Inyección de Dependencias en Java: Ejemplo Práctico Conversor de divisas

El siguiente código ilustra de manera clara el concepto de inyección de dependencias en Java, un patrón de diseño que promueve la flexibilidad y la extensibilidad del código al separar la creación de objetos y su comportamiento.

Clase Principal (Principal.java)

El método principal main crea instancias de la clase Usuario y utiliza la clase Inyector para inyectar la dependencia de la conversión en cada usuario. Luego, invoca el método realizaConversion en cada usuario.

```
public class Principal {
    public static void main(String[] args) {
        Usuario u1 = new Usuario('D',10.5);
        Usuario u2 = new Usuario('E',10);
        Usuario u3 = new Usuario('Y',10);

        Inyector.inyectarConversion(u1);
        Inyector.inyectarConversion(u2);
        Inyector.inyectarConversion(u3);

        u1.realizaConversion();
        u2.realizaConversion();
        u3.realizaConversion();
    }
}
```

Clase Usuario (Usuario.java)

La clase Usuario modela a un usuario con propiedades como tipo, cantidad y myDivisa. La inyección de dependencias se evidencia en el método realizaConversion, donde se invoca el método conversion de la interfaz Divisa. Los métodos get y set facilitan el acceso y la modificación de myDivisa.

```java
public class Usuario {

    private char tipo;
    private double cantidad;
    private Divisa myDivisa;

    public Usuario(char tipo, double cantidad){
        this.tipo = tipo;
        this.cantidad = cantidad;
    }

    public Divisa getmyDivisa(){
        return myDivisa;
    }

    public void setmyDivisa(Divisa myDivisa){
        this.myDivisa = myDivisa;
    }

    public char getTipo(){
        return this.tipo;
    }

    void realizaConversion(){
        myDivisa.conversion(cantidad);
    }
}
```

Clase Inyector (Inyector.java)

Esta clase encapsula la lógica de inyección de dependencias. Dependiendo del tipo de usuario, instancia la clase de divisa correspondiente (DivisaDolar, DivisaEuro, o DivisaYuan) y la asigna al usuario. Esta separación permite modificar el comportamiento de la conversión sin cambiar el código principal.

```java
public class Inyector{
    static void inyectarConversion(Usuario user){
        if (user.getTipo() == 'D') {
            user.setmyDivisa(new DivisaDolar());
        }

        if (user.getTipo() == 'E') {
            user.setmyDivisa(new DivisaEuro());
        }

        if (user.getTipo() == 'Y') {
            user.setmyDivisa(new DivisaYuan());
        }
    }
}
```

Interfaz Divisa (Divisa.java)

La interfaz Divisa define el método conversion, asegurando que todas las clases de divisa implementen este comportamiento.

```java
public interface Divisa {
    public void conversion(double cantidad);
}
```

Clase DivisaDolar (DivisaDolar.java)

Esta clase concreta implementa la interfaz Divisa y proporciona una implementación específica para la conversión de dólares a pesos mexicanos. Las demás clases (DivisaEuro y DivisaYuan) son similares a esta

```java
public class DivisaDolar implements Divisa{

    double tipoDeCambio = 20;

    @Override
    public void conversion(double cantidad){
        System.out.println(" 1 USD -> MXP : " +
tipoDeCambio + " pesos mex");
        System.out.println(cantidad +
        " Dolares en pesos mexicanos son: " +
(cantidad*tipoDeCambio) + "\n" );
    }

}
```