# HITO 2-POO variables, arrays, clases, packages

Ing. William Barra
Estudiante: Iris Michelle Velasco Aruquipa

# ¿A que se refiere Cuando se habla de POO?

Es un paradigma de programación que usa objetos y sus interacciones para diseñar aplicaciones y Programas de computadora. Está basado en varias técnicas, incluyendo herencia, modularidad, polimorfismo, y encapsulamiento.

# Cuáles son los 4 Componentes que Componen POO

Clases: Las clases pueden ser definidas como un molde que contendrá todas las características y acciones con las cuales podemos.

Propiedades: Las propiedades son las características de una clase

Métodos: Los métodos son las acciones que una clase puede realizer.

Objetos: Son aquellos que tienen propiedades y comportamientos, estos pueden ser físicos o conceptuales.


# Cuáles son los pilares de POO

Abstracción: Es cuando separamos los datos de un objeto para luego generar un molde .

Encapsulamiento: Lo puedes utilizar cuando deseas que ciertos métodos o propiedades sean inviolables o inalterables


Herencia: Nos permite crear nuevas clases a partir de otras.

Polimorfismo: Se utiliza para crear métodos con el mismo nombre pero con diferente comportamiento.

# ¿Qué es encapsulamiento?

Se refiere a la agrupación de datos con los métodos que operan en esos datos, o la restricción del acceso directo a algunos de los componentes de un objeto.

```java
public class MiClase{

    private int tipo;

        public void setTipo(int t) {

            tipo = t;

        }

    public int getTipo() {

        return tipo;

    }

}

class AccesoIndirecto {

    public static void main(String[] args) {

        MiClase mc = new MiClase();
        mc.setTipo(5);
        System.out.println("El tipo es:" + mc.getTipo());

    }

}
```

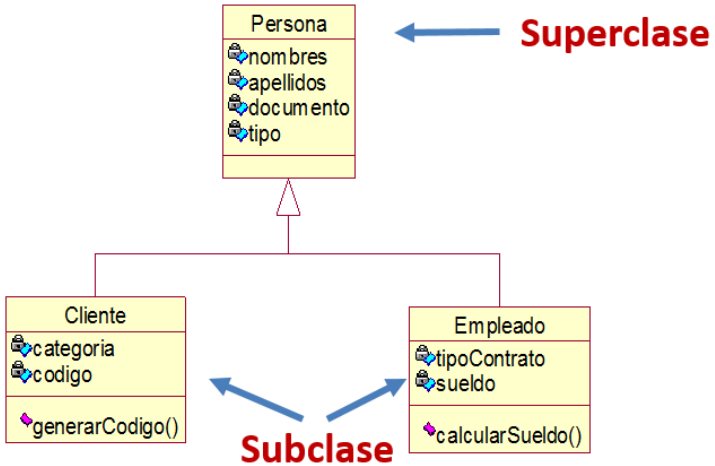
# ¿Qué es Abstracción?

```python
class Perro(Animal):

Animal.__nombre = 'Perro'
Animal.__patas = 4
Animal.__sonido = 'Guau guau'

def presentarse(self):
    print("Hola. Soy un " + self.__nombre)

def emitirSonido(self):
    print(self.__sonido)

def comer(self):
    print(self.__nombre + ': ¡Yumi! Estoy comiendo :D')

def caminar(self):
    print(self.__nombre + ': Estoy caminando.')

def volar(self):
    print(self.__nombre + ': Estoy volando.')

def nadar(self):
    print(self.__nombre + ': Estoy nadando.')
```

Se refiere al proceso por el cuál la interfaz de un objeto muestra su comportamiento específico y nada más, absolutamente nada más.

# ¿Qué es Herencia?

Es un concepto de la programación orientada a objetos. El cual es un mecanismo que permite derivar una clase a otra clase.




# ¿Qué es Polimorfismo?

el polimorfismo se refiere a la propiedad por la que es posible enviar mensajes sintácticamente iguales a objetos de tipos distintos.

```
public class Polimorfismo {
    public static void main(String[] args) {

        mostrarObjeto(new Rectangulo(1, 1, "negro", true));
        mostrarObjeto(new Circulo(1, "rojo", false));
    }

    public static void mostrarObjeto(FiguraGeometrica object) {
        System.out.println("Creado en  " + object.obtenerFechaCreado() +
        ". su color es " + object.obtenerColor());
    }
}
```

# ¿Qué es ARRAY?

En programación, se le denomina vector, formación, matriz, a una zona de almacenamiento contiguo que contiene una serie de elementos del mismo tipo, los elementos de la matriz

# ¿Qué son los paquetes en JAVA?

Un Paquete en Java es un contenedor de clases que permite agrupar las distintas partes de un programa y que por lo general tiene una funcionalidad y elementos comunes, definiendo la ubicación de dichas clases en un directorio de estructura jerárquica.

# ¿Cómo se define una clase main en JAVA?

**El método main es el punto de entrada de un programa ejecutable, es donde se inicia y finaliza el control del programa.**

```
{
    static int Main(string[] args)
    {
        // Test if input arguments were supplied.
        if (args.Length == 0)
        {
            Console.WriteLine("Please enter a numeric argument.");
            Console.WriteLine("Usage: Factorial <num>");
            return 1;
        }

        // Try to convert the input arguments to numbers. This will throw
        // an exception if the argument is not a number.
        // num = int.Parse(args[0]);
        int num;
        bool test = int.TryParse(args[0], out num);
        if (!test)
        {
            Console.WriteLine("Please enter a numeric argument.");
            Console.WriteLine("Usage: Factorial <num>");
            return 1;
        }

        // Calculate factorial.
        long result = Functions.Factorial(num);

        // Print result.
        if (result == -1)
            Console.WriteLine("Input must be >= 0 and <= 20.");
        else
            Console.WriteLine($"The Factorial of {num} is {result}.");

        return 0;
    }
}
// If 3 is entered on command line, the
// output reads: The factorial of 3 is 6.
```

# Practica Generamos la clase Provincia

```java
public class Provincia {
    4 usages
    private String NombreProvincia;

    public Provincia (String NombreProvincia) {
        this.NombreProvincia = NombreProvincia;
    }
    2 usages
    public Provincia() {
        this.NombreProvincia = "";
    }

    1 usage
    public String getNombreProvincia() {
        return this.NombreProvincia;
    }

    2 usages
    public void setNombreProvincia(String NuevoNombre) {
        NombreProvincia = NuevoNombre;
    }

    1 usage
    public void mostrarProvincia() {
        System.out.println("Mostrando datos de la provincia");
        System.out.println("Nombre Provincia: " + this.getNombreProvincia());
        System.out.println("\n");
```

# Generamos la clase Departamento

```java
public class Departamento {
    3 usages
    private String nombreDepartamento;
    5 usages
    private int noProvincias;
    3 usages
    private Provincia[] provincias;

    public Departamento(String nombreDepartamento, int noProvincias, Provincia[] provincias) {
        this.nombreDepartamento = nombreDepartamento;
        this.noProvincias = noProvincias;
        this.provincias = provincias;
    }
    2 usages
    public Departamento () {}
    1 usage
    public String getNombreDepartamento() {
        return this.nombreDepartamento;
    }
    1 usage
    public Provincia[] getProvincias() {
        return this.provincias;
    }

    public int getNoProvincias() {
        return noProvincias;
```

```java
public int getNoProvincias() {
    return noProvincias;
}

1 usage
public void setNoProvincias(int noProvincias) {
    this.noProvincias = noProvincias;
}

2 usages
public void setNombreDepartamento(String nombreDepartamento) {
    this.nombreDepartamento = nombreDepartamento;
}

1 usage
public void setProvincias(Provincia[] provincias) {
    this.provincias = provincias;
}

1 usage
public void mostrarDepartamento() {
    System.out.println("\nMOSTRANDO DATOS DEL DEPARTAMENTO");
    System.out.println("Nombre Departamento: " + this.getNombreDepartamento());
    System.out.println("No Provincias: " + this.noProvincias);
    for (int i=0; i < this.noProvincias; i++) {
        this.getProvincias()[i].mostrarProvincia();
    }
```



# GENERAR UNA CLASE PAIS

```java
package PAIS;
public class pais {
    3 usages
    private String nombrePais;
    2 usages
    private int noDepartamentos;
    3 usages
    private Departamento[] departamentos;

    1 usage
    public pais(String nombrePais,int noDepartamentos, Departamento[] departamentos) {
        this.nombrePais = nombrePais;
        this.noDepartamentos = noDepartamentos;
        this.departamentos = departamentos;
    }
    1 usage
    public String getNombrePais() {
        return this.nombrePais;
    }
    1 usage
    public Departamento[] getDepartamentos() {
        return this.departamentos;
    }

    public void setNombrePais(String nombrePais) {
        this.nombrePais = nombrePais;
    }
```



```java
}
1 usage
public Departamento[] getDepartamentos() {
    return this.departamentos;
}

public void setNombrePais(String nombrePais) {
    this.nombrePais = nombrePais;
}

public void setDepartamentos(Departamento[] departamentos) {
    this.departamentos = departamentos;
}
1 usage
public void mostrarPais() {
    System.out.println("\nMOSTRANDO DATOS DEL PAIS");
    System.out.println("Nombre Pais: " + this.getNombrePais());
    for (int i = 0; i < this.noDepartamentos; i++) {
        this.getDepartamentos()[i].mostrarDepartamento();
    }
}
```

# Main

```java
package PAIS;

import java.util.Scanner;

public class main2 {
    public static void main(String [] args) {
        Scanner leer = new Scanner(System.in);
        String nombreProvincia;
        int i, nProvincias;
        nProvincias = 2;
        String nombreDepartamento;
        int j, nDepartamentos = 2;

        Departamento[] departamentos = new Departamento[100];
```

```java
for (j = 0; j < nDepartamentos; j = j + 1) {
    System.out.println("Ingrese el nombre del departamento " + (j + 1) + ": ");
    nombreDepartamento = leer.next();
    Provincia[] provincias = new Provincia[100];

    for (i = 0; i < nProvincias; i = i + 1) {
        System.out.println("Ingrese el nombre de la provincia " + (i + 1) + ": "
        nombreProvincia = leer.next();

        Provincia prov = new Provincia();
        prov.setNombreProvincia(nombreProvincia);

        provincias[i] = prov;
    }
    System.out.println("Ingrese el nombre de la nueva provincia: ");
    nombreProvincia = leer.next();

    Provincia prov = new Provincia();
    prov.setNombreProvincia(nombreProvincia);

    provincias[nProvincias] = prov;
```

```
        provincias[nProvincias] = prov;

        Departamento dep = new Departamento();
        dep.setNombreDepartamento(nombreDepartamento);
        dep.setProvincias(provincias);
        dep.setNoProvincias(nProvincias + 1);
        departamentos[j] = dep;
    }

    System.out.println("Ingrese el nombre del nuevo departamento: ");
    nombreDepartamento = leer.next();

    Departamento dep = new Departamento();
    dep.setNombreDepartamento(nombreDepartamento);

    departamentos[nDepartamentos] = dep;
    pais p1 = new pais( nombrePais: "BOLIVIA",  noDepartamentos: nDepartamentos + 1, departamentos);
    p1.mostrarPais();
}
```