

Exercícios Obrigatórios:

Agora o objetivo é criar um sistema para gerenciar funcionários do banco. Este modelo será utilizado e modificado em exercícios posteriores. **Não deixe de fazer!**

- 1) Modele (apenas esboce uma classe em um papel) um funcionário que deve ter os seguintes atributos:
 - nome;
 - departamento (onde o funcionário trabalha);
 - salário;
 - ativo (um atributo que indica se o funcionário ainda trabalha na empresa ou não).Você vai precisar criar alguns métodos. Dentre deles, crie o método `aumentarSalario`, que deverá aumentar o salário do funcionário de acordo com o percentual passado como argumento. Crie também um método `demite`, que não recebe parâmetro algum, apenas muda o valor booleano que indica que o funcionário não trabalha mais no banco.
- 2) Abra o projeto chamado projeto-banco, criado anteriormente;
- 3) Escreva uma classe Java a partir do modelo acima. Teste-a usando a bancada do BlueJ e depois um programa (outra classe que tenha o método `main`). Um esboço da classe:

```
public class Funcionario {  
    private String nome, departamento;  
    private double salario;  
    private boolean ativo;  
  
    public void aumentarSalario(double percentual){  
        //Implementar...  
    }  
  
    public void demite(){  
        //Implementar...  
    }  
  
    //Métodos get e set...
```

Boa prática de programação: você deve compilar seu arquivo Java sem que tenha terminado de escrever toda a classe. Isso evitará que você siga por um caminho errado ou que encontre dezenas de erros de uma vez só na primeira compilação.

Crie os atributos e compile. Se estiver tudo certo a cada novo método declarado compile novamente.

3.1) Crie uma classe chamada `TestaFuncionario`:

Um esboço do programa (classe que possui o `main`):

```

public class TestaFuncionario {

    public static void main(String[] args) {
        Funcionario funcionario1 = new Funcionario();
        Funcionario funcionario2 = new Funcionario();

        funcionario1.setNome("Fulano de Tal");
        funcionario1.setSalario(3000);
        System.out.print("O funcionário "+funcionario1.getNome());
        System.out.println(" tem um salário de "+funcionario1.getSalario());
        funcionario1.aumentarSalario(15);
        System.out.print("Depois do aumento, o funcionário "+funcionario1.getNome());
        System.out.println(" tem um salário de "+funcionario1.getSalario());

        funcionario2.setNome("Rafael");
        funcionario2.setSalario(1000);
        System.out.print("O funcionário "+funcionario2.getNome());
        System.out.println(" tem um salário de "+funcionario2.getSalario());
    }
}

```

LEMBRE-SE: As classes de teste (que possuem o método main) não precisam ser instanciadas. Basta chamar o método estático main usando o botão direito do mouse sobre a classe.

Faça outros testes. Use todos os atributos e métodos. Imprima outros atributos e de forma mais completa.

IMPORTANTE: não se esqueça de seguir a convenção Java, isto é: NomeDeClasse, nomeDeAtributo, nomeDeVariavel, nomeDeMetodo, etc...

- 4) Crie um método mostra(), que simplesmente imprime, linha a linha, todos os atributos de um funcionário. Assim, você evita ter que ficar copiando e colando um System.out.println a cada mudança de estado de seus atributos. Você apenas vai chamar:

```
funcionario1.mostra();
```

Produza as saídas desejadas.

Veja abaixo:

```

Nome: Paulo
Salário: 2100.0
Departamento: Compras
Está na empresa? Sim

```

```

Nome: Paulo
Salário: 2500.0
Departamento: Compras
Está na empresa? Sim

```

```

Nome: Paulo
Salário: 2500.0
Departamento: Compras
Está na empresa? Não

```

A implementação do método ficaria mais ou menos assim:

```

void mostra() {
    System.out.println("Nome: "+this.nome);
    // Imprimir outros atributos ...
}

```

Mais adiante veremos uma solução muito mais elegante para mostrar um objeto como string através do método ToString.

- 5) Escreva um novo programa TestaFuncionario2, instanciando dois funcionários através do new e comparando-os com ==. E se eles tiverem os mesmos valores para seus atributos? Veja:

```

public class TestaFuncionario2 {
    public static void main(String[] args) {
        Funcionario funcionario1 = new Funcionario();
        Funcionario funcionario2 = new Funcionario();

        funcionario1.setNome("Fulano de Tal");
        funcionario1.setSalario(3000);

        funcionario2.setNome("Fulano de Tal");
        funcionario2.setSalario(3000);

        if(funcionario1==funcionario2)
            System.out.println("IGUAIS");
        else
            System.out.println("DIFERENTES");
    }
}

```

O que vai ser impresso?

- 6) Desta vez crie duas referências para o mesmo funcionário e compare-os novamente:

```

public class TestaFuncionario2 {
    public static void main(String[] args) {
        Funcionario funcionario1 = new Funcionario();
        Funcionario funcionario2 = new Funcionario();

        funcionario1.setNome("Fulano de Tal");
        funcionario1.setSalario(3000);

        funcionario2 = funcionario1;

        if(funcionario1==funcionario2)
            System.out.println("IGUAIS");
        else
            System.out.println("DIFERENTES");
    }
}

```

E agora? O que vai ser impresso?

O que aconteceu no exercício 5? Quantos objetos foram criados? Quantos objetos vão ficar na memória?

- 7) Digamos que agora um funcionário vai ter mais um atributo chamado dataDeNascimento. Em vez de criar um atributo do tipo String para representá-lo,

vamos criar uma classe Data que vai conter 3 atributos do tipo String (dia, mês, ano). Nesta mesma classe, crie um método que retorne uma String que representará a data no formato dd/mm/aaaa.

- 8) Faça com que o atributo dataNascimento de Funcionário seja do tipo Data e que na declaração do atributo um objeto do tipo Data já seja criado:

```
private Data dataDeNascimento = new Data();
```

- 9) Modifique seu programa TestaFuncionario para que seja definida a data de nascimento do funcionário e que através do método mostra() da classe Funcionário seja exibida a data de nascimento do funcionário.

- 10) O que acontece quando você tenta acessar diretamente um atributo da classe? Como, por exemplo:

```
Funcionario.nome = "Paulo";
```