

Exercício projeto bancov07. Exercício para ser feito em classe. (parte 3)

Ainda no projeto bancov07, vamos explorar um pouco o poder do polimorfismo.

- 1) Crie a classe Tesoureiro que tem tudo que um Funcionário tem, além de uma Bonificação de 20%.

```
3 public class Tesoureiro extends Funcionario {
4     @Override
5     public double getBonificacao() {
6         //super é mais descritivo uma vez que o atributo está na superclasse
7         return super.salario * 0.20;
8     }
9 }
```

- 2) Perceba que substituímos a palavra chave this pela palavra chave super. Assim o código fica mais legível. Afinal, salário é um atributo que vem da super classe. Faça a mesma alteração no getBonificação das demais subclasses.
- 3) Imagine que em nosso banco temos um departamento que gerencia gastos com bonificação e uma sala onde fazemos a contabilização das bonificações natalinas. Crie a classe ContabilizadorDeBonificacoes para representar esse departamento e o método contabilizaBonificacao para representar esta sala. Por hora trabalharemos apenas com bonificações de Gerentes.

```
3 public class ContabilizadorDeBonificacoes {
4     private double totalEmBonificacoes=0;
5
6     public void contabilizaBonificacao(Gerente g) {
7         this.totalEmBonificacoes += g.getBonificacao();
8     }
9
10    public double getTotalEmBonificacoes() {
11        return totalEmBonificacoes;
12    }
13 }
```

- 4) Crie a classe TestaContabilizadorDeBonificacoes.

```
7 public class TestaGerenciadorDeBonificacoes {
8     public static void main(String[] args) {
9         Gerente g1 = new Gerente();
10        g1.setNome("Rafael");
11        g1.setSalario(2000);
12        Gerente g2 = new Gerente();
13        g2.setNome("Fulano");
14        g2.setSalario(3000);
15
16        ContabilizadorDeBonificacoes contabilizador = new ContabilizadorDeBonificacoes();
17        contabilizador.contabilizaBonificacao(g1);
18        contabilizador.contabilizaBonificacao(g2);
19
20        System.out.println("Total em bonificacoes: R$" + contabilizador.getTotalEmBonificacoes());
21    }
22 }
```

- 5) Vamos acrescentar a essa classe de testes um Caixa e um Tesoureiro e vamos contabilizar a bonificação de um Caixa, além dos dois gerentes. Vamos esbarrar com um erro. O método contabilizaBonificacao não foi projetado para contabilizar bonificações de um caixa.

```
9 public class TestaGerenciadorDeBonificacoes {
10    public static void main(String[] args) {
11        Gerente g1 = new Gerente();
12        g1.setNome("Rafael");
13        g1.setSalario(2000);
14        Gerente g2 = new Gerente();
15        g2.setNome("Fulano");
16        g2.setSalario(3000);
17        Tesoureiro t1 = new Tesoureiro();
18        t1.setNome("Renata");
19        t1.setSalario(1500);
20        Caixa c1 = new Caixa();
21        c1.setNome("Pedro");
22        c1.setSalario(1000);
23
24        ContabilizadorDeBonificacoes contabilizador = new ContabilizadorDeBonificacoes();
25        contabilizador.contabilizaBonificacao(g1);
26        contabilizador.contabilizaBonificacao(g2);
27        contabilizador.contabilizaBonificacao(c1);
28
29        System.out.println("Total em bonificacoes: R$" + contabilizador.getTotalEmBonificacoes());
30    }
31 }
```

- 6) Uma boa solução para o problema acima seria sobrecarregar o método contabilizaBonificacao para que ele também possa contabilizar a bonificação de um Caixa. Veja:

```
3 public class ContabilizadorDeBonificacoes {
4     private double totalEmBonificacoes=0;
5
6     public void contabilizaBonificacao(Gerente g) {
7         this.totalEmBonificacoes += g.getBonificacao();
8     }
9     //Sobrecarregando...
10    public void contabilizaBonificacao(Caixa c) {
11        this.totalEmBonificacoes += c.getBonificacao();
12    }
13
14    public double getTotalEmBonificacoes() {
15        return totalEmBonificacoes;
16    }
17 }
```

- 7) Tente rodar novamente e verá que agora tudo funcionou perfeitamente. Agora queremos contabilizar a bonificação de um Tesoureiro. Para isso, basta sobrecarregar novamente o mesmo método. No entanto, criar um método novo para cada subtipo de Funcionario que queremos contabilizar não é uma boa prática. Gerente, Caixa e Tesoureiro possuem algumas coisas em comum. Todos tem um método getBonificacao e todos são Funcionários!! Use o polimorfismo a seu favor!!! Veja uma solução mais elegante:

```
3 public class ContabilizadorDeBonificacoes {
4     private double totalEmBonificacoes=0;
5     //Agora posso receber todos os subtipos de Funcionario
6     public void contabilizaBonificacao(Funcionario f) {
7         //O getBonificacao invocado vai depender do objeto que foi passado como argumento
8         //Pode ser Gerente, Caixa, Tesoureiro ....
9         this.totalEmBonificacoes += f.getBonificacao();
10    }
11
12    public double getTotalEmBonificacoes() {
13        return totalEmBonificacoes;
14    }
15 }
```

- 8) Modifique a classe TestaContabilizadorDeBonificacoes para contabilizar também um Tesoureiro.

```
9 public class TestaGerenciadorDeBonificacoes {
10    public static void main(String[] args) {
11        Gerente g1 = new Gerente();
12        g1.setNome("Rafael");
13        g1.setSalario(2000);
14        Gerente g2 = new Gerente();
15        g2.setNome("Fulano");
16        g2.setSalario(3000);
17        Tesoureiro t1 = new Tesoureiro();
18        t1.setNome("Renata");
19        t1.setSalario(1500);
20        Caixa c1 = new Caixa();
21        c1.setNome("Pedro");
22        c1.setSalario(1000);
23
24        ContabilizadorDeBonificacoes contabilizador = new ContabilizadorDeBonificacoes();
25        contabilizador.contabilizaBonificacao(g1);
26        contabilizador.contabilizaBonificacao(g2);
27        contabilizador.contabilizaBonificacao(c1);
28        contabilizador.contabilizaBonificacao(t1);
29
30        System.out.println("Total em bonificacoes: R$"+contabilizador.getTotalEmBonificacoes());
31    }
32 }
```

- 9) Nosso banco agora deve ter um Diretor. Todo Diretor é um Gerente e tem uma bonificação de 50% sobre o salário. Crie essa classe.

```
3 public class Diretor extends Gerente {  
4     @Override  
5     public double getBonificacao() {  
6         return super.salario * 0.50;  
7     }  
8 }
```

10) Se quisermos contabilizar a bonificação de um Diretor não teremos nenhum problema. Afinal, todo Diretor é um Gerente e, conseqüentemente, é um Funcionário! Quando criamos contabilizaBonificacao, nem imaginamos que haveria um Caixa, um Tesoureiro ou um Diretor. Ainda assim, tudo funciona perfeitamente. Assim que deve ocorrer com um sistema bem projetado. As modificações futuras devem causar pouco ou nenhum impacto para a aplicação.

11) Experimente mudar as regras do contabilizaBonificacao. Exemplo: Agora só serão contabilizadas as bonificações de Gerente. Você poderá mudar a assinatura do método e ele agora passará a aceitar apenas Diretor e Gerente. Afinal, Diretor é Diretor e Gerente. Faça esse Teste e depois volte o código ao normal.