

CLASES Y OBJETOS

AYUDANTIA 1

DIEGO MENA



Clases

Una clase puede interpretarse como una colección de variables (Atributos) y funciones (Métodos), los cuales en si permiten representar un conjunto de datos y especificar las operaciones o procedimientos que permiten manipular tales datos.



Ejemplo de clases: “Perros”.



Atributos (características)

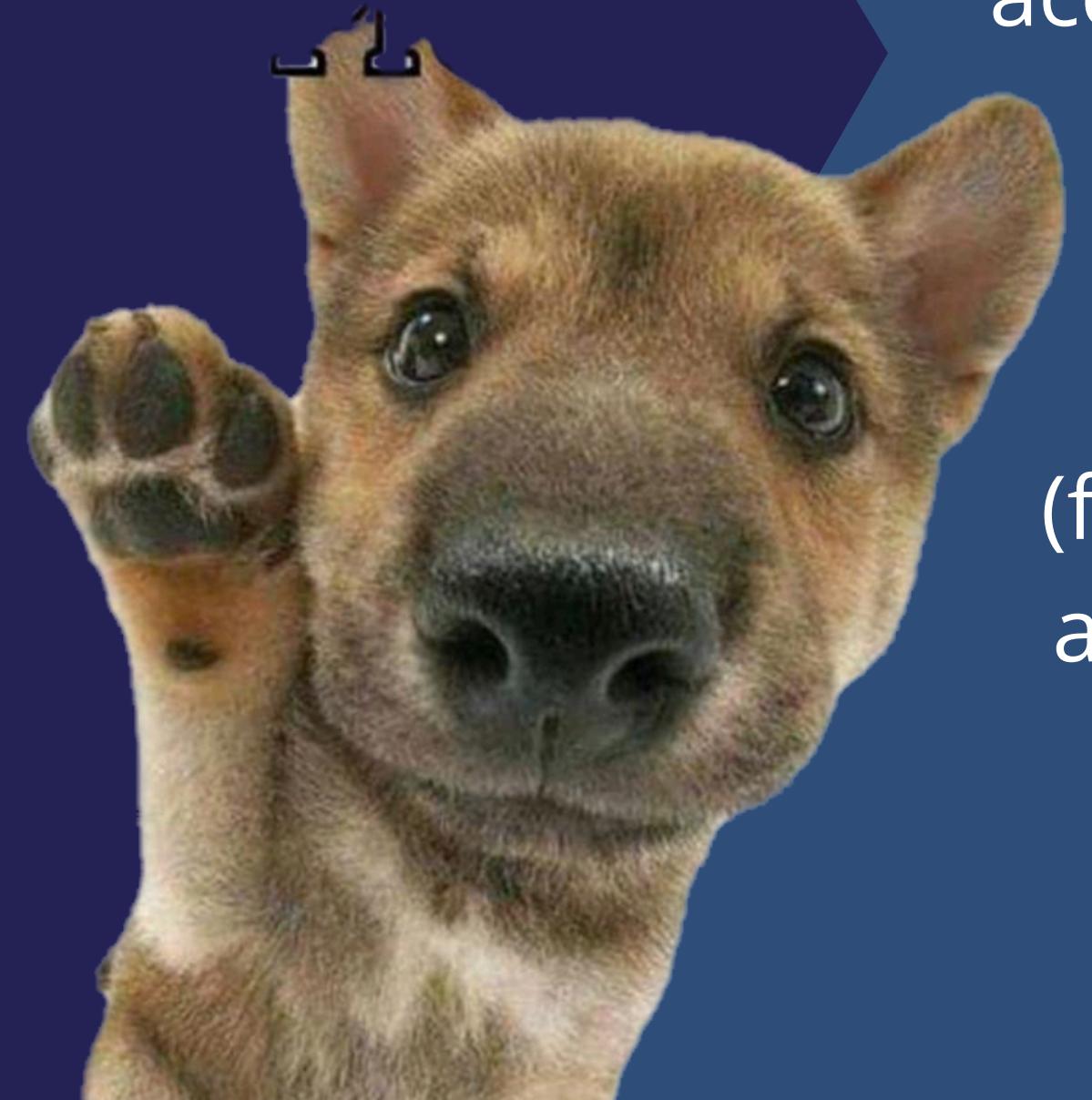
Nombre	Color
Peso	Raza
Dueño	Tamaño

Métodos (Funciones)

Ladrar()	Jugar()
Dormir()	
Comer()	

Sintaxis de las clases

```
Class nombredelaclase  
{  
    private:  
    //atributos  
    public:  
    //Métodos  
}
```



Accesos

Acceso privado: En este apartado irán los atributos, los cuales solo podrán ser accesibles para la misma clase.

Acceso público: En este apartado irán los métodos (funciones). Estos podrán ser accesibles tanto para dentro como afuera de la clase.

Ejemplo de la clase Perros



```
#include <iostream>
using namespace std;
class perros
{
private:
    int edad;
    string nombre;
    string dueño;
public:
    void Ladrar()
    {
        cout<<"woof woof" << endl;
    }
    void dormir()
    {
        cout<<"zzzzzzzzzz" << endl;
    }
};

int main()
{
}
```

Objetos

Un objeto es una instancia de una clase. Esto significa que es una entidad creada a partir del modelo de esta, básicamente un objeto pertenece a una clase, en donde gracias a esto esta puede acceder a los datos y funciones definidos por la esta misma.



Sintaxis para objetos

Class perro

```
{  
...  
};  
int main()  
{  
nombreClase *nombreObjeto = new nombreClase;  
}
```

Ejemplo para perros



Nombre: Tejo
Edad: 5 años
Dueño: Aceituno

Nombre: Alfredo
Edad: 7 años
Dueño: Xavi



```
#include <iostream>
using namespace std;
class perros
{
private:
    int edad;
    string nombre;
    string dueño;
public:
    perros(int Edad, string Nombre, string Dueno)
    {
        this->edad=Edad;
        this->nombre=Nombre;
        this->dueño=Dueno;
    }
    void setNombre(string nuevoNombre)
    {
        nombre=nuevoNombre;
    }
    string getNombre()
    {
        return nombre;
    }
};
int main()
{
    perros *objeto = new perros(4,"Kiara","Mena");
    cout<<objeto->getNombre()<<endl;
    objeto->setNombre("Deisy");
    cout<<objeto->getNombre()<<endl;
}
```

Constructor

El constructor sirve para inicializar un objeto de una clase específica, declarando además a su paso las variables.

Set

El método set, se ocupa para poder modificar el valor de una variable, desde fuera de la clase.

Get

El metodo get, sirve para obtener el valor contenido por un atributo de la clase, desde fuera de este.

Objetos multiples

A la hora de la creación de objetos, se pueden generar objetos uno a uno, como también se puede hacer uso de los ya conocidos arreglos, para así dejar una sola “variable”, la cual contenga en su interior varios objetos.

```
3 perros *Arreglo[3];
4 Arreglo[0]=new perros(1,"Deisy","Araceli");
5 Arreglo[1]=new perros(2,"Nina","Britanic");
6 Arreglo[2]=new perros(3,"Loki","Miguel");
7 for(int i=0;i<3;i++)
8 {
9     cout<<Arreglo[i]->getNombre()<<endl;
10 }
```

Ejercicio

Clase

- Implemente un programa para una peluqueria, en la cual permita crear un listado de perros.
- Cada perro debe tener un nombre, su edad y el nombre de su dueño.
- Se necesita un método el cual calcule el promedio de las edades de los perros.
- También para un post-análisis de la peluqueria se busca saber cual es el perro mas viejo.

Main

- **El main debe pedir la cantidad de perros en la lista y el ingreso de datos de estos.**
- **Además, de la impresión de los métodos hechos en la clase.**



Contacto



+56 9 6070 8865



diego.mena2@mail_udp.cl



<https://github.com/Dmena1/Ayudantia-prograAvanzada>



dmena