PROGRAMAÇÃO I

INTRODUÇÃO À ORIENTAÇÃO A OBJETOS II

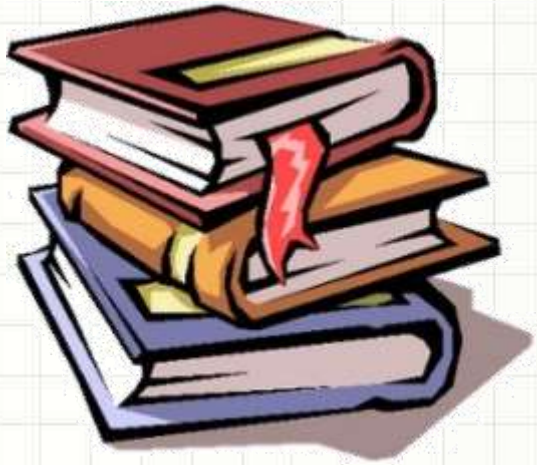
Prof. Dr. Daniel Caetano

2017 - 1

Objetivos

- Formalizar o conceito de método
- Compreender o conceito de sobrecarga de métodos
- Formalizar o uso da referência *this* e do "operador" ponto
- Conhecer um pouco mais sobre a classe Math e String

Material de Estudo



Material	Acesso ao Material
Apresentação	http://www.caetano.eng.br/ (Programação I – Aula 4)
Material Didático	Programação I – Págs 59 a 65

Retomando Classes em Java

O que são Classes?

- Podemos imaginar uma classe assim:

Atributos

Métodos

Privados ou Públicos

Setters e *Getters*

- Atributos: em geral **private**
  - **public**, **protected** e **private**
- **Setters**: métodos que modificam atributos.
  - Atributo é **idade**, o *setter* será **setIdade()**.
- **Getters**: métodos que leem os atributos.
  - Atributo é **idade**, o *getter* será **getIdade()**.

Como criar objetos?

- Operador **new**

Classe

new

Objeto

- Ex.: classe Cliente

```
Cliente oCliente = new Cliente();
```

O objeto **oCliente** é uma instância da classe **Cliente**

Construtor

- Construtor: configura objeto
  - Construtor com parâmetro

```
public Cliente(String cpf) {
...
}
```

- Passando parâmetros de criação

```
Cliente oCliente;
oCliente = new Cliente(“012.345.678-90”);
```

Métodos em Java

Métodos

- Um método:“função” associada à classe
  - Carro
    - Acelerar, Frear, Virar
  - Pedido
    - CalcularDesconto, ValorTotal
- Getters e Setters são métodos
- Construtores são métodos

Declarando Métodos

- Como declarar um método?

```
[escopo] tipoRetorno nomeMétodo(
    tipoParam1 nomeParam1,
    tipoParam2 nomeParam2,
    ...) {

}
```

No construtor, não se indica tipo do retorno!

- Exemplo

```
public int calculaPreco(
    double desconto, int tipoPagamento) {
}
```

Chamando Métodos

- O método em uma classe

```
public class Produto {
    public int calculaPreco(double desconto,
        int tipoPagamento) {
            ...
    }
}
```

- Chamando o método:

```
Produto umProduto = new Produto();
int preco = umProduto.calculaPreco(0.10, 1);
```

Sobrecarga de Métodos

Métodos Comuns

Como a linguagem sabe qual usar?

- É possível ter várias versões de um método
  - Recebendo diferentes tipos de parâmetros
- Exemplo

```
public void setCpf(String cpf) {
    ...
}
public void setCpf(long cpf) {
    ...
}
```

ASSINATURA

Métodos Comuns

- É possível ter várias versões de um método
- Exemplo

objeto.setCpf("012.345.678-90");

```
public void setCpf(String cpf) {
    ...
}
public void setCpf(long cpf) {
    ...
}
```

Métodos Comuns

- É possível ter várias versões de um método
- Exemplo

```
public void setCpf(String cpf) {
    ...
}
public void setCpf(long cpf) {
    ...
}
```

objeto.setCpf(01234567890L);

Eles não podem ser diferentes só pelo tipo de retorno!

Métodos Comuns

- Para não repetir código...
  - É comum um método chamar outro:

```
private String cpf;
public void setCpf(String cpf) {
        this.cpf = cpf;
        }
public void setCpf(long cpf) {
        setCpf(Long.toString(cpf));
        }
```

Construtores

- Construtores; versão completa / parciais

```
public Cliente(String cpf, int credito) {
        ...
        }

public Cliente (String cpf) {
        this(cpf, 0);
        }
```

this é uma referência para o objeto atual!

this, neste caso, chama um outro construtor!

Alguns Métodos Úteis: Classes Math e String

Atributos e Métodos: Math/String

- Math.PI
- Math.pow(base, exp)
- Math.sqrt(n)
- String t1 = “aba”;
- String t2 = “cate”;
- t1.length()
- t1.charAt(pos)
- t1.replace(oque, comoque)
- t1.toLowerCase()
- t1.toUpperCase()

PERGUNTAS?

Parte Prática

Experimentando

- Continuar a implementação do Produto
  - Validação
  - Checagem de preenchimento
- Aprimorar a classe Produto com a implementação um segundo construtor vazio.

Experimentando

- Vamos implementar uma classe Retangulo:
  - Dois construtores (x1, y1, x2, y2) / (x,y,b,h)
    - Não serão simultâneos!
  - Calcule seu perímetro
  - Calcule sua área

Experimentando

- Vamos implementar uma classe Círculo:
  - Um construtor (x,y,r)
  - Calcule seu perímetro
  - Calcule sua área

PROJETO / ATIVIDADE

Projeto / Atividade

- Objetivo: criar cadastro de clientes simples
- Quem: Duplas especificadas
- A primeira entrega será 17/04
  - Classe cliente (com validações)
  - Criação/Preenchimento dos dados do cliente por modo texto (Scanner)
- A segunda entrega será 05/06
  - Listagem de clientes
  - Busca de clientes
  - Edição de clientes

Projeto / Atividade

- Hoje
  - Crie um projeto CadCli
  - Comecem criando a classe Cliente
    - cpf (validação: 11 dígitos, só números)
      - Dica: use Character.isDigit(caractere)
    - Nome (validação: pelo menos 2 letras)
    - Idade (validação: de 0 a 100 anos)
    - Crie um construtor sem parâmetros que inicialize os atributos com “” e 0, dependendo do tipo.
    - Crie um método que imprima a descrição do objeto
  - Use o método *main* para criar um cliente e imprimir sua descrição