

1.1 Introducción a C++

1.1 Introducción a C++

1. Introducción
2. Estructura del fichero
3. IDE C++
4. Elementos básicos.

1. Introducción

C++ es un lenguaje de programación de alto nivel, que usa como traductor un **compilador**.

En general, un programa C++ suele estar escrito en diferentes ficheros. Durante el proceso de compilación estos ficheros serán combinados adecuadamente y traducidos a código objeto, obteniendo el programa ejecutable.

2. Estructura del fichero

El fichero suele comenzar con unas líneas para incluir las definiciones de los módulos de biblioteca que utilice nuestro programa, e irá seguido de declaraciones y definiciones de tipos, de constantes y de subprogramas. El programa debe contener un subprograma especial (la función main) que indica dónde comienza la ejecución. Las instrucciones contenidas en dicha función main se ejecutarán una tras otra hasta llegar a su fin. La función main devuelve un valor que indica si el programa ha sido ejecutado correctamente o, por el contrario, ha ocurrido un error. En caso de no aparecer explícitamente una sentencia return, por defecto, se devolverá un valor que indica terminación normal.

```
#include <iostream>
using namespace std ;
const double EUR_PTS = 166.386 ;
int main()
{
    cout << "Introduce la cantidad (en euros): " ;
    double euros ;
    cin >> euros ;
    double pesetas = euros * EUR_PTS ;
    cout << euros << " Euros equivalen a " << pesetas << " Pts" << endl ;
    // return 0 ;
}
```

3. IDE C++

Para obtener el programa ejecutable necesitamos dos herramientas básicas: un editor con el que crear un archivo con el texto del mismo (el código fuente) y un compilador para traducir el código fuente a código ejecutable. Para realizar estos pasos podemos seguir dos enfoques diferentes: usar directamente la línea de comandos del sistema operativo o usar un entorno integrado de desarrollo (IDE).

4. Elementos básicos.

- **Bibliotecas.** Una biblioteca es un programa c++ ya creado, que se puede incluir en mi proyecto para así poder beneficiarme de funciones y métodos ya creados.

El lenguaje C++ consta de un reducido número de instrucciones, pero ofrece un amplio repertorio de bibliotecas con herramientas que pueden ser importadas por los programas cuando son necesarias. Por este motivo, un programa suele comenzar por tantas líneas `#include` como bibliotecas se necesiten.

- **Constantes simbólicas y constantes literales.** Una constante es un valor que no puede ser cambiado. Por ejemplo, 1 euro equivale 166.386 pesetas, eso es un valor invariable por lo que se considera como constante. Para declarar una variable es necesario usar la palabra reservada `const`, el tipo de dato de la constante y su nombre identificativo (normalmente en mayúsculas).

```
const double EUR_PTS = 166.386 ;
```

- **Identificadores.** Para cada elemento que introduzcamos en nuestro programa debemos definir un nombre (identificador) con el que hacer referencia al mismo y disponer de algún mecanismo para informar al compilador de dicho nombre y de sus características.
- **Palabras reservadas.** Algunas palabras tienen un significado especial en el lenguaje y no pueden ser utilizadas con otro sentido. Por este motivo no pueden ser utilizados para designar elementos definidos por el programador. Son palabras reservadas, como por ejemplo: `using`, `namespace`, `const`, `double`, `int`, `char`, `bool`, `void`, `for`, `while`, `do`, `if`, `switch`, `case`, `default`, `return`, `typedef`, `enum`, `struct`, etc.
- **Delimitadores.** Son símbolos que indican comienzo o fin de una entidad (`() { } ; , < >`). Por ejemplo, en nuestro programa `euros.cpp` usamos `{ }` para delimitar el comienzo y el final de la función `main`, y el símbolo `;` para delimitar el final de una sentencia.
- **Operadores.** Son símbolos con significado propio según el contexto en el que se utilicen. Ejemplo: `= << >> / % + - < > <= >= == != ++ -- . ,` etc.
- **Comentarios.** Un comentario es una utilidad para el programador, nunca se ve por pantalla, y por ende no tiene funcionalidad. Existen dos tipos de comentarios: los de una línea (`//`) y los de bloque (`/**/`).
- **Variables.** Las variables son información que se guardan en memoria. Los datos se almacenan en memoria en variables de un cierto tipo. El programador debe decidir qué variables va a utilizar, pensar en un identificador para referirse a ellas y comunicarlo al compilador.
- **Entrada/Salida de datos.** En general, un programa necesitará tomar datos de entrada y mostrar resultados en algún dispositivo de salida. En C++ disponemos de flujos de entrada (`cin`) y de salida (`cout`), que nos permiten efectuar entrada y salida de datos, respectivamente. El operador `>>` se usa para tomar un valor de la entrada y el operador `<<` se usa para sacar un valor por la salida.
- **Asignación.** Una asignación es darle un valor a una variable o a una constante. Si se le da por primera vez, se le suele llamar declaración.

