Clase test

```
import java.io.IOException;
import Abstract_Factory.*;
public class Test{
    public static void main(String[] args) throws IOException {
        ProductoSalida salida = new SalidaTerminal();
        ProductoEntrada entrada = new EntradaTerminal();
        Figura figura;
        double area = 0;
        FabricaFiguras cuadrado = new FabricaCuadrado();
        figura = cuadrado.crearFigura();
        area = figura.CalcularArea();
        salida.enviar("El area del cuadrado es: " + area);

        FabricaFiguras circulo = new FabricaCirculo();
        figura = circulo.crearFigura();
        area = figura.CalcularArea();
        salida.enviar("El area del circulo es: " + area);

        FabricaFiguras triangulo = new FabricaTriangulo();
        figura = triangulo.crearFigura();
        area = figura.CalcularArea();
        salida.enviar("El area del triangulo es: " + area);

        double operador1 , operador2;
        Operaciones operaciones;
        FabricaOperaciones suma = new FabricaSuma();
        operaciones = suma.crearOperaciones();
        salida.enviar("Ingresa dos numeros: ");
        operador1 = Double.parseDouble(entrada.capturar());
        operador2 = Double.parseDouble(entrada.capturar());
        double resultado = operaciones.Operar(operador1, operador2);
        salida.enviar("El resultado de la suma es: " + resultado);

        FabricaOperaciones resta = new FabricaResta();
        operaciones = resta.crearOperaciones();
        salida.enviar("Ingresa dos numeros: ");
        operador1 = Double.parseDouble(entrada.capturar());
        operador2 = Double.parseDouble(entrada.capturar());
        resultado = operaciones.Operar(operador1, operador2);
        salida.enviar("El resultado de la resta es: " + resultado);

        FabricaOperaciones multiplicacion = new FabricaMultiplicacion();
        operaciones = multiplicacion.crearOperaciones();
        salida.enviar("Ingresa dos numeros: ");
        operador1 = Double.parseDouble(entrada.capturar());
```

```
        operador2 = Double.parseDouble(entrada.capturar());
        resultado = operaciones.Operar(operador1, operador2);
        salida.enviar("El resultado de la multiplicacion es: " + resultado);

        FabricaOperaciones division = new FabricaDivision();
        operaciones = division.crearOperaciones();
        salida.enviar("Ingresa dos numeros: ");
        operador1 = Double.parseDouble(entrada.capturar());
        operador2 = Double.parseDouble(entrada.capturar());
        resultado = operaciones.Operar(operador1, operador2);
        salida.enviar("El resultado de la division es: " + resultado);
    }
}
```

**Ingresa el lado del cuadrado:**

**4**

**El area del cuadrado es: 16.0**

**Ingresa el radio del circulo:**

**54**

**El area del circulo es: 9160.884177867836**

**Ingresa un lado del triagnulo:**

**12**

**Ingresa otro lado del triangulo:**

**23**

**Ingresa el ultimo lado del triangulo:**

**34**

**El area del triangulo es: 66.80896272207794**

**Ingresa dos numeros:**

**3**

**4**

**El resultado de la suma es: 7.0**

**Ingresa dos numeros:**

**2**

**3**

**El resultado de la resta es: 1.0**

**Ingresa dos numeros:**

**45**

**5**

**El resultado de la multiplicacion es: 225.0**

**Ingresa dos numeros:**

**34**

**55**

**El resultado de la division es: 0.0**

**Clase Figura**

```
public abstract class Figura {
    public abstract double CalcularArea();
}
```

Clase FabricaFiguras

```
import java.io.IOException;
public abstract class FabricaFiguras {
    public abstract Figura crearFigura() throws IOException;
}
```

Clase FabricaCuadrado

```
import java.io.IOException;
import Abstract_Factory.*;
public class FabricaCuadrado extends FabricaFiguras {
    public Figura crearFigura() throws IOException{
        ProductoEntrada entrada = new EntradaTerminal();
        ProductoSalida salida = new SalidaTerminal();
        salida.enviar("Ingresa el lado del cuadrado: ");
        double lado = Double.parseDouble(entrada.capturar());
        return new FiguraCuadrado(lado);
    }
}
```

Clase FiguraCuadrado

```
public class FiguraCuadrado extends Figura {
```

```
    private double lado;
    public FiguraCuadrado(double lado) {
        this.lado = lado;
    }
    public double CalcularArea() {
        return lado * lado;
    }
}
```

Clase FabricaCirculo

```
import Abstract_Factory.*;
import java.io.IOException;

public class FabricaCirculo extends FabricaFiguras {
    public Figura crearFigura() throws IOException {
        ProductoEntrada entrada = new EntradaTerminal();
        ProductoSalida salida = new SalidaTerminal();
        salida.enviar("Ingresa el radio del circulo: ");
        double radio = Double.parseDouble(entrada.capturar());
        return new FiguraCirculo(radio);
    }
}
```

Clase FiguraCirculo

```
public class FiguraCirculo extends Figura {
    private double radio;
    public FiguraCirculo(double radio) {
        this.radio = radio;
    }
    public double CalcularArea() {
        return Math.PI * radio*radio;
    }
}
```

Clase FabricaTriangulo

```
import Abstract_Factory.*;
import java.io.IOException;

public class FabricaTriangulo extends FabricaFiguras {
    public Figura crearFigura() throws IOException {
        ProductoSalida salida = new SalidaTerminal();
        ProductoEntrada entrada = new EntradaTerminal();
        salida.enviar("Ingresa un lado del triagnulo: ");
        double lado1 = Double.parseDouble(entrada.capturar());
        salida.enviar("Ingresa otro lado del triangulo: ");
```

```
        double lado2 = Double.parseDouble(entrada.capturar());
        salida.enviar("Ingresa el ultimo lado del triangulo: ");
        double lado3 = Double.parseDouble(entrada.capturar());
        return new FiguraTriangulo(lado1, lado2, lado3);
    }
}
```

Clase FiguraTriangulo

```
public class FiguraTriangulo extends Figura {
    private double lado1;
    private double lado2;
    private double lado3;

    public FiguraTriangulo(double lado1, double lado2, double lado3) {
        this.lado1 = lado1;
        this.lado2 = lado2;
        this.lado3 = lado3;
    }
    public double CalcularArea() {
        double SemiPerimetro = (lado1 + lado2 + lado3) / 2;
        return Math.sqrt(SemiPerimetro*(SemiPerimetro-lado1)*(SemiPerimetro-
      lado2)*(SemiPerimetro-lado3));
    }
}
```

Clase Operaciones

```
public abstract class Operaciones{
    public abstract double Operar(double Operador1, double Operador2);
}
```

Clase FabricaOperaciones

```
public abstract class FabricaOperaciones{
    public abstract Operaciones crearOperaciones();
}
```

Clase FabricaSuma

```
public class FabricaSuma extends FabricaOperaciones {
    public Operaciones crearOperaciones() {
        return new OperacionSuma();
    }
```

```
}
```

Clase OperacionSuma

```
public class OperacionSuma extends Operaciones {
    public double Operar(double a, double b) {
        return a + b;
    }
}
```

Clase FabricaResta

```
public class FabricaResta extends FabricaOperaciones {
    public Operaciones crearOperaciones() {
        return new OperacionResta();
    }
}
```

Clase OperacionResta

```
public class OperacionResta extends Operaciones {
    Operaciones suma = new OperacionSuma();
    public double Operar(double a, double b) {
        return Math.abs(suma.Operar(a, -b));
    }
}
```

Clase FabricaMultiplicacion

```
public class FabricaMultiplicacion extends FabricaOperaciones {
    public Operaciones crearOperaciones() {
        return new OperacionMultiplicacion();
    }
}
```

Clase OperacionMultiplicacion

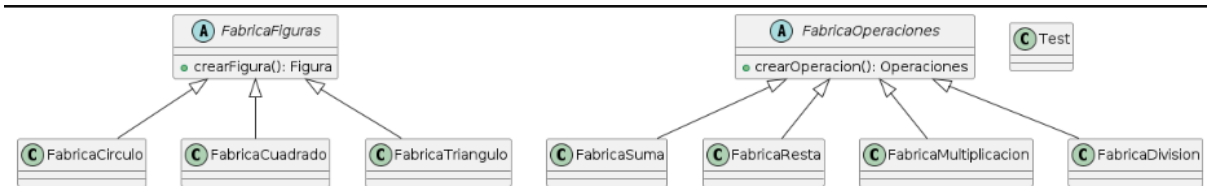
```
public class OperacionMultiplicacion extends Operaciones {
    Operaciones suma = new OperacionSuma();
    public double Operar(double a, double b) {
        double c = 0;
        for (int i = 0; i < b; i++) {
            c = suma.Operar(c, a);
        }
        return c;
    }
}
```

Clase FabricaDivision

```java
public class FabricaDivision extends FabricaOperaciones {
    public Operaciones crearOperaciones() {
        return new OperacionDivision();
    }
}
```

Clase OperacionDivision

```java
public class OperacionDivision extends Operaciones {
    Operaciones resta = new OperacionResta();
    public double Operar(double a, double b) {
        if (b == 0) {
            throw new ArithmeticException("No se puede dividir por cero");
        }
        boolean esNegativo = (a < 0) != (b< 0);
        double dividendo = Math.abs(a);
        double divisor = Math.abs(b);
        double cociente = 0;
        while (dividendo >= divisor) {
            a = resta.Operar(a, b);
            cociente++;
        }
        return esNegativo ? -cociente : cociente;
    }
}
```



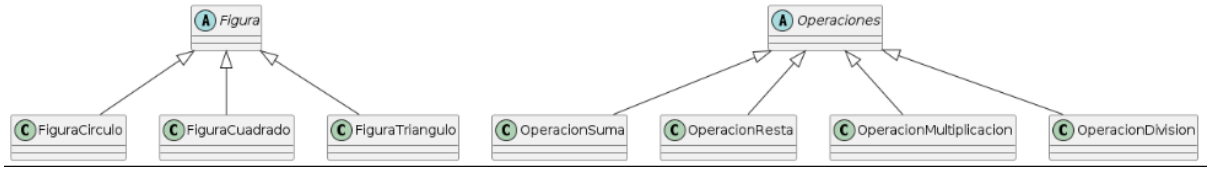


Figura
FiguraCirculo
FiguraCuadrado
FiguraTriangulo
Operaciones
OperacionSuma
OperacionResta
OperacionMultiplicacion
OperacionDivision