

Programacion orientado a objetos:

Existen:

Atributos: Cosas que describen el objeto y enseña como es diferente respecto a los otros objetos

Métodos: Son parámetros para añadir más información necesaria para que se ejecute

Cuando tu creas un programa aplicado a objetos, lo que estas haciendo es definir las clases, y cuando tu corres el programa, las instancias son creadas

Ventajas: Reusabilidad, fácil de debugear, modularidad

Abstraccion: “separar por medio de una operación intelectual las cualidades de un objeto para considerarlas aisladamente o para considerar el mismo objeto en su pura esencia o noción

Encapsulación: Este concepto consiste en la ocultación del estado o de los datos miembro de un objeto, de forma que sólo es posible modificar los mismos mediante los métodos definidos para dicho objeto.

Herencia: Es la forma en que un objeto puede heredar el comportamiento y los atributos de otros objetos que son similares a ella

Polimorfismo: Comportamientos diferentes, asociados a objetos distintos, pueden compartir el mismo nombre; al llamarlos por ese nombre se utilizará el comportamiento correspondiente al objeto que se esté usando

## **TODOS LOS ATRIBUTOS DEBEN PERTENECER A UNA CLASE**

Campos:

Almacenan valores de un objeto, también conocidos como instancias, definen el estado de un objeto:

Ej: Private int Price

1º Modificador de visibilidad, 2º el tipo, 3º El nombre que toma la variable

### **Cosntructores:**

Inicializan el objeto, tienen el mismo nombre que la clase, normalmente reciben valores de los campos externos:

```
Ej: public TicketMachine(int ticketCost)
    {
        price = ticketCost;
        balance = 0;
        total = 0;
    }
```

Los metodos implementan el comportamiento de los objetos, tienen cabecera y un cuerpo,

```
Cabecera: public int GetPrice(){  
/*Body: todo lo que esta dentro de los corchetes*/  
Return Price  
}
```

Los métodos mutantes sirven para cambiar el valor de 1 o mas campos, tienen la misma estructura cabecera, cuerpo

Ej:

```
Public void insertmoney (int amount){  
Balance= balance+amount  
}
```

Método estático: Declarando un método o atributo como estático, significa que va a ser justo una instancia de ese método o atributo

Librerías: grupo de clases que son relacionados

Ej: java.io es una librería que agrupa clases para que puedas hacer un input/output

Una librería puede más librerías dentro

**Import:** Si queremos usar una clase dentro de un paquete, podemos usar el import, podemos utilizarlo tanto para importar solo una clase o todas las clases dentro de una libreria

Paquete o librería	Descripción
java.io	Librería de Entrada/Salida. Permite la comunicación del programa con ficheros y periféricos.
java.lang	Paquete con clases esenciales de Java. No hace falta ejecutar la sentencia import para utilizar sus clases. Librería por defecto.
java.util	Librería con clases de utilidad general para el programador.
java.applet	Librería para desarrollar <i>applets</i> .
java.awt	Librerías con componentes para el desarrollo de interfaces de usuario.
java.swing	Librerías con componentes para el desarrollo de interfaces de usuario. Similar al paquete awt.
java.net	En combinación con la librería java.io, va a permitir crear aplicaciones que realicen comunicaciones con la red local e Internet.
java.math	Librería con todo tipo de utilidades matemáticas.
java.sql	Librería especializada en el manejo y comunicación con bases de datos.
java.security	Librería que implementa mecanismos de seguridad.
java.rmi	Paquete que permite el acceso a objetos situados en otros equipos (objetos remotos).
java.beans	Librería que permite la creación y manejo de componentes <i>javabeans</i> .