



Unidad V Manejo de clases

M.C. Juan Carlos Olivares Rojas



Agenda

- 5.1 Fundamentos de las clases
- 5.2 Datos públicos y privados
- 5.3 Funciones constructor y destructor
- 5.4 Sobrecarga de operadores
- 5.5 Funciones estáticas y miembros estáticos

5.1 Fundamentos de las clases

- Las clases son el elemento básico en el cual se desarrolla la programación orientada a objetos.
- Las clases son el tipo de datos en las cuales se declaran los objetos.
- Un objeto es la instanciación de una clase. Es decir un objeto es una clase en memoria.

Clases

- Las clases se parecen mucho a las estructuras en el sentido de que pueden almacenar datos distintos. La diferencia es que las clases pueden almacenar además procedimientos.
- La otra diferencia radica en la forma en como son visibles los elementos de una clase.

Clases

```
class auto
{
    char nombre[];
    unsigned short tipo;
    unsigned short color;

    void arrancar();
    unsigned short parar();
    float gasolina();
}
```

Clases

- Para declarar un objeto, este debe hacerse dentro de un bloque de programa. Es necesario crear ese objeto para que pueda ser utilizable.
- `main()`
- `{`
- `auto honda;`
- `honda = new auto();`
- `}`

Clases

- Los elementos que puede contener una clase son de dos tipos propiedades (tipos de datos, características del objeto) y métodos (funciones, comportamiento de los objetos).
- En general los métodos se definen dentro de una clase pero no se implementan (aunque podrían hacerlo). Esto con el objetivo de hacerlo más portable.

Clases

- Para acceder a los elementos se utiliza el operador `.` y `-->` en caso de apuntadores.
- Para acceder al elemento gasolina:
`honda.gasolina(); honda.color`
- Pero esto nos marcaría un error el compilador ¿por qué?

5.2 Datos públicos y privados

- Por que los miembros de una clase son privados de manera general.
- Para acceder a datos privadas se utilizan métodos para acceder a los elementos. Los métodos más utilizados son set y get.
- Los métodos set colocan un valor y los métodos get obtienen un valor;

Datos públicos y privados

Class clase

{

private:

int a;

public:

getA();

setA();

}

- Clase C;
- int f = C.getA();
- C.setA(10);

5.3 Funciones constructor y destructor

- Las funciones constructor y destructor sirven para configurar de manera predeterminada el funcionamiento de un objeto.
- Las funciones constructor se ejecutan antes de crearse el objeto por lo que se utilizan para inicializar propiedades del objeto

Funciones constructor y destructor

- Una función constructor lleva el mismo nombre de un clase pero sin devolver ningún tipo de datos ni void.
- Class clase
- {
- public:
- **clase()**
- {
- código del constructor
- }
- }

Funciones constructor y destructor

- El constructor de un objeto nunca se llama por el usuario. Se inicializa cuando se crea un objeto. Objeto = new Clase();
- El destructor se utiliza cuando un objeto finaliza su vida ya sea de manera implícita o explícita. Generalmente se utiliza para limpiar memoria asignada al objeto.

Funciones constructor y destructor

- Un destructor se define con el nombre de la clase sin devolver tipo (ni void) generalmente no acepta parámetros pero puedo hacerlo al igual que el constructor.
- `class clase {`
- `public`
- `~clase() { // código del destructor}`
- `}`

5.4 Sobrecarga de operadores

- La sobrecarga además de funciones puede realizarse a través de operadores.
- Por ejemplo el operador aritmético $+$, está sobrecargado para poder realizar sumas con cualquier tipo de datos numérico.
- Los operadores se sobrecargan con la palabra reservada `operator` seguido del operados

Sobrecarga de operadores

- El operador + es un operador binario por lo que se necesitarían dos argumentos. Por ejemplo:
operator + (int a, int b);
- Dentro del método operator sobrecargado realizamos la lógica en nuestro caso se utilizará para concatenar cadenas.
- Los operadores sobrecargados para cada clase son =.

Sobrecarga de operadores

- Casi todos los operadores del lenguaje se pueden sobrecargar. El detalle radica en saber si son operadores unarios, binarios o ternarios y saber como se pasa el primer argumento.
- En algunos operadores binarios el primer argumento se puede omitir para utilizar alguna propiedad de un objeto.

5.5 Funciones estáticas y miembros estáticos

- Las funciones estáticas son aquellas que se pueden ejecutar sin instanciar un objeto de una clase.
- Las funciones estáticas radican siempre en memoria.
- `static clase::calularEdad(void);`

Funciones y variables estáticas

- Las variables estáticas también residen en memoria. La característica principal radica en el hecho de que son globales a todos los objetos de la clase.
- `static int voip =5;`
- Todos los objetos comparten el mismo valor. Si se cambia el valor siguen con el nuevo valor.

¿Preguntas?

