

Revisão sobre orientação a objetos

Fabio Takeshi Matsunaga

Universidade Estadual do Paraná

3 de outubro de 2016

Conteúdo

Missão da aula de hoje

Identificar classes em um sistema de software.

- Classes em Java
- Construtores em Java
- Atividades a serem desenvolvidas em sala e para próxima aula

Orientação a objetos

- Paradigma amplamente utilizado e difundido entre as aplicações reais;
- Conceitos de programação úteis
 - Segurança dos dados
 - Reutilização de código
 - Facilidade de abstração
- Linguagens de programação: C++, Java, Python e PHP.

Sistemas de software

- Um software não é um bloco inteiro de um sistema - é constituído de diversas partes.
- Sistema de gerenciamento de pet-shop: animais, produtos, usuários (donos dos animais).
- Um sistema é constituído de diversas entidades e colaboram para criar a funcionalidade do software.
- Essas entidades são representadas por classes. Classes são abstrações para representar objetos.

Exemplo de classe e objeto

- Uma cadeira é uma classe (entidade) para representar um objeto.
- Existem diversos tipos de cadeiras, cada uma possui uma característica própria. Entretanto, todos representam a mesma entidade.
- Cada cadeira particular é um objeto, pois representa uma instância concreta da classe cadeira.

Exemplo de classe e objeto

- Classe é uma denominação genérica.
- Objetos são únicos. Mesmo que tenham cadeiras iguais, representam dois objetos diferentes.
 - Uma cadeira pode estar quebrada, outra em uma posição diferente. Mesmo cadeiras iguais podem ter estados diferentes. Isso é o que caracteriza objetos diferentes.

Estados e comportamentos de classes

- Temos que diferenciar estados de classes e comportamento de classes.
- Cadeira é uma abstração (classe).
- Essa abstração pode ser definida assim: uma cadeira possui encosto, pés e assento. Isso é genérico para todas as cadeiras – que define os atributos.
- Já objetos do tipo cadeira possuem formas diferentes desses atributos – define o estado dos atributos.
- Uma ação pode modificar os atributos da classe: exemplo: mover (ação) uma cadeira altera sua posição (atributo). Ações são representadas por métodos.

Estados e comportamentos de classes

- Uma classe tem seu estado (atributos) e seu comportamento (métodos).
- Vamos ver a classe Carro: possui atributos de potência e velocidade atual, que definem seu estado atual.
- A ação 'acelerar' altera a velocidade do carro de acordo com a potência do motor.
- Já frear diminui a velocidade, até chegar no máximo 0.
- Velocidade zero define o estado do carro como 'parado' ou 'estacionado'.

Classes em Java

- Uma classe deve armazenar suas informações em atributos.

```
1 public class Carro {  
2     int potencia;  
3     float velocidade;  
4 }
```

Acima, percebemos o modo tradicional e convencional como declarar uma variável para definir os atributos.

Acesso a itens da Classe

- Para acessar um atributo, utiliza-se o operador '.' para acessar atributos e métodos.

```
1 Carro c = new Carro();  
2 Carro fusca = new Carro();  
3  
4 fusca.potencia = 2;  
5 float v = fusca.velocidade;
```

Este modo é similar ao acesso à itens de um registro e classe em C++.

Métodos de classes

- Para definir métodos, vamos criar ações dentro da Classe Carro.

```
1 public class Carro {  
2     ...  
3  
4     public void acelerar() {  
5         velocidade += potencia;  
6     }  
7  
8     public void frear() {  
9         velocidade *= 0.5;  
10    }  
11 }
```

No método acima, void não retorna nada, não tem parâmetros mas manipula um estado da classe.

Chamada de métodos em um objeto

- Para chamar o método, é de forma similar à chamada de um atributo.

```
1 Carro c = new Carro();  
2 Carro fusca = new Carro();  
3  
4 fusca.acelerar();  
5 fusca.frear();
```

No método acima, void não retorna nada, não tem parâmetros mas manipula um estado da classe.

Construtores de classes em Java

- Construtores são métodos especiais para criar e inicializar objetos de classes.
- Serve para parametrizar o objeto criado e inicializar objetos.
- O parâmetro é passado durante a instância do objeto e este passa a ter as características passadas por parâmetro.
- Sem construtores, criamos classes sem passar parâmetros, por default ().

Métodos de classes

- Criando um construtor com parâmetro na classe Carro.

```
1 public class Carro {  
2     ...  
3  
4     Carro(int potencia) {  
5         this.potencia = potencia;  
6         velocidade = 0;  
7     }  
8  
9 }
```

Construtores não tem retorno e tem o mesmo nome da classe. A palavra-chave `this` é usado para referenciar elementos da classe (atributos ou métodos) e foi utilizado para diferenciar o atributo do parâmetro.

Construtores de classes em Java

- Toda classe tem um construtor definido, mesmo que não seja explicitamente definido. Esse é o construtor default `Carro()`.
- A linguagem Java já coloca um construtor vazio sem parâmetro na classe.
- Não existe classe sem construtor, apenas não é explícito.
- Podemos ter mais de um construtor na classe, desde que todos tenham tipos diferentes e quantidade de parâmetros diferentes.

Construtores múltiplos

- Podemos ter mais de um construtor na classe, desde que todos tenham tipos diferentes e quantidade de parâmetros diferentes.

```
1 public class Carro {  
2     ...  
3  
4     Carro(int pot){}  
5     Carro(String marc){}  
6     Carro(int pot, String marc){}  
7     Carro(int vel){} // Erro, pois tem um  
                        construtor com int  
8 }
```

Para ter mais de um construtor em uma classe, os parâmetros necessariamente precisam ter tipos diferentes, mesmo que a quantidade seja a mesma.

Construtores múltiplos

- Podemos ter mais de um construtor na classe, desde que todos tenham tipos diferentes e quantidade de parâmetros diferentes.

```
1 public class Carro {  
2     ...  
3  
4     Carro(int pot){}  
5     Carro(String marc){}  
6     Carro(int pot, String marc){}  
7     Carro(int vel){} // Erro, pois tem um  
                        construtor com int  
8 }
```

Para ter mais de um construtor em uma classe, os parâmetros necessariamente precisam ter tipos diferentes, mesmo que a quantidade seja a mesma.

Exemplos práticos no Netbeans

- Classe Carro.
- Classe Comércio.
- Atividades de classes em Java.

Obrigado! Fabio Takeshi Matsunaga ftakematsu@gmail.com