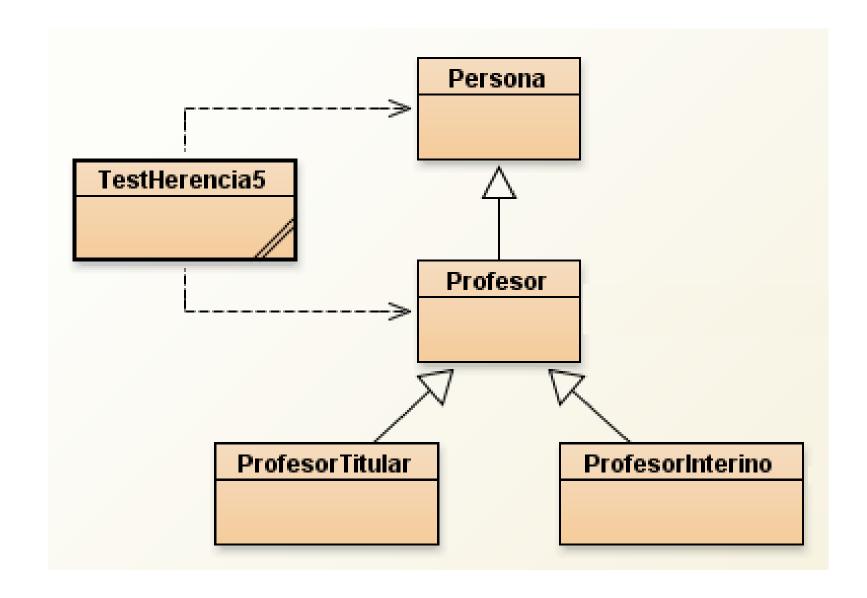
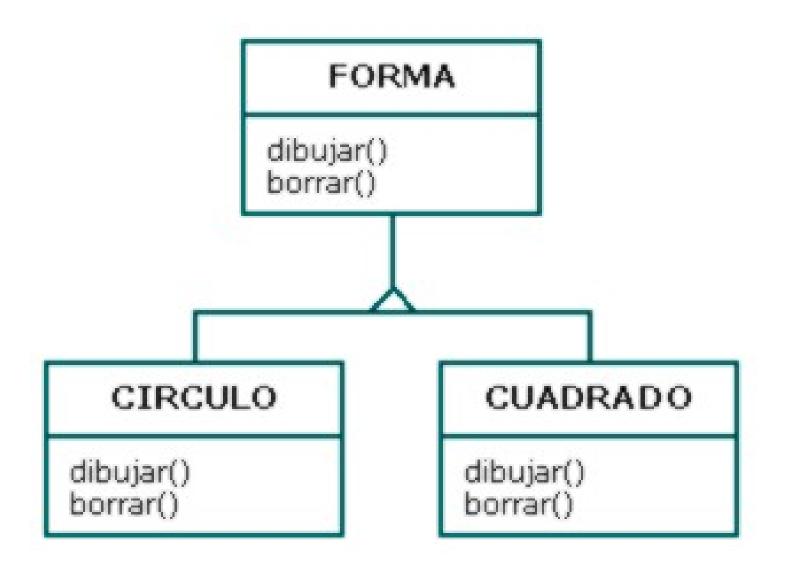
## clases abstractas POLIMORFISMO

Danna Valeria Morales Aguilar

## j de la una CLASE ABSTRACTA?

Una clase abstracta es una clase que no puede ser instanciada directamente, en otras palabras, no se pueden crear objetos directamente a partir de ella. Se define utilizando la palabra clave "abstract".







Capacidad de un objeto de una clase base para ser usado como objeto de sus clases derivadas, permite que un objeto pueda tomar diferentes formas y se aplica a variables o a métodos.



Este código hace referencia al tema de clases abstractas y polimorfismo, lo que realiza el codigo es que a partir de la clase abstracta **Deporte** tiene un método abstracto llamado **getCategoria()**.

Las subclases son diferentes deportes que se heredan de la clase **Deporte** y dan su propia implementación para el método **getCategoria()**.

## Jay DEPORTE

Se declara un método abstracto **getCategoria()**, este metodo no tiene implementación en esta clase y cada subclase que herede de **Deporte** debe dar su propia implementación.

El **private String** nombre se almacenara el nombre del **Deporte** y **private int vo**l almacenara el volumen de entrenamiento que se tenga.

```
public abstract class Deporte {
    abstract String getCategoria();
    private String nombre;
    private int vol;
    public Deporte(String nombre, int vol) {
        this.nombre=nombre;
        this.vol=vol;
```

public Deporte(String nombre, int vol), es el constructor de la clase Deporte, este método se invoca cuando se crea una instancia de una subclase de Deporte.

```
public class Pesas extends Deporte{
    private String musculo;

public Pesas(String nombre, int vol, String musculo) {
        super(nombre, vol);
        this.musculo=musculo;
    }

public String getCategoria() {
        return "Pesas";
    }
}
```



Se define la clase **Pesas, Pilates** y **Ballet** como subclase abstracta de **Deporte**, mediante la palabra **extends**.

Se declara musculo de tipo String con acceso privado, almacenara información relacionada con el grupo muscular especifico que se enfoque en el entrenamiento de pesas.

El constructor de la clase **Pesas** y **Pilates**, toma 3 parámetros: **nombre, vol** y **musculo**, se utiliza la palabra **super** para llamar al constructor de la clase base **Deporte**.

En el método abstracto **getCategoria()** se devuelve el nombre de **Pesas, Pilates** o **Ballet** como categoria del deporte.

```
Problems @ Javadoc ■ Declaration ■ Console ×
<terminated> Principal (9) [Java Application] C:\Users\Dann
Pesas
Pilates
Ballet
```

En la clase **Principal**, mandamos llamar el constructor de las subclases para inicializar la instancia con los valores.

Llamamos al método **getCategoria()** e imprimira la categoria del deporte.

```
public class Principal {
   public static void main(String[] args) {
        Deporte pesas = new Pesas("Pesas", 10, "cuadricep");
        System.out.println(pesas.getCategoria());
        Deporte pilates = new Pilates("Pilates", 2, "abs");
        System.out.println(pilates.getCategoria());
        Deporte ballet = new Ballet("Ballet", 3, 6);
        System.out.println(ballet.getCategoria());
}
```

