

```
package teste;
```

```
import empresa.Funcionario;
```

```
import empresa.OperacaoFuncionario;
```

```
import java.util.Calendar;
```

```
import java.util.GregorianCalendar;
```

```
import java.util.Scanner;
```

```
public class App {
```

```
    public static void main(String[] args) {
```

```
        Scanner scan = new Scanner(System.in);
```

```
        Funcionario vet [] = new Funcionario[2];
```

```
        int ano, mes, dia;
```

```
        Calendar hoje = Calendar.getInstance();
```

```
        /*System.out.println("Data/Hora atual: "+c.getTime());
```

```
        System.out.println("Ano: "+c.get(Calendar.YEAR));
```

```
        System.out.println("Mês: "+c.get(Calendar.MONTH));
```

```
        System.out.println("Dia: "+c.get(Calendar.DAY_OF_MONTH));*/
```

```
        for(int i = 0; i < vet.length; i++) {
```

```
            vet[i] = new Funcionario();
```

```
            System.out.println("Digite o nome do funcionario "+(i+1)+" : ");
```

```
            vet[i].nome = scan.nextLine();
```

```
            vet[i].dataNascimento = new GregorianCalendar();
```

```
            System.out.println("Digite a data de nascimento: dd mm aaaa: ");
```

```
            vet[i].dia = scan.nextInt();
```

```
            vet[i].mes = scan.nextInt() - 1;
```

```

        vet[i].ano = scan.nextInt();

        scan.nextLine();

        vet[i].dataNascimento.set(vet[i].ano, vet[i].mes, vet[i].dia);

        System.out.println("Digite o cpf: ");

        vet[i].cpf = scan.nextLine();

//        System.out.println("Digite o sexo do funcionario "+(i+1)+" (m pra
masculino e f pra feminino): ");
//        vet[i].sexo = scan.next().charAt(0);
//        scan.nextLine();
//        System.out.println("Digite o salario bruto: ");
//        vet[i].salarioBruto = scan.nextFloat();
//        scan.nextLine();
    }

    System.out.println("A idade do "+vet[0].nome+" é: "+vet[0].CalculaIdade(vet,
hoje));

    System.out.println("A idade do "+vet[1].nome+" é: "+vet[1].CalculaIdade(vet,
hoje));

    /*if(vet[0].FuncionarioIgual(vet) == 1) {
        System.out.println("\nHá Funcionarios iguais!");
    } else {
        System.out.println("\nNão há funcionários iguais.");
    }

    Funcionario.MaiorMenorSalario(vet);

    Funcionario.ContaSexo(vet);

    vet[0].ValidaNascimento(vet);
    vet[0].ValidaAdmissao(vet, hoje);*/

    //Funcionario.MaisVelhoNovo(vet, hoje);

```

```
Funcionario.ValidaFuncionario(vet, hoje);

for(int i = 0; i < vet.length; i++) {
    System.out.println(vet[i].toString());
}
}
}
```

```
package empresa;
```

```
import java.util.Calendar;
```

```
import java.util.GregorianCalendar;
```

```
public class Funcionario extends OperacaoFuncionario {
    String cpf, nome;
    char sexo;
    float salarioBruto;
    GregorianCalendar dataAdmissao;
    GregorianCalendar dataNascimento;
    int dia, mes, ano;

    public Funcionario() {
        this.dataAdmissao = new GregorianCalendar();
    }
}
```

```

        public Funcionario(String cpf, String nome, GregorianCalendar
dataNascimento) {

            this.cpf = cpf;

            this.nome = nome;

            this.dataNascimento = dataNascimento;

        }

```

```

        public Funcionario(String cpf, String nome, char sexo, double salarioBruto,
GregorianCalendar dataAdmissao) {

        }

```

```

        public int CalculaIdade(Funcionario vet[], Calendar hoje) {

            int idade = 0;

            if(this.mes < hoje.get(Calendar.MONTH)) {

                idade = hoje.get(Calendar.YEAR) - this.ano;

            } else if(this.mes > hoje.get(Calendar.MONTH)) {

                idade = (hoje.get(Calendar.YEAR) - this.ano) - 1;

            } else if(this.mes == hoje.get(Calendar.MONTH)) {

                if(this.dia > hoje.get(Calendar.DAY_OF_MONTH)) {

                    idade = (hoje.get(Calendar.YEAR) - this.ano) - 1;

                } else {

                    idade = hoje.get(Calendar.YEAR) - this.ano;

                }

            }

            return idade;

        }

```

```

        int ValidaCpf(Funcionario vet[]) {

            if(this.cpf.length() == 14) {

                System.out.println("CPF do funcionário "+this.nome+"
válido!");

```

```

        return 1;
    } else {
        System.out.println("CPF do funcionário "+this.nome+"
inválido.");
        return 0;
    }
}

```

```

int ValidaNascimento(Funcionario vet[]) {
    if(this.ano < 1920) {
        System.out.println("\nData de nascimento do
funcionário "+this.nome+" inválida.");
        return 0;
    } else {
        System.out.println("\nData de nascimento do
funcionário "+this.nome+" válida!");
        return 1;
    }
}

```

```

int ValidaAdmissao(Funcionario vet[], Calendar hoje) {
    if(hoje.get(Calendar.YEAR) < 1995) {
        System.out.println("Data de admissão inválida.");
        return 0;
    } else {
        System.out.println("Data de admissão válida!");
        return 1;
    }
}

```

```

public double SalarioLiquido(float salarioBruto) {
    double sLiquido;

```

```

        if (salarioBruto <= 3000) {
            sLiquido = salarioBruto - (salarioBruto * 0.17);
        } else {
            sLiquido = salarioBruto - (salarioBruto * 0.27);
        }
        return sLiquido;
    }

    public int FuncionarioIgual(Funcionario vet[]) {
        for(int i = 0; i < vet.length; i++) {
            for(int j = 1; j < vet.length; j++) {
                if(vet[i].nome.equals(vet[j].nome) &&
vet[i].cpf.equals(vet[j].cpf) && vet[i].sexo == vet[j].sexo) {
                    return 1;
                } else {
                    return 2;
                }
            }
        }
        return 0;
    }

    public String toString() {
        return ("Nome: "+this.nome+", CPF: "+this.cpf+", Data de Nascimento:
"+this.dia+"/"+this.mesNas+"/"+this.ano+", Salario Bruto: "+this.salarioBruto+", Sexo:
"+this.sexo+"\n");
    }
}

```

```
package empresa;
```

```
import java.util.Calendar;
```

```
public class OperacaoFuncionario {
```

```
    public static void ValidaFuncionario(Funcionario vet[], Calendar hoje) {  
        for(int i = 0; i < vet.length; i++) {  
            if(vet[i].ValidaNascimento(vet) + vet[i].ValidaAdmissao(vet, hoje) +  
vet[i].ValidaCpf(vet) == 3) {  
                System.out.println("Dados do funcionário "+vet[i].nome+"  
válidos!");  
            } else {  
                System.out.println("Dados do funcionário "+vet[i].nome+"  
inválidos.");  
            }  
        }  
    }  
}
```

```
    public static void ContaSexo(Funcionario vet[]) {  
        int contM = 0, contF = 0;  
        for(int i = 0; i < vet.length; i++) {  
            if(vet[i].sexo == 'm' || vet[i].sexo == 'M') {  
                contM++;  
            } else if(vet[i].sexo == 'f' || vet[i].sexo == 'F') {  
                contF++;  
            }  
        }  
        System.out.println("\nHá "+contM+" funcionários homens e "+contF+"  
funcionárias mulheres!");  
    }  
}
```

```

public static void MaiorMenorSalario(Funcionario vet[]) {

    double maior = vet[0].SalarioLiquido(vet[0].salarioBruto);

    double menor = vet[0].SalarioLiquido(vet[0].salarioBruto);

    String maiorSalario = "NULL", menorSalario = "NULL";

    for(int i = 0; i < vet.length; i++) {

        if(maior <= vet[i].SalarioLiquido(vet[i].salarioBruto)) {

            maior = vet[i].SalarioLiquido(vet[i].salarioBruto);

            maiorSalario = vet[i].nome;

        }

        if (menor >= vet[i].SalarioLiquido(vet[i].salarioBruto)) {

            menor = vet[i].SalarioLiquido(vet[i].salarioBruto);

            menorSalario = vet[i].nome;

        }

    }

    System.out.println("\nO maior salário líquido é de "+maiorSalario+" e o menor
    salário líquido é de "+menorSalario);

}

```

```

public static void MaisVelhoNovo(Funcionario vet[], Calendar hoje) {

    int novo = 200, velho = 0;

    String nomeVelho = "NULL", nomeNovo = "NULL";

    for(int i = 0; i < vet.length; i++) {

        if(vet[i].CalculaIdade(vet, hoje) < novo) {

            novo = vet[i].CalculaIdade(vet, hoje);

            nomeNovo = vet[i].nome;

        }

        if(vet[i].CalculaIdade(vet, hoje) > velho) {

            velho = vet[i].CalculaIdade(vet, hoje);

            nomeVelho = vet[i].nome;

        }

    }

}

```



```
        System.out.println("O funcionário mais velho é o "+nomeVelho+" com  
        "+velho+" anos. E o funcionário mais novo é o "+nomeNovo+" com "+novo+" anos.");  
    }
```

```
}
```