

Lista de Exercícios – unidade 3

1 – Ler o arquivo funcionarios.txt e mostrar a relação de funcionários do arquivo na console e no final mostrar o total de imposto de renda que será pago por funcionários do sexo feminino, masculino e o total de imposto pago por todos os funcionários.

Dica: Ler cada linha do arquivo como um String e usar o método **split** para separar o String em partes `String[] campos = linha.split("#")`. Para cada linha, instanciar um objeto da classe Funcionario abaixo, usar os métodos mostraFuncionario para mostrar os dados e o método de calculo de imposto para cada um.

Base de Cálculo (R\$)	Alíquota (%)	Parcela a deduzir do IR (em R\$)
Até 1.903,98	-	-
De 1.903,99 até 2.826,65	7,5	142,80
De 2.826,66 até 3.751,05	15	354,80
De 3.751,06 até 4.664,68	22,5	636,13
Acima de 4.664,68	27,5	869,36

```
public class Funcionario {

    String nome;
    char sexo;
    double salario;
    int dependentes;

    Funcionario(String nome, char sexo, double salario, int dependentes) {
        this.nome = nome;
        this.sexo = sexo;
        this.salario = salario;
        this.dependentes = dependentes;
    }

    public double impostoRenda() {
        double reducaoDep = 189.59 * dependentes;
        if (salario <= 1903.98)
            return 0;
        else if (salario <= 2826.65)
            return (salario * 7.5 / 100 - 142.80) - reducaoDep;
        else if (salario <= 3751.05)
            return (salario * 15 / 100 - 354.80) - reducaoDep;
        else if (salario <= 4664.68)
            return (salario * 22.5 / 100 - 636.13) - reducaoDep;
        else
            return ((salario * 27.5 / 100) - 869.36) - reducaoDep;
    }

    public String mostraFuncionario() {
        return nome + ' ' + sexo + ' ' + salario + ' ' + dependentes + ' ' +
        impostoRenda();
    }
}
```

2 – Fazer um programa para ler o arquivo e gravar todos os dados como objetos em outro arquivo usando ObjectOutputStream e serializando a classe Funcionario. Fazer outro programa que irá ler os objetos do novo arquivo (ObjectInputStream) e repetir o resultado do item 1.

3 - Executar testes de unidade em um método de cálculo de imposto de renda de uma classe Pessoa.

Para esse contexto, uma Pessoa tem como características: seu nome, salário, número de dependentes e idade.

As regras para o cálculo do imposto de renda são:

- Uma pessoa com idade maior ou igual a 65 anos não paga imposto.
- Para pessoas com menos de 65 anos o imposto é calculado de acordo com a seguinte tabela:

Salário	Percentual	Dedução
Até 1.787,77	-	-
De 1.787,78 até 2.679,29	7,5%	134,08
De 2.679,30 até 3.572,43	15%	335,03
De 3.572,44 até 4.463,81	22,5%	602,96
Acima de 4.463,81	27,5%	826,15

Etapas 1 do cálculo:

Verificar em faixa se enquadra o salário da pessoa, aplicar o percentual da faixa e do resultado, deduzir o valor de dedução da faixa.

Exemplo de cálculo do imposto:

salário 2000,00 → entra na faixa 2 (De 1.787,78 até 2.679,29)
imposto = $2000,00 * 7,5\% - 134,08 = 15,91$

Etapas 2 do cálculo:

Do total de imposto calculado, é deduzido R\$ 179,71 por dependente.

Ou seja, se o imposto calculado foi 300,00 e a pessoa tem 1 dependente, o imposto passa a ser $300,00 - 179,71$.

De acordo com esse contexto faça:

- uma classe **Pessoa** com seus atributos e um método **calculaImposto** que retorna o valor do imposto (*double*), para passar na série de teste da classe **TestePessoa**;
 - o código a seguir apresenta uma classe de teste (JUnit) - **TestePessoa** - com um método de teste para cada situação das regras apresentadas para o problema. Copie o código, crie um novo método para a regra de pessoas maiores de 65 anos com salários em todas as faixas.
 - Implemente o método **calculaImposto** da classe **Pessoa** com as regras necessárias para passar na série de teste.
-

Classes de Testes a ser usada:

TestePessoa.java

```
import static org.junit.Assert.*;

import org.junit.After;
import org.junit.AfterClass;
import org.junit.Before;
import org.junit.BeforeClass;
import org.junit.Test;

public class TestePessoa {

    @Test
    public void test1() {
        Pessoa p = new Pessoa("joão",1500.00,0,20);
        assertEquals(0,p.calculaImposto(),0);

        p = new Pessoa("maria",1500.00,1,50);
        assertEquals(0,p.calculaImposto(),0);

        System.out.println("Passou Teste 1 - primeira faixa");
    }

    @Test
    public void test2() {

        Pessoa p = new Pessoa("joão",2000.00,0,60);
        assertEquals(15.91,p.calculaImposto(),0.1);

        p = new Pessoa("joão",2000.00,2,50);
        assertEquals(0,p.calculaImposto(),0);

        System.out.println("Passou Teste 2 - segunda faixa");
    }

    @Test
    public void test3() {
        Pessoa p = new Pessoa("paulo",3500.00,0,20);
        assertEquals(189.97,p.calculaImposto(),0.1);

        p = new Pessoa("paulo",3500.00,1,20);
        assertEquals(10.26,p.calculaImposto(),0.1);

        p = new Pessoa("paulo",3500.00,2,40);
        assertEquals(0,p.calculaImposto(),0);
    }
}
```

```
        System.out.println("Passou Teste 3 - terceira faixa");
    }
```

```
@Test
public void test4() {
    Pessoa p = new Pessoa("paulo",4400.00,0,40);
    assertEquals(387.04,p.calculaImposto(),0.1);

    p = new Pessoa("paulo",4400.00,1,30);
    assertEquals(207.33,p.calculaImposto(),0.1);

    p = new Pessoa("paulo",4400.00,2,40);
    assertEquals(27.62,p.calculaImposto(),0.1);

    p = new Pessoa("paulo",4400.00,3,20);
    assertEquals(0,p.calculaImposto(),0);

    System.out.println("Passou Teste 4 - quarta faixa");
}
```

```
@Test
public void test5() {
    Pessoa p = new Pessoa("jose",7000.00,0,50);
    assertEquals(1098.85,p.calculaImposto(),0.1);

    p = new Pessoa("jose",7000.00,2,30);
    assertEquals(739.43,p.calculaImposto(),0.1);

    p = new Pessoa("paulo",7000.00,6,40);
    assertEquals(20.59,p.calculaImposto(),0.1);

    p = new Pessoa("paulo",7000.00,7,20);
    assertEquals(0,p.calculaImposto(),0);

    System.out.println("Passou Teste 5 - quinta faixa");
}
```

```
@BeforeClass
public static void inicializaClass() {
    System.out.println("iniciou série de teste");
}
```

```
@AfterClass
public static void finalizaClass() {
    System.out.println("Finalizou série de teste");
}
```

```
}
```