

TITULO

PROGRAMACION ORIENTADA A OBJETOS

Semana 05

MIEMBROS DE CLASE Y DE INSTANCIA

OBJETIVOS

- ◆ Entender miembros de instancia.
- ◆ Entender miembros de clase.
- ◆ Aplicar la programación haciendo uso de miembros de instancia y clase.



AGENDA

- **Miembros de instancia.**
- **Miembros de clase.**



Miembros de instancia

- ◆ Los miembros de instancia se refiere a:
 - Atributos de instancia.
 - Métodos de instancia.
- ◆ Para asignar y utilizar los valores de los atributos de instancia, se debe implementar métodos de instancia.
- ◆ Los atributos y métodos de instancia existen desde el momento que se crea un objeto.
- ◆ El valor asignado a un atributo de instancia puede ser diferente para cada uno de los objetos.
- ◆ Para invocar a un atributo o método se hace referencia a través de un objeto.

Miembros de instancia

◆ Veamos el siguiente ejemplo:

```
public class Operacion
{
    //Atributo de instancia
    public int x;

    //Método de instancia
    public String mostrar()
    {
        return "\n El valor de x es = " + x;
    }
}
```

Miembros de instancia

- ◆ Veamos el siguiente ejemplo (continua):

```
public class MainActivity extends AppCompatActivity {  
protected void onCreate(Bundle savedInstanceState) {  
    //Declaración del objeto  
    Operacion objOpe;  
    //Creación del objeto  
    objOpe = new Operacion();  
    //invocación al Atributo de instancia y asignar valor  
    objOpe.x = 5;  
    // invocación al Método de instancia  
    txv.setText (objOpe.mostrar());  
    //Creamos otro objeto.  
    Operacion obj2= new Operacion();  
    obj2.x = 20;  
    txv.append (obj2.mostrar());  
}  
}
```

Miembros de clase

- ◆ Los miembros de clase se refiere a:
 - Atributos de clase.
 - Métodos de clase.
- ◆ Para asignar y utilizar los valores de los atributos de clase, se debe implementar métodos de clase.
- ◆ Los métodos de clase sólo pueden acceder a los atributos de clase.

Miembros de clase

- ◆ El valor asignado a un atributo de clase es el mismo para todos los objetos de dicha clase.
- ◆ Para invocar a un atributo o método se hace referencia a la clase.
- ◆ Para indicar que un atributo ó método es un miembro de clase se utiliza la palabra reservada **static**.

- ◆ Veamos el siguiente ejemplo:

```
public class Operacion
{
    //Atributo de clase
    public static int x;

    //Método de clase
    public static String mostrar()
    {
        return "\n El valor de x es = " + x;
    }
}
```

- ◆ Veamos el siguiente ejemplo (continua):

```
public class MainActivity extends AppCompatActivity {  
  
    protected void onCreate(Bundle savedInstanceState) {  
  
        //invocación al Atributo de clase  
        Operacion.x = 5;  
        // invocación al Método de clase  
        txv.setText(Operacion.mostrar());  
  
        Operaciones.x = 20;  
        txv.append (Operacion.mostrar());  
    }  
}
```

Ejercicio

◆ Enunciado

◆ De los datos que se muestran para una Factura, indicar cuáles serían atributos de clase:

- Fecha
- Importe
- Igv (en porcentaje, no en soles)
- Ruc del cliente
- Código del vendedor (asumir que son varios)
- Nombre de la empresa (la que vende)

◆ Respuesta

- ◆ Igv
- ◆ Nombre de la empresa

Implementar una Aplicación Móvil que permita registrar los datos de la Factura utilizando miembros de instancia y de clase .

- ◆ En esta sesión, aprendió que:
 - ◆ Los atributos y métodos de instancia recién existen al momento que se crea un objeto.
 - ◆ Para invocar a un atributo ó método de instancia, se debe hacer referencia a través de un objeto.
 - ◆ El valor asignado a un atributo de clase será el mismo para todos los objetos de dicha clase.
 - ◆ Para acceder a los atributos de clase se debe utilizar métodos de clase.
 - ◆ Para declarar atributos y métodos de clase se utiliza la palabra reservada **static**.