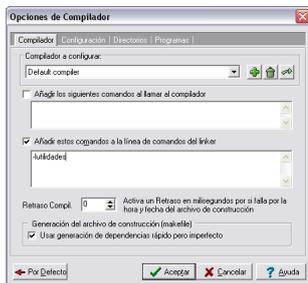
```
    cout > a;
```

```
    cout > b;
```

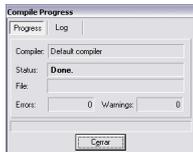
```
    resultado = suma(a, b);
```

```
    cout
```

Opciones del compilador activa la casilla Añadir estos comandos a la línea de comandos del link



3. Ahora compilamos el programa, el resultado de la compilación es:



4. Ahora ejecutamos el programa, para esto presionamos

```
Ingreso a: 2
Ingreso b: 3
Resultado a+b=5
Ingreso cadena: ala
Es palindroma
Presione una tecla para continuar . . .
```