

Programación y Estructuras de Datos

SEMINARIO C++

Guía rápida

SEMINARIO - Generalidades

Método propio del libro.

En el libro se estructuran los contenidos de la siguiente manera :

En cada capítulo, se trabaja básicamente sobre la clase **TCoordenada**.

Al final de cada capítulo se proponen como ejercicio las clases **TVector** (de enteros) y **TCalendarario**. Para las dos se incluye la solución.

El código de las cuatro clases que se desarrollan durante el seminario (**TCoordenada**, **TLinea**, **TVector** y **TCalendarario**) aparece implementado al final del libro.

Se propone como ejercicio al final la clase **TComplejo** (no se incluye la solución).

Formas de COMPILAR.

Estas dos formas serán las que más se usan al seguir el libro :

- "g++ -o prog main.cc" → genera un EJECUTABLE llamado "**prog**"
- "g++ main.cc" → genera un EJECUTABLE llamado "**main.o**"

Aparte , están estas otras formas :

- "g++ -o main -c main.cc" → genera un FICHERO BINARIO (no ejecutable) llamado "**main**"
- "g++ -c main.cc" → genera un FICHERO BINARIO (no ejecutable) llamado "**main.o**"

SEMINARIO - SESIÓN 1

"C++ paso a paso" - Capítulo 2 : "Clases y objetos"

- **2.1 Introducción**

- Fíjate en los ejemplos de ***ptrP3** y **&refP4** y su diferencia : **&refP4** actúa como un nombre alternativo (alias) para la variable **objetoP1**. Ambas tienen la misma dirección , y si cambia **objetoP1** , cambia también **refP4** (pág.8).
- Teclea y compila el **ejemplo 2.1** y comprueba resultado.

- **2.2 Declaración de una clase**

- Teclea y compila el **ejemplo 2.2** . Este ejemplo completa al anterior (**ejemplo 2.1**), ya probado antes.

- **2.3 Acceso a miembros de una clase**

- Fíjate en el ejemplo de **TCoordenada p1** , accediendo a sus valores **p1.x** , etc. (pág.11)
- Fíjate en el ejemplo de **TCoordenada *ptr1** , accediendo a sus valores **ptr1->x** , etc. Repara en cuán similar es esto a acceder a una **struct** mediante un **puntero**. (pág.11)
- Teclea y compila el **ejemplo 2.3** y comprueba resultado (errores). Este ejemplo completa al anterior (**ejemplo 2.2**), ya probado antes.

- **2.4 Control de acceso**

- Fíjate en la descripción de las etiquetas **public:** , **private:** y **protected:** , especialmente **private:** , que es el predeterminado. (cuadro pág.14).
- Teclea y compila el **ejemplo 2.4** y comprueba resultado . Este ejemplo completa al anterior (**ejemplo 2.3**), ya probado antes.
- Fíjate en la necesidad de incluir funciones en la clase (pág.15).
- Teclea y compila el **ejemplo 2.5** (incluyendo las funciones nuevas y pasando a **private:** las variables de clase) y comprueba resultado . Este ejemplo completa al anterior (**ejemplo 2.4**), ya probado antes.

- **2.5 Visualización de un objeto**

- Teclea y compila el **ejemplo 2.6** y comprueba resultado . Este ejemplo completa al anterior (**ejemplo 2.5**), ya probado antes.
- (Este ejemplo aporta la función **Imprimir** , para visualizar los elementos de la clase) .

- **2.6 Empleo de punteros**

- Fíjate en las 2 formas de acceder con punteros a un miembro de clase : **ptr1→x** y con el operador de desreferencia (***ptr1**).x (pág.18).
- Teclea y compila el **ejemplo 2.7** y comprueba resultado (error). Tecléalo NUEVO, sin borrar el **ejemplo 2.6** de antes, que hemos ido progresivamente enriqueciendo.
- Ahora haz la corrección en la línea 18 del **ejemplo 2.7** que propone al final del capítulo, y recompila, comprobando la bondad de dicha corrección.

- **2.7 Separación interfaz-implementación**

- Fíjate en la descripción de los 3 tipos de fichero en los que se escindirá la definición de cada clase : el ".h", el ".cc" y el ".o" (pág.20).
- Teclea el **ejemplo 2.8** , llevando la definición de clase exclusivamente ahí.
- Teclea el **ejemplo 2.9** , llevando los métodos de clase ahí.
- Teclea el **ejemplo 2.10** , definiendo el MAIN que ejecute la clase. Compíllalo de la forma en que se dice en la pág.22 .

- **2.8 Herramienta MAKE**

- Fíjate en la descripción de MAKEFILE dedicado a una clase (pág.23).
- Teclea el **ejemplo 2.11** (MAKEFILE sin parámetros) y ejecútalo para la clase TCoordenada , que hemos estado tratando.
- Teclea el **ejemplo 2.12** (MAKEFILE con parámetros) y ejecútalo para la misma clase TCoordenada.

- **2.9 Ficheros encabezado**

- Fíjate en los ficheros de encabezado propuestos (pág.24).
- Prueba a completar el **ejemplo 2.10** (antes probado) , con los #INCLUDE de <iostream> y <string>.

- **2.10 Espacios de nombres**

- Fíjate en la definición de espacios de nombres y en sus formas de declaración (pág.25).
- Fíjate cómo el **ejemplo 2.10** antes probado, ya incluye la cláusula **using namespace std.**

- **2.12 Ejercicios de PROGRAMACIÓN / 2.14 Respuesta a ejercicios de PROGRAMACIÓN**

- Puedes revisar en principio la definición de las clases **TCalendarario** y **TVector**.