FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

sexta-feira, 24 de novembro de 2017

Praias Fluviais em Portugal

Relatório Algoritmos e Estruturas de Dados
2017/18

2MIEIC01_02:

Carlos Daniel Gomes	up201603404	carlosdcfgomes@hotmail.com
Miguel Duarte	up201606298	miguelpduarte98@gmail.com
Tiago Castro	up201606186	tiagoaraujocastro@gmail.com

ÍNDICE

Descrição

Este programa é um guia turístico para as Praias Fluviais Portuguesas. Porém, as praias não recebem todo o foco. A ajuda para encontrar zonas de alojamento, por exemplo, também é fornecida.

Assim, tendo as informações necessárias sobre as praias, alojamento, restauração e pontos de interesse turístico, o utilizador pode consultar quais são os pontos de restauração, alojamento ou, de forma geral, outros pontos de interesse mais perto de uma certa praia. Pode, também, consultar as informações sobre cada um destes tipos de dados. Estas informações vão desde uma descrição sobre o elemento selecionado até ao seu horário, se este existir.

Normalmente, uma praia tem-lhe associada um tipo de serviço: a existência de Nadador Salvador, um café, um restaurante, por exemplo. Cada serviço pode sempre ser inspecionado, sendo a data da última inspeção efetuada acessível ao utilizador como informação sobre o serviço em questão. Cada praia possui informações importantes para que o utilizador escolha com o maior critério possível: a existência de bandeira azul, indicando que a praia é limpa, a sua localização em coordenadas gps, a sua capacidade populacional e informações sobre o rio ou albufeira, dependendo se a praia a ser consultada é uma praia fluvial associada a um rio ou a uma albufeira.

Indiretamente associados às praias estão diversos serviços. Os que têm especial destaque são Pontos de Interesse turístico, Restaurantes e Alojamento. Cada um deles possui informações de interesse a quem consulta este guia. Por exemplo, um Ponto de Interesse possui uma descrição sobre ele mesmo para apelar à sua visita e o Alojamento indica se tem quartos disponíveis. Informações interessantes e convenientes associadas a este serviço são se este se encontra aberto, isto é, se está acessível ao público. Cada estabelecimento pode fechar a qualquer momento, quer seja por um determinado intervalo de tempo quer seja indefinidamente.

Solução Implementada

Para estruturar o código foram criadas 7 classes, sendo uma dedicada para o Menu. Esta última terá a responsabilidade de estabelecer a ligação entre o utilizador e o programa. Também foi implementado um namespace (chamado Utilities) que contém funções úteis, utilizadas em vários sítios ao longo do projeto.

As restantes classes estão divididas em:

- LeisureGuide - Todas as estruturas de dados que são usadas pelo programa são ser geridas neste elemento;
- Beach - Uma Super-Classe que possui como classes derivadas BayouBeach e RiverBeach. Esta implementação teve origem no facto de tal como os próprios nomes indicam, existirem elementos em comum entre um praia em rio e uma praia em albufeira;
- Restaurant - Agrega a informação sobre os elementos de restauração.
- POI (“Points Of Interest”) - Condensa a informação sobre os pontos de interesse que poderão estar presentes no guia de lazer;
- Lodging - Possui a informação sobre o alojamento referenciado no guia de lazer.

Para guardar e organizar toda esta informação temos quatro estruturas de dados, set, vetor, priority queue e hash table:

- As praias são guardadas numa árvore binária de pesquisa, implementada através do container *set* da STL (usa uma Red-Black tree). A BST possui em cada nódulo um par com o primeiro elemento a ser uma string que possui o nome do concelho ao qual a praia pertence, e o segundo elemento um apontador para uma praia. A ordem com que são inseridas na árvore é primeiramente por concelho, seguido de bandeira azul e nome da praia em último caso. (ConcelhoBeachBST beaches);
- Vetor para guardar os elementos de restauração (elementos do tipo Restaurant);
- Vetor para guardar os pontos de interesse (elementos do tipo POI);
- Vetor para guardar o alojamento (elementos do tipo Lodging);
- Priority queue (Inspection) guarda um par de Service, string; a string é o nome da praia na qual se encontra o serviço em questão. No topo da priority queue está sempre o último serviço que foi inspecionado. Existe uma priority queue para cada diferente tipo de serviço existente no guia, sendo que é utilizado um vetor para guardar as diferentes Inspections geradas;
- Hash table (closed Touristic Points) guarda elementos da struct TouristicPointPointer que têm como membro um pointer para um

TouristicPoint e a inserção dos elementos na tabela de dispersão é feita através do nome do objeto em questão.

Descrição de cada uma das classes utilizadas:

Menu

O Menu é uma classe que serve de interface entre o utilizador e o programa. Esta classe retira de um ficheiro de texto a estrutura do Menu e cria as opções lá especificadas, de forma mutável (baseado nos conteúdos do ficheiro de texto) e hierarquizada (a opção 1.1 surge após selecionar a opção 1, etc)

Leisure Guide (Guia Turístico)

É neste ponto que tudo se inicia. Todas as estruturas essenciais são criadas e vão ser utilizadas neste ponto. Esta classe vai gerir tudo: a adição, eliminação e alteração de praias, de restaurantes, de alojamento, de Pontos de Interesse e Inspeções. Vai permitir carregar informações de e para ficheiros. Portanto, é o que vai proceder à classe Menu.

Beach

Esta classe é uma classe abstrata que possui como classes derivadas as classes com maior foco neste trabalho: River Beach (praia fluvial em rio) e Bayou Beach (praia fluvial em albufeira). Por serem bastante semelhantes foi a decisão do grupo torná-las subclasses da superclasse Beach. Todas as praias possuem em seus parâmetros o nome da Praia, as coordenadas de onde se situa, uma capacidade, uma bandeira azul e os serviços (que também é uma classe) prestados na praia em questão. A partir deste momento passamos para as subclasses nas quais a Bayou Beach possui mais um parâmetro, a área aquática utilizável, e a River Beach mais três: largura, profundidade e corrente. Todas estas informações estão sempre disponíveis para o utilizador.

Service

Por questões de facilidade de implementação, foi criada esta classe que vai ser utilizada somente nas praias pois só estas apresentam serviços. Possui três

parâmetros: o nome do serviço, o tipo de serviço prestado e uma breve descrição sobre o serviço.

Restaurant

Possui como membros de dados o nome do local, o seu horário, as suas localização em coordenadas e uma breve descrição. Esta classe é derivada da classe TouristicPoint.

POI (Point Of Interest)

Possui como membros de dados o nome do local, a sua localização em coordenadas e uma breve descrição sobre o local. Não possui horário pois foi considerado que pode ser visitado a qualquer altura do dia. Esta classe é derivada da classe TouristicPoint.

Lodging

Possui como membros de dados o nome do local, a sua localização em coordenadas, um indicativo de se está totalmente ocupado e uma breve descrição sobre o local. Não possui horário porque se encontra vinte e quatro horas por dia aberto. Esta classe é derivada da classe TouristicPoint.

TouristicPoint

Possui como membros de dados a data de fecho da do ponto turistico (string vazia se estiver aberto) e o tipo de ponto turistico. Esta classe foi criada com o objetivo de unir as suas classes derivadas (POI, Restaurant e Lodging) de forma a tornar possivel a inserção na hash table.

Casos de Utilização

A estrutura do menu é a seguinte, sendo estas todas as opções e subopções:

- 1 View
  - 1.1 Unconditional Listing
    - 1.1.1 All Beaches sorted by Concelho and Blue Flag
    - 1.1.2 All POIs
    - 1.1.3 All Restaurants
    - 1.1.4 All Lodging
    - 1.1.5 Inspections of a Beach
    - 1.1.6 Inspections of a Service Type
  - 1.2 Conditional Listing
    - 1.2.1 List Beaches by Concelho
    - 1.2.2 Recommendations near a Beach
    - 1.2.3 Closed Touristic Points
- 2 Manage
  - 2.1 Beaches
    - 2.1.1 View Details
    - 2.1.2 Add
    - 2.1.3 Remove
    - 2.1.4 Modify
  - 2.2 POIs
    - 2.2.1 View Details
    - 2.2.2 Add
    - 2.2.3 Remove
    - 2.2.4 Modify
  - 2.3 Restaurants
    - 2.3.1 View Details
    - 2.3.2 Add
    - 2.3.3 Remove
    - 2.3.4 Modify
  - 2.4 Lodging
    - 2.4.1 View Details
    - 2.4.2 Add
    - 2.4.3 Remove
    - 2.4.4 Modify
  - 2.5 Touristic Points
    - 2.5.1 Close
    - 2.5.2 Reopen
  - 2.6 Inspections
    - 2.6.1 Add
- 3 File I/O
  - 3.1 Load
  - 3.2 Save

O menu da opção 1 (View) permite a visualização não condicional (1.1 - Unconditional Listing) ou condicional (1.2 - Conditional Listing) dos dados do programa, podendo nas subopções da opção 1.1 visualizar todos os detalhes de cada um dos elementos do programa e nas subopções da opção 1.2 visualizar as praias para um certo concelho ou as recomendações próximas de uma praia (restaurantes, alojamento e pontos de interesse mais próximos).

O menu da opção 2 (Manage) possui vários submenus relativos a cada tipo de dados principal do programa (Praias, Restaurantes, Alojamento e Pontos de Interesse), permitindo ao utilizador fazer uma gestão do tipo CRUD (Create Read Update Delete) de cada um destes. É possível, então, para cada um deles, observar os detalhes de um elemento específico (2.x.1 - View Details), adicionar um novo elemento do tipo selecionado (2.x.2 - Add), remover um elemento deste mesmo tipo (2.x.3 - Remove) ou modificar um elemento existente (2.x.4 - Modify). A partir do ponto 2.5 as opções são ligeiramente diferentes. No ponto 2.5 existe a opção 2.5.1 que permite fechar um estabelecimento e a 2.5.2 que permite reabrir um. A opção 2.6 apresenta apenas uma opção de adicionar uma inspeção a um serviço.

O menu da opção 3 (File I/O) permite fazer operações com ficheiros relativos aos dados internos do programa, sendo possível efetuar a gravação dos dados atuais do programa para ficheiros de texto (3.2 - Save) e também o carregamento dos dados para a memória atual do programa (3.1 - Load).

Para além destas opções é sempre apresentada a opção 0, que quando se está num submenu serve para sair até ao menu principal (0 - Back to main menu), e quando se está já no menu principal serve para sair do programa (0 - Exit).

Principais Dificuldades

Uma dificuldade propriamente dita que tivemos foi a interpretação do enunciado e o decifrar o que ele pedia, temo-nos restringido somente à nossa interpretação deste.

Não tivemos dificuldades específicas que tenham sido especialmente pronunciadas, talvez apenas alguns problemas de compatibilidade no ambiente de desenvolvimento, pois eram todos diferentes entre os vários elementos do grupo, quer sistemas operativos quer IDEs.

Porém, aproveitamos este problema para transformar isto numa vantagem, podendo testar algo que funcionasse em todas as configurações e portanto assegurasse a portabilidade do nosso projeto.

No início do projeto tivemos alguns problemas a nível de estabelecer o workflow colaborativo que iríamos usar mas eventualmente conseguimos entender melhor como trabalhar simultaneamente, usando ferramentas como o GitHub.

Distribuição do trabalho dentro do grupo

O trabalho dentro do grupo foi distribuído de forma maioritariamente igual. Porém, alguns membros do grupo estiveram responsáveis por algumas funcionalidades mais específicas:

Carlos Daniel Gomes - Priority queue

Miguel Duarte - Binary Search Tree

Tiago Castro - Hash Table

Conclusão

Em suma, este trabalho foi interessante como uma forma de pormos em prática os conceitos teóricos numa situação real, tornando-os mais palpáveis e, de certa forma, também mais fáceis de assimilar

Tínhamos concluído na primeira parte que faria sentido usar outro tipo de estruturas de dados, usadas em parte nesta segunda parte. Verificamos que, realmente, fez sentido organizar o programa de forma diferente, o que o tornou mais estruturado.

Também extraímos que é bastante importante fazer os nossos programas da forma mais modular possível pois foi uma grande vantagem aquando da reestruturação do projeto para a segunda parte.