



Temas Puntuales:

Objetos ,Clases, atributos, métodos, mensaje. Etc.

Concepto de POO:

La programación orientada a objetos es un enfoque de programación que tiene un estilo en que cada programa es visto como un objeto se forma por una serie de componentes, auto contenidos que cooperan para realizar las acciones de la aplicación completa.

Diferencias con la Programación Estructurada.

- Las principales **diferencias** entre la **programación estructurada y la orientada a objetos** son:
- La programación orientada a objetos es más moderna, es una evolución de la programación estructurada
- La programación orientada a objetos se basa en lenguajes que soportan sintáctica y semánticamente la unión entre los tipos abstractos de datos y sus operaciones (a esta unión se la suele llamar *clase*).

La programación orientada a objetos incorpora en su entorno de ejecución mecanismos tales como el polimorfismo y el envío de mensajes entre objetos.

Video explicativo Prof. Jorge I. Fumarola. <https://youtu.be/dF0LxyRRiq4>

Conceptos orientados a objetos

- Objeto
- Clase
- Método
- Evento
- Mensaje
- Propiedad o atributo
- Componentes de un objeto
- Representación de un objeto



Características de la POO:

Las características mas importantes son:

- Abstracción
- Encapsulamiento
- Principio de ocultación
- Polimorfismo (lo trataremos en la 2da parte de esta Unidad)
- Herencia

Clases y Objetos:

- **Objeto** es la entidad provista de un conjunto de propiedades o atributos (datos) y de comportamiento o funcionalidad (métodos). Corresponden a los objetos reales del mundo que nos rodea, o a objetos internos del sistema (del programa). Es una instancia a una clase.

Componentes de un objeto;

Los componentes de un objeto son:

- atributos,
- identidad,
- relaciones
- y métodos.

Un objeto se representa por medio de una tabla o entidad que esté compuesta por sus atributos y funciones correspondientes y el tipo del objeto puede ser definido mediante una Clase o una interface.

```
Public class Alumno { Nombre de la Clase
```

```
String nombre; ← Atributos
```

```
Int edad; ←
```

```
String domicilio; ←
```



Las Clases son definiciones de las propiedades y comportamiento de un tipo de objeto concreto. La instanciación es la lectura de estas definiciones y la creación de un objeto a partir de ellas.

Video tutorial gentileza Prof. Daniel Balbi.

<https://www.youtube.com/watch?v=QVcZ8nFzyXY&t=405s>

Método:

- Es un algoritmo asociado a un objeto (o a una clase de objetos), cuya ejecución se desencadena tras la recepción de un "mensaje". Desde el punto de vista del comportamiento, es lo que el objeto puede hacer. Un método puede producir un cambio en las propiedades del objeto, o la generación de un "evento" con un nuevo mensaje para otro objeto del sistema.

Método Constructor es una función, método, etc, de las clases, la cual es llamada automáticamente cuando se crea un objeto de esa clase.

- Por ser métodos, los constructores también aceptan parámetros. Cuando en una clase no especificamos ningún tipo de constructor, el compilador añade uno público por omisión sin parámetros, el cual NO hace nada.

Características de los Constructores

1. Un constructor, tiene el mismo nombre de la clase a la cual pertenece.
2. No puede ser Heredado.
3. No retorna ningún valor (Ni void), por lo cual no debe especificarse ningún tipo de dato.
4. Debe declararse como public, sólo en casos realmente extraordinarios será de otro tipo.

Supongamos una clase llamada Datos

Public class Datos

```
{  
//declaracion de atributos  
Public Datos();//constructor sin parámetros  
Public Datos(int Edad)  
{//Constructor con parámetros
```



```

    EdadM=Edad;
}
}

```

Ejemplo.

- Se Desea crear un objeto del tipo Coordenadas, que indique, la posición de un punto en el plano cartesiano, y permita calcular la distancia existente con otro punto, además del punto medio entre ellos.

```

1 /*Archivo Coordenadas.java*/
2
3 public class Coordenadas {
4     private int x,y;//Atributos
5     public Coordenadas (int x1 , int y1)//constructor
6     {
7         //asignamos esos valores
8         x=x1;
9         y=y1;
10    }
11    public double Distancia (int x1, int y1)
12    {
13        //este método recibe como parametros
14        //las coordenadas del segundo punto
15        double D;
16        D=Math.sqrt((x-x1)*(x-x1)+(y-y1)*(y-y1));
17        return D;//Retornamos el valor de la distancia
18    }
19    public void Pmedio(int x1, int y1)
20    {
21        double p1, p2;
22        p1=(x+x1)/2;
23        p2=(y+y1)/2;
24        System.out.println("El puntomedio es:
"+p1+", "+p2);

```



```
25     }
26 }
```

```
1 /*Archivo UsaCoordenadas.java*/
2 public class UsaCoordenadas {
3     public static void main (String args []){
4         Coordenadas P=new Coordenadas (10,20);
5         //calculamos la Distancia:
6         System.out.println("La Distancia es: "+P.Distancia(5,6));
7         //Punto Medio
8         P.Pmedio(5,6);
9     }
10 }
```

Explicación:

Un método, si va a devolver algún valor, éste debe especificarse cuando se hace la definición del mismo, para el caso, el método Distancia, devuelve el valor de la distancia, es por ello que lo especificamos como *public double*. Además que, cuando realizamos la llamada de ese método, es en ese preciso lugar que aparecerá el valor que devuelve. Por otra parte, los atributos, son de acceso privado, lo cual implica que, sólo podemos acceder a ellos, en la clase Coordenadas

Video tutorial gentileza Prof. Daniel Balbi. Sobre armado de un método e instanciar un método cómo dijimos en la explicación de Objeto.

<https://www.youtube.com/watch?v=MMl8GlaVVul&t=7s>

Evento:

- Es un suceso en el sistema (tal como una interacción del usuario con la máquina, o un mensaje enviado por un objeto). El sistema maneja el evento enviando el mensaje adecuado al objeto pertinente. También se puede definir como evento, a la reacción que puede desencadenar un objeto, es decir la acción que genera.



Mensaje:

- Es una comunicación dirigida a un objeto, que le ordena que ejecute uno de sus métodos con ciertos parámetros asociados al evento que lo generó.

Propiedad o atributo:

- Es contenedor de un tipo de datos asociados a un objeto (o a una clase de objetos), que hace los datos visibles desde fuera del objeto y esto se define como sus características predeterminadas, y cuyo valor puede ser alterado por la ejecución de algún método.

Estado interno:

- Es una propiedad invisible de los objetos, que puede ser únicamente accedida y alterada por un método del objeto, y que se utiliza para indicar distintas situaciones posibles para el objeto (o clase de objetos).

También tenemos las clases de tipo Abstractas:

Una clase que declara la existencia de métodos pero no la implementación de dichos métodos (o sea, las llaves { } y las sentencias entre ellas), se considera una clase abstracta. Una clase abstracta puede contener métodos no-abstractos pero al menos uno de los métodos debe ser declarado abstracto.

Para declarar una clase o un método como abstractos, se utiliza la palabra reservada **abstract**.

```
abstract class Drawing
{
    abstract void miMetodo(int var1, int var2);
    String miOtroMetodo( ){ ... }
}
```

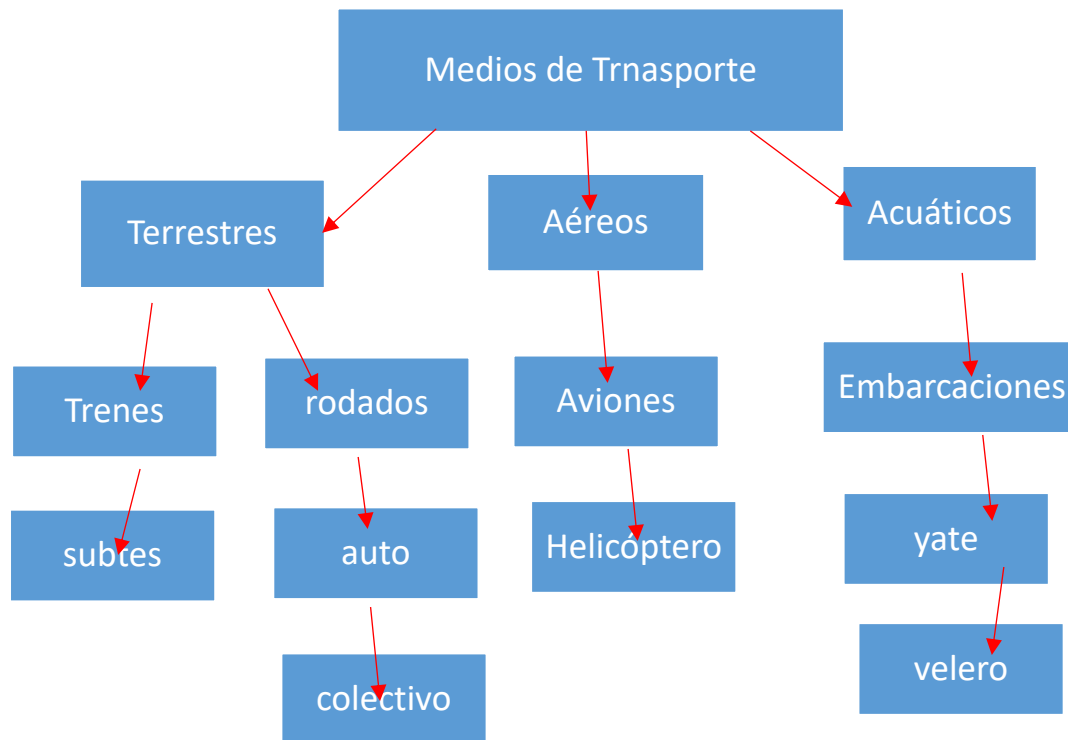
Una clase abstracta no se puede instanciar pero si se puede heredar y las clases hijas serán las encargadas de agregar la funcionalidad a los métodos abstractos. Si no lo hacen así, las clases hijas deben ser también abstractas.



Estas llamadas Clases, generan Jerarquía , que por ejemplo: supongamos la clase medio de Transporte (Sería la **clase Padre o Superclase**) , y luego las clases Terrestres , Aéreos , Acuáticos , las (**Subclases**) , y así vemos en el esquema como se siguen formando las clases de las **subclases**....

Video ejemplo de cómo funcionan las clases en un proyecto terminado que utiliza una base de datos en Mysql. Prof. Jorge Fumarola

<https://youtu.be/LsgFt6pONfY>



Ampliaremos este esquema y conceptos al siguiente Unidad4 Herencia