

Ejercicios Java 6

60.

• Saludable.java

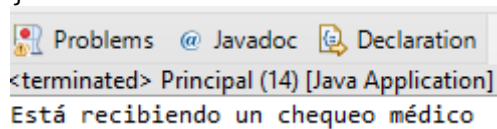
```
public interface Saludable {  
    void realizarChequeo(); //Método abstracto  
}
```

• Persona.java

```
public class Persona implements Saludable {  
    public void realizarChequeo() { //Imprime un mensaje  
        System.out.println("Está recibiendo un chequeo médico");  
    }  
}
```

• Principal.java

```
public class Principal {  
    public static void main(String[] args) {  
        Persona persona = new Persona();  
        persona.realizarChequeo(); //Imprime un mensaje  
    }  
}
```



Problems @ Javadoc Declaration
<terminated> Principal (14) [Java Application]
Está recibiendo un chequeo médico

61.

• Volador.java

```
public interface Volador {  
    void volar(); //Método abstracto  
}
```

• Avion.java

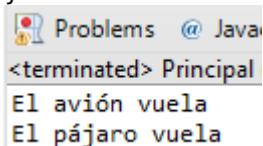
```
public class Avion implements Volador {  
    public void volar() { //Imprime un mensaje  
        System.out.println("El avión vuela");  
    }  
}
```

• Pajaro.java

```
public class Pajaro implements Volador {  
    public void volar() { //Imprime un mensaje  
        System.out.println("El pájaro vuela");  
    }  
}
```

• Principal.java

```
public class Principal {  
    public static void main(String[] args) {  
        Avion avion = new Avion();  
        Pajaro pajaro = new Pajaro();  
        avion.volar(); //Imprime un mensaje  
        pajaro.volar(); //Imprime un mensaje  
    }  
}
```



Problems @ Java
<terminated> Principal
El avión vuela
El pájaro vuela

62.

• Imprimible.java

```
public interface Imprimible {  
    void imprimir(); //Método abstracto  
}
```

• Factura.java

```
public class Factura implements Imprimible{
```

```

        public void imprimir() { //Imprime una factura

            System.out.println("Agua: 1.45 €");
            System.out.println("Patatas: 2.13 €");
            System.out.println("Total: 3.58 €");
        }
    }
}

```

• Principal.java

```

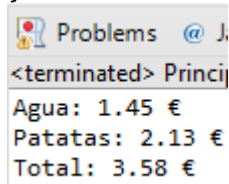
public class Principal {

    public static void main(String[] args) {

        Factura factura = new Factura();

        factura.imprimir(); //Imprime una factura
    }
}

```



63.

• Saludador.java

```

public interface Saludador {

    void saludar(); //Método abstracto
}

```

• Robot.java

```

public class Robot implements Saludador {

    public void saludar() { //Imprime un saludo

        System.out.println("Hola, soy un robot");
    }
}

```

• Principal.java

```

public class Principal {

```

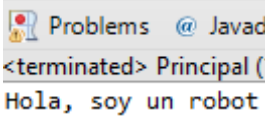
```

        public static void main(String[] args) {

            Robot robot = new Robot();

            robot.saludar(); //Imprime un saludo
        }
    }
}

```



64.

• Cantante.java

```

public interface Cantante {

    void cantar(); //Método abstracto
}

```

• Bailarin.java

```

public interface Bailarin {

    void bailar(); //Método abstracto
}

```

• Artista.java

```

public class Artista implements Cantante, Bailarin {

    public void cantar() { //Imprime el mensaje

        System.out.println("Está cantando");
    }

    public void bailar() { //Imprime el mensaje

        System.out.println("Está bailando");
    }
}

```

• Principal.java

```

public class Principal {

    public static void main(String[] args) {

        Artista artista = new Artista();
    }
}

```

```

        artista.cantar(); //Imprime un mensaje
        artista.bailar(); //Imprime un mensaje
    }
}

```



Problems @

<terminated> Print

Está cantando

Está bailando

65.

• Accion.java

```

public interface Accion {

    void ejecutar(); //Método abstracto
}

```

• Boton.java

```

public class Boton implements Accion{

    public void ejecutar() { //Imprime un mensaje

        System.out.println("Botón presionado");
    }

}

```

• Principal.java

```

public class Principal {

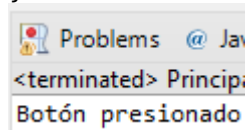
    public static void main(String[] args) {

        Boton boton = new Boton();

        boton.ejecutar(); //Imprime un mensaje
    }

}

```



Problems @ Java

<terminated> Principi

Botón presionado

66.

• Notificable.java

```
public interface Notificable {  
    void enviarNotificacion(); //Método abstracto  
}
```

• CorreoElectronico.java

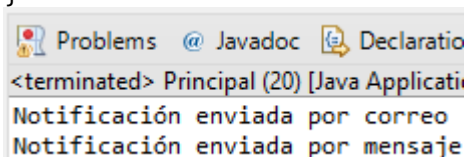
```
public class CorreoElectronico implements Notificable {  
    public void enviarNotificacion() { //Imprime un mensaje  
        System.out.println("Notificación enviada por correo");  
    }  
}
```

• MensajeTexto.java

```
public class MensajeTexto implements Notificable {  
    public void enviarNotificacion() { //Imprime un mensaje  
        System.out.println("Notificación enviada por mensaje");  
    }  
}
```

• Principal.java

```
public class Principal {  
    public static void main(String[] args) {  
        CorreoElectronico correo = new CorreoElectronico();  
        MensajeTexto mensaje = new MensajeTexto();  
        correo.enviarNotificacion(); //Imprime un mensaje  
        mensaje.enviarNotificacion(); //Imprime un mensaje  
    }  
}
```



```
<terminated> Principal (20) [Java Applicati  
Notificación enviada por correo  
Notificación enviada por mensaje
```

67.

• Figura.java

```
public interface Figura {  
  
    void calcularArea(); //Método abstracto  
}
```

• Cuadrado.java

```
public class Cuadrado implements Figura{  
  
    double lado;  
  
    public Cuadrado(double lado) { //Constructor  
  
        this.lado = lado;  
    }  
  
    public void calcularArea() { //Calcula el área del cuadrado  
  
        System.out.println("Área del cuadrado: " + lado * lado);  
    }  
}
```

• Triangulo.java

```
public class Triangulo implements Figura {  
  
    double base;  
    double altura;  
  
    public Triangulo(double base, double altura) { //Constructor  
  
        this.base = base;  
        this.altura = altura;  
    }  
  
    public void calcularArea() { //Calcula el área del triángulo  
  
        System.out.println("Área del triángulo: " + base * altura / 2);  
    }  
}
```

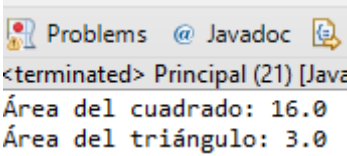
• Principal.java

```
public class Principal {  
  
    public static void main(String[] args) {  
  
        Cuadrado cuadrado = new Cuadrado(4);  
        Triangulo triangulo = new Triangulo(3, 2);  
    }  
}
```

```

        cuadrado.calcularArea(); //Muestra el área del cuadrado
        triangulo.calcularArea(); //Muestra el área del triángulo
    }
}

```



68.

• Logger.java

```

public interface Logger {

    void registrar(String mensaje); //Método abstracto

    public void separador(); //Método abstracto
}

```

• ConsoleLogger.java

```

public class ConsoleLogger implements Logger{

    public void registrar(String mensaje) { //Imprime un mensaje

        System.out.println(mensaje);
    }

    public void separador() { //Imprime una línea de separacion

        System.out.println("-----");
    }
}

```

• Principal.java

```

public class Principal {

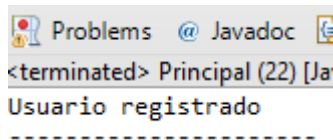
    public static void main(String[] args) {

        ConsoleLogger c = new ConsoleLogger();

        c.registrar("Usuario registrado"); //Muestra el área del
        cuadrado
        c.separador(); //Imprime una línea de separación

    }
}

```

```
<terminated> Principal (22) [Ja
Usuario registrado
-----
```

69.

• Identificable.java

```
public interface Identificable {

    void mostrarIdentidad(); //Método abstracto
}
```

• Empleado.java

```
public class Empleado implements Identificable {

    String nombre;
    int id;

    public Empleado(String nombre, int id) { //Constructor

        this.nombre = nombre;
        this.id = id;
    }

    public void mostrarIdentidad() { //Muestra los datos del empleado

        System.out.println("Nombre: " + nombre);
        System.out.println("ID: " + id);
    }
}
```

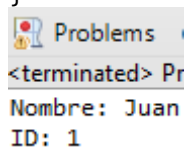
• Principal.java

```
public class Principal {

    public static void main(String[] args) {

        Empleado empleado = new Empleado("Juan", 001);

        empleado.mostrarIdentidad(); //Muestra los datos del empleado
    }
}
```



```
<terminated> Pr
Nombre: Juan
ID: 1
```

<https://github.com/Rodrigo-Garcia-Ortiz/Programacion.git>