

Programación orientada a objetos - Informática II

Clase 3 – Definición de una clase

¡Hola! ¿Cómo están? Continuamos bajo la modalidad en línea de Informática II, esta semana comenzamos a trabajar sobre un nuevo lenguaje y el paradigma de la POO, en la clase de hoy nos centraremos en la sintaxis del lenguaje y el concepto de clase.

¿Qué es una clase?

- Una clase es un descriptor (matriz) de un conjunto de objetos con estructura, comportamiento y relaciones similares.
- Es una abstracción de la realidad
- Cómo se logra ese comportamiento no es tema de quien usa la clase, sino, de quien la implementa.
- Sirve como molde o plantilla para crear “instancias” o como ya sabemos objetos reales descritos por la clase.

¿Cómo se implementa?

```
class nombre_clase
{
    modificador_de_acceso_1:
        miembro_1;
    ...
    modificador_de_acceso_2:
        miembro_2;
    ...
};
```

Veamos un ejemplo de la definición de la clase **Hora**

```
class Hora
{
    public:
        Hora();
        void ConfigurarHora(int, int, int);
        void MostrarHora(void);

    private:
        int hora ;
        int minuto;
        int segundo;
} ;
```

Las etiquetas *public* y *private* se denominan **especificadores de acceso a miembros**.

Cualquier atributo (característica) o método (comportamiento) declarada luego de *public* será accesible desde cualquier parte del programa (dentro y fuera de la clase). En el caso del especificador *private*, los atributos y métodos solo serán accesibles dentro de la clase.

Cuestiones importantes:

- Definimos una clase llamada **Hora**, la cual posee atributos y métodos públicos y privados.
- Dentro del especificador público encontramos el constructor por defecto de la clase, y dos métodos, uno para configurar o establecer la hora y otro para mostrar o informar la hora.
- Dentro del especificador privado, encontramos 3 atributos asociados a la hora, minuto y segundo.
- Esta definición se encontrará en el archivo **Hora.h**
- Su implementación la realizaremos en el archivo **Hora.cpp**

¿Puedo encontrar algo más dentro de una clase? ¿Existen otros modificadores de acceso? ¿Cómo organizo y jerarquizo la información?

```
class nombre_clase
{
    private:
        // típicamente datos sensibles...

    public:
        nombre_clase ();    //constructor de la clase
        ~nombre_clase ();  //destructor de la clase

        // típicamente métodos y datos que interactúan "puertas afuera"

    protected:
        /* relacionado con el concepto de "herencia" , paciencia ya
        llegaremos ☺ */
};
```

En estos breves ejemplos han aparecido algunos conceptos o palabras a las que no estamos habituados: constructor, destructor, métodos, visibilidad. De a poco iremos viendo cada uno de ellos.

¿Qué es un constructor?

Un constructor es una función con el mismo nombre de la clase que se ejecuta tan pronto como una instancia de la clase es creada.

Cuestiones que necesitamos saber:

- Un constructor es una función miembro especial que inicializa la instancia de una clase.
- Una clase puede (y suele) tener más de un constructor (sobrecargados).
- Un constructor tiene el mismo nombre de la clase que construye.
- Un constructor no retorna tipo alguno, ni siquiera **void**.
- Instanciar una clase provoca la llamada al constructor por default.

¿Qué es un destructor?

Un destructor es una función con el mismo nombre de la clase y un ~ antepuesto, que se ejecuta tan pronto como una instancia de la clase es destruida.

Cuestiones que necesitamos saber:

- Un destructor es la contraparte del constructor.
- Limpia el objeto cuando no se requiere más su uso, liberando los recursos que este ocupaba.
- Un destructor se identifica por el nombre de la clase anteponiéndole una tilde ~.
- Un destructor no toma ni regresa parámetros.
- Dejar de utilizar el objeto provoca la llamada al destructor.

Tipos de métodos

- Un consultor (*get*) es una función que retorna un valor desde su objeto, pero no cambia el objeto (sus atributos). Permite acceder a los atributos del objeto.
- Un modificador (*set*) es una función que modifica su objeto.

Final de esta clase

Bueno, este es el final de la clase de C++ sobre clases y su implementación.

Los y las invitamos a ver la video-clase que complementa este material de lectura inicial.

Si surgen dudas o consultas, pueden escribirnos en el [foro](#) correspondiente.

En la próxima clase, comenzaremos a trabajar sobre los constructores, el destructor y sus características.

¡Nos leemos!