

Atividade 2

(fonte: Apostila Caelum)

Em relação a classe `Funcionario` implementada na aula passada, faça o que se pede:

- 1) Crie um método `mostra()`, que não recebe nem devolve parâmetro algum e simplesmente imprime todos os atributos do nosso funcionário. Dessa maneira, você não precisa ficar copiando e colando um monte de `System.out.println()` para cada mudança e teste que fizer com cada um de seus funcionários.
- 2) Construa dois funcionários com o `new` e compare-os com o `==`. E se eles tiverem os mesmos atributos? Para isso você vai precisar criar outra referência:

```
Funcionario f1 = new Funcionario();
f1.nome = "Fiodor";
f1.salario = 100;

Funcionario f2 = new Funcionario();
f2.nome = "Fiodor";
f2.salario = 100;

if (f1 == f2) {
    System.out.println("iguais");
} else {
    System.out.println("diferentes");
}
```

- 3) Crie duas referências para o mesmo funcionário, compare-os com o `==`. Tire suas conclusões. Para criar duas referências pro mesmo funcionário:

```
f1.nome = "Fiodor";
f1.salario = 100;

Funcionario f2 = f1;

O que acontece com o if do exercício anterior?
```

- 4) Em vez de utilizar uma `String` para representar a data, crie uma outra classe, chamada `Data`. Ela possui 3 campos `int`, para dia, mês e ano. Faça com que seu funcionário passe a usá-la. Teste usando o método `main` do programa principal.
- 5) Modifique seu método `mostra` para que ele imprima o valor da `dataDeEntrada` daquele Funcionário.
- 6) O que acontece se você tentar acessar um atributo diretamente na classe? Como, por exemplo:
`Conta.saldo = 1234;`
Esse código faz sentido? E este:
`Conta.saca(50);`
Faz sentido pedir para o esquema da conta sacar uma quantia?
- 7) Um método pode chamar ele mesmo. Chamamos isso de recursão. Resolva a série de fibonacci usando um método que chama ele mesmo. O objetivo é você criar uma classe, que possa ser usada da seguinte maneira:

```
Fibonacci fibo = new Fibonacci();
int i = fibo.calculaFibonacci(5);
System.out.println(i);
```

Aqui imprimirá 8, já que este é o sexto número da série.

Este método `calculaFibonacci` não pode ter nenhum laço, só pode chamar ele mesmo como método. Pense nele como uma função, que usa a própria função para calcular o resultado.