

PARADIGMA DE PROGRAMACIÓN ORIENTADO A OBJETOS

Ing. Edgar Gerardo Salinas Gurrión

POO

- Es un paradigma de programación centrada en los objetos. Pretende que se facilite el mantenimiento de software a través del ocultamiento de la información y la encapsulación.
- ¿Qué es un objeto?
- Operaciones comunes
 - ▣ Crear nuevos objetos
 - ▣ Copiar objetos o comprobar su igualdad
 - ▣ Realizas operaciones de E/S con esos objetos.



Paradigma Orientado a Objetos - POO

- Un programa está organizado en base a Objetos
- Los Objetos están contenidos en una clase
- Un Objeto es una instancia de la clase y es un elemento independiente con:
 - ▣ Características propias - atributos
 - ▣ Realiza tareas específicas
- Ejemplos de Clases
 - ▣ Persona
 - ▣ Silla
 - ▣ Carro



Objetos

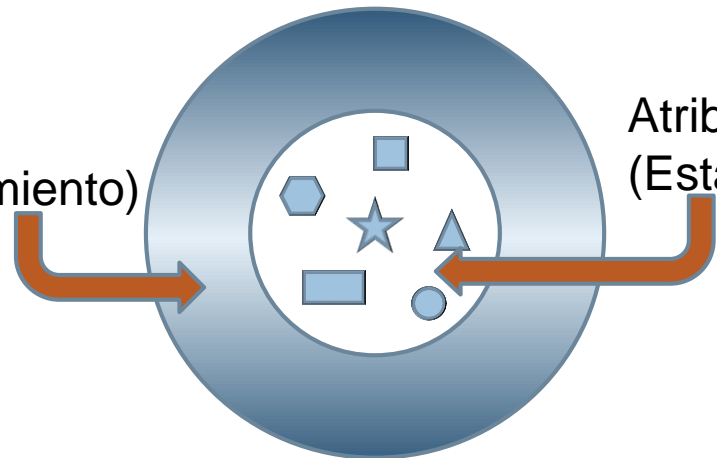
- Los objetos son entes, entidades, sujetos o cosas que encontramos en el dominio del problema de nuestra realidad.
- El mundo real está lleno de objetos y estos tienen 2 características.

- ▣ Estado

- ▣ Comportamiento

Métodos
(Comportamiento)

Atributos
(Estado)



Abstracción

- Considerar los aspectos más representativos de un problema y generar una soluciones en esos términos.
- Pensar en el qué y no en el cómo.

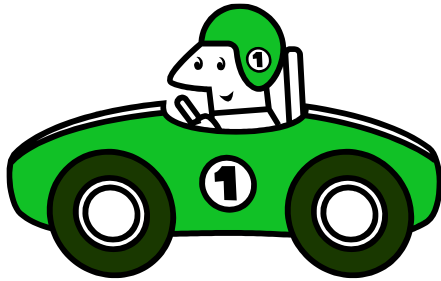


Clase

- Una clase es el modelo que define como está conformado un Objeto y a través de ella los objetos son creados.
- Los atributos almacenan la información inherente a la clase.
- Los métodos son las acciones que un objeto puede realizar. Regularmente modifican y acceden a los atributos del objeto.



Clases



```
class Carro {  
    double tamaño;  
    String marca;  
    int puertas;  
  
    void camina () {  
    }  
}
```

Atributos

Métodos



Objetos



```
Carro miCarro = new Carro();  
miCarro.tamaño = 3.00;  
miCarro.marca = "VW";  
miCarro.puertas = 4;  
  
miCarro.camina();
```

- Objeto es la instancia de una clase
- Objeto define los valores para cada atributo
 - ▣ tamaño = 3.00;
 - ▣ marca = "VW";
 - ▣ puertas = 4;
- Un Objeto manda llamar a los métodos para realizar acciones



Encapsulamiento

- El *Encapsulamiento* es la capacidad que tienen los objetos de agrupar métodos y variables ocultando los detalles de implementación al usuario del objeto. Esto ofrece principalmente 2 beneficios:
 1. Modularidad
 2. Mejor administración de la información

```
class Carro {  
    int tamaño;  
    String marca;  
    int puertas;  
  
    void camina () {  
    }  
}
```



Polimorfismo

- “En programación orientada a objetos se denomina **polimorfismo** a la capacidad que tienen objetos de diferentes clases de responder al mismo mensaje. Esto significa que puede haber muchos mensajes con el mismo nombre, en diferentes clases. Cada Clase responde al mensaje con su código propio (o método).”

Wikipedia

`miCarro.camina();`



`persona.camina();`

`tux.camina();`



Mensajes

- Los objetos interactúan entre si sólo a través del paso de *mensajes*.

- Un *mensaje* está compuesto por tres partes:
 1. El **nombre del objeto** al que se desea enviar el mensaje.
 2. El **nombre del método** que se desea ejecutar.
 3. Cualquier **parámetro** que se necesite para ejecutar el método.



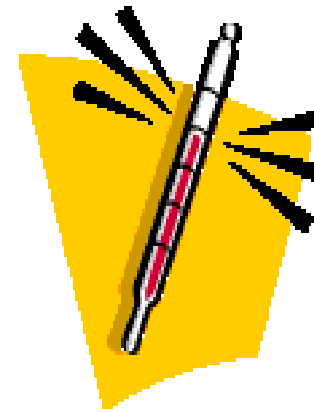
Constructores

- Son los métodos que se utilizan para crear nuevos objetos.
- Su principal función es la de reservar la memoria necesaria para almacenar el objeto e inicializar correctamente los atributos de un objeto.
- Los constructores siempre tienen el mismo nombre de la clase y nunca devuelven nada por lo tanto **NUNCA** se pone en la firma del constructor algún tipo de dato a regresar.
- Podemos tener más de un constructor y se llamará al que correspondan los parámetros que se pasen.
- El constructor se llama junto con el operador new y sólo se puede llamar para crear un nuevo objeto.



Actividad: Clase Termómetro

- Atributo:
 - temp.
- Métodos:
 - Constructor default
 - Constructor con parámetro
 - setTemp
 - getTemp



Tarea Clase Libro

□ Atributos

- ▣ titulo
- ▣ autor
- ▣ editorial
- ▣ isbn

(Todos son de tipo String)

□ Métodos

- ▣ Constructor por default
- ▣ Constructor con parámetros(Recibe los 4 valores titulo, autor, editorial e isbn)
- ▣ Setters
- ▣ Getters

Un setter y getter por cada atributo.

