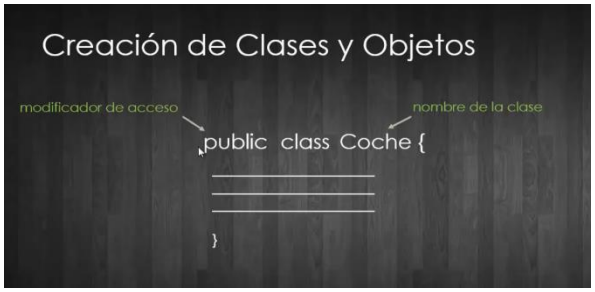
TEMA 8.
INTRODUCCIÓN A LA PROGRAMACIÓN ORIENTADA A OBJETOS

8.1. Definición de clases

Una clase es el elemento básico de la programación orientada a objetos, representa a una plantilla desde lo que se puede crear objetos.

Una clase es un tipo definido por el usuario; son los bloques de construcción fundamentales de los programas orientados a objetos.

Técnicamente una clase es una declaración de objetos, también se podría definir como una abstracción de objetos; esto quiere decir que la definición de un objeto es la clases. Cuando programamos sobre un objeto y se le define características y funcionalidades, en realidad lo que estamos haciendo es programar una clase.



8.2. Definición de objeto

✍ Booch[1] define un objeto como “una entidad (algo) que tiene un estado, un comportamiento y una identidad”.

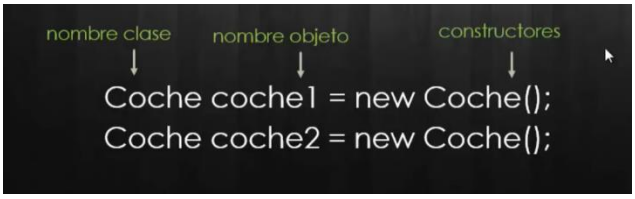
En el diseño de un programa orientado a objetos, se crea una abstracción o modelo simplificado de la máquina basado en las propiedades y comportamiento que son útiles en el tiempo.

✍ Martin y Odell definen un objeto como “cualquier cosa, real o abstracta, en la que se almacenan datos y aquellos métodos (operaciones) que manipulan los datos”.

Para realizar esa actividad se añaden a cada objeto de la clase los propios datos y los asociados con sus propios métodos miembro que pertenecen a la clase.

✍ Los objetos son copias o ejemplares de una clase. El hecho de crear un objeto permite tener acceso a los atributos y métodos definidos en la clase. La acción de crear un objeto de una clase es llamada instanciade clase.

Como se crean Objetos



8.3. Abstracción de clases

- Una clase describe un grupo de objetos que comparten propiedades y métodos comunes
- Una clase es una plantilla que define qué forma tienen los objetos de la clase
- Una clase se compone de:
  -Información: campos (atributos, propiedades)
  -Comportamiento: métodos (operaciones, funciones)

Nombre de la Clase
Lista de Atributos
Lista de Métodos

En java se visualiza con el siguiente script:

Class nombre de la clase

```
class Nombre de la clase
{
    Lista de atributos
    Lista de Métodos
}
```

8.3.1. Atributos

Un atributo de clase es la característica que puede tener un objeto. Se puede decir que las propiedades son algo así como variables donde se almacenan datos relacionados con los objetos. *Formato:*

```
Visibilidad tipoDatos nombreAtributo;
```

Donde:

- Visibilidad: puede ser private, public, protected o sin definición
- TipoDatos: son los tipos de datos primitivos, pero también se puede hacer referencia a clases específicas.
- NombreAtributo: es el nombre que se le asignará al atributo de clase.

Ejemplo: Suponga que tiene la clase vendedor y se necesita declarar el atributo código. Vea cuántas posibilidades tiene para declarar y cuál es la consecuencia de ello:

DECLARACIÓN DE UN ATRIBUTO	EXPLICACIÓN

int codigo;	Al declarar código sin visibilidad solo podrá ser accedida desde la misma clase y no desde el exterior de la clase.
public int codigo;	Al declarar código como público todas las clases del mismo proyecto podrán hacer referencia al valor contenido en dicho atributo.
private int codigo;	Al declarar como private solo podrá ser accedida desde la clase que se definió . Hay que mencionar que la implementación del atributo como privado es lo más recomendado, puesto que se protege la información contenida en el atributo, es decir, no será manipulada la información.
protected int codigo;	Al declarar como protected solo se podrá acceder desde la clase que se difinió y las clases descendientes de esta.

Vea el script que declara los atributos de la clase Producto como privados:

Producto
-codigo: int -descripcion: String -categoria: int -precioUnitario: double

En java tendría el siguiente script:

```
public class Producto
{
    private int codigo;
    private String descripcion;
    private int categoria;
    private double precioUnitario
}
```

Vea el script que declara los atributos de la clase Producto como pública:

```
public class Producto
{
    public int codigo;
    public String descripcion;
    public int categoria;
    public double precioUnitario;
}
```

8.3.2. Métodos

Considerar el tema de programación modular, la cual dejaba claro que una aplicación puede ser administrada por medio de módulos que dividen el problema en pequeñas

porciones que ayudan al programador a realizar un mejor desarrollo de la aplicación. Los métodos de la clase tienen el mismo propósito dentro de la clase, la diferencia es que ahora tomará interés en la visibilidad de los métodos.

Formato:

```
Visibilidad tipoDatos nombreMetodo();

Visibilidad tipoDatos nombreMetodo(parametros);
```

Donde:

- Visibilidad: puede ser private, public, protected o sin definición
- TipoDatos: es el tipo de datos que retornará el método, cuando no retorna valor alguno se colocará la cláusula void.
- NombreMetodo: es el nombre que se le asignará al método de clase, hay que tener en cuenta que dependiendo del trabajo que realice el método se deberá implementar los parámetros.

Ejemplos de declaración de métodos.

DECLARACIÓN DE UN MÉTODO	EXPLICACIÓN
int getCodigo(){ }	Se declara el método getCodigo sin visibilidad; por tanto, solo será accesible dentro de la clase.
public int getCodigo(){ }	Al declararse como público el método getCodigo podrá ser accesible desde todas las clases de la misma aplicación.
private int getCodigo(){ }	Al declararse como privado el método getCodigo solo será accesible dentro de clases de la misma clase.
protected int getCodigo(){ }	Al declararse como protected el método getCodigo podrá ser accesible desde la misma clase y las clases descendientes.

Si necesita implementar los métodos de la clase Producto en un sistema de ventas tendría:

Producto
-getCodigo():int -getDescripcion():String -getCategoria():int -getPrecioUnitario():double -setCodigo(codigo:int): void -setDescripcion(descripción:String):void -setCategoria(categoria:int):void -setPrecioUnitario(doublé precioUnitario):void

En java tendría el siguiente script de la clase Producto:

```
public class Producto
{
    public int getCodigo(){
       return codigo;
    }
    public String getDescripcion(){
```

8.3.3. Problemas de aplicación

Caso Práctico: Crear clases y objetos

```
package ClasesObjetos2;

//Crear una clase
public class Telefono {
 //valor a los atributos
    String Marca;
    String Modelo;
    String Numero;
    double costo;
    //Metodo main
    public static void main(String[] args ){
       //Crear el objeto fono 1
       Telefono fono1=new Telefono();
       fono1.Marca="Samsung";
       fono1.Modelo="Galaxy Z";
       fono1.Numero="7589666";
       fono1.costo=2300;
```

```
        System.out.println("La marca es: "+ fono1.Marca);
        System.out.println("El modelo es: "+fono1.Modelo);
        System.out.println("El nùmero es: "+fono1.Numero);
        System.out.println("El costo es: "+fono1.costo);
        Telefono fono2=new Telefono();
        fono2.Marca="Apple";
        fono2.Modelo="iPhone 15 Pro Max";
        fono2.Numero="6580000";
        fono2.costo=5200.3;

        System.out.println("----Telefono 2------");

        System.out.println("La marca es: "+ fono2.Marca);
        System.out.println("El modelo es: "+fono2.Modelo);
        System.out.println("El nùmero es: "+fono2.Numero);
        System.out.println("El costo es: "+fono2.costo);

    }

     }
```

```java
package ClasesObjetos2;

//Crear una clase
public class Telefono {
  //valor a los atributos
      String Marca;
      String Modelo;
      String Numero;
      double costo;
      //Metodo main
      public static void main(String[] args ){
          //Crear el objeto fono 1
          Telefono fono1=new Telefono();
          fono1.Marca="Samsung";
          fono1.Modelo="Galaxy Z";
          fono1.Numero="7589666";
          fono1.costo=2300;

          System.out.println("La marca es: "+ fono1.Marca);
          System.out.println("El modelo es: "+fono1.Modelo);
          System.out.println("El nùmero es: "+fono1.Numero);
          System.out.println("El costo es: "+fono1.costo);
          Telefono fono2=new Telefono();
          fono2.Marca="Apple";
          fono2.Modelo="iPhone 15 Pro Max";
          fono2.Numero="6580000";
          fono2.costo=5200.3;

          System.out.println("----Telefono 2------");

          System.out.println("La marca es: "+ fono2.Marca);
          System.out.println("El modelo es: "+fono2.Modelo);
          System.out.println("El nùmero es: "+fono2.Numero);
          System.out.println("El costo es: "+fono2.costo);

    }

      }
```

Sin parámetros y sin retorno

Caso Práctico 2: Crear Métodos
//Crear la clase Operación

```
package ClasesyObjetos;

import javax.swing.JOptionPane;

public class Operacion {
   //Atributos
   int numero1;
   int numero2;
```

```
   int suma;
   int resta;
   int multiplicacion;
   int division;

   //Metodos

   //Mètodo para pedirle al usuario que nos digite 2 nùmeros
   public void leernumeros(){
      numero1=Integer.parseInt(JOptionPane.showInputDialog("Digite un Nùmero"));
      numero2=Integer.parseInt(JOptionPane.showInputDialog("Ingrese Segundo
Nùmero"));
   }
   public void sumar(){
      suma=numero1+numero2;
   }
   public void restar(){
     resta=numero1-numero2;
   }
   public void multiplicar(){
      multiplicacion=numero1*numero2;
   }
   public void dividir(){
      division=numero1/numero2;
   }
   public void mostrarResultados(){
      System.out.println("La suma es: "+suma);
      System.out.println("La Resta es:.."+resta);
      System.out.println("La Multiplicaciòn es:.."+multiplicacion);
      System.out.println("La divisiòn  es:."+division);
   }
}
```

```java
package ClasesyObjetos;

import javax.swing.JOptionPane;

public class Operacion {
    //Atributos
    int numero1;
    int numero2;
    int suma;
    int resta;
    int multiplicacion;
    int division;

    //Metodos

    //Mètodo para pedirle al usuario que nos digite 2 nùmeros
    public void leernumeros(){
        numero1=Integer.parseInt(JOptionPane.showInputDialog("Digite un Nùmero"));
        numero2=Integer.parseInt(JOptionPane.showInputDialog("Ingrese Segundo Nùmero"));
    }
    public void sumar(){
        suma=numero1+numero2;
    }
    public void restar(){
       resta=numero1-numero2;
    }
    public void multiplicar(){
        multiplicacion=numero1*numero2;
    }
    public void dividir(){
        division=numero1/numero2;
    }
    public void mostrarResultados(){
        System.out.println("La suma es: "+suma);
        System.out.println("La Resta es:.."+resta);
        System.out.println("La Multiplicaciòn es:.."+multiplicacion);
        System.out.println("La divisiòn  es:."+division);
    }
}
```

```java
//Crear una nueva clase Main
package ClasesyObjetos;

public class main {
   public static void main(String[] args ){
   Operacion op=new Operacion();
   op.leernumeros();
   op.sumar();
   op.restar();
   op.multiplicar();
   op.dividir();
   op.mostrarResultados();
}
   }
```

```
package ClasesyObjetos;

public class main {
    public static void main(String[] args ){
    Operacion op=new Operacion();
    op.leernumeros();
    op.sumar();
    op.restar();
    op.multiplicar();
    op.dividir();
    op.mostrarResultados();
}
    }
```

Con Parámetros y con retorno

```
package ClasesyObjetos;

import javax.swing.JOptionPane;

public class Operacion2 {

    //Metodos

    //Mètodo para pedirle al usuario que nos digite 2 nùmeros
    public void leernumeros(){

    }
    public int sumar(int numero1,int numero2){
      int  suma=numero1+numero2;
      return suma;
    }
    public int restar(int numero1,int numero2){
       int resta=numero1-numero2;
       return resta;
    }
    public int multiplicar(int numero1,int numero2){
        int multiplicacion=numero1*numero2;
        return multiplicacion;
    }
    public int dividir(int numero1,int numero2){
        int division=numero1/numero2;
        return division;
    }
    public void mostrarResultados(int suma, int resta, int multiplicacion, int division){
        System.out.println("La suma es: "+suma);
        System.out.println("La Resta es:.."+resta);
        System.out.println("La Multiplicaciòn es:.."+multiplicacion);
        System.out.println("La divisiòn  es:."+division);
    }
}
```

```
package ClasesyObjetos;

import javax.swing.JOptionPane;

public class Operacion2 {

   //Metodos

   //Mètodo para pedirle al usuario que nos digite 2 nùmeros
   public void leernumeros(){

   }
   public int sumar(int numero1,int numero2){
    int  suma=numero1+numero2;
    return suma;
   }
   public int restar(int numero1,int numero2){
     int resta=numero1-numero2;
     return resta;
   }
   public int multiplicar(int numero1,int numero2){
      int multiplicacion=numero1*numero2;
      return multiplicacion;
   }
   public int dividir(int numero1,int numero2){
      int division=numero1/numero2;
      return division;
   }
   public void mostrarResultados(int suma, int resta, int multiplicacion, int division){
      System.out.println("La suma es: "+suma);
      System.out.println("La Resta es:.."+resta);
      System.out.println("La Multiplicaciòn es:.."+multiplicacion);
      System.out.println("La divisiòn  es:."+division);
   }
}
```

Main

```
package ClasesyObjetos;

import javax.swing.JOptionPane;

public class main {
    public static void main(String[] args ){
      int   numero1=Integer.parseInt(JOptionPane.showInputDialog("Digite un Nùmero"));
        int numero2=Integer.parseInt(JOptionPane.showInputDialog("Ingrese Segundo Nùmero"));
    Operacion2 op=new Operacion2();
    op.leernumeros();
    int sumar=op.sumar(numero1,numero2);
     int restar=op.restar(numero1,numero2);
     int multi=op.multiplicar(numero1,numero2);
     int divide=op.dividir(numero1,numero2);
    op.mostrarResultados(sumar,restar,multi,divide);
}
    }
```

package ClasesyObjetos;

import javax.swing.JOptionPane;

public class main {
   public static void main(String[] args){
     int   numero1=Integer.parseInt(JOptionPane.showInputDialog("Digite un Nùmero"));
      int numero2=Integer.parseInt(JOptionPane.showInputDialog("Ingrese Segundo Nùmero"));
   Operacion2 op=new Operacion2();
   op.leernumeros();
   int sumar=op.sumar(numero1,numero2);
    int restar=op.restar(numero1,numero2);
    int multi=op.multiplicar(numero1,numero2);
    int divide=op.dividir(numero1,numero2);
   op.mostrarResultados(sumar,restar,multi,divide);
}
   }

8.4. Abstracción de datos y encapsulamiento

8.5. Constructores

> ✍ "Un constructor es un método especial de una clase que se invoca siempre que se crea un objeto de esa clase."

> ✍ "Un constructor es un método que se ejecuta automáticamente cuando se crea un objeto de una clase; sirve para inicializar los miembros de la misma."

El constructor tiene el mismo nombre que clase; cuando se define, no se puede especificar un valor de retorno porque nunca devuelve uno; sin embargo, puede tomar cualquier número de argumentos.

Cuando se crea un objeto ocurren 3 cosas:

- Se asigna memoria para el objeto
- Se inicializan los atributos de ese objeto.
- Se invoca al constructor de la clase que puede ser uno entre varios.

```
Persona p1 = new Persona();
```

Características de los Constructores

- Tienen el mismo nombre de la Clase
- No devuelven ningún valor.
- Debe declararse como público.

Parámetros = Atributos	Parámetros <> Atributos

Constructor
`package ClasesObjteto3;`
`public class Persona {`
`//Atributos`
`String nombre;`
`int edad;`
`//Metodos`
`//Metodo Constructor`
`public Persona(String nombre,int edad){`
`this.nombre=nombre;`
`this.edad=edad;`
`}`
`public void mostrarDatos(){`
`System.out.println("El nombre es:"+nombre);`
`System.out.println("La edad es:"+edad);`
`}`
`}`
	`package ClasesObjteto3;`
	`public class Persona {`
	`//Atributos`
	`String nombre;`
	`int edad;`
	`//Metodos`
	`//Metodo Constructor`
	`public Persona(String nombre1,int edad1){`
	`nombre=nombre1;`
	`edad=edad1;`
	`}`
	`public void mostrarDatos(){`
	`System.out.println("El nombre es:"+nombre);`
	`System.out.println("La edad es:"+edad);`
	`}`
	`}`
	Constructor

Main	Main

```java
package ClasesObjteto3;

public class Main {


    public static void main(String[] args) {
        Persona P1=new Persona("Carlos",30);
        P1.mostrarDatos();
    }

}
```

```java
package ClasesObjteto3;

public class Main {


    public static void main(String[] args) {
        Persona P1=new Persona("Carlos",30);
        P1.mostrarDatos();
    }

}
```

Sobrecarga de Constructores y métodos

```java
package ClasesObjteto3;


public class Persona {
    //Atributos
    String nombre;
    int edad;
    String ci;

    //Metodos

//Metodo Constructor

public Persona(String nombre, int edad){
        this.nombre = nombre;
        this.edad = edad;
}

    public Persona(String ci) {
        this.ci = ci;
    }

    public void corre(){
        System.out.println("Mi nombre es: "+nombre+" tengo "+edad+" años ");
    }
    public void corre(int km){
        System.out.println("Corrì "+km+" corridos");
     }

    public void mostrarDatos() {
        System.out.println("El nombre es:"+nombre);
        System.out.println("La edad es:"+edad);
    }

}
```

```
package ClasesObjteto3;

public class Main {



    public static void main(String[] args) {
        Persona P1=new Persona("Carlos",25);
        P1.corre();
        Persona P2=new Persona("1124556");
        P2.corre(100);
        //P1.mostrarDatos();P
    }

}
```