

# **CAPÍTULO 3**

## **CLASES Y OBJETOS**

- 3.1 Definición de clases y objetos**
- 3.2 Constructores, destructores y gestión de memoria**
- 3.3 Miembros de clases estáticos**
- 3.4 Manipulación de objetos por referencia y por valor.**

## 3.1 DEFINICIÓN DE CLASES Y OBJETOS

### CLASE.

Se pueden declarar variables, propiedades, , métodos, etc.  
Cada elemento puede tener un modificador de acceso.

**private**(solamente en la clase puede ser utilizado el exterior no lo conoce)

**public**(todos lo pueden utilizar).

**protected**(los que tienen permiso de acceso pueden utilizar lo que tiene la clase)

POR DEFECTO TODOS LOS ATRIBUTOS SON PRIVADOS, Y PARA LOS METODOS DEBEN ESPECIFICARSE.

# EJEMPLO: VARIABLES Y MODIFICADORES

```
class Ejemplo
```

```
{
```

```
    int nro; //sino se indica, es private automáticamente.
```

```
    private int edad;
```

```
    public string nombre;
```

```
    protected float peso;
```

```
};
```

Ámbito de objeto (variables que están dentro de la clase)

# EJEMPLO: MODIFICADORES Y MÉTODOS

```
void ArregloAtleta::llenar()  
{  
    int g,e;  
    for (int i=0;i<tam*2;i=i+2)  
    {  
        g = 1+rand()% (2+1-1); // valores de 1 a 2  
        e = 18+ rand()%(100+1-18); // de 18 a 100  
        arreglo[i]=g;  
        arreglo[i+1]=e;  
    }  
}
```

Ámbito local (variables que están dentro del método)

## 3.1 DEFINICIÓN DE CLASES Y OBJETOS

Un objeto es una instancia de una clase, creada en tiempo de ejecución y formada por tantos campos como atributos tenga la clase.

NombreClase nombreObjeto1 (lista de parámetros/ sin parámetros);

NombreClase nombreObjeto2 (lista de parámetros/ sin parámetros);

NombreClase nombreObjeto3 (lista de parámetros/ sin parámetros);

Etc,etc.

**Con parámetros se usa constructor.**

**Sin parámetros no se necesita constructor, este por defecto es vacío.**

## 3.2 CONSTRUCTOR

- Método especial que es invocado(llamado) automáticamente cada vez que se crea un objeto.
- Debe tener el mismo nombre que el de la clase.
- No posee tipo de dato de retorno (no es void ni función con return)
- Si la clase no tiene constructor el c# crea uno vacío por defecto.
- Es empleado muchas veces para inicializar los valores de los atributos.

## 3.2 CONSTRUCTOR

```
class Atleta
{
    int gen, edad;    // 1 = hombre
public:
    // constructor
    Atleta(int g, int e);
    void imprimir();
private:
    float pulsaciones();
};
Atleta::Atleta(int g, int e)
{
    gen = g;
    edad = e;
}
```

Lleva el mismo nombre que la clase y recibe por parámetro los valores para los atributos



## 3.2 CONSTRUCTOR

```
class ArregloAtleta
{
    int *arreglo;
    int tam;
public:
    ArregloAtleta(int dim);
    void llenar();
    void imprime();
    float pulsaciones(int p);
};

ArregloAtleta::ArregloAtleta(int dim)
{
    arreglo = new int[dim*2];
    tam=dim;
}
```

Lleva el mismo nombre que la clase y recibe por la dimensión del arreglo



## 3.2 DESTRUCTOR

- Se invoca automáticamente cuando el objeto es destruido
- No es necesario escribir código para el.
- La memoria se libera automáticamente al cerrar la ejecución del programa.

### 3.3 MIEMBROS ESTÁTICOS

- No requieren el uso de un CONSTRUCTOR.
- En cada método se usa la palabra **static**.
- No requiere la creación de un objeto, se llama al método como si fuese un módulo creado por el main.
- Los valores se envían por parámetro.

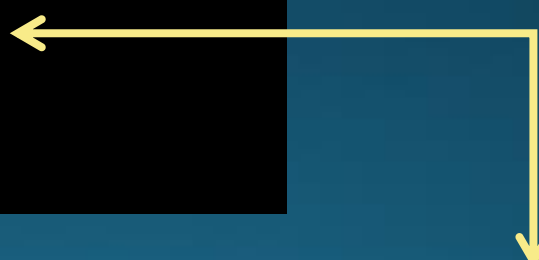
```
cout<< "Altura = "<<alt<<endl;
imprimeIMC(pes,alt);
cout<<"Escreva 1- segue"
```

## 3.3 MIEMBROS ESTÁTICOS

```
using namespace std;
class CalculaIMC
{    //sin atributos
    // sin constructor
    private:
        static float imc(float peso,float altura);
    public:
        static void imprimeIMC(float peso,float altura);
};

static float imc(float peso,float altura)
{    return (peso/(altura*altura));
}

static void imprimeIMC(float peso,float altura)
{    cout<<"Su IMC = "<<imc(peso,altura)<<endl;
}
```



```
cout<<"Altura = "<<alt;
imprimeIMC(pes,alt);
cout<<"Escriba 1 = para salir"
```

# EJERCICIOS

1. Calcular el numero de pulsaciones que debe tener una persona por cada 10 segundos de ejercicio aeróbico; la formula que se aplica cuando son hombres o mujeres es:

Para hombres:  $\text{NroPuls} = (210 - \text{edad}) / 10$

Para mujeres:  $\text{NroPuls} = (220 - \text{edad}) / 10$

Realizar un programa para calcular las pulsaciones de 4 personas.