

INF319 — Projeto e Implementação Orientados a Objetos

Solução: **Lista de Materiais**

Luiz E. Busato

Instituto de Computação – UNICAMP
{buzato}@ic.unicamp.br

Especialização em Engenharia de Software

Especificação

O cliente da lista de materiais requisitou a inclusão de mais uma operação: a listagem dos componentes de uma lista de materiais. A operação de listagem corresponde a geração de uma cadeia de caracteres (*string*) simples, com o seguinte formato:

```
Parte: cod; Descrição: desc; Custo: custo\n
```

No caso de uma lista de materiais composta, os componentes devem ser listados, em ordem crescente de código da peça, deslocados um espaço para a direita:

```
Parte: cod; Descrição: desc; Custo: custo\n
```

```
  Parte: subcod1; Descrição: desc1; Custo: custo1\n
```

```
  Parte: subcod2; Descrição: desc2; Custo: custo2\n
```

Exemplo de Listagem

Exemplo: Disquete

- Disquete
 - Disco
 - Disco magnético
 - Anel de tração
 - Caixa plástica
 - Tampa corrediça
 - Folha de alumínio
 - Mola

Exercício E_1

Exemplo: Disquete

Parte: 1; Descrição: Disquete; Custo: 26.0\n

Parte: 11; Descrição: Disco; Custo: 17.0\n

Parte: 5; Descrição: Anel de tração; Custo: 2.0\n

Parte: 6; Descrição: Disco magnético; Custo: 15.0\n

Parte: 12; Descrição: Caixa plástica; Custo: 5.0\n

Parte: 13; Descrição: Tampa corrediça; Custo: 4.0\n

Parte: 3; Descrição: Folha de alumínio; Custo: 3.0\n

Parte: 4; Descrição: Mola; Custo: 1.0\n

Problema

Implementar `public String list()` um método que produz uma listagem ordenada crescentemente dos itens de uma lista de materiais.

De onde veio a especificação da assinatura?

Da análise do código fonte do teste (junit) fornecido.

Solução

- o método `list()` deve **produzir** uma **listagem indentada dos itens**, com itens **ordenados** em ordem crescente.

Solução

Problema pode ser resolvido como a composição da solução de três sub-problemas:

- 1 algoritmo de **percurso** da lista de materiais;
- 2 geração da **listagem**: um texto (cadeia de caracteres);
- 3 **ordenação** os itens de uma (sub-)lista em ordem crescente.

Solução

- **percurso** (navegação): inspire-se na implementação do percurso (algoritmo) implementado para o cálculo do custo (`cost()`) de uma lista de materiais.
- **listagem**: é uma *cadeia de caracteres* `String`. Assim, devemos nos perguntar se Java contém alguma classe que permite a construção (manipulação) de cadeias de caracteres?
- **ordenação**: linha 16 de `Assembly`: `List<Part> parts`. `List`, `LinkedList` fazem parte de `Collection`. `Collection` permite ordenação?

Solução

- **percurso**: percurso em largura, praticamente idêntico ao utilizado por `cost()`.
- **listagem**: `StringBuilder`. Durante o percurso, o que cada classe adiciona à cadeia de caracteres (listagem)?
- **ordenação**: A classe `Collections` contém métodos estáticos que atuam sobre `Collection`. Assim, para obter a ordenação basta implementar a comparação em `Part` (`equals`, `hashCode`). IDEs (eclipse, intellij, netbeans, etc) já trazem implementações de referência para esses métodos, basta acionar a geração.