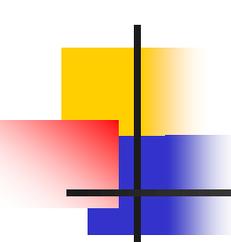


Modificador static e Pacotes

Alberto Costa Neto
DComp - UFS



Roteiro

- Modificador static
- Pacotes

Modificador static

- Instanciando a classe...

Funcionário
matricula nome cpf salario <u>salario-min</u>
getMatricula() getSalario() <u>obterProximaMat()</u> ...

mat = 389
nom= João
cpf = 93823...
salario = 2.000,00
sal_min = 520,00



mat = 420
nom= Paulo
cpf = 4873...
salario = 1.300,00
sal_min = 520,00



mat = 421
nom= Pedro
cpf = 32445...
salario = 520,00
sal_min = 520,00

Modificador Static

- As vezes, queremos que uma variável seja compartilhada por todas as instâncias de uma classe

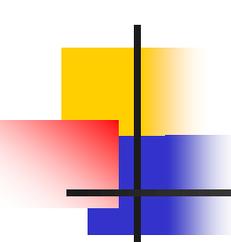
mat = 389
nom= João
cpf = 93823...
salario = 2.000,00
sal_min = 520,00



mat = 420
nom= Paulo
cpf = 4873...
salario = 1.300,00
sal_min = 520,00



mat = 421
nom= Pedro
cpf = 32445...
salario = 520,00
sal_min = 520,00



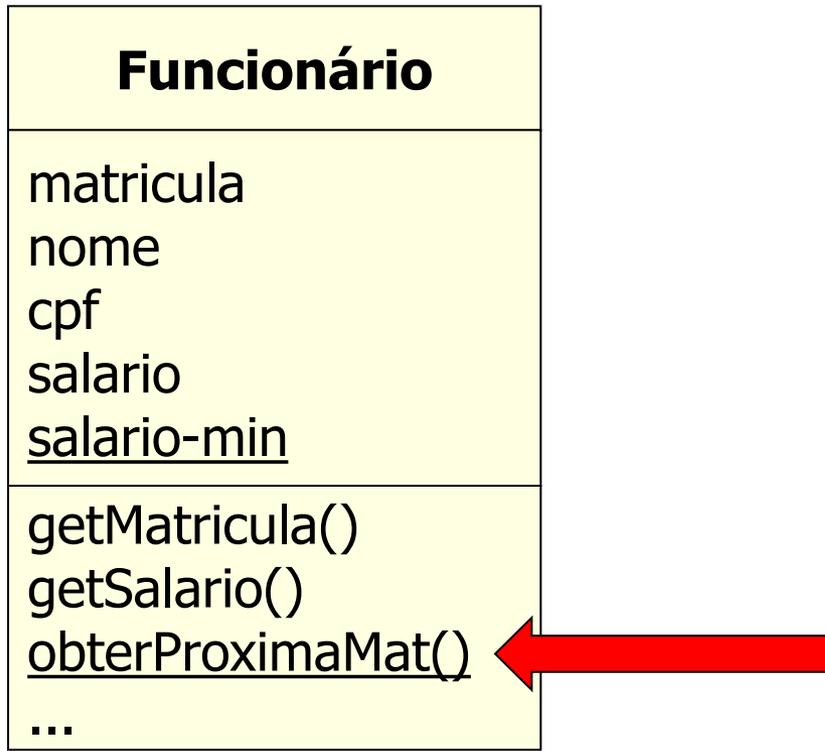
Modificador Static

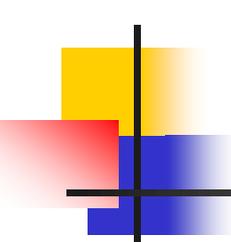
- Como faço para definir variáveis que tenham este comportamento?

```
public class Funcionario {  
  
    private int matricula;  
    private String nome;  
    private String cpf;  
    private float salario;  
    private static float salario_min;  
    ...  
}
```

Modificador Static

- As vezes, precisamos utilizar métodos de uma classe sem que seja necessário instanciá-la

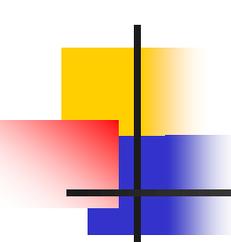




Modificador Static

- Como faço para definir métodos que tenham este comportamento?

```
public class Funcionario {  
    ...  
    public int getMatricula() {  
        return this.matricula;  
    }  
  
    public static int obterProximaMatricula(){  
        ...  
    }  
    ...  
}
```



Modificador Static

- Este código compila?

```
public class Funcionario {  
    ...  
    public static int obterProximaMatricula(){  
        return this.getMatricula() + 1;  
    }  
    ...  
}
```

Modificador Static

- Uma solução seria...

Funcionário
matricula
nome
cpf
salario
<u>salario-min</u>
<u>ultimaMatricula</u>
getMatricula()
getSalario()
<u>obterProximaMat()</u>
...

```
public class Funcionario {  
  
    private int matricula;  
    private String nome;  
    private String cpf;  
    private float salario;  
    private static float salario_min;  
    private static int ultimaMatricula;  
  
    ...  
}
```

Modificador Static

■ Instanciando a classe...

Funcionário
matricula
nome
cpf
salario
<u>salario-min</u>
<u>ultimaMatricula</u>
<u>getMatricula()</u>
<u>getSalario()</u>
<u>obterProximaMat()</u>
...

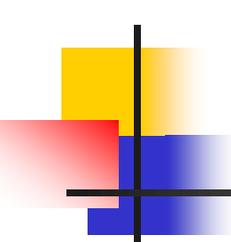
mat = 389
nom= João
cpf = 93823...
salario = 2.000,00
sal_min = 520,00
ultimaMat = 421



mat = 420
nom= Paulo
cpf = 4873...
salario = 1.300,00
sal_min = 520,00
ultimaMat = 421



mat = 421
nom= Pedro
cpf = 32445...
salario = 520,00
sal_min = 520,00
ultimaMat = 421



Modificador Static

- O código resultante seria...

```
public class Funcionario {  
    ...  
    public static int obterProximaMatricula(){  
        ultimaMatricula += 1;  
        return ultimaMatricula;  
    }  
    ...  
}
```

Modificador Static

- Escopo de Variáveis e Métodos

Escopo de Instância

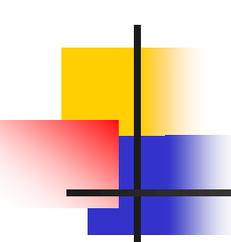


**Cada instância tem
o seu valor próprio**

Escopo de Classe



**Todas as instâncias
têm o mesmo valor**



Modificador Static

- Métodos e Atributos estáticos pertencem à Classe
- Não precisam de uma instância da classe para serem acessados
- Métodos estáticos não podem acessar atributos ou invocar métodos da instância da classe

Modificador Static

■ Outro exemplo

Atributo estático

```
public class CalculaArea {
```

```
    static float PI = 3.14f;
```

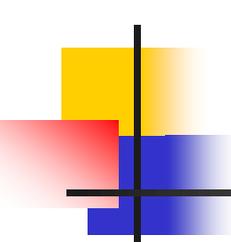
```
    public static float area(float raio) {  
        return PI * raio * raio;  
    }
```

```
    public static void main(String[] args) {  
        System.out.println(CalculaArea.PI );  
        System.out.println(CalculaArea.area(5.3f));  
    }
```

```
}
```

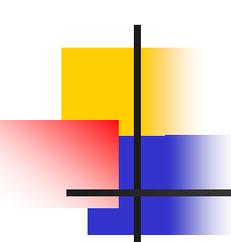
Acesso à variável e à
operação

O método estático
só pode acessar
métodos e
atributos
estáticos.



Dever de Sala

- Qual seria a solução para:
 - Não é permitido criar um funcionário sem que seja especificado um cpf válido
 - O seu código deve garantir que o cpf do funcionário seja válido
 - Não será criada nenhuma classe utilitária para isso
-

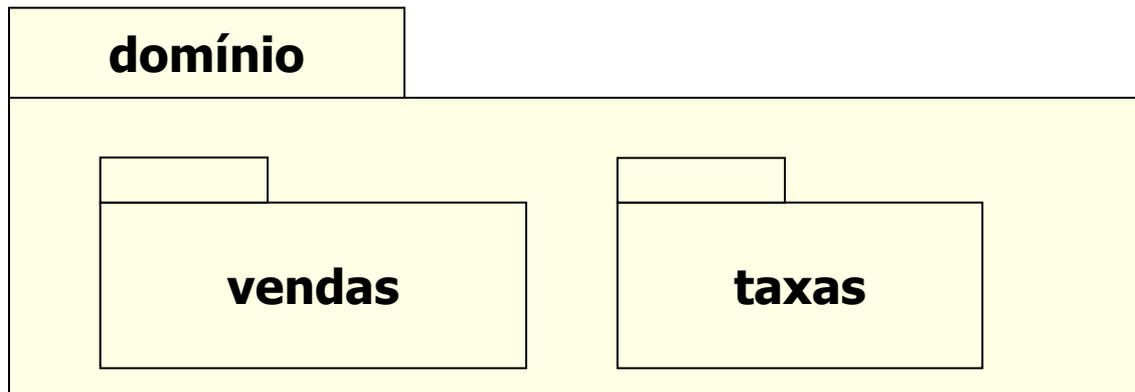


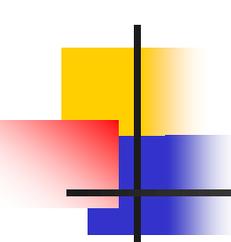
Pacote

- O que é um pacote?
 - Mecanismo genérico de agrupamento

Pacote

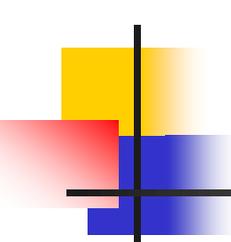
- Para que serve, na POO?
 - Agrupar classes de forma lógica
 - Organizar o código





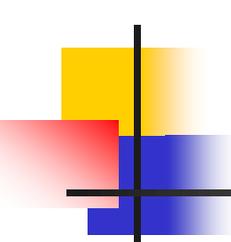
Pacote

- Como criar um pacote em Java?
 - Cláusula package
 - Exemplos
 - package negocio;
 - package java.util;



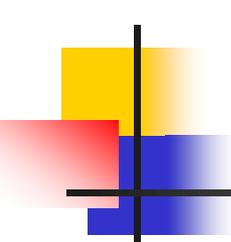
Pacote

- É comum organizar pacotes da seguinte forma:
 - Identificação da instituição + sistema + função
- Exemplos:
 - `br.ufs.dcomp.referencias.dados`
 - `br.ufs.dcomp.referencias.negocio`
 - `br.ufs.dcomp.referencias.telas`
 - `br.ufs.dcomp.referencias.util`



Pacote

- Detalhes:
 - Classes em pacotes diferentes podem ter o mesmo nome
 - Uma classe pública pode ser usada por classes de outros pacotes
 - Uma classe não pública só pode ser usada por outras classes do mesmo pacote



Pacote

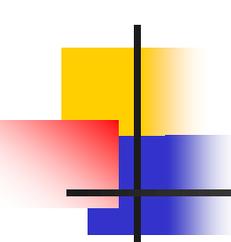
- Como faço para usar classes públicas de outros pacotes?
 - Usando a cláusula ***import***

```
import java.util.Scanner;

public class Exemplo {
    ...
    Scanner s = new Scanner(...);
    ...
}
```

- Usando o **nome completo** da classe

```
java.util.Scanner s = new java.util.Scanner(...);
```



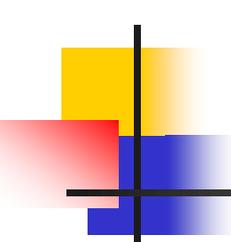
Pacote

- Detalhe:
 - Importando apenas a classe especificada

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

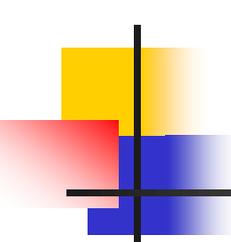
- Importando todas as classes do pacote

```
import java.util.*;
```



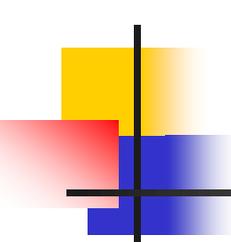
Dever de Sala

- Definir o pacote `br.ufs.poo.util` e colocar a classe `CalculaArea` dentro do pacote
 - Definir o pacote `br.ufs.poo.tela` e criar dentro desse pacote a classe `TestePacote` com um método `main`.
 - Acessar dessa classe os métodos da classe `CalculaArea`
-



Dever de Sala

```
package br.ufs.poo.tela;  
  
import br.ufs.poo.util.CalculaArea;  
  
public class TestePacote {  
  
    public static void main(String[] args) {  
        System.out.println( CalculaArea.area(5) );  
    }  
  
}
```



Referências

- Slides “Aula22-00- Static, Final, Pacotes e Listas” Prof. Marcos Dósea. UFS. 2010.
- Slides “Modificador static e Pacotes”, Prof^a. Débora. UFS. 2010
- Slides “07Essenciais” Prof Giovanni . Java.UFS. 2009.
- Java How to Program
 - Capítulo – 8