Design Patterns

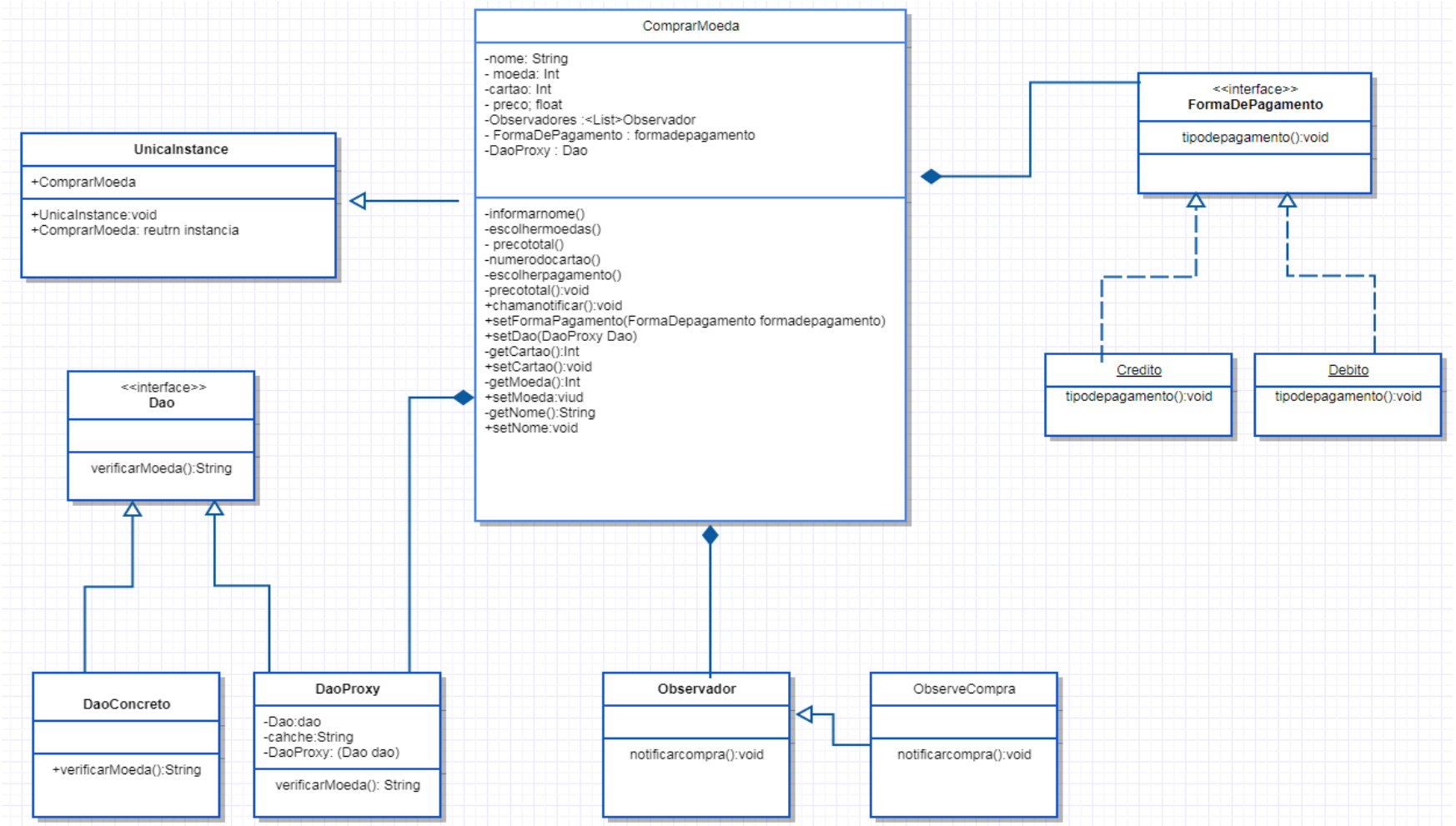
Projeto


Loja
Xingu
Compre, presenteia e desfrute do
melhor que a moeda virtual Xingu
pode te oferecer

Patterns Utilizados

- **Template**
- **Strategy**
- **Observer**
- **Proxy**
- **Singleton**

UML


ComprarMoeda
-nome: String
- moeda: Int
-cartao: Int
- preco; float
-Observadores :<List>Observador
- FormaDePagamento : formadepagamento
-DaoProxy : Dao
-informarnome()
-escolhermoedas()
- precototal()
-numerodocartao()
-escolherpagamento()
-precototal():void
+chamanotificar():void
+setFormaPagamento(FormaDepagamento formadepagamento)
+setDao(DaoProxy Dao)
-getCartao():Int
+setCartao():void
-getMoeda():Int
+setMoeda:viud
-getNome():String
+setNome:void
UnicaInstance
+ComprarMoeda
+UnicaInstance:void
+ComprarMoeda: reutrn instancia
<<interface>>
FormaDePagamento
tipodepagamento():void
Credito
tipodepagamento():void
Debito
tipodepagamento():void
<<interface>>
Dao
verificarMoeda():String
DaoConcreto
+verificarMoeda():String
DaoProxy
-Dao:dao
-cahche:String
-DaoProxy: (Dao dao)
verificarMoeda(): String
Observador
notificarcompra():void
ObserveCompra
notificarcompra():void

Template

```
public void setFormaPagamento(FormaDePagamento formapagamento) {

    this.formapagamento = formapagamento;

}

public void setDao(DaoProxy dao) {

    this.dao = dao;

}

private List<Observador> obs = new ArrayList<>();

public void addobservador(Observador obs) {

    this.obs.add(obs);
}

public void comprando() {

    informarnome();
    escolhermoedas();
    precototal();
    numerodocartao();
    escolherpagamento();

}
```

Strategy

```java
public class Credito implements FormaDePagamento {



    @Override
    public void tipodepagamento(){

        System.out.println("Forma de Pagamento : Credito");
              System.out.println("Data de Validade : 25/19");
                System.out.println("Codigo CSV: 004");


    }
```

```java
public class Debito implements FormaDePagamento{

    @Override
    public void tipodepagamento(){

        System.out.println("Forma de Pagamento : Credito");
         System.out.println("Banco : Banco Do Brasil");
           System.out.println("Agencia : 0258");
    }
```

Observer

```
private List<Observador> obs = new ArrayList<>();

public void addobservador(Observador obs) {

    this.obs.add(obs);
}

 public void chamanotificar() {

    for (Observador o : obs) {

        o.notificarcompra();
    }}
```

```
public class ObserveCompra extends Observador{
public static final String ANSI_GREEN = "\u001B[32m";
public static final String ANSI_BLACK_BACKGROUND = "\u001B[40m";



    @Override
    public void notificarcompra()
    {

        System.out.println(ANSI_GREEN + "Compra Aprovada");

    }

}
```

Proxy

```java
public class DaoProxy implements Dao {

    private Dao dao;
    private String cache;

    public DaoProxy(Dao dao) {

        this.dao = dao;

    }

    @Override
    public String verificarMoeda() {
        System.out.println("Calculando os Pontos");

        if (cache == null) {

            cache = dao.verificarMoeda();

        } else {
            return cache;

        }
        return "Obrigado Pela Compra";

    }

}
```

```java
public class DaoConcreto implements Dao {

    public String cache;
    public static final String ANSI_RED = "\u001B[31m";

    @Override
    public String verificarMoeda() {

        if (cache == null) {

            System.out.println("Pontos Entregado Com Sucesso");

        }
        return ANSI_RED + "Você já recebeu os pontos , Não e possível receber noamente";

    }

}
```

Singleton

```
public class UnicaInstance {

    public static ComprarMoeda instancia;

    public UnicaInstance() {
        System.out.println("Criando uma Compra ...");
    }

    public static ComprarMoeda getInstancia() {
        //Se nao criei,crio
        if (instancia == null) {
            instancia = new ComprarMoeda();
            //Se ja criei, envio a instancia anteriormente
        }
        return instancia;

    }
}
```

Resultado

```
Criando uma Compra ...
            Dados Da Compra

Nome do comprador: Raphael
Quantidade de Moedas escolhidas 300
Valor Total: 600 R$
Numero do Cartão: 1891397
Forma de Pagamento : Credito
Data de Validade : 25/19
Codigo CSV: 004
Compra Aprovada

            Moedas

Calculando os Pontos
Pontos Entregado Com Sucesso
Obrigado Pela Compra
Calculando os Pontos
Você já recebeu os pontos , Não e possível receber novamente
CONSTRUÍDO COM SUCESSO (tempo total: 0 segundos)
```

```
Criando uma Compra ...
            Dados Da Compra

Nome do comprador: Raphael
Quantidade de Moedas escolhidas 1500
Valor Total: 3000 R$
Numero do Cartão: 1891397
Forma de Pagamento : Debito
Banco : Banco Do Brasil
Agencia : 0258
Compra Aprovada

            Moedas

Calculando os Pontos
Pontos Entregado Com Sucesso
Obrigado Pela Compra
Calculando os Pontos
Você já recebeu os pontos , Não e possível receber novamente
CONSTRUÍDO COM SUCESSO (tempo total: 0 segundos)
```