

```

package teste;

import empresa.Funcionario;
import empresa.OperacaoFuncionario;
import empresa.Programador;
import empresa.Analista;
import empresa.Gerente;

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[12];
        Programador pro[] = new Programador[6];
        Analista ana[] = new Analista[4];
        Gerente ger[] = new Gerente[2];
        int ano, mes, dia, x, op;

        Calendar hoje = Calendar.getInstance();

        int contPro = 0, contAna = 0, contGer = 0;
        for(int i = 0; i < vet.length; i++) {

            System.out.println("Que tipo de funcionário deseja registrar?
1 - Programador // 2 - Analista // 3 - Gerente");
            op = scan.nextInt();
            scan.nextLine();
            if(op == 1) {
                pro[contPro] = new Programador();
                System.out.println("Digite o nome do programador: ");
                pro[contPro].setNome(scan.nextLine());
                pro[contPro].setDataNascimento(new
GregorianCalendar());
                System.out.println("Digite a data de nascimento: dd mm
aaaa: ");
                pro[contPro].setDia(scan.nextInt());
                pro[contPro].setMes(scan.nextInt());
                pro[contPro].setAno(scan.nextInt());
                scan.nextLine();

                pro[contPro].getDataNascimento().set(pro[contPro].getAno(),
pro[contPro].getMes(), pro[contPro].getDia());
                System.out.println("Digite o cpf: ");
                pro[contPro].setCpf(scan.nextLine());
                System.out.println("Digite o sexo do funcionario
"+(contPro+1)+" (m pra masculino e f pra feminino): ");
                pro[contPro].setSexo(scan.next().charAt(0));
                scan.nextLine();
                System.out.println("Digite o salario bruto: ");
                pro[contPro].setSalarioBruto(scan.nextFloat());
                scan.nextLine();
            }
        }
    }
}

```

```

        contPro++;

    } else if(op == 2) {
        ana[contAna] = new Analista();
        System.out.println("Digite o nome do analista: ");
        ana[contAna].setNome(scan.nextLine());
        ana[contAna].setDataNascimento(new
GregorianCalendar());
        System.out.println("Digite a data de nascimento: dd mm
aaaa: ");

        ana[contAna].setDia(scan.nextInt());
        ana[contAna].setMes(scan.nextInt());
        ana[contAna].setAno(scan.nextInt());
        scan.nextLine();

        ana[contAna].getDataNascimento().set(ana[contAna].getAno(),
ana[contAna].getMes(), ana[contAna].getDia());
        System.out.println("Digite o cpf: ");
        ana[contAna].setCpf(scan.nextLine());
        System.out.println("Digite o sexo do funcionario
" +(contAna+1)+ " (m pra masculino e f pra feminino): ");
        ana[contAna].setSexo(scan.next().charAt(0));
        scan.nextLine();
        System.out.println("Digite o salario bruto: ");
        ana[contAna].setSalarioBruto(scan.nextFloat());
        System.out.println(ana[contAna].toString());
        contAna++;

    } else if(op == 3) {
        ger[contGer] = new Gerente();
        System.out.println("Digite o nome do gerente: ");
        ger[contGer].setNome(scan.nextLine());
        ger[contGer].setDataNascimento(new
GregorianCalendar());
        System.out.println("Digite a data de nascimento: dd mm
aaaa: ");

        ger[contGer].setDia(scan.nextInt());
        ger[contGer].setMes(scan.nextInt());
        ger[contGer].setAno(scan.nextInt());
        scan.nextLine();

        ger[contGer].getDataNascimento().set(ger[contGer].getAno(),
ger[contGer].getMes(), ger[contGer].getDia());
        System.out.println("Digite o cpf: ");
        ger[contGer].setCpf(scan.nextLine());
        System.out.println("Digite o sexo do funcionario
" +(contGer+1)+ " (m pra masculino e f pra feminino): ");
        ger[contGer].setSexo(scan.next().charAt(0));
        scan.nextLine();
        System.out.println("Digite o salario bruto: ");
        ger[contGer].setSalarioBruto(scan.nextFloat());
    }
}

```

```

    }
    int aux = 0;
    for(int i = 0; i < 6; i++) {
        vet[i] = pro[i];
    }
    aux = 0;
    for(int i = 6; i < 10; i++) {
        vet[i] = ana[aux];
        aux++;
    }
    aux = 0;
    for(int i = 10; i < 12; i++) {
        vet[i] = ger[aux];
        aux++;
    }
    for(int i = 0; i < vet.length; i++) {
        System.out.println(vet[i].toString());
    }

    System.out.println("A idade do "+vet[0].getNome()+" é:
"+vet[0].CalculaIdade(vet, hoje));
    System.out.println("A idade do "+vet[1].getNome()+" 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.");
    }

    OperacaoFuncionario.MaiorMenorSalario(vet);
    OperacaoFuncionario.ContaSexo(vet);

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

    OperacaoFuncionario.MaisVelhoNovo(vet, hoje);

    OperacaoFuncionario.ValidaFuncionario(vet, hoje);

    for(int i = 0; i < vet.length; i++) {
        System.out.println(vet[i].toString());
        vet[i].dataAposentadoria(vet, hoje);
    }
    System.out.println("Digite qual funcionario deseja comparar: ");
    x = scan.nextInt();
    vet[1].eMaisVelho(vet, x, hoje);

    System.out.println(vet[0].getBonificacao(vet));

    OperacaoFuncionario.AposentaFuncionarios(vet, hoje);

    System.out.println(ana.toString());
    System.out.println(ger.toString());
    System.out.println(pro.toString());

```

```

        OperacaoFuncionario.TotalBonificacoes(ana);
    }
}

```

```

package empresa;

import java.util.Calendar;
import java.util.GregorianCalendar;

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

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

    public Funcionario(String cpf, String nome, GregorianCalendar
dataNascimento) {
        this.cpf = cpf;
        this.setNome(nome);
        this.dataNascimento = dataNascimento;
    }

    public Funcionario(String cpf, String nome, char sexo, float
salarioBruto, GregorianCalendar dataAdmissao,
GregorianCalendar dataNascimento, int dia, int mes,
int ano) {
        this.cpf = cpf;
        this.nome = nome;
        this.sexo = sexo;
        this.salarioBruto = salarioBruto;
        this.dataAdmissao = dataAdmissao;
        this.dataNascimento = dataNascimento;
        this.dia = dia;
        this.mes = mes;
        this.ano = ano;
    }

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

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

    public 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].getNome().equals(vet[j].getNome()) &&
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.getNome()+", CPF: "+this.cpf+", Data de
Nascimento: "+this.dia+"/"+this.mes+"/"+this.ano+", Salario Bruto:
"+this.salarioBruto+", Sexo: "+this.sexo+"\n");
        }

        public double getBonificacao(Funcionario vet[]) {
            double bonificacao = (this.salarioBruto * 12);
            bonificacao = bonificacao * 0.02;
            System.out.println("A bonificação do funcionario
"+this.getNome()+" é "+bonificacao);
            return bonificacao;
        }

        public void dataAposentadoria(Funcionario vet[], Calendar hoje) {
            int anoAposenta = 0;
            if(this.sexo == 'm' || this.sexo == 'M') {
                anoAposenta = hoje.get(Calendar.YEAR) + 35;
            } else if(this.sexo == 'f' || this.sexo == 'F') {
                anoAposenta = hoje.get(Calendar.YEAR) + 30;
            }
            System.out.println("Data de Aposentadoria prevista do
funcionário " +this.getNome()+ ":
"+hoje.get(Calendar.DAY_OF_MONTH)+"/"+hoje.get(Calendar.MONTH)+"/"+anoAposenta);
        }

        public void eMaisVelho(Funcionario vet[], int x, Calendar hoje) {
            if(vet[x].CalculaIdade(vet, hoje) > this.CalculaIdade(vet,
hoje)){
                System.out.println(vet[x].getNome()+" é mais velho do
que "+this.getNome());
            } else if (vet[x].CalculaIdade(vet, hoje) <
this.CalculaIdade(vet, hoje)) {
                System.out.println(this.getNome()+" é mais velho do
que "+vet[x].getNome());
            }
        }

        public String getNome() {
            return nome;
        }

```

```
}

public void setNome(String nome) {
    this.nome = nome;
}

public String getCpf() {
    return cpf;
}

public void setCpf(String cpf) {
    this.cpf = cpf;
}

public char getSexo() {
    return sexo;
}

public void setSexo(char sexo) {
    this.sexo = sexo;
}

public float getSalarioBruto() {
    return salarioBruto;
}

public void setSalarioBruto(float salarioBruto) {
    this.salarioBruto = salarioBruto;
}

public GregorianCalendar getDataAdmissao() {
    return dataAdmissao;
}

public void setDataAdmissao(GregorianCalendar dataAdmissao) {
    this.dataAdmissao = dataAdmissao;
}

public GregorianCalendar getDataNascimento() {
    return dataNascimento;
}

public void setDataNascimento(GregorianCalendar dataNascimento) {
    this.dataNascimento = dataNascimento;
}

public int getDia() {
    return dia;
}

public void setDia(int dia) {
    this.dia = dia;
}

public int getMes() {
```

```

        return mes;
    }

    public void setMes(int mes) {
        this.mes = mes;
    }

    public int getAno() {
        return ano;
    }

    public void setAno(int ano) {
        this.ano = ano;
    }
}

```

```
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].getNome()+" válidos!");
            } else {
                System.out.println("Dados do funcionário
"+vet[i].getNome()+" 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].getNome();
            }
            if (menor >= vet[i].SalarioLiquido(vet[i].salarioBruto)) {
                menor = vet[i].SalarioLiquido(vet[i].salarioBruto);
                menorSalario = vet[i].getNome();
            }
        }
        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].getNome();
            }
            if(vet[i].CalculaIdade(vet, hoje) > velho) {
                velho = vet[i].CalculaIdade(vet, hoje);
                nomeVelho = vet[i].getNome();
            }
        }
        System.out.println("O funcionário mais velho é o "+nomeVelho+" com
"+velho+" anos. E o funcionário mais novo é o "+nomeNovo+" com "+novo+" anos.");
    }

    public static void TotalBonificacoes(Analista vet[]) {
        double total = 0;
        for(int i = 0; i < vet.length; i++) {
            total += vet[i].getBonificacao(vet);
        }
        System.out.println("Soma das bonificações de todos os funcionarios:
"+total);
    }

    public static void AposentaFuncionarios(Funcionario vet[], Calendar hoje)
    {
        for(int i = 0; i < vet.length; i++) {
            vet[i].dataAposentadoria(vet, hoje);
        }
    }
}

```

```

package empresa;

import java.util.GregorianCalendar;

public class Programador extends Funcionario{
    public String linguagem;
    public Programador() {
        super();
        String[] linguagem = new String[5];
    }

    public Programador(String cpf, String nome, char sexo, float
salarioBruto, GregorianCalendar dataAdmissao, GregorianCalendar dataNascimento,
int dia, int mes, int ano, String linguagem) {
        this.cpf = cpf;
        this.nome = nome;
        this.sexo = sexo;
        this.salarioBruto = salarioBruto;
        this.dataAdmissao = dataAdmissao;
        this.dataNascimento = dataNascimento;
        this.dia = dia;
        this.mes = mes;
        this.ano = ano;
        this.linguagem = linguagem;
    }

    public String getLinguagem() {
        return linguagem;
    }

    public void setLinguagem(String linguagem) {
        this.linguagem = linguagem;
    }

    public boolean programaMesmasLinguagens(Programador vet[], int x) {
        int cont = 0;
        for(int i = 0; i < vet.length; i++) {
            if(vet[x].linguagem.equals(this.linguagem)) {
                cont++;
            }
            if(cont == 5) {
                System.out.println("O funcionário "+this.getNome()+"e
o funcionario "+vet[x].getNome()+" programam as mesmas linguagens!");
                return true;
            }
        }
        return false;
    }

    public String toString() {
        return "Programador: nome=" + nome + ", CPF=" + cpf + ", sexo=" +
sexo + ", salarioBruto=" + salarioBruto + ", dataNascimento: "+ dia + "/" + mes
+ "/" + ano;
    }
}

```

```

package empresa;
import java.util.GregorianCalendar;

public class Analista extends Funcionario{
    String diagrama;
    public Analista() {
        super();
        String[] diagrama = new String[5];
    }

    public Analista(String cpf, String nome, char sexo, float salarioBruto,
GregorianCalendar dataAdmissao, GregorianCalendar dataNascimento, int dia, int
mes, int ano, String diagrama) {
        this.cpf = cpf;
        this.nome = nome;
        this.sexo = sexo;
        this.salarioBruto = salarioBruto;
        this.dataAdmissao = dataAdmissao;
        this.dataNascimento = dataNascimento;
        this.dia = dia;
        this.mes = mes;
        this.ano = ano;
        this.diagrama = diagrama;
    }

    public boolean dominaMesmosDiagramas(Analista vet[], int x) {
        int cont = 0;
        for(int i = 0; i < vet.length; i++) {
            if(vet[x].diagrama.equals(this.diagrama)) {
                cont++;
            }
            if(cont == 5) {
                System.out.println("O analista "+this.getNome()+"e o
analista "+vet[x].getNome()+" dominam os mesmos diagramas!");
                return true;
            }
        }
        return false;
    }

    public String toString() {
        return "Analista: nome=" + nome + ", CPF=" + cpf + ", sexo=" + sexo
+ ", salarioBruto=" + salarioBruto + ", dataNascimento: "+ dia + "/" + mes + "/" +
ano;
    }

    public double getBonificacao(Analista ana[]) {
        double bonificacao = (this.salarioBruto * 12);
        return (bonificacao * 0.03);
    }
}

```

```

package empresa;

import java.util.GregorianCalendar;

public class Gerente extends Funcionario{
    String metodologiaDesenvolvimento;
    public Gerente() {
        super();
        String[] metodologiaDesenvolvimento = new String[5];
    }

    public Gerente(String cpf, String nome, char sexo, float salarioBruto,
GregorianCalendar dataAdmissao, GregorianCalendar dataNascimento, int dia, int
mes, int ano, String metodologiaDesenvolvimento) {
        this.cpf = cpf;
        this.nome = nome;
        this.sexo = sexo;
        this.salarioBruto = salarioBruto;
        this.dataAdmissao = dataAdmissao;
        this.dataNascimento = dataNascimento;
        this.dia = dia;
        this.mes = mes;
        this.ano = ano;
        this.metodologiaDesenvolvimento = metodologiaDesenvolvimento;
    }

    public boolean dominaMesmasMetodologias(Gerente vet[], int x) {
        int cont = 0;
        for(int i = 0; i < vet.length; i++) {

            if(vet[x].metodologiaDesenvolvimento.equals(this.metodologiaDesenvolvimen
to)) {
                cont++;
            }
            if(cont == 5) {
                System.out.println("O Gerente "+this.getNome()+"e o
Gerente "+vet[x].getNome()+" dominam os mesmos metodologiaDesenvolvimentos!");
                return true;
            }
        }
        return false;
    }

    public String toString() {
        return "Gerente: nome=" + nome + ", CPF=" + cpf + ", sexo=" + sexo
+ ", salarioBruto=" + salarioBruto + ", dataNascimento: " + dia + "/" + mes + "/" +
ano;
    }

    public double getBonificacao(Gerente vet[]) {
        double bonificacao = (this.salarioBruto * 12);
        return (bonificacao * 0.05);
    }
}

```