

Trabalho Final - Consultas

Gabriel Ribas Mocelin

Pedro Leão

Faculdade de Informática - PUCRS

90619-900 - Porto Alegre - Brasil

18 de novembro de 2015

Resumo

Este documento tem como fundamento apresentar um aplicativo para ajudar na consulta sobre as linhas de ônibus de Porto Alegre, explorando os conceitos estudados na cadeira de programação orientada a objetos.

Introdução

O aplicativo desenvolvido realiza diversas consultas por paradas e linhas de ônibus de Porto Alegre, disponibilizando uma simples interface gráfica com uma biblioteca capaz de exibir um mapa das ruas para a interação com o usuário.

Para seguirmos com o desenvolvimento do trabalho é necessário saber as etapas que devem ser realizadas para o total desenvolvimento do mesmo.

Leitura e Modelagem

A partir de arquivos disponibilizados pela prefeitura de Porto Alegre, possuímos os dados necessários para o armazenamento das linhas e paradas, como por exemplo seus identificadores e posições geográficas (coordenadas).

Para a implementação foi necessário ler 4 arquivos, na qual iremos explicar a seguir:

Paradas - este arquivo possui informações como código, id, latitude, longitude e terminal de cada parada. Sabendo isto, criamos a classe *parada* para armazenar os dados referentes a cada parada e efetuar as operações sobre ela, e então criamos a classe *paradas* com um *HashMap* de paradas para agrupá-las e efetuar as operações sobre o grupo.

Linhas - este arquivo possui as informações das linhas, como id, nome, código e tipo. Semelhante a estrutura que criamos para as paradas, também criamos para as linhas, onde possui uma classe *linha* para o armazenamento e as operações de cada linha, e a classe *linhas* para o agrupamento em um *HashMap* das linhas e as operações sobre o grupo.

Coordenadas - cada linha possui muitas coordenadas e então este arquivo serve para relacionar as linhas com suas respectivas coordenadas. Este arquivo contém dados da coordenada como id, latitude e longitude e também o id da linha na qual esta parada deve ser relacionada, então para este relacionamento criamos primeiramente uma classe *Coordenadas* para o armazenamento dos dados, e então um ArrayList de *Coordenadas* na classe *linha*, na qual adiciona em cada Linha as suas respectivas Coordenadas.

ParadaLinha – Este arquivo serve apenas para relacionar quais linhas passam em quais paradas, possuindo os dados como o id da linha e o id da parada. Para isso criamos na classe *Linha* um ArrayList de *Paradas*, para adicionar quais paradas esta linha percorre, e na classe *Parada* um ArrayList de *Linhas*, para adicionar quais linhas passam por esta parada.

Agora que possuímos os dados armazenados corretamente podemos efetuamos as consultas.

Consultas e GUI

O aplicativo utiliza a biblioteca do JXMapView e a funcionalidade básica adicionada pelo professor da cadeira, e também disponibiliza 4 diferentes consultas para o usuário, na qual são:

- Consulta 1: Exibir todas as paradas e o traçado de uma linha de ônibus selecionada.

Para a realização desta consulta, criamos primeiramente um Jlist informando todas as linhas de ônibus, quando o usuário selecionar uma linha o método *consultaLinha* é chamado recebendo esta linha como parâmetro, e este método armazena em um ArrayList as coordenadas desta linha, adicionando cada uma como um ponto em um traçado, criando assim o desenho da linha no mapa, e depois para cada parada da linha, adiciona um MyWaypoint, que é um ponto específico no mapa.

- Consulta 2: Exibir as linhas de ônibus que passam em uma determinada parada.

Quando o usuário seleciona a consulta 2, todas as paradas são mostradas no mapa como um MyWaypoint, ao clicar em um MyWaypoint no mapa, as coordenadas clicadas são passadas como parâmetro para o método *encontraPorLoc* da classe *paradas* que retorna à referência da *parada* mais próxima, por isso é necessário o usuário clicar bem próximo do MyWaypoint desejado, quando o método acha a *parada*, então é acessado o ArrayList de *linha* da parada, adicionando as linhas ao Jlist, e então simulando o processo da consulta 1, ao clicar em uma linha esta é desenhada no mapa.

- Consulta 3: Exibir as linhas de ônibus que passam próximas a determinada localização no mapa, onde a proximidade é dada pela distância a um ponto qualquer da linha.

Semelhante a consulta 2, quando uma localização é clicada no mapa, as coordenadas da localização são passadas para o método *getLinhasProximas*, na

qual para cada linha de ônibus, verifica a partir do método *passaProximo* da classe *Linha* se esta possui alguma coordenada até 500 metros das coordenadas desejadas, se esta linha possui então é adicionada ao *Jlist*, no final permitindo selecionar e então desenhar as linhas no mapa.

- Consulta 4: Exibir as linhas de ônibus que passam em um conjunto de paradas selecionadas.

Quando o usuário seleciona a consulta 4, cada parada selecionada no mapa é adicionada a um *ArrayList*, o qual é enviado como parâmetro para o método *conjuntoDeParadas* ao clicar no botão “Efetuar consulta 4”, e então percorremos todas as linhas, e verificamos para cada uma se a mesma passa em todas as paradas selecionadas, se sim, populamos um *ArrayList* de *Linhas*, o qual é adicionado ao *Jlist*.

Conclusão

Conclui-se que a linguagem Java orientada a objetos, é uma ótima linguagem de programação, na qual em sua versão 8 possui uma API imensa e com diversos recursos, a praticidade dessa linguagem é levada em conta pelo fato de possuímos diversas classes e interfaces prontas para utilizarmos, como o exemplo do ArrayList que implementa a interface collections, imagine se a cada momento que criamos um ArrayList no programa precisássemos criar um vetor, controlando índices e tamanho, provavelmente seria um trabalho bem maior.