Residência em Software

Residência em Tecnologia da Informação e Comunicação

Classes e Objetos

Professor:

Alvaro Degas Coelho

Java

INSTITUIÇÃO EXECUTORA cepedi UESC

COORDENADORA MCTI FUTURO Softex

APOIO MINISTÉRIO DA CIÊNCIA, TECNOLOGIA E INOVAÇÃO GOVERNO FEDERAL BRASIL UNIÃO E RECONSTRUÇÃO

Residência em Software

Sobrecarga

- Métodos com o mesmo nome, mas com comportamentos diferentes de acordo com os argumentos passados, é o conceito chamado de sobrecarga de métodos.

- A sobrecarga pode ser de quantidade de argumentos (quantidades diferentes para métodos diferentes) ou de tipos de dados diferentes ou retorno de valores diferentes.

Residência em Software

Exemplo

- Suponha uma classe Estudante
- Há um método matricular(semestre) que renova sua matrícula a cada semestre
- Há um método matricular(turma) que matricula o estudante em uma dada turma
- Isto é um exemplo de Sobrecarga

Residência em Software

Exercício

- Suponha que você tenha uma classe que implementa uma lista de usuários (ArrayList da classe Usuario) e você quer criar o método listar() com três formas diferentes
  - listar() → lista todos os usuários
  - listar(int x) → lista todos os usuários a partir da posição x
  - listar (int x, int y) → lista todos os usuários entre as posições x e y (mesmo que y seja menor que x)

Residência em Software

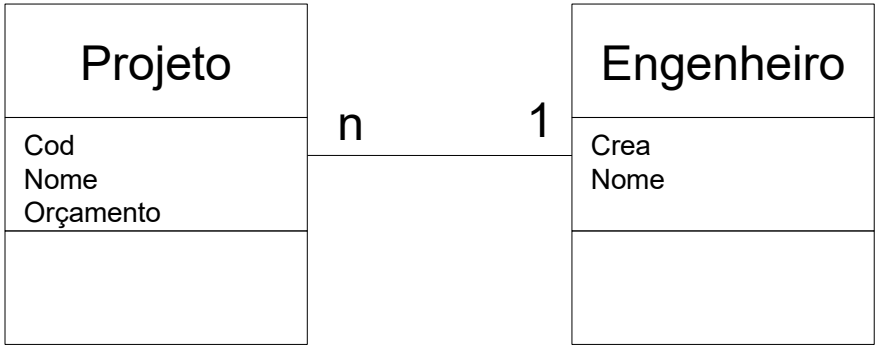
Sobrecarga de Construtores

- Como qualquer método, um construtor também pode ser sobrecarregado
- Exemplo
  - Um objeto do tipo Cliente pode ser inicializado tendo seu CPF passado como parâmetro. Mas também pode ser inicializado tendo o Nome e o CPF passados como parâmetro.
  - Implemente

Residência em Software

Exercício

- A classe Projeto ao lado tem três construtores
- Um que recebe apenas o código e o nome
- Um que recebe o código, o nome e o Orçamento
- Um que recebe o código, o nome, o orçamento e o Engenheiro responsável
- Implemente



Residência em Software

Herança

- Ocorre quando uma classe passa a herdar características (atributos e métodos) definidas em outra classe, especificada como *ancestral* ou *superclasse*. A classe receptora de recursos é denominada subclasse.

Residência em Software

Implementando

- Heranças em Java: palavra reservada *extends*
- Java propõe que uma classe não *herda* de outra. Mas a *estende*.
  - Faz tudo o que a outra faz e acrescenta/modifica algo
- Na prática é a mesma coisa. Mas filosoficamente dá o que pensar...

Residência em Software

Observe o exemplo abaixo

```
//Classe base Veiculo
class Veiculo {
 private String modelo;
 private String cor;
 private int ano;

 public Veiculo(String modelo, String cor, int ano) {
      this.modelo = modelo;
      this.cor = cor;
      this.ano = ano;
 }

 public void exibirInformacoes() {
      System.out.println("Modelo: " + modelo);
      System.out.println("Cor: " + cor);
      System.out.println("Ano: " + ano);
 }
}
```

Residência em Software

Exemplo

- Temos uma classe genérica (ou classe mãe, ou super classe) veículo
  - Atributos: modelo, cor, ano
  - Um construtor
  - Um método pra exibir informações

Residência em Software

Queremos estender (herdar)

- Uma classe Carro
- Que tem um atributo a mais: número de portas
- Isto (possivelmente) força a alteração do construtor e do método de exibir informações

Residência em Software

Como fica

```java
//Classe derivada Carro, que herda de Veiculo
class Carro extends Veiculo {
 private int numeroPortas;

 public Carro(String modelo, String cor, int ano, int numeroPortas) {
     // Chama o construtor da classe base (Veiculo)
     super(modelo, cor, ano);
     this.numeroPortas = numeroPortas;
 }

 // Sobrescreve o método exibirInformacoes da classe base
 @Override
 public void exibirInformacoes() {
     // Chama o método da classe base para exibir informações básicas
     super.exibirInformacoes();
     System.out.println("Número de Portas: " + numeroPortas);
 }
}
```

Residência em Software

Polimorfismo

- Numa relação de herança
- Um mesmo método pode ter comportamentos diferentes
  - Um para a classe base
  - Outro para a classe derivada
- É um tipo de sobrecarga...

E se eu quisesse permitir que Carro fosse instanciado também sem o número de portas?

Residência em Software

E se eu quisesse permitir que Carro exibisse informações também sem o número de portas (com um boolean informando se é pra mostrar ou não o número de portas)?

Residência em Software

Exercício

- Implemente as classes Veículo e Carro, criando o construtor e o método exibirInformações() conforme previsto nos dois slides anteriores.

Residência em Software

Exercício

- Implemente a classe Caminhão, que também herda de Veículo, mas tem atributos para o número de eixos que o caminhão possui, bem como para sua capacidade máxima.
- Construtor e exibirInformacoes() com formatos variados como no caso da classe Carro