INSTITUTO MAUÁ DE TECNOLOGIA

Linguagens I

Atributo de classes

Profº. Tiago Sanches da Silva

Atributo de classes

Atributo de classes

Caso eu queira armazenar de alguma forma quantos objetos contas foram criados o que eu devo fazer?

Atributo de classes

Discuta com o professor as implicações dessa implementação e avalie se é uma solução viável.

Quando criarmos duas contas, qual será o valor do totalDeContas de cada uma delas?

Vai ser 1. O atributo é de cada objeto.

```
class Conta {
    private int totalDeContas;
    //...

    Conta() {
        this.totalDeContas = this.totalDeContas + 1;
    }
}
```

Atributo de classes

Seria interessante então, que essa variável fosse única, compartilhada por todos os objetos dessa classe. Dessa maneira, quando mudasse através de um objeto, o outro enxergaria o mesmo valor.

Atributo de classes

Para fazer isso em java, declaramos a variável como **static**.

```
private static int totalDeContas;
```

Atributo de classes

Quando declaramos um atributo como static, ele passa a não ser mais um atributo de cada objeto, e sim um atributo da classe, a informação fica guardada pela classe, não é mais individual para cada objeto.

Atributo de classes

Para acessarmos um atributo estático, não usamos a palavra chave this, mas sim o nome da classe:

```
class Conta {
    private static int totalDeContas;
    //...

    Conta() {
        Conta.totalDeContas = Conta.totalDeContas + 1;
    }
}
```

Atributo de classes

Já que o atributo é privado, como podemos acessar essa informação a partir de outra classe? Precisamos de um getter para ele!

```
class Conta {
    private static int totalDeContas;
    //...

    Conta() {
        Conta.totalDeContas = Conta.totalDeContas + 1;
    }

    public int getTotalDeContas() {
        return Conta.totalDeContas;
    }
}
```

Atributo de classes

Como fazemos então para saber quantas contas foram criadas?

```
Conta c = new Conta();
int total = c.getTotalDeContas();
```

Precisamos criar uma conta antes de chamar o método! **Isso não é legal**, pois gostaríamos de saber quantas contas existem sem precisar ter acesso a um objeto conta.

Atributo de classes

A ideia aqui é a mesma, transformar esse método que todo objeto conta tem em um método de toda a classe. Usamos a palavra **static** de novo, mudando o método anterior.

```
public static int getTotalDeContas() {
    return Conta.totalDeContas;
}
```

Para acessar esse novo método a partir da classe usuária:

```
int total = Conta.getTotalDeContas();
```

Repare que estamos chamando um método não com uma referência para uma Conta, e sim usando o nome da classe.

Atributo de classes

Use realmente apenas quando necessário.

Métodos e atributos estáticos

Métodos e atributos estáticos só podem acessar outros métodos e atributos estáticos da mesma classe, o que faz todo sentido já que dentro de um método estático não temos acesso à referência `this`, pois um método estático é chamado através da classe, e não de um objeto.

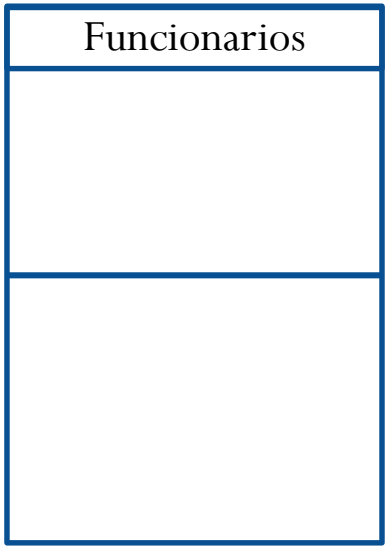
O `static` realmente traz um "cheiro" procedural, porém em muitas vezes é necessário.

Perguntas?

Exercícios

Prática 1 - Exercício 2

- Modelar a classe Funcionarios de uma concessionaria.
- Crie um novo projeto: Concessionaria
- Inicie a implementação
- Discuta a solução com o professor se necessário
- RH deve conseguir acessar essas informações
  - (13°, comissão, férias, salario do mês)
- Comissão = 5% de todas as vendas
- salario do mês = salario base + horas extras + comissão



Prática 2 - Exercício 2

A concessionaria possui 2 cargos principais:

Vendedor:

- Recebe comissão por venda, como o funcionário do exercício passado. Mas agora armazena quem é o gerente responsável.

Gerente de vendas:

- Pode vender, dar aumento a funcionários mas apenas se forem de sua responsabilidade.
- Recebe comissão diferenciada: 10% de suas vendas + 50% do salario base.

Modele as duas classes.