



FACULTAD DE INFORMÁTICA

El lenguaje C++

Una introducción para programadores

TALLER

Programación orientada a objetos — Unidad 0

Autor: Luis Hernández Yáñez

FdI
UCM

La sintaxis del lenguaje C++

Un ejercicio sobre sintaxis (*resolución individual*)

El siguiente programa contiene bastantes errores de sintaxis. Intenta detectar el mayor número de errores que puedas (señala dónde se encuentra la causa). Aunque el compilador te ayudaría mucho a encontrar los errores, intenta encontrarlos por ti mismo.

```
#include <istream>
using namespace std
#define MAX 100

chr menu();
boolean divisivol(int; int);
void prueba_for
void prueba_while();

int main() {
    cout << "Bienvenido al programa de prueba\n";
    char c == menu();
}
```

Programación orientada a objetos (Facultad de Informática)

Unidad 0 - Taller - 1

FdI
UCM

La sintaxis del lenguaje C++

```
switch(c) {
    case 1:
        prueba_for();
        break
    case 2:
        prueba_while();
        break;
    case 3:
        integer num1, num2;
        cout << "Dame un entero: ";
        cin > num1;
        cout << "Dame otro: ";
        cin > num2;
        if(divisible(num1, num2) then
            cout << "El primero es divisible por el segundo\n"
        else
            cout << "El primero no es divisible por el primero\n";
        break;
}
return 0;
};
```

Programación orientada a objetos (Facultad de Informática)

Unidad 0 - Taller - 2

FdI
UCM

La sintaxis del lenguaje C++

```
char menu {
    char c;
    do
        cout << "Lo que podemos mostrarte:\n";
        cout << "1 - Prueba del for\n";
        cout << "2 - Prueba del while\n";
        cout << "3 - Prueba de funcion\n";
        cout << "Quieres... (1 a 3): ";
        cin >> c;
        if((c < '1') or (c > '3')) cout << "Intenta otra vez...\n"
    while((c < '1') or (c > '3'));
    return c
}

bool divisible(int numero1, int numero2)
    if(numero1 mod numero2 = 0) return true;
    return false;
}
```

Programación orientada a objetos (Facultad de Informática)

Unidad 0 - Taller - 3

```
void prueba_for() {
    cout << "Uso un for para repetir algo...\n";
    for(int i = 1, i <=MAX, i++) cout >> (i % 10);
    cout << "\n";
}

void prueba_while() {
    cout << "Mientras que 'letra' no sea la 'x' yo sigo...\n";
    char letra = 'a';
    while(letra <> "x") {
        cout << "Dame 'letra': ";
        cin >> letra;
    };
}
```

Y, de paso, ¿has entendido lo que hace el programa?

Sobre las diferencias entre Pascal y C++ (*resolución individual*)

¿Qué mecanismos has visto en C++ que no tengan equivalente en Pascal?

De las construcciones de bucles de C++, ¿has visto alguna que no sea exactamente equivalente a las que hay en Pascal?

Respecto de lo que conoces del lenguaje Pascal, ¿hay algo que eches de menos en lo que hemos visto del lenguaje C++?

Ejercicio práctico sobre el lenguaje C++ (*resolución en grupo*)

Se trata de desarrollar un TAD (Tipo abstracto de datos) **Complejo** que permita trabajar con números complejos en los programas.

Usaremos sólo lo que hemos visto hasta ahora de C++.

La funcionalidad que se desea para el TAD es la básica:

- ✓ Construcción de un complejo a partir de su partes real e imaginaria
- ✓ Suma de complejos
- ✓ Resta de complejos
- ✓ Multiplicación de complejos
- ✓ División de complejos
- ✓ Visualización de complejos en una forma del tipo $(2 + 4i)$

Pasos

1. Lo primero que tenéis que hacer es centraros en los datos necesarios.
2. Una vez que hayáis establecido la estructura de datos adecuada, deberéis pasar a implementar las seis operaciones.
3. Superado el paso anterior deberéis convertir el TAD en una biblioteca, de forma que esté disponible para cualquier programador.
4. Para terminar, desarrollad un sencillo programa principal que use todas las características del TAD **Complejo**.

Otros requisitos

Para implementar las operaciones aritméticas utilizad funciones operadoras.

Utiliza comentarios donde te parezcan útiles (*piensa en los demás*).