



Classe Abstrata

Prof^a. Rachel Reis
rachel@inf.ufpr.br



Problema

- Uma empresa de tecnologia contrata empregados por meio de uma das seguintes modalidades de pagamento:



mensalista



horista



Problema

- O empregado mensalista recebe um salário mensal fixo + as horas-extra que ele fizer no mês. Lembrando que o valor da hora-extra corresponde a $\frac{1}{160}$ do salário mensal.



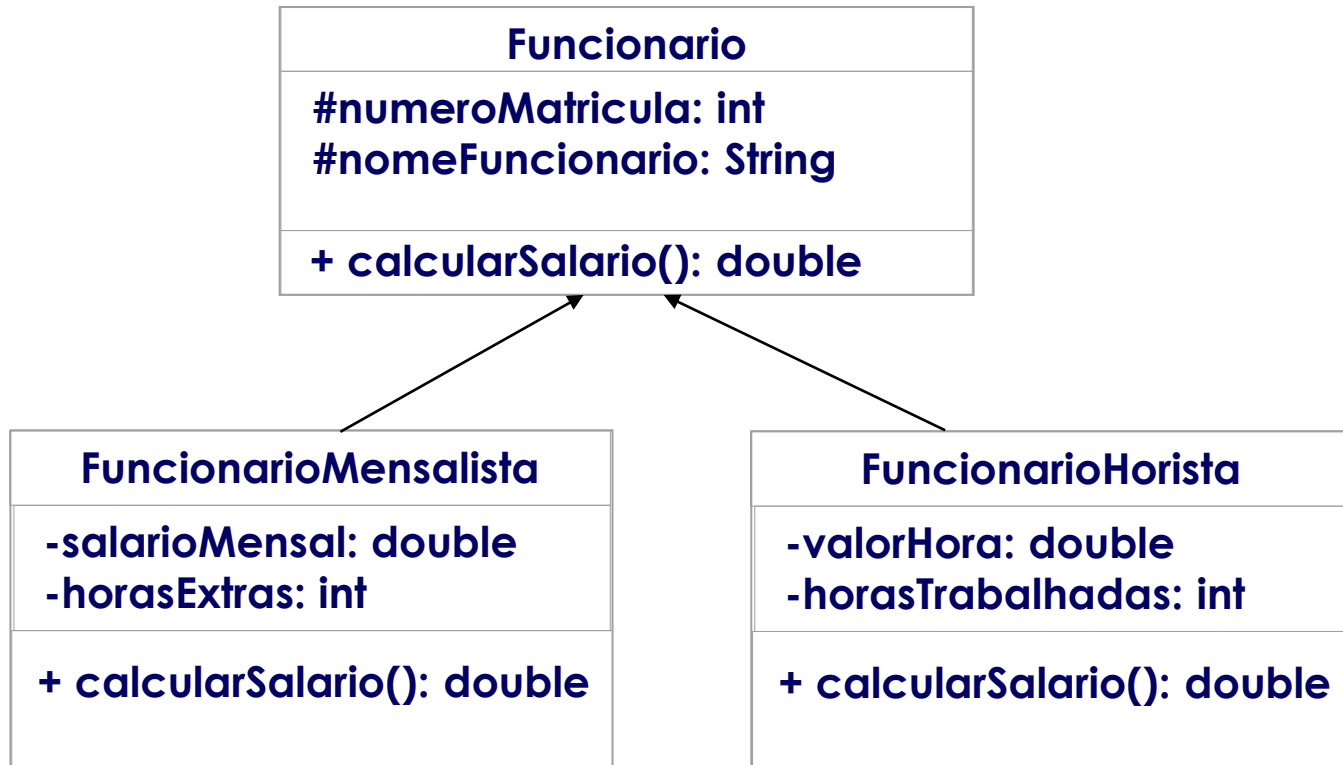


Problema

- O empregado horista simplesmente recebe pelas horas trabalhadas no mês, não existindo o recurso de hora extra.



Solução



```
public class Funcionario
```

```
{
```

```
    // Atributos
```

```
    protected int numeroMatricula;
```

```
    protected String nomeFuncionario;
```

```
    // Métodos get e set
```

```
    ...
```

```
    // Outros métodos
```

```
    public double calcularSalario() {
```

```
        ...
```

```
    }
```

```
}
```

Funcionario
#numeroMatricula: int #nomeFuncionario: String
+ calcularSalario(): double

```
public class FuncionarioMensalista extends Funcionario
{
    // Atributos
    private double salarioMensal;
    private int horasExtras;

    // Métodos get e set
    ...

    // Outros métodos
    public double calcularSalario() {
        double extraF = horasExtras*(salarioMensal/160);
        double salarioT = salarioMensal + extraF;
        return salarioT;
    }
}
```

FuncionarioMensalista
-salarioMensal: double
-horasExtras: int
+ calcularSalario(): double

```
public class FuncionarioHorista extends Funcionario
{
    // Atributos
    private double valorHora;
    private int horasTrabalhadas;

    // Métodos get e set
    ...

    // Outros métodos
    public double calcularSalario() {
        double salarioT = valorHora * horasTrabalhadas;
        return salarioT;
    }
}
```

FuncionarioHorista
-valorHora: double -horasTrabalhadas: int
+ calcularSalario(): double


```
public class Principal{  
    public static void main(String args[]){  
  
        FuncionarioMensalista obj1 = new  
                                FuncionarioMensalista();  
  
        obj1.setNomeFuncionario("Flash");  
        obj1.setNumeroMatricula(123);  
        obj1.setSalarioMensal(1000.0);  
        obj1.setHorasExtras(10);  
  
        System.out.println("Salario: R$" +  
                            obj1.calcularSalario());  
    }  
}
```



- 123
- Flash
- 1000.0
- 10

→ obj1



```
public class Principal{  
    public static void main(String args[]){  
  
        FuncionarioHorista obj2 = new  
                                FuncionarioHorista();  
  
        obj2.setNomeFuncionario("Hulk");  
        obj2.setNumeroMatricula(456);  
        obj2.setValorHora(90.0);  
        obj2.setHorasTrabalhadas(100);  
  
        System.out.println("Salario: R$" +  
                            obj2.calcularSalario());  
    }  
}
```



- 456
- Hulk
- 90.0
- 100

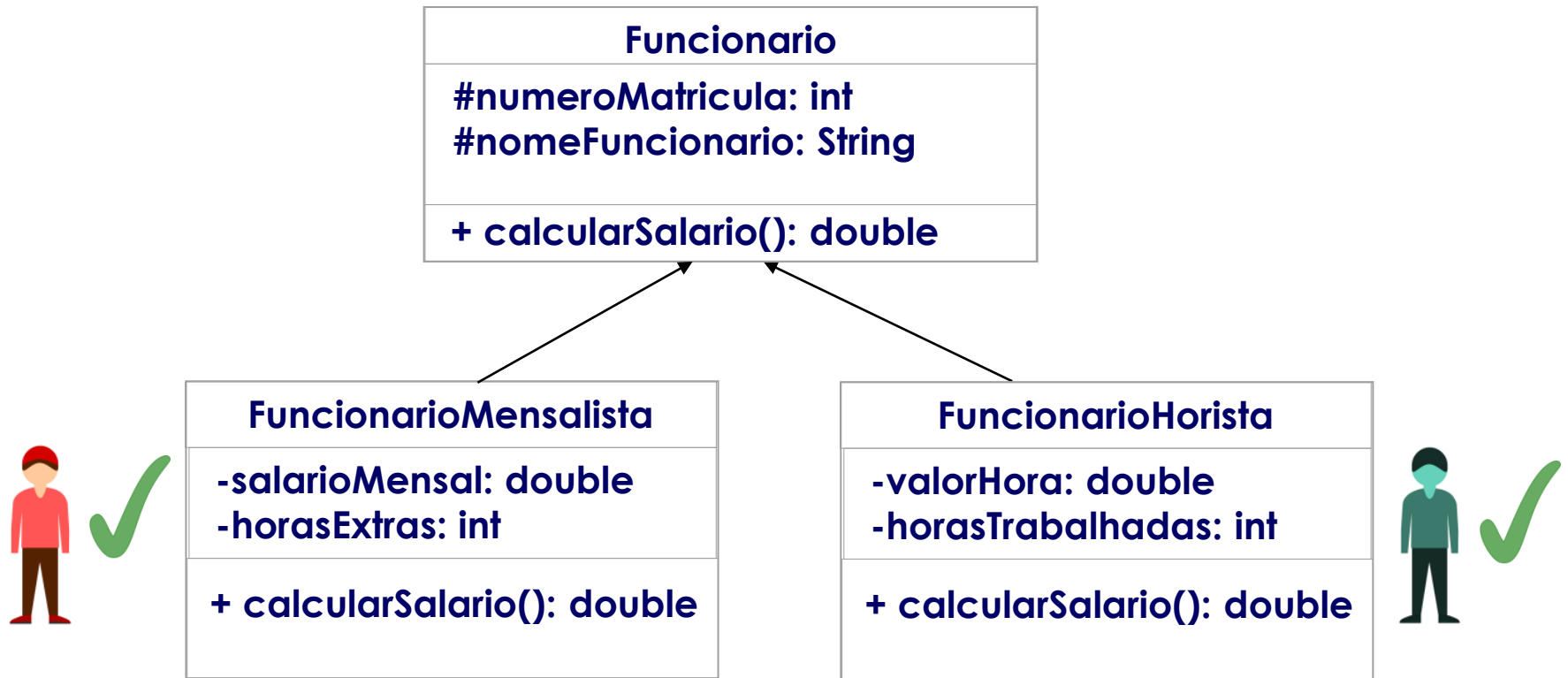
→ obj2



Funcionario
#numeroMatricula: int #nomeFuncionario: String
+ calcularSalario(): double

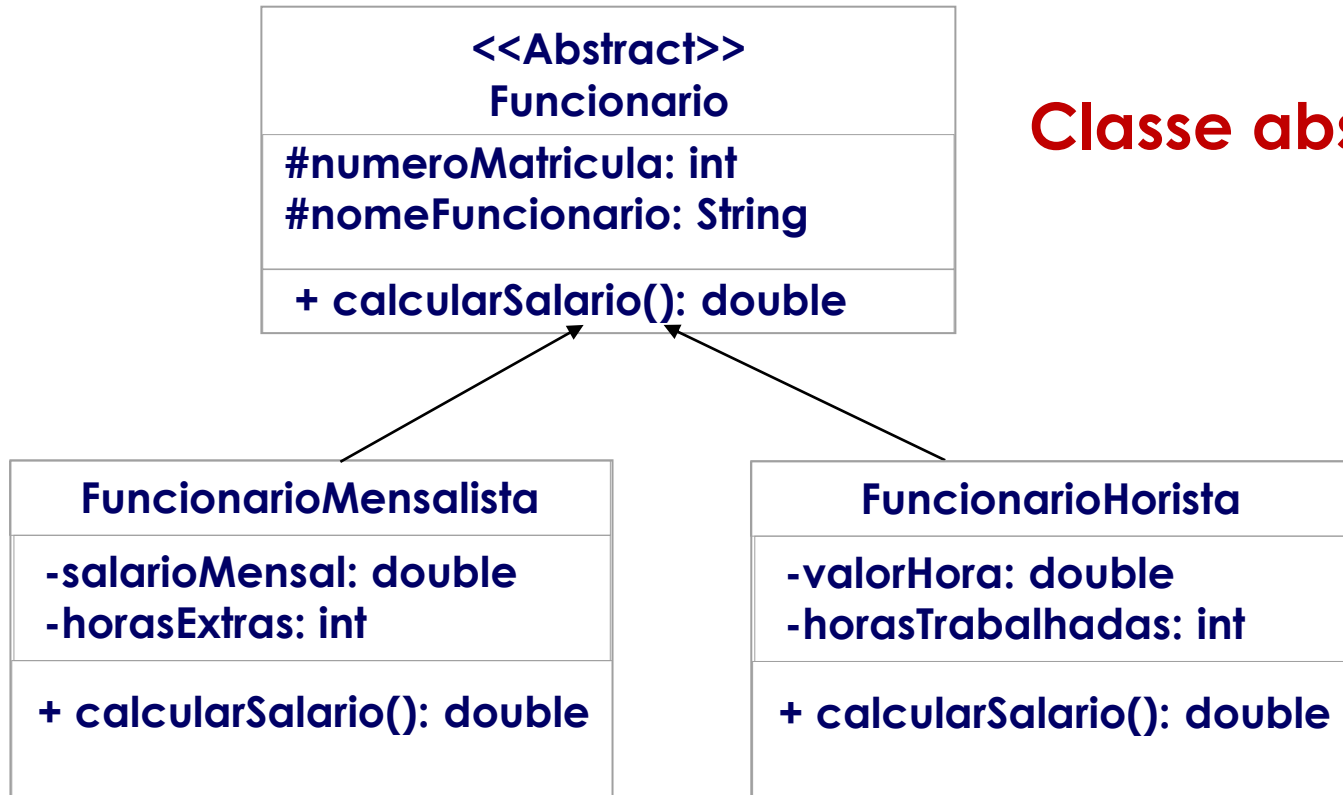


- Nenhum empregado jamais será instanciado na superclasse.



- Objetos serão criados somente a partir das subclasses.

Solução



Classe abstrata

Classes concretas



Classe Abstrata – o que é?

- Uma superclasse **extremamente genérica**, que não pode ser instanciada.

```
Funcionario objeto1 = new Funcionario();
```





Classe Abstrata – o que é?

- Formada por métodos:

- Com implementação:

```
public double calcularSalario() {  
    // implementação do método  
}
```

- Sem implementação (chamados de métodos abstratos):

```
public abstract double calcularSalario();
```



Classe Abstrata – o que é?

- É um meio termo entre *classe* e *interface*
 - Classe: possui implementação para todos os métodos declarados.
 - Interface: não possui nenhum método implementado.



Classe Abstrata – como usar

- Para declarar uma classe como abstrata deve-se usar a palavra-chave *abstract*:

```
public abstract class Funcionario{}
```

- Uma classe abstrata contém zero ou mais métodos abstratos.

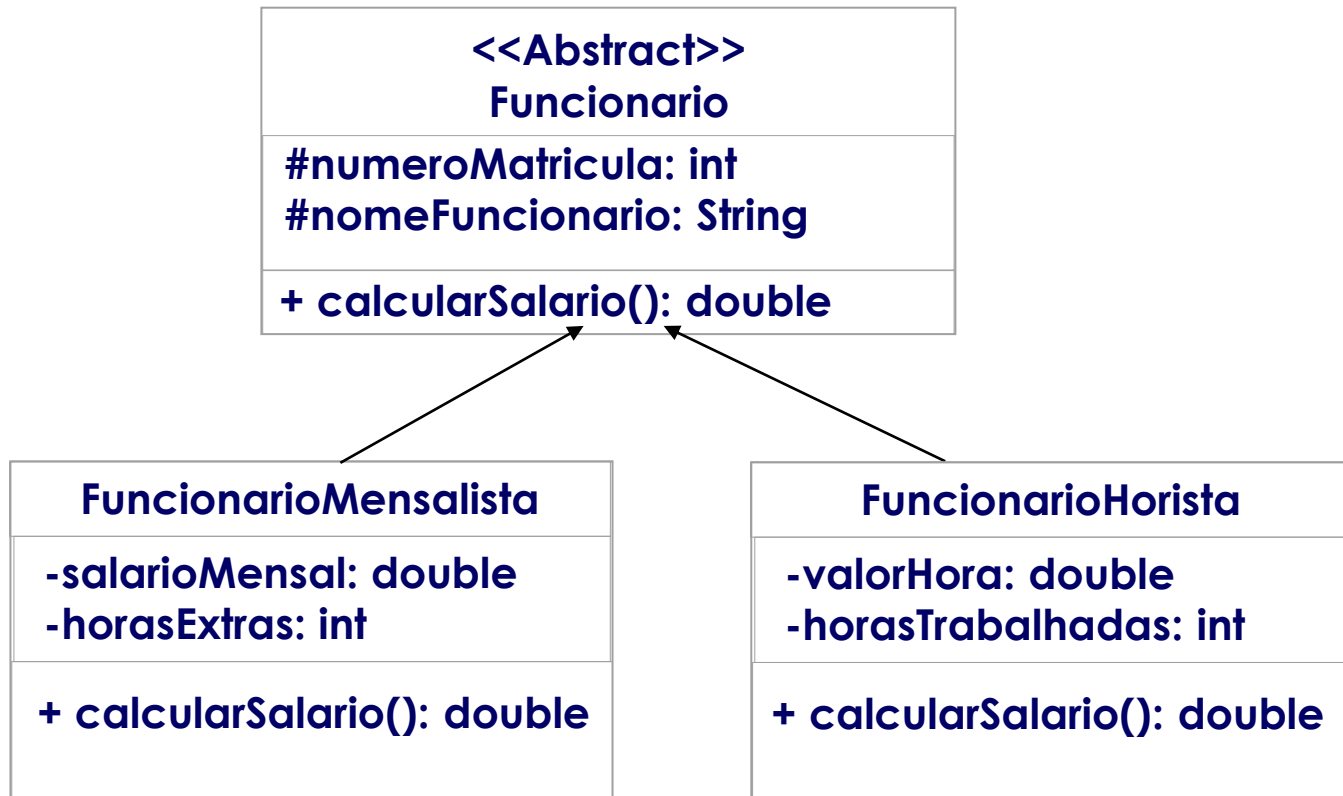


Classe Abstrata – como usar

- Um método abstrato é definido com a palavra-chave *abstract*.

```
public abstract double calcularSalario();
```

- O método abstrato define apenas a assinatura do método e, portanto, não contém código (veja que não há chaves e corpo do método)





- Vamos revisar nossa implementação...

```
public abstract class Funcionario
{
    // Atributos
    protected int numeroMatricula;
    protected String nomeFuncionario;

    // Métodos get e set
    ...

    // Outros métodos
    public abstract double calcularSalario();
}
```



```
public class FuncionarioMensalista extends Funcionario
{
    // Atributos
    private double salarioMensal;
    private int horasExtras;

    // Métodos get e set
    ...

    // Outros métodos
    public double calcularSalario() {
        double extraF = horasExtras*(salarioMensal/160);
        double salarioT = salarioMensal + extraF;
        return salarioT;
    }
}
```

```
public class FuncionarioHorista extends Funcionario
{
    // Atributos
    private double valorHora;
    private int horasTrabalhadas;

    // Métodos get e set
    ...

    // Outros métodos
    public double calcularSalario() {
        double salarioT = valorHora + horasTrabalhadas;
        return salarioT;
    }
}
```



Classe Abstrata – como usar

- Atributos e Construtores podem ser *abstract*?



No!



Se não é abstrata, implemente!

- Uma classe concreta (não abstrata), que estende uma classe abstrata, deve fornecer a implementação dos métodos abstratos.

Classe abstrata

<<abstract>> Funcionario

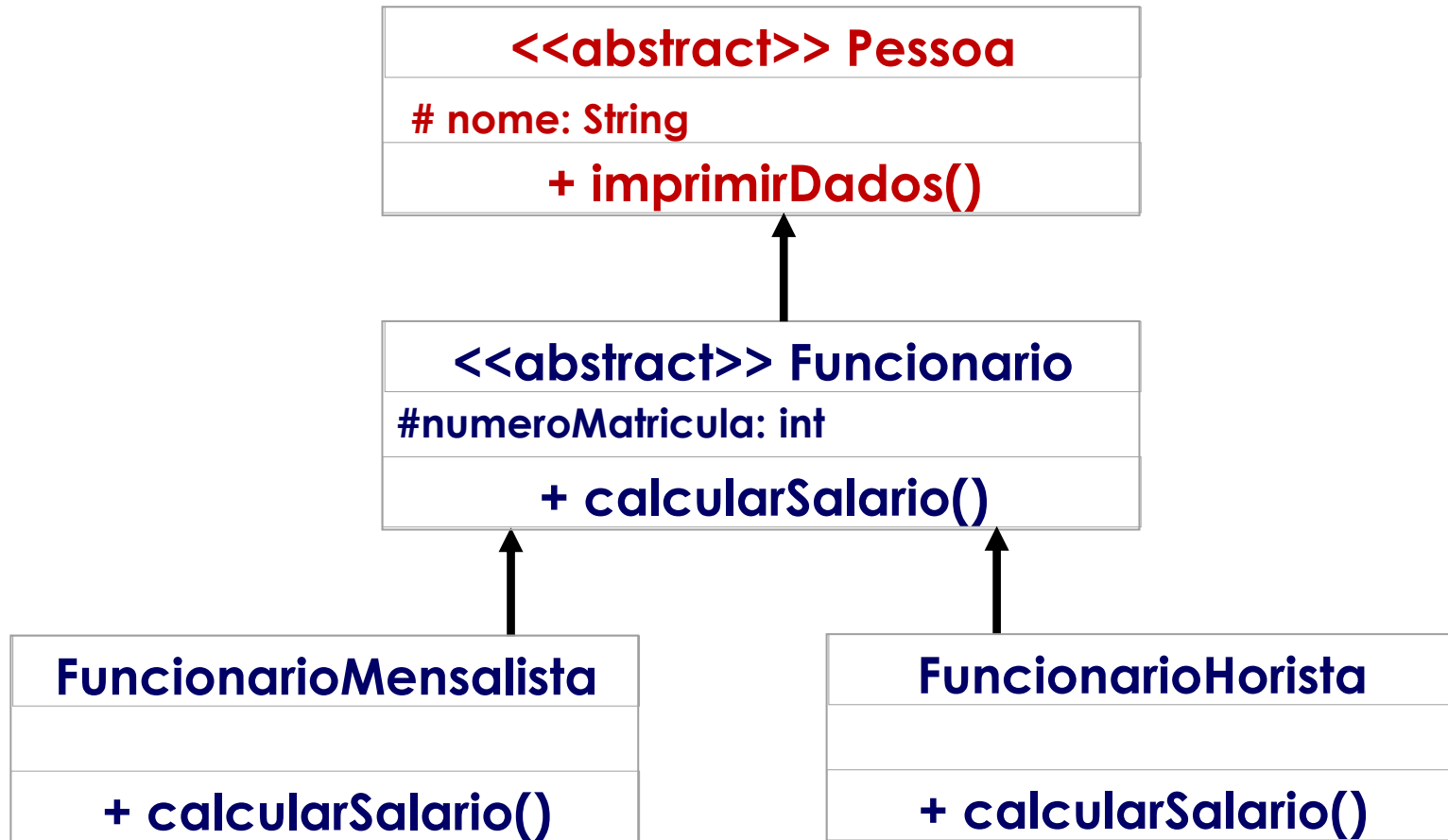
+ calcularSalario()

Classe concreta

FuncionarioMensalista

+ calcularSalario()





```
public abstract class Pessoa
```

```
{
```



```
    // Atributos
```

```
    protected String nome;
```

```
    // Métodos get e set
```

```
    ...
```

```
    // Outros métodos
```

```
    public abstract void imprimirDados();
```

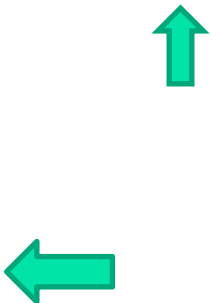


```
}
```

```
public abstract class Funcionario extends Pessoa
{
    // Atributos
    protected int numeroMatricula;
protected String nomeFuncionario;

    // Métodos get e set
    ...

    // Outros métodos
    public abstract double calcularSalario();
}
```



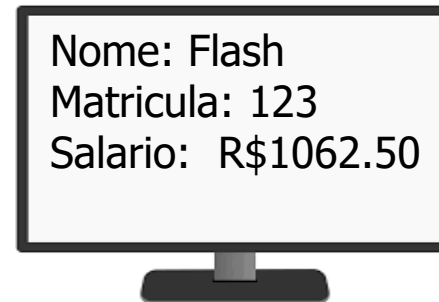
```
public class FuncionarioMensalista extends Funcionario{  
    // Atributos  
    private double salarioMensal;  
    private int horasExtras;  
  
    // Métodos get e set  
    ...  
  
    // Outros métodos  
    public double calcularSalario(){...}  
  
    ➡ public void imprimirDados() {  
        System.out.println("Nome: " + this.nome) ;  
        System.out.println("Matricula: " + this.numeroMatricula) ;  
        System.out.println("Salario: " + this.calcularSalario()) ;  
    }  
}
```

```
public class Principal{  
    public static void main(String args[]){  
  
        FuncionarioMensalista obj1 = new  
                                FuncionarioMensalista();  
  
        obj1.setNome("Flash");  
        obj1.setNumeroMatricula(123);  
        obj1.setSalarioMensal(1000.0);  
        obj1.setHorasExtras(10);  
  
        obj1.imprimirDados();  
    }  
}
```



- 123
- Flash
- 1000.0
- 10

→ obj1



```
public class FuncionarioHorista extends Funcionario{
    // Atributos
    private double valorHora;
    private int horasTrabalhadas;

    // Métodos get e set
    ...

    // Outros métodos
    public double calcularSalario(){...}

    ➡ public void imprimirDados() {
        System.out.println("Nome: " + this.nome);
        System.out.println("Matricula: " + this.numeroMatricula);
        System.out.println("Salario: " + this.calcularSalario());
    }
}
```

```
public class Principal{  
    public static void main(String args[]){  
  
        FuncionarioHorista obj2 = new  
                                FuncionarioHorista();  
  
        obj2.setNome("Hulk");  
        obj2.setNumeroMatricula(456);  
        obj2.setValorHora(90.0);  
        obj2.setHorasTrabalhadas(100);  
  
        obj2.imprimirDados();  
    }  
}
```



- 456
- Hulk
- 90.0
- 100

→ obj2





É possível herdar método abstrato?

- Herdar método abstrato ou não
 - Se uma classe abstrata estende outra que é abstrata, esta pode fornecer ou não implementação para os métodos herdados.
 - A primeira classe concreta (não abstrata) que herdar esse hierarquia tem que obrigatoriamente fornecer a implementação dos métodos abstratos herdados.



Classe Abstrata - Resumo

- Atributos e construtores não podem ser abstratos.
- Uma classe que contém um ou mais métodos abstratos deve ser declarada como abstrata.
- Toda subclasse concreta de uma superclasse abstrata deve implementar os métodos abstratos da superclasse.
- Se uma classe estende outra que tem método abstrato e não o implementa, então ela também deve ser declarada como abstrata.



Referências

- Deitel, P. J.; Deitel, H. M. (2017). Java como programar. 10a edição. São Paulo: Pearson Prentice Hall.
- Barnes, D. J. (2009). Programação orientada a objetos com Java: uma introdução prática usando o BlueJ (4. ed.). São Paulo, SP: Prentice Hall.
- Boratti, I. C. (2007). Programação orientada a objetos em Java. Florianópolis, SC: Visual Books.