

Exercício projeto bancov06. Exercício para ser feito em classe.

- 1) Abra o eclipse e crie um projeto chamado bancov06 a partir do projeto bancov05. Mantenha as classes Cliente e Conta.
- 2) Crie a classe Funcionario com nome e salário. A classe Funcionario também deve ter um método getBonificacao() que retorna o valor da bonificação natalina de um funcionário que é de 10% sobre seu salário.

```
2 public class Funcionario {
3     private String nome;
4     private double salario;
5     public Funcionario() {
6     }
7     public double getBonificacao() {
8         return this.salario * 0.10;
9     }
10    public String getNome() {
11        return this.nome;
12    }
13    public void setNome(String nome) {
14        this.nome = nome;
15    }
16    public double getSalario() {
17        return this.salario;
18    }
19    public void setSalario(double salario) {
20        this.salario = salario;
21    }
22 }
```

- 3) Crie a classe TestaFuncionario conforme código abaixo.

```
2 public class TestaFuncionario {
3     public static void main(String[] args) {
4         Funcionario func = new Funcionario();
5         func.setNome("Rafael");
6         func.setSalario(2000);
7
8         System.out.println("Nome: "+func.getNome());
9         System.out.println("Salario: R$"+func.getSalario());
10        System.out.println("Bonificacao: R$"+func.getBonificacao());
11    }
12 }
```

- 4) Agora nosso banco também deve poder representar o Gerente que tem as mesmas características de um Funcionario, além do atributo senha do tipo inteiro e de um método autentica que recebe uma senha, compara com a própria senha e retorna um booleano informando se as senhas são iguais.

```

1 public class Funcionario {
2     private String nome;
3     private double salario;
4     private int senha;
5     public int getSenha() {
6         return this.senha;
7     }
8     public void setSenha(int senha) {
9         this.senha = senha;
10    }
11    public boolean autentica(int senha) {
12        if(this.senha==senha)
13            return true;
14        return false;
15    }
16    public Funcionario() {}
17    public double getBonificacao() {
18        return this.salario * 0.10;
19    }
20    public String getNome() {
21        return this.nome;
22    }
23    public void setNome(String nome) {
24        this.nome = nome;
25    }
26    public double getSalario() {
27        return this.salario;
28    }
29    public void setSalario(double salario) {
30        this.salario = salario;
31    }
32 }

```

- 5) A bonificação de um Gerente é de 30% sobre o salário. Para resolver esse problema criaremos um atributo chamado tipoDeFuncionario. Se seu valor for 0 trata-se de um funcionário comum. Se o valor for 1, trata-se de um gerente, se for 2 trata-se de um caixa cuja bonificação deve ser de 15%. Além do novo atributo, acrescente um if-else ao método getBonificacao().

```

1 public class Funcionario {
2     private String nome;
3     private double salario;
4     private int senha;
5     private int tipoDeFuncionario=0;
6     //construtor
7     public Funcionario() {}
8
9     public double getBonificacao() {
10        if(this.tipoDeFuncionario==2)
11            return this.salario * 0.15;
12        else if(this.tipoDeFuncionario==1)
13            return this.salario * 0.30;
14        else
15            return this.salario * 0.10;
16    }
17    public int getTipoDeFuncionario() {
18        return this.tipoDeFuncionario;
19    }
20    public void setTipoDeFuncionario(int tipoDeFuncionario) {
21        this.tipoDeFuncionario = tipoDeFuncionario;
22    }
23 }

```

- 6) Faça pequenas alterações me TestaFuncionario e teste definir o tipo com os valores 0,1 e 2.

```

1
2 public class TestaFuncionario {
3     public static void main(String[] args) {
4         Funcionario func = new Funcionario();
5         func.setNome("Rafael");
6         func.setSalario(2000);
7         func.setTipoDeFuncionario(1);
8
9         System.out.println("Nome: "+func.getNome());
10        System.out.println("Salario: R$"+func.getSalario());
11        System.out.println("Bonificacao: R$"+func.getBonificacao());
12    }
13 }

```

```

<terminated> TestaFuncionario [Java Applicat
Nome: Rafael
Salario: R$2000.0
Bonificacao: R$600.0

```

Perceba que tudo vai funcionar corretamente. No entanto, um funcionário comum não tem senha e nem um método autentica. O mesmo vale para um caixa. Isso fere o princípio da alta coesão e o princípio da responsabilidade única se pensarmos em nossa classe. Se pensarmos que um caixa também deve ter um atributo numeroDoGuiche, a coisa fica pior. Nossa classe Funcionário está uma bagunça.

Se pararmos para pensar, o que gerente e caixa tem em comum?

Resposta: Ambos são funcionários e possuem nome, salário e uma bonificação natalina.

Ambos são funcionários!! Gerente é um Funcionário. Caixa é um Funcionário!

Logo, tudo o que for comum fica na classe Funcionário e o que for específico fica nas classes Gerente e Caixa. Essas classes são uma extensão de Funcionário. Um caso clássico de Herança.

- 7) Agora que já ficou claro o conceito de Herança, enxugue a classe Funcionario e crie as classes Gerente e Caixa.

Funcionario

```
1 public class Funcionario {
2     private String nome;
3     private double salario;
4     //construtor
5     public Funcionario() {}
6
7     public double getBonificacao() {
8         return this.salario * 0.10;
9     }
10    public String getNome() {
11        return this.nome;
12    }
13    public void setNome(String nome) {
14        this.nome = nome;
15    }
16    public double getSalario() {
17        return this.salario;
18    }
19    public void setSalario(double salario) {
20        this.salario = salario;
21    }
22 }
```

Gerente (com erro)

```
2 public class Gerente extends Funcionario {
3     private int senha;
4
5     public int getSenha() {
6         return this.senha;
7     }
8
9     public void setSenha(int senha) {
10        this.senha = senha;
11    }
12
13    @Override
14    public double getBonificacao() {
15        return this.salario * 0.30;
16    }
17
18    public boolean autentica(int senha) {
19        if (this.senha == senha)
20            return true;
21        return false;
22    }
23 }
```

Caixa(com erro)

```

2 public class Caixa extends Funcionario {
3     private int numDoGuiche;
4
5     public int getNumDoGuiche() {
6         return numDoGuiche;
7     }
8
9     public void setNumDoGuiche(int numDoGuiche) {
10        this.numDoGuiche = numDoGuiche;
11    }
12
13    @Override
14    public double getBonificacao() {
15        return this.salario * 0.15;
16    }
17 }

```

- 8) Os métodos getBonificacao de Gerente e Caixa estão com erro porque estão definidos como private em Funcionario. Altere o nível de visibilidade dos atributos de Funcionario para protected e os erros irão sumir.

```

1 public class Funcionario {
2     protected String nome;
3     protected double salario;
4     //construtor
5     public Funcionario() {}

```

- 9) Em TestaFuncionario, experimente criar func como Funcionario, Gerente e Caixa (um de cada vez) e verá que todos funcionam perfeitamente.

```

2 public class TestaFuncionario {
3     public static void main(String[] args) {
4         //Funcionario func = new Funcionario();
5         //Caixa func = new Caixa();
6         Gerente func = new Gerente();
7         func.setNome("Rafael");
8         func.setSalario(2000);
9
10        System.out.println("Nome: "+func.getNome());
11        System.out.println("Salario: R$"+func.getSalario());
12        System.out.println("Bonificacao: R$"+func.getBonificacao());
13    }
14 }

```