UNIVERSIDADE
FEDERAL DO CEARÁ

Programação Orientada a Objetos
Trem vol. 1

David Sena Oliveira

Quixadá, Outubro de 2014

1 Descrição

O **Trem de Ferro** é um dos transportes públicos mais comuns na Europa. Um trem é composto por uma locomotiva e uma quantidade variáveis de vagões, chamada de composição. Uma locomotiva possui um limite máximo de vagões que pode locomover e é responsável por movimentar a composição. Cada vagão pode transportar uma quantidade máxima de passageiros. Assim, vagões são ocupados gradativamente, onde um passageiro só pode ir para o segundo vagão quando o primeiro já estiver totalmente ocupado. O objetivo desse programa é permitir que passageiros possam ser embarcados, desembarcados, buscados e listados nos vagões do trem.

2 Diagrama

O diagrama da Figura 1 apresenta as classes a serem implementadas.

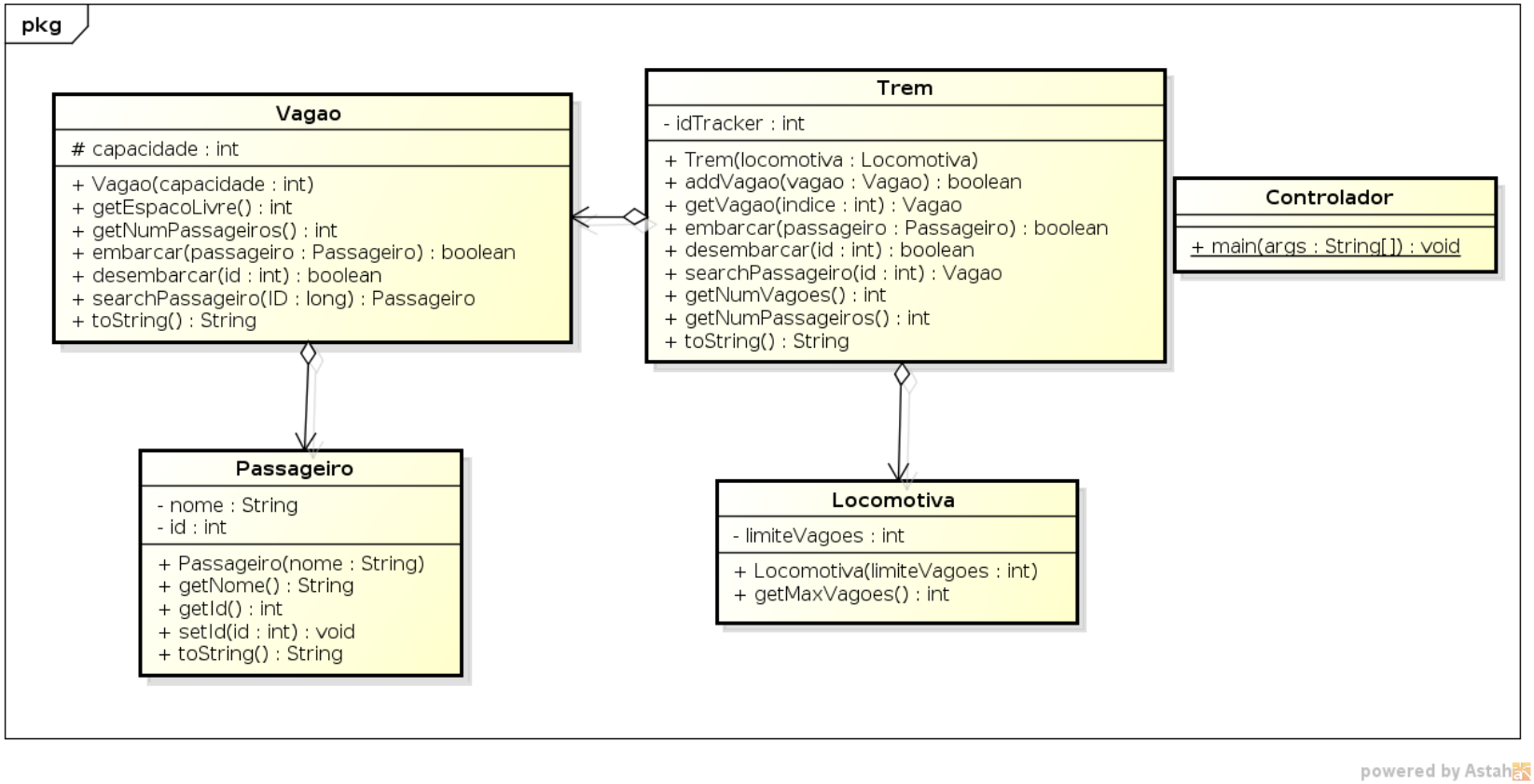

Figura 1: Diagrama de Classes

3 Métodos

A maioria dos métodos, principalmente os **get** e **set** são de implementação trivial. Os métodos não triviais estão descritos abaixo.

3.1 Passageiro

- Construtor recebe apenas o nome do passageiro. O `id` deve ser iniciado com 0.
- Método `toString()` deve ser sobrescrito para o seguinte:

```
public String toString() {
    return id + ":" + nome;
}
```

3.2 Locomotiva

- A Locomotiva é trivial.

3.3 Vagão

- Método `int getEspacoLivre()` calcula o espaço livre no vagão, dado pela **capacidade** - numero de passageiros.
- Método `boolean embarcar(Passageiro passageiro)` adiciona um passageiro no vagão observando o espaço livre disponível. Ele retorna verdadeiro se operação foi concluída com sucesso ou falso caso contrário;
- Método `boolean desembarcar(int id)` desembarca o passageiro do vagão que possui o id igual ao passado como parâmetro. Ele retorna verdadeiro se a operação foi concluída com sucesso ou falso caso contrario;
- Método `Passageiro searchPassageiro(int id)` retorna o passageiro do vagão que possui o id igual ao passado como parâmetro. Caso ele não encontre o passageiro, ele retorna `null`;
- Método `String toString()`

  Retorna uma String contendo a descrição do vagão. Deve iniciar com um [. Depois colocar o nome de cada um dos passageiros na ordem em que eles foram embarcados separando por espaço. Coloque _ para cada cadeira livre. Exemplo para vagão com capacidade 5 e duas pessoas embarcadas.

  `[ 1:Carlos 2:Mario _ _ _ ]`

3.4 Trem

- Método `void addVagao(Vagao vagao)` que adiciona um vagão ao trem observando o número máximo de vagões suportado pela locomotiva;
- Método `boolean embarcar(Passageiro passageiro)` que embarca um passageiro no trem no primeiro vagão com espaço livre disponível. Ele chama os métodos embarcar do vagão, delegando as verificações de espaço livre ao vagão. Encontrando um vagão disponível, ele usa o método `passageiro.setId(int id)` configurar o id do passageiro. Cada novo passageiro que embarca deve receber um novo id. Use o atributo idTracker para guardar manter o controle do próximo id. Ele retorna verdadeiro se o passageiro embarcou ou falso caso contrário;
- Método `boolean desembarcar(int id)` faz o desembarque do passageiro no vagão onde ele se encontra. Ele retorna verdadeiro se o passageiro desembarcou ou falso caso contrário;
- Método `Vagao searchPassageiro(int id)` procura o passageiro pelo id percorrendo todos os vagões até o encontrar. Caso encontre, retorna a referência ao vagao que ele está, caso ele não encontre o passageiro, retorna null;
- Sobrescreva o método `String toString()` para gerar uma String com iniciando com "Trem {", chamando o método `toString()` de cada vagão e finalizando com um "}". A saída do comando resulta em algo como

  ```
  Trem{[ 4:Davi ][ 2:Bruno 3:Carlos _ _ ]}
  ```

3.5 Controlador

O controlador é brinde. Dado, o controlador abaixo, a saída gerada deve ser algo como:

Listing 1: Saida Esperada

```
Trem: {[ 1:A ][ 2:B 3:C ]}
Trem: {[ _ ][ 2:B 3:C ]}
Trem: {[ 4:D ][ 2:B 3:C ]}
```

Classe Controlador

```
package controle;

import trem.Locomotiva;
import trem.Passageiro;
import trem.Trem;
import trem.Vagao;

public class Controlador {
    public static void main(String[] args) {
        Locomotiva locomotiva = new Locomotiva(2);

        Vagao vagao = new Vagao(1);
        Vagao vagao2 = new Vagao(2);
        Vagao vagao3 = new Vagao(3);

        Trem trem = new Trem(locomotiva);
        trem.addVagao(vagao);
        trem.addVagao(vagao2);
        trem.addVagao(vagao3);

        trem.embarcar(new Passageiro("A"));
        trem.embarcar(new Passageiro("B"));
        trem.embarcar(new Passageiro("C"));
        Passageiro pass = new Passageiro("D");

        trem.embarcar(pass);
        System.out.println(trem);
        trem.desembarcar(1);
        System.out.println(trem);
        trem.embarcar(pass);
        System.out.println(trem);
    }
}
```