

Java Advanced

06/08/2024

Prof. Dr. Marcel Stefan Wagner

Aula 02 – Revisão de POO

FIAP

Tópicos Abordados

- 1 Introdução à Programação Orientada a Objetos
- 2 Classes
- 3 Encapsulamento
- 4 Herança
- 5 Temas para a Próxima Aula
- 6 Referências Bibliográficas

Introdução



Introdução

- Abstração é a habilidade de concentrar nos aspectos essenciais de um contexto qualquer, ignorando características menos importantes.
- Em modelagem orientada a objetos, uma classe é uma abstração de entidades existentes no domínio do sistema de *software*.
- As entidades em Programação Orientada a Objetos (POO) devem representar entidades no mundo real.
- O principal benefício da utilização da programação por meio de classes é a reutilização do código, pois a cada objeto criado você não precisa criar sua estrutura novamente.

Introdução



Classes são estruturas que definem as características e comportamentos dos seres ou coisas do mundo real; quando implementados em Java nas classes, essas características passam a se chamar atributos, e os comportamentos se transformam nos nossos métodos.



Atributos são as características inerentes ao meu ser ou coisa do mundo real quando são definidas no mundo computacional.



Objetos são as instâncias das minhas classes, os seres que existem no meu mundo real quando são inseridos no mundo computacional.



Métodos são os comportamentos ou funções de um objeto do mundo real na forma como ele é tratado no mundo computacional.

Introdução

A PROGRAMAÇÃO ORIENTADA A OBJETO

A **Programação Orientada a Objeto (POO)** tem como pilar teórico o conceito de **objeto**, ou seja, um sistema orientado a objeto é um conjunto de objetos que representam os seres e coisas do mundo real, interagindo computacionalmente com as mesmas características e comportamentos reais.

À estrutura computacional que define o modelo de um objeto do mundo real damos o nome de **classe**, e a partir dela cópias são construídas para serem utilizadas para cada objeto real que tenho no meu sistema.

As **características** dos objetos do mundo real são chamadas de **atributos** no mundo computacional, enquanto a seus **comportamentos** chamamos métodos.

CLASSE CARRO		OBJETO CARRO A	OBJETO CARRO B
Atributos de objeto	Marca	Ford	Mitsubishi
	Modelo	Fiesta	L-200
	Cor	branco	azul royal
	Combustível	gasolina	diesel
Métodos	ligar		
	acelerar		
	frear		

Assim, durante a execução do sistema, a cada carro que cadastramos uma cópia nova da classe carro é criada, e damos a ela características que irão diferenciá-la dos demais carros, ou seja, criamos a cada vez um novo objeto do tipo de carro.

Tipo Abstrato de Dados (TAD)

- Um Tipo Abstrato de Dados (TAD) refere-se ao conceito de definição de um tipo de dado. A definição de TAD leva à criação de um novo tipo de dado.
- Exemplo: Pode-se criar um tipo Racional, onde os valores armazenados têm a forma $\frac{1}{2}$, $\frac{2}{3}$, etc.
 - E sobre esse conjunto podem ser especificadas operações: +, -, *, /.

Tipo Abstrato de Dados (TAD)

- Um determinado objeto deve representar uma entidade do mundo real em um sistema computacional.
 - Deve ser descrito através de suas características desejáveis (atributos) e as operações que são realizadas nele (métodos).

Classes

- Uma classe é uma declaração de tipo que agrega:
 - constante, variáveis e funções.
- Classes possuem basicamente dois grupos de elementos:
 - atributos;
 - comportamentos.

Classes

- Atributos
 - definem as características que cada objeto de uma classe deve ter.
- Comportamentos
 - definem as ações que cada objeto de uma classe pode executar.

Classes em Java

```
1 public class RegistraAluno {  
2     // demais partes do código  
3 }
```

Onde:

- **public:** é um modificador de acesso (será mais bem explicado futuramente) e define que qualquer classe pode acessar a classe **RegistraAluno**;
- **class:** é a palavra-chave usada para criação de uma classe;
- **RegistraAluno:** é o identificador único que nomeia a classe.

Classes em Java

Classes em Java

```
public class Pessoa {

    private String nome;
    private int idade;

    public Pessoa(String nome, int idade) {

        this.nome = nome;
        this.idade = idade;
    }

    // ----- Métodos -----
    public String getNome() {
        return nome;
    }
    public void setNome(String nome) {
        this.nome = nome;
    }

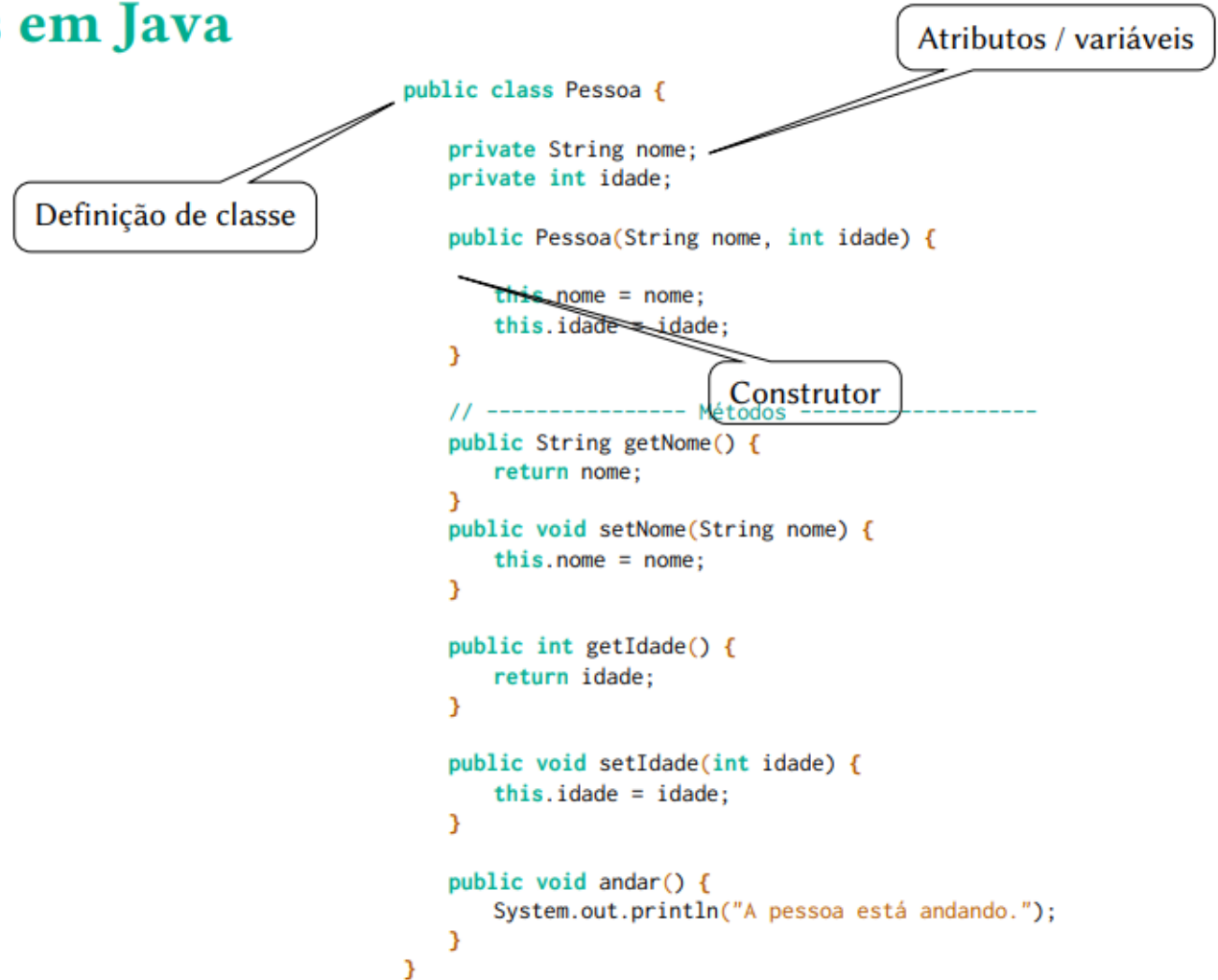
    public int getIdade() {
        return idade;
    }

    public void setIdade(int idade) {
        this.idade = idade;
    }

    public void andar() {
        System.out.println("A pessoa está andando.");
    }
}
```

Classes em Java

Classes em Java



Classes em Java

- Definição de classe *private* / *public*:
 - encapsulamento;
 - define para quais classes o atributo é visível;
 - *public*: visível para todas as classes;
 - *private*: visível apenas para a própria classe.
- Atributos *get* / *set*:
 - utilizados para acessar as variáveis *private*;
 - *set* muda o valor da variável;
 - *get* acessa o valor da variável.

Classes em Java

Classes em Java

```
public class Main {  
  
    public static void main (String[] args) {  
  
        Pessoa pedro = new Pessoa("Pedro", 24);  
  
        System.out.println(pedro.getNome() + " " + pedro.getIdade());  
  
        pedro.andar()  
        pedro.setIdade(25);  
  
        System.out.println(pedro.getNome() + " " + pedro.getIdade());  
    }  
}
```

Classes em Java

Classes em Java

```
public class Main {

    public static void main (String[] args) {

        Pessoa pedro = new Pessoa("Pedro", 24);

        System.out.println(pedro.getNome() + " " + pedro.getIdade());

        pedro.andar()
        pedro.setIdade(25);

        System.out.println(pedro.getNome() + " " + pedro.getIdade());

    }
}
```

Declaração do Objeto

Para criar um Objeto

Classes em Java

Classes em Java

- Saída do Programa:

Pedro 24

A pessoa está andando.

Pedro 25

Classes em Java

Exercícios

- **Exercício 1** - Faça um programa em Java para termos um tipo abstrato de dado chamado Produto. No produto poderemos armazenar o nome, marca, preço e peso.
 - Crie pelo menos 3 objetos diferentes para o produto e em seguida altere os seus valores.

Referências

George Coulouris, Jean Dollimore, Tim Kindberg, and Gordon Blair. **Sistemas Distribuídos: Conceitos e Projeto**. Bookman Editora, 5 edition, 2013.

Harvey M Deitel, Paul J Deitel, David R Choffnes, et al. **Sistemas Operacionais**. Pearson/Prentice Hall, 3 edition, 2005.

Maarten Van Steen and A Tanenbaum. **Sistemas Distribuídos: Princípios e Paradigmas**. Pearson/Prentice Hall, 2 edition, 2007.

Harvey M Deitel and Paul J Deitel. **Java, como programar**. Ed. Pearson/Prentice Hall, 8 edition, 2010.

.....

Obrigado!

Agradecimento pela parceria e elaboração de materiais aos professores:
Prof. Me. Gustavo Torres Custódio
Prof. Thiago Yamamoto

.....

Contato: profmarcel.wagner@fiap.com.br

Cursos:

Tecnologia em Análise e Desenvolvimento de Sistemas (TDS)

Tecnologia em Defesa Cibernética (TDC)

Engenharia de Software (ES)

