

ARQUITETURA E PROJETO DE SISTEMAS

Trabalho Final

Prazo: 12/08

Valor: 4.0

Prof. Igor Avila Pereira

Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Sul (IFRS)
Divisão de Computação

igor.pereira@riogrande.ifrs.edu.br

1. (1.0) Loja Virtual

Uma loja virtual possui alguns produtos a venda. Os produtos são livros, DVDs e brinquedos. Cada produto possui nome e preço. A mesma loja oferece promoções diferentes a cada mês. Uma promoção regular desconta cada produto em 10% mais um desconto extra varia de 5% a 10% dependendo do mês. Uma liquidação desconta 30% ao preço de cada produto. Há meses que não há promoção descrita.

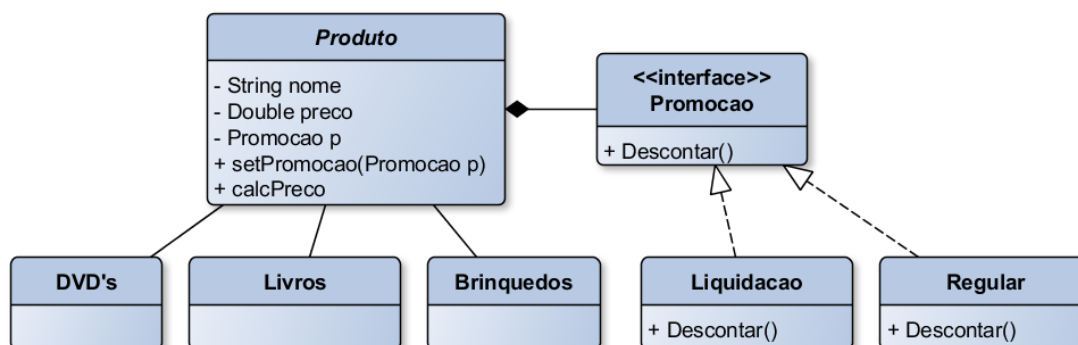


Figura 1. UML: Loja Virtual

2. (1.0) Empresa Pública

Em uma empresa pública, um cargo possui um nome e um valor de salário. Os cargos de ingresso são auxiliar, especialista, e gerente. Se alguém com um cargo entrar para um cargo político (Secretário, Prefeito ou Vereador) o salário deve ser incorporado. Um cargo pode ter mais de uma incorporação, os salários base são calculados como se segue:

- Auxiliar = Inicial + 1000
- Especialista = Inicial + 2500
- Gerente = Inicial + 3000

e as incorporações:

- Prefeito = Base + 4000

- Secretário = Base + 2000
- Vereador = Base + 5000

Para o cargo: Se alguém entra como especialista e incorpora vereador e prefeito o cargo fica: especialista incorporado como prefeito incorporado como vereador.

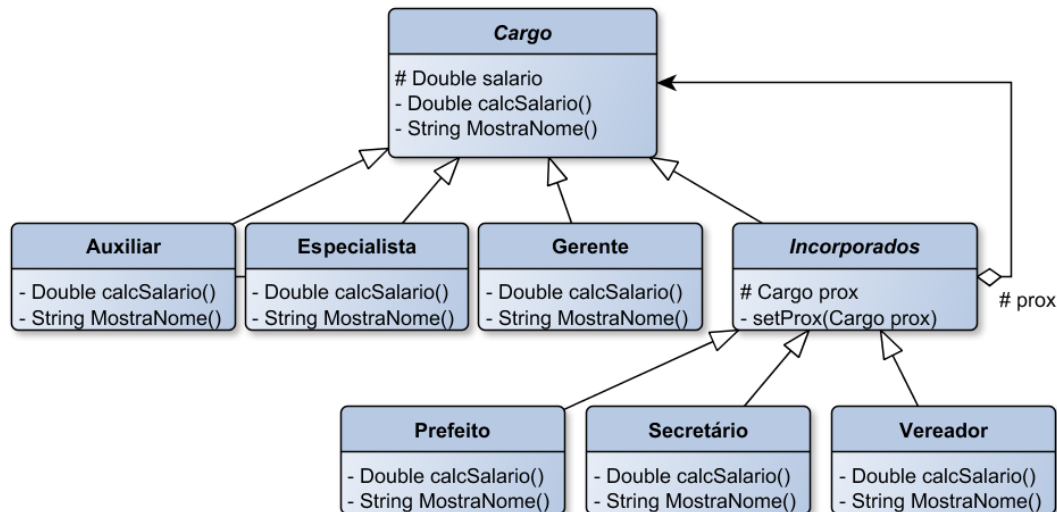


Figura 2. UML: Empresa Pública

3. (1.0) Revista de Fofoca Online

Pense em uma site de uma revista de fofocas online. Nela há uma lista de assinantes. Quando uma nova notícia é cadastrada, todos os assinantes são notificados. Há vários tipos de assinantes: pessoa física, pessoa jurídica, idosos e etc.

4. (1.0) JUnit

Utilize a biblioteca *JUnit* para desenvolver a classe e os casos de teste. Há casos de teste que provocam quebras no código. Encontre-os. O aluno que encontrar os casos de teste de falha receberá a integralidade da questão. Crie métodos de teste para cada método da classe. Declare asserções que provoquem erros. Uma outra exigência da questão é dizer o que cada método faz.

```

1 public class Oi {
2     public int x(int z) {
3         if (z == 1 || z == 0) {
4             return 1;
5         } else {
6             return z * x(z - 1);
7         }
8     }
9     public boolean w(int z) {
10        if (z == 2) {
11            return true;
12        }
  
```

```

13         if (z % 2 == 0) {
14             return false;
15         }
16         int o = 1;
17         while (z % o != 0 || o != z) {
18             o++;
19         }
20         return o == z;
21     }
22     public int y(int z) {
23         if (z == 0) {
24             return 0;
25         }
26         if (z == 1 || z == 2) {
27             return 1;
28         } else {
29             return y(z - 1) + y(z - 2);
30         }
31     }
32 }

```

Os métodos podem ser invocados da seguinte forma:

```

1 public class Test {
2     public static void main(String[] args) {
3         Oi oi = new Oi();
4         System.out.println(oi.x(2));
5         System.out.println(oi.w(2));
6         for (int i = 0; i < 10; i++) {
7             System.out.println(i+": "+oi.y(i));
8         }
9     }
10 }

```

Dica: Algumas Anotações JUnit

@Test, @Test(timeout = 1000), @Test(expected = Exception.class) e etc...

Anexos: UML

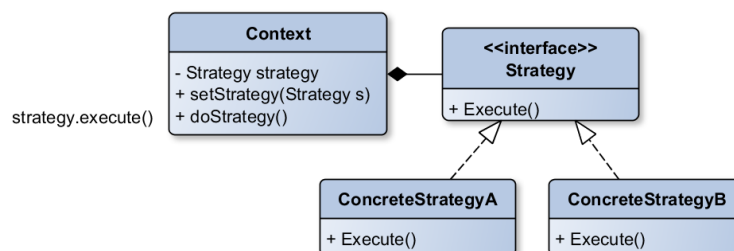


Figura 3. Padrão Strategy

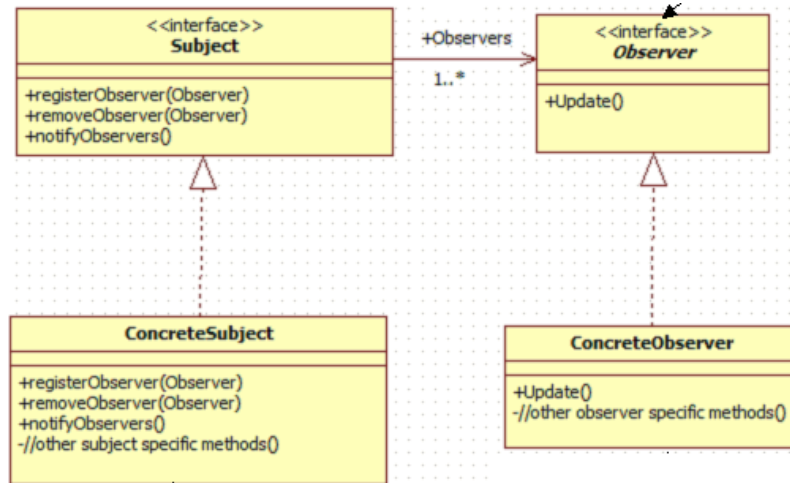


Figura 4. Padrão Observer

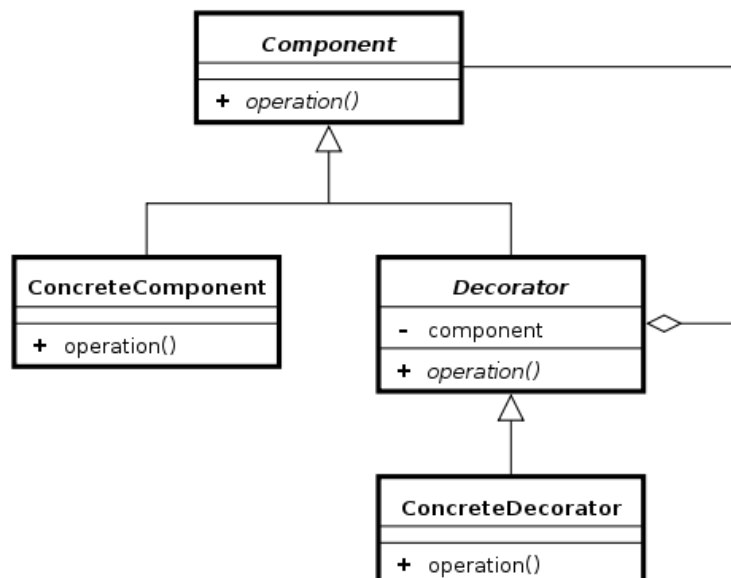


Figura 5. Padrão Decorator