Abstract Factory

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

public class Test {
    public static void main(String[] args) throws IOException {
        SumaCliente suma = new SumaCliente();
        suma.sumar();
        suma.restar();
        suma.multiplicar();
        suma.dividir();
    }
}
Salida terminal
```

Ingresa un numero:

656

Ingresa otro numero:

646

El resultado es: 1302

Ingresa un numero:

213

Ingresa otro numero:

798

El resultado es: 585

Ingresa un numero:

456432

Ingresa otro numero:

641

El resultado es: 292572912

Ingresa el dividendo:

32185

Ingresa el divisor:

54

Clase SumarCliente

```
package Abstract_Factory;
import java.io.IOException;
public class SumaCliente {
    int operando1, operando2;
    Operaciones operaciones = new Operaciones();
    public void sumar() throws IOException{
        Cliente cliente = new Cliente(new FabricaTerminal());
        cliente.salida.enviar("Ingresa un numero: ");
        operando1 = Integer.parseInt(cliente.entrada.capturar());
        cliente.salida.enviar("Ingresa otro numero: ");
        operando2 = Integer.parseInt(cliente.entrada.capturar());
        cliente.salida.enviar("El resultado es: " +
operaciones.sumar(operando1, operando2));
    }
    public void restar() throws IOException{
        Cliente cliente = new Cliente(new FabricaTerminal());
        cliente.salida.enviar("Ingresa un numero: ");
        operando1 = Integer.parseInt(cliente.entrada.capturar());
        cliente.salida.enviar("Ingresa otro numero: ");
        operando2 = Integer.parseInt(cliente.entrada.capturar());
        cliente.salida.enviar("El resultado es: " +
operaciones.restar(operando1, operando2));
    }
    public void multiplicar() throws IOException{
        Cliente cliente = new Cliente(new FabricaTerminal());
        cliente.salida.enviar("Ingresa un numero: ");
        operando1 = Integer.parseInt(cliente.entrada.capturar());
        cliente.salida.enviar("Ingresa otro numero: ");
        operando2 = Integer.parseInt(cliente.entrada.capturar());
        cliente.salida.enviar("El resultado es: " +
operaciones.multiplicar(operando1, operando2));
    }
    public void dividir() throws IOException{
        Cliente cliente = new Cliente(new FabricaTerminal());
        cliente.salida.enviar("Ingresa el dividendo: ");
        operando1 = Integer.parseInt(cliente.entrada.capturar());
        cliente.salida.enviar("Ingresa el divisor: ");
        operando2 = Integer.parseInt(cliente.entrada.capturar());
        cliente.salida.enviar("El resultado es: " +
operaciones.dividir(operando1, operando2));
    }
}
```

Clase operaciones

```java
package Abstract_Factory;
public class Operaciones {
    public int sumar(int a, int b) {
        return a + b;
    }
    public int multiplicar(int a, int b) {
        int c = 0;
        for (int i = 0; i < b; i++) {
            c = sumar(c, a);
        }
        return c;
    }
    public int restar(int a, int b) {
        return Math.abs(sumar(a, -b));
    }
    public int dividir(int a, int b) {
        if (b == 0) {
            throw new ArithmeticException("No se puede dividir por cero");
        }
        int dividendo = Math.abs(a);
        int divisor = Math.abs(b);
        int cociente = 0;
        while (dividendo >= divisor) {
            a = restar(a, b);
            cociente++;
        }
        return cociente;
    }
}
```

Clase Cliente

```java
package Abstract_Factory;
import java.io.IOException;
public class Cliente{
        FabricaAbstracta fabrica;
        ProductoEntrada entrada;
        ProductoSalida salida;
        public Cliente (FabricaAbstracta fabrica) throws IOException{
            this.fabrica = fabrica;
            entrada = fabrica.crearEntrada();
            salida = fabrica.crearSalida();
    }
}
```

Clase FabricaAbstracta

```
package Abstract_Factory;
public abstract class FabricaAbstracta {
    public abstract ProductoSalida crearSalida();
    public abstract ProductoEntrada crearEntrada();
}
```

Fabricas Terminal, Grafico, Archivo

```
package Abstract_Factory;
public class FabricaTerminal extends FabricaAbstracta{
    public ProductoSalida crearSalida(){
        return new SalidaTerminal();
    }
    public ProductoEntrada crearEntrada(){
        return new EntradaTerminal();
    }
}
```

```
package Abstract_Factory;
public class FabricaGrafico extends FabricaAbstracta{
    public ProductoSalida crearSalida(){
        return new SalidaGrafico();
    }
    public ProductoEntrada crearEntrada(){
        return new EntradaGrafico();
    }
}
```

```
package Abstract_Factory;
public class FabricaArchivo extends FabricaAbstracta {
    public SalidaArchivo crearSalida(){
        return new SalidaArchivo();
    }

    public EntradaArchivo crearEntrada(){
        return new EntradaArchivo();
    }
}
```

Clase ProductoSalida

```java
package Abstract_Factory;
public abstract class ProductoSalida{
    public abstract void enviar(String mensaje);
}
```

Clase ProductoEntrada

```java
package Abstract_Factory;
import java.io.IOException;
public abstract class ProductoEntrada {
    public abstract String capturar() throws IOException;
}
```