Programación Orientada a Objetos

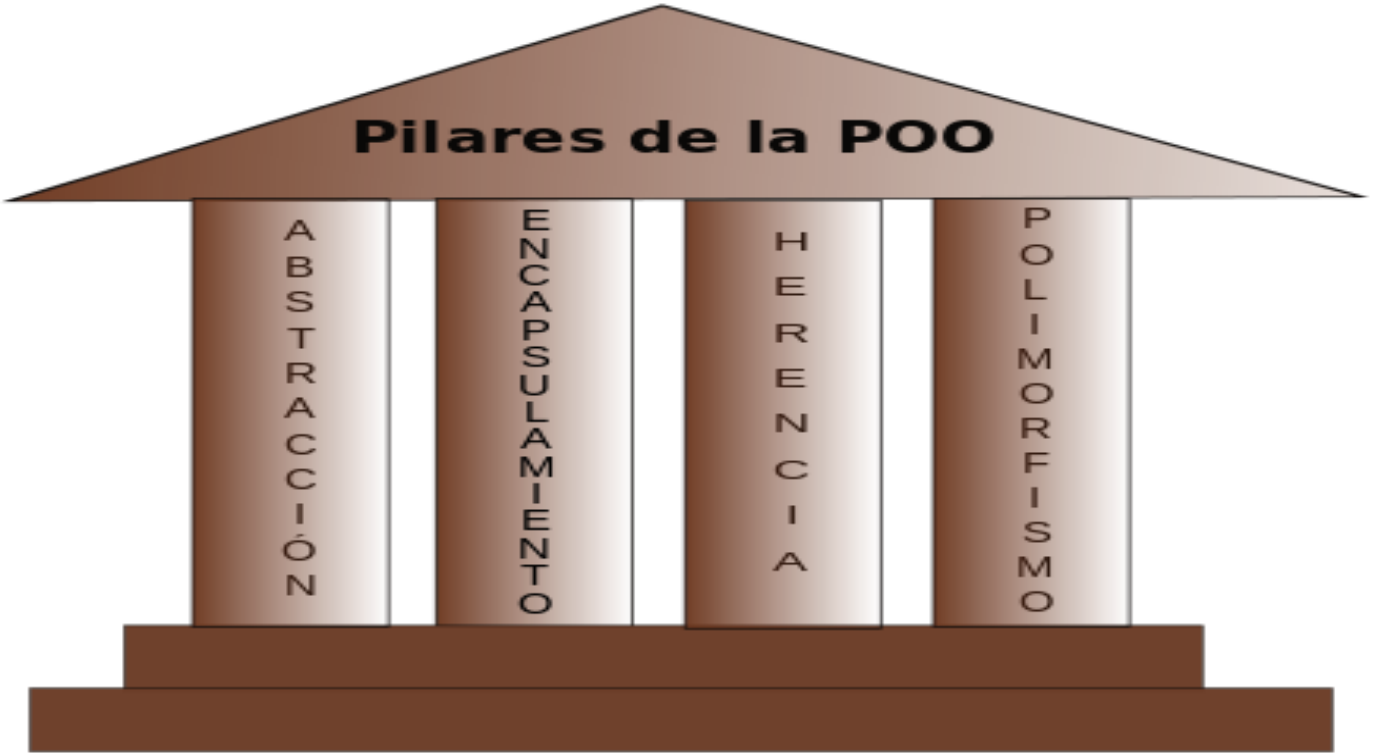
Es un paradigma de programación que usa objetos y sus interacciones, para diseñar aplicaciones y programas de computadora.

Es una forma especial de programar, más cercana a como expresaríamos las cosas en la vida real.

Se basa en la idea natural de la existencia de un mundo lleno de objetos, de modo que la resolución del problema se realiza en términos de objetos.

Programación Orientada a Objetos

Pilares Fundamentales de la POO



Programación Orientada a Objetos

Abstracción

- Consiste en captar las características esenciales de un objeto, así como su comportamiento, al mismo tiempo que se ignoran los detalles no esenciales.
- En programación, el término se refiere al énfasis en el "**¿qué hace?**" más que en el "¿cómo lo hace?".

Ejemplo: el proceso para calcular el promedio de 3 números puede explicarse así:

Abstracción: ¿Qué es y que Hace?

i. *Sumamos los números y dividimos entre 3*

Programación Orientada a Objetos

No es Abstracción: ¿Cómo debe Implementarse?

i. *Declarar variables*
ii. *Mostrar un mensaje en pantalla para pedir cada número*
iii. *Leer los números*
iv. *Asignarlos la suma a una variable Total*
v. *Dividir el total entre 3 y asignarlo a la variable Promedio*
vi. *Mostrar un mensaje en pantalla indicando que se presentará el resultado*
vii. *Mostrar la variable Promedio.*

Programación Orientada a Objetos

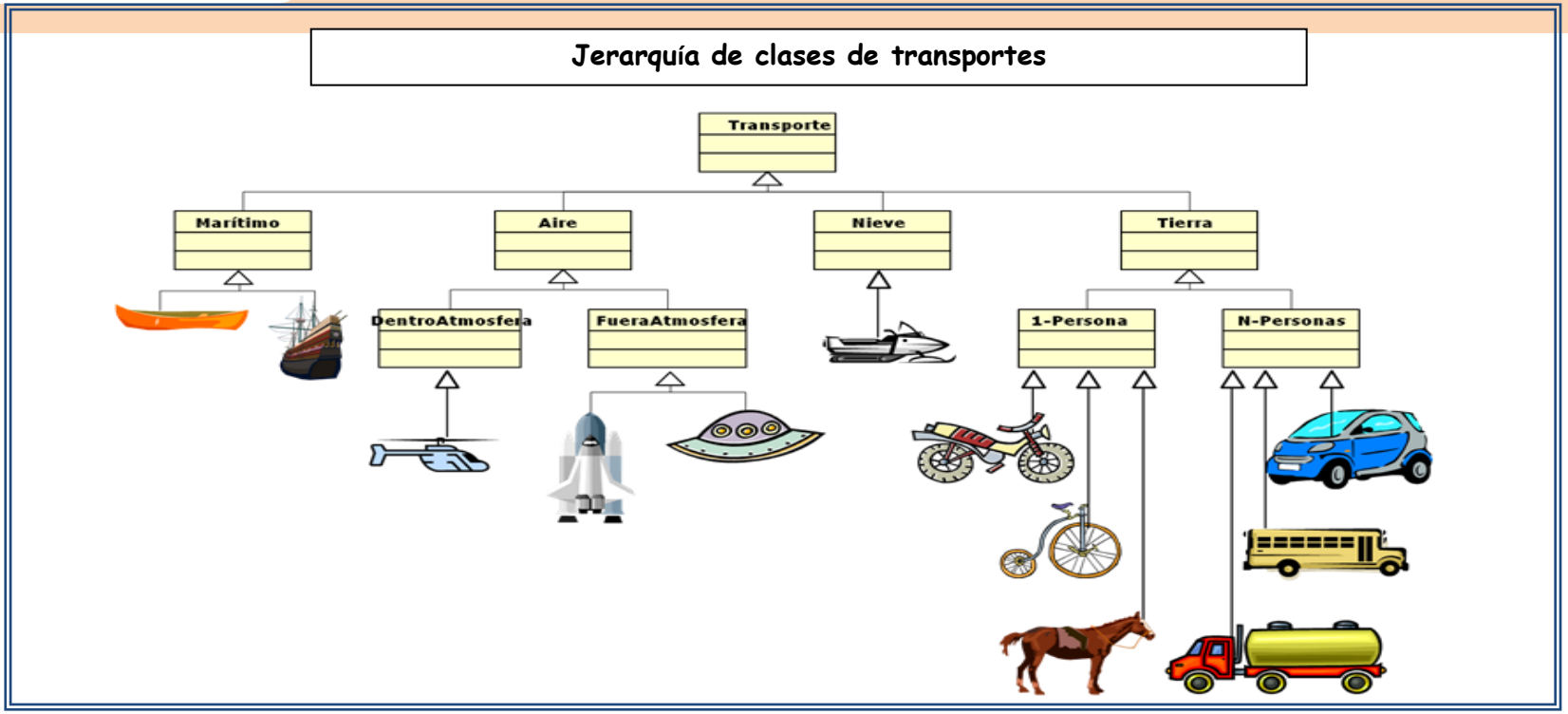
Encapsulamiento

- Es una propiedad que ayuda a mantener juntos, en una única entidad, los atributos o propiedades (datos) y las funciones (métodos) que definen el comportamiento del objeto.

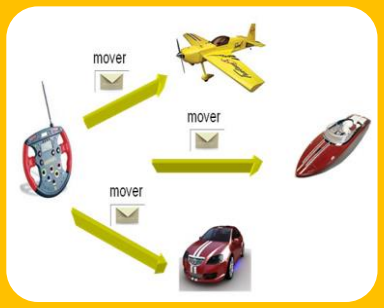
Programación Orientada a Objetos

Herencia

- Es una propiedad que permite que los objetos sean creados a partir de otros ya existentes, obteniendo características (métodos y atributos) similares a los ya existentes.

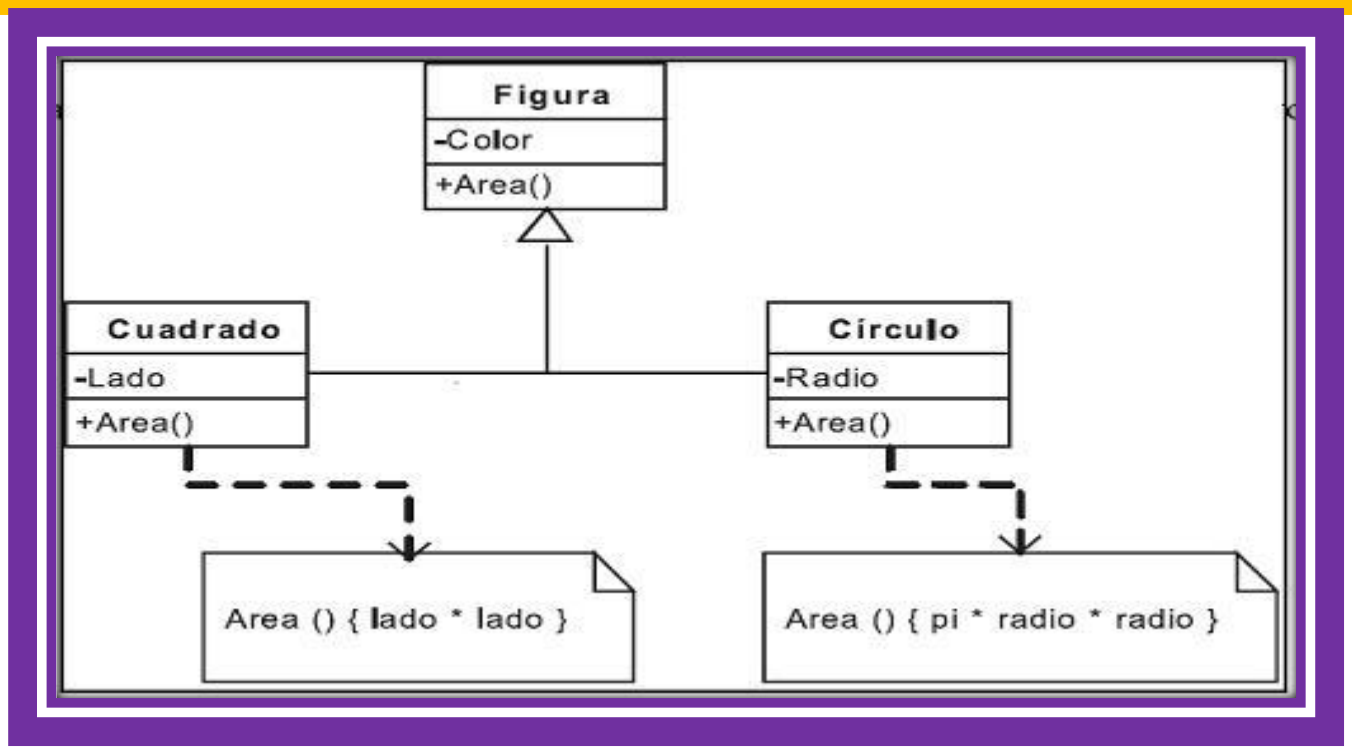


Programación Orientada a Objetos



Polimorfismo

- Consiste en la posibilidad de definir en una jerarquía de clases métodos y/o atributos denominados de forma idéntica, pero que se comportan de manera distinta.



Programación Orientada a Objetos

¿Qué es un Objeto?

- Todo objeto del mundo real tiene 2 componentes: **características (atributos) y comportamiento (métodos).**

Según Booch

- Es algo que tiene estado, un comportamiento y una identidad.

¿Cuáles son sus Atributos?

¿Cuáles son sus Métodos?

Programación Orientada a Objetos

¿Qué es un Clase?

- Es la descripción de un conjunto de objetos; consta de métodos y atributos (datos) que resumen características comunes de un conjunto de objetos.
- Cuando programamos un objeto y definimos sus características y funcionalidades en realidad lo que estamos haciendo es programando una clase.

Programación Orientada a Objetos

Elementos de una Clase

Propiedades o atributos

- Son las características de los objetos. Cuando definimos una propiedad normalmente especificamos su **nombre** y su **tipo**.
- Las propiedades son algo así como variables donde almacenamos datos relacionados con los objetos.

Métodos

- Son las funcionalidades asociadas a los objetos, es decir, describen el comportamiento asociado a un objeto.
- Las funciones (métodos) residen en el objeto y determinan como actuará éste cuando reciba un mensaje.

Programación Orientada a Obj

Sintaxis para crear una Clase en C++

```
class NombreClase {
  private:    //Visibilidad o Acceso
    Atributos

  public:   //Visibilidad o Acceso
    Métodos
};
```

Sintaxis para crear un Objeto en C++

NombreClase NombreObjeto;

Sintaxis para enviar un mensaje a un objeto en C++

NombreObjeto.Metodo();

Clases y Objetos: Constructores

Es un método especial que se ejecuta automáticamente al momento de la creación de un objeto; su propósito es la inicialización de los atributos del objeto, bien sea con valores predefinidos o con valores que se "pasan" al objeto a través de parámetros.

Consideraciones

- Se denominan exactamente igual que la clase.
- Pueden haber varios constructores en una clase, con la diferenciación entre ellos de los argumentos que reciben (sobrecarga de métodos).
- Para efectos de este curso, principalmente se usarán **2 constructores**: (valores predeterminados, y otro que recibe los argumentos de inicialización).
- En caso de que sea un constructor de inicialización predeterminado, colocarás valores base.

Clases C++: Setters y Getters

La función **Set** es un método que nos permite modificar el valor (asignar) de una variable privada (atributo) de una clase.

La función **Get** es un método que nos permite visualizar el valor de una variable privada (atributo) de una clase, en otras palabras provee el valor del atributo a quien lo solicite.

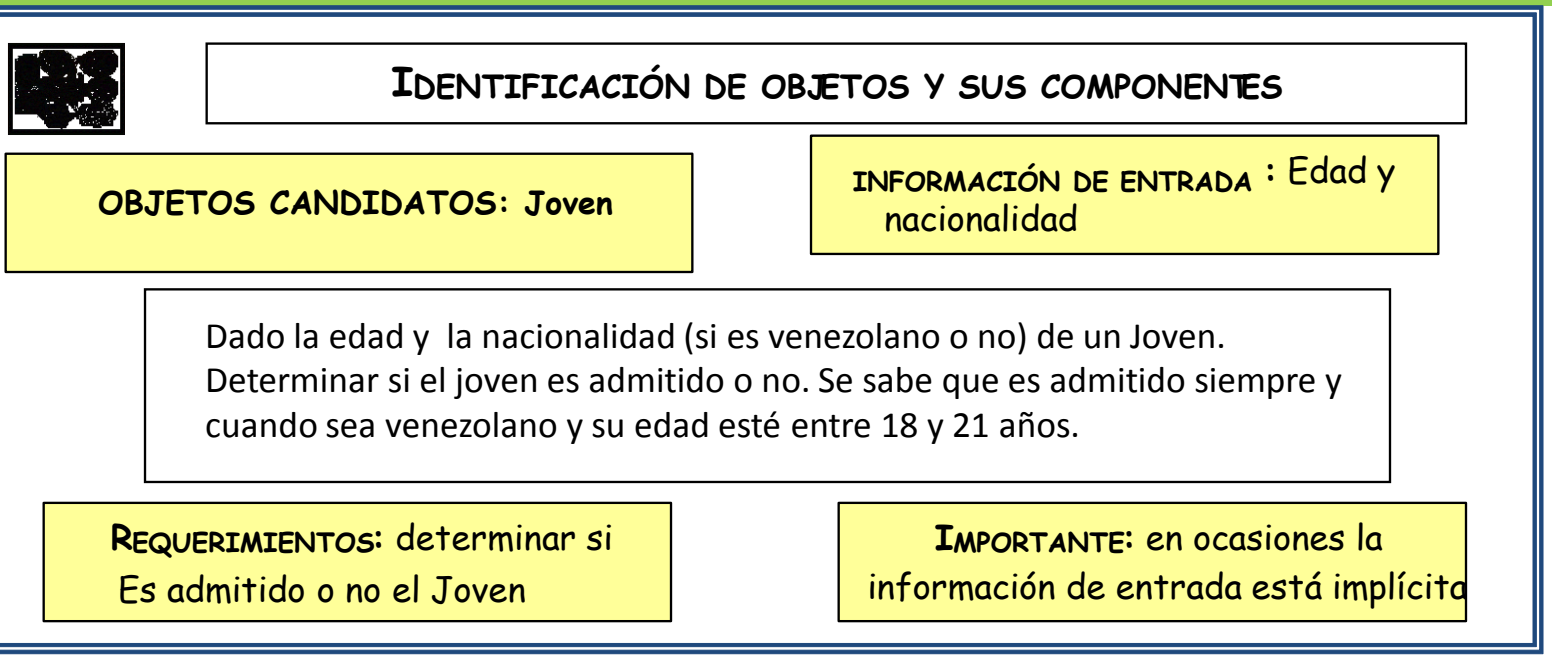
Consideraciones

- Por **cada atributo** debería existir un Set y un Get.
- Los Sets constituyen un método cuya única función será asignarle un valor a un atributo en particular, el cual viene por parámetro. Dado que no tiene que retornar valor, los Sets son procedimientos, o funciones **void**.
- Los Gets son métodos que se invocarán para que retornen el valor de una atributo en particular, de manera que el tipo de retorno de cada Get dependerá del tipo del atributo.

Identificación de Objetos

POO
Objective C

- Una técnica a seguir para lograr la identificación de los objetos es subrayar cada sustantivo (nombre) presente en el planteamiento del problema.
- ***Ejemplo:*** Dado la edad y la nacionalidad (si es venezolano o no) de un Joven. Determinar si el joven es admitido o no. Se sabe que es admitido siempre y cuando sea venezolano y su edad esté entre 18 y 21 años.



Programación de Objetos en C++

Un Programa Orientado a Objetos en C++, está compuesto por:

- Varias librerías de uso genérico, bien sea las incorporadas al **lenguaje** o programadas por el **usuario**.
- Una carpeta en el disco duro (o pendrive) con el nombre del proyecto.
- Clases propias para el programa. Cada clase se almacena en 2 archivos: uno para la **interfaz** (extensión .h) y otro para la **implementación** (extensión .cpp). Estos archivos van dentro de la carpeta del proyecto

Cada clase que se diseña tendrá 2 partes:

- La **interfaz**: contiene solamente la declaración de los componentes de la clase, siguiendo la sintaxis de C++.
- La **implementación**: contiene la programación de los métodos de la clase