Clases y objetos de C++

Los **objetos** y las **clases** se utilizan para encapsular *funciones y datos relacionados* en un solo lugar en C++.

Supongamos que necesitamos almacenar el *largo*, el *ancho* y la *altura* de una *habitación* rectangular y calcular su *área* y *volumen*.

Para manejar esta tarea, podemos crear tres variables , digamos, *largo*, *ancho*, y *altura*, junto con las funciones **`calcula_area()`** y **`calcula_volume()`**.

Sin embargo, en C++, en lugar de crear variables y funciones separadas, también podemos envolver los datos y funciones relacionados en un solo lugar (**creando objetos**).

Este paradigma de programación se conoce como **Programación Orientada a Objetos**, (**POO**).

Pero antes de que podamos crear objetos y usarlos en C++, primero necesitamos aprender sobre las clases.

1. Clase C++

Una clase es un modelo para el objeto. Podemos pensar en una clase como un boceto (*prototipo*) de una casa.

Contiene todos los detalles sobre los pisos, puertas, ventanas, etc. Construimos la casa basándonos en estas descripciones. **La casa es el objeto**.

2. Crear una clase

Una **clase** se define en C++ utilizando la palabra clave **`class`** seguida del nombre de la clase.

El cuerpo de la clase se define dentro de llaves y **termina con un punto y coma** al final.

```
class ClassName {
   // algunos datos
   // algunas funciones
};
```

Por ejemplo,

```
class Habitacion {
    public:
```

```
        double largo;
        double ancho;
        double altura;

        double calcula_area()
        {
            return largo * ancho;
        }

        double calcula_volume()
        {
            return largo * ancho * altura;
        }
};
```

Aquí definimos una clase llamada **`Habitacion`**.

Las variables son **`largo`**, **`ancho`**, y **`altura`**. Los elementos declarados dentro de la clase se conocen como **datos miembros**, y las funciones **`calcula_area()`** y **`calcula_volume()`** se conocen como **funciones miembro de una clase**.

3. Objetos de C++

Cuando se define una clase, **solo se define la especificación del objeto**; no se asigna memoria ni almacenamiento.

Para utilizar los datos y las funciones de acceso definidas en la clase, necesitamos **crear objetos**.

3.1. Sintaxis para definir objetos en C++

```
ClassName nombre_objeto;
```

Podemos crear objetos de la clase Habitacion (definida en el ejemplo anterior) de la siguiente manera:

```
// function_ejemplo
void function_ejemplo()
{
    // crea objectos
    Habitacion habitacion1, habitacion2;
}

int main()
```

```
{
    // crea objectus
    Habitacion habitacion3, habitacion4;
}
```

Aquí, dos objetos **habitación1** y **habitación2** de la clase **Habitacion** que se crean en la **function_ejemplo()**.

De manera similar, los objetos **habitacion3** y **habitacion4** se crean en **main()**

Como podemos ver, podemos crear objetos de una clase en cualquier función del programa. También podemos crear objetos de una clase dentro de la propia clase o en otras clases.

Además, podemos crear tantos objetos como queramos a partir de una sola clase.

3.2. Miembros de datos y funciones miembro de C++ Access

Podemos acceder a los datos miembros y a las funciones miembro de una clase utilizando un operador **.** (**punto**) .

Por ejemplo,

```
habitacion2.calcula_area();
```

Esto llamará a la función ***calcula_area()*** dentro de la clase Habitacion para el objeto **habitación2**.

De manera similar, se puede acceder a los miembros de datos como:

```
habitacion1.largo = 5.5;
```

En este caso, inicializa la variable **largo** de **habitación1** en **5.5**

3.3. Ejemplo: Objeto y clase en programación C++

```
// Programa para ilustrar el funcionamiento de
// objetos y clases en la programación en C++

#include <iostream>
using namespace std;

// crea una clase (class)
class Habitacion
{
   public:
    double largo;
```

```
    double ancho;
    double altura;

    double calcula_area()
    {
        return largo * ancho;
    }

    double calcula_volumen()
    {
        return largo * ancho * altura;
    }
};

int main()
{
    // crea un objecto de la clase Habitacion
    Habitacion habitacion1;

    // asigna valores a los datos miembro
    habitacion1.largo = 42.5;
    habitacion1.ancho = 30.8;
    habitacion1.altura = 19.2;

    // Calcular y mostrar el área y volumen de la habitación
    cout << "Area de la habitacion =  "
         << habitacion1.calcula_area()
         << endl;
    cout << "Volumen de la habitacion =  "
         << habitacion1.calcula_volumen()
         << endl;

    return 0;
}
```

3.4. Producción

```
Área de la habitación = 1309
Volumen de la habitación = 25132,8
```

En el programa, hemos utilizado la clase **`Habitacion`** y su objeto **`habitacion1`** para calcular el **`área`** y el **`volumen`** de una habitación.

En **`main()`**, asignamos los valores de **`largo`**, **`ancho`**, y **`altura`** con el código:

```
room1.largo = 42.5;
room1.ancho = 30.8;
room1.altura = 19.2;
```

Luego llamamos a las funciones **`calculate_area()`** y **`calculate_volume()`** para realizar los cálculos necesarios.

Tenga en cuenta el uso de la palabra clave **`public`** en el programa. Esto significa que los miembros son públicos y se puede acceder a ellos desde cualquier lugar del programa.