

Trabalho Prático 01 - AEDS 1 - Trios

Professora: Thais R. M. Braga Silva

Valor: 10 pontos

Data de Entrega: 20/09/2018

Forma de Entrega: PVANet (formato .zip ou .tar.gz)

Você foi contratado para fazer a implementação de um sistema para um aeroporto. Basicamente este sistema tem por objetivo possibilitar o registro de todos os voos que decolaram das pistas deste aeroporto em um determinado dia, separando-os por faixas de horário de decolagem e pouso. Como desenvolvedor experiente, você sabe que deve fazer o desenvolvimento baseado em Tipos Abstratos de Dados (TADs). A seguir, estão definidos os TADs que você deve implementar, bem como as estruturas de dados a serem utilizadas e a lista de operações necessárias para cada um deles. É esperado que você faça a implementação COMPLETA dos TADs solicitados, utilizando a linguagem C e os arquivos .c e .h, conforme explicado em sala de aula. Seu código deve ser organizado e comentado, assim como o projeto bem estruturado. Também é esperado que você implemente, conforme definido abaixo, o programa que vai utilizar os TADs desenvolvidos.

TIPO ABSTRATO DE DADOS VOO

Para começar, será necessário que você implemente o TAD Voo. Cada voo deverá ser representado por VID (identificador do voo, gerado aleatoriamente), Horário da decolagem, Horário previsto de pouso, Aeroporto de decolagem, Aeroporto previsto de pouso e identificador da pista de decolagem (número inteiro, considerando que o aeroporto tenha N pistas de decolagem – você define o valor de N). As operações disponibilizadas, além da operação iniciar, devem ser aquelas que permitem acessar os dados do voo para leitura (operação Get) e alteração (operação Set). Exemplo: GetVID() e SetVID(int VID).

TIPO ABSTRATO DE DADOS LISTA_DE_VOOS

A finalidade deste TAD é gerenciar uma lista de voos, ou seja, uma lista linear cujos itens são do tipo Voo. A estrutura de dados que deve ser utilizada é uma lista simplesmente encadeada, implementada com célula cabeça.

Este TAD possui as seguintes operações:

a) Iniciar nova lista de voos. Esta operação deve inicializar a estrutura de dados a ser utilizada, criando uma lista linear vazia.

b) Inserir voo. A operação de inserção deverá ser implementada de maneira que um novo voo seja inserido sempre imediatamente antes do primeiro voo encontrado na lista com horário de decolagem maior ou igual ao dele. Para isso, será necessário percorrer a lista de voos desde o início, buscando o local correto de inserção.

c) Remover voo pelo identificador. Essa operação deve receber como parâmetro o identificador do voo a ser removido, retirando-o da lista e retornando-o para que possa ser utilizado.

d) Implemente a operação (c) acima sem realizar a remoção do voo da lista, apenas retornando-o para utilização. Essa operação deve se chamar Procura voo pelo identificador.

TIPO ABSTRATO DE DADOS ITEM_MATRIZ

Este TAD será utilizado para fazer a criação da matriz de voos que decolaram do aeroporto em um determinado dia, segmentados por faixas de horários de decolagem e de previsão de pouso. Cada elemento dessa matriz será do tipo Item_Matriz. Um Item_Matriz possui os seguintes campos: uma Lista_de_Voos (um elemento do tipo Lista_de_Voos, conforme definido acima), número de voos (quantidade de voos cadastrados na lista de voos), horário da última atualização. As operações, além da operação iniciar, devem ser aquelas que permitem acessar os dados do voo para leitura (operação Get) e alteração (operação Set).

TIPO ABSTRATO DE DADOS MATRIZ_VOOS

O TAD Matriz_Voos irá efetivamente cadastrar todos os voos que partiram de um determinado aeroporto em um certo dia, organizados por faixas de horários de decolagem e pouso. Para tanto, você deverá criar uma matriz $M[24][24]$. Cada linha e cada coluna dessa matriz representa a faixa de horário de 1 hora do dia. Assim, a linha 0 e a coluna 0 representam a faixa de horário de 0h (inclusive) a 1h (exclusive), a linha 1 e a coluna 1 representam a faixa de horário de 1h (inclusive) a 2h (exclusive) e assim sucessivamente. Considere que as linhas representam faixas de horário de decolagem e colunas representam faixas de horário de previsão de pouso.

Os elementos da matriz devem ser do tipo Tipo_Matriz, ou seja, a matriz deve ser declarada com esse tipo, definido anteriormente por você. Dessa maneira, cada posição i,j da matriz terá os atributos definidos por esse TAD. Além disso, cada posição i,j representa o fato de que, os voos cadastrados na lista de voos dessa posição decolaram em um horário entre i (inclusive) e $i+1$ (exclusive) e tem previsão de pouso em um horário entre j (inclusive) e $j+1$ (exclusive).

A estrutura de dados desse TAD deve conter, além da matriz $M[24][24]$, os atributos data, numero total de voos (cadastrados em todas as listas da matriz) e horário da última atualização.

As operações definidas para esse TAD são:

- a) Inicializar. Inicializa a estrutura de dados do TAD.
- b) Inserir voo. Recebe como parâmetro as informações de um voo e faz a inserção do mesmo na lista de voos da posição correta i e j da matriz.
- c) Remover voo. Deve receber como parâmetro o identificador do voo a ser removido.
- d) Procurar voo. Deve receber como parâmetro o identificador do voo a ser procurado.
- e) Imprimir voos, dados horário de decolagem e horário de pouso previsto.
- f) Imprimir voos, dado horário de decolagem apenas. A operação deve considerar todos os possíveis horários de previsão de pouso.
- g) Imprimir voos, dado horário previsto de pouso apenas. A operação deve considerar todos os possíveis horários de decolagem.
- h) Imprimir toda a matriz. Mostra todos os voos de todas as listas cadastradas.
- i) Encontrar faixa de horário de decolagem e previsão de pouso com maior número de voos cadastrados. Mostra índices i e j da posição encontrada e quantidade de voos.
- j) Encontrar faixa de horário de decolagem e previsão de pouso com menor número de voos cadastrados. Mostra índices i e j da posição encontrada e quantidade de voos.
- k) Encontrar lista de voos mais recentemente alterada. Mostra índices i e j da posição encontrada e horário da última alteração.
- l) Encontrar lista de voos menos recentemente alterada. Mostra índices i e j da posição encontrada e horário da última alteração.
- m) Verificar se matriz é esparça. Para essa operação, você deve verificar se o número de posições da matriz sem voos cadastrados é, no mínimo, o dobro daquelas com pelo menos 1 voo cadastrado.

SISTEMA DE TESTE

Para testar a implementação dos seus TADs, você deverá implementar um programa principal que representa o sistema de aeroporto que você foi contratado para desenvolver. Para tanto, este programa deverá possuir as seguintes funcionalidades:

- Inicialização da Matriz de Voos
- Seleção de dois modos de utilização do sistema: interativo e por arquivo. No primeiro, o usuário deve escolher manualmente cada operação que deseja realizar com a Matriz de Voos e passar os valores dos parâmetros necessários. No segundo, o usuário informa o nome e caminho do arquivo a ser carregado, o qual estará formatado linha a linha com as operações e parâmetros a serem executados.
- Para o modo interativo, apresentação de um menu de funcionalidades para o usuário, exibindo as operações do TAD Matriz_Voos, que podem ser solicitadas pelo mesmo. Entrar em loop de escolha de execução de operações, até que o usuário solicite a saída do modo interativo ou do sistema como um todo.

Todos os resultados deverão ser exibidas na saída padrão.

ARQUIVO DE ENTRADA

Para o modo de utilização por arquivo, o arquivo de entrada deverá possuir o seguinte formato:

```
Num_ops // Número total de operações constantes no arquivo

Op-1 param1 ... param N // Op-1 é a letra que identifica a operação do TAD
                        // Matriz_Voos, conforme acima. Os parâmetros
                        // devem ser separados por espaço

Op-2 param 1 ... param N
...

Op-[Num_ops] param 1 ... param N

//Note que a primeira operação deverá ser sempre a de inicialização (a)
```

TESTES

Será disponibilizado junto com a especificação desse trabalho um pequeno arquivo de teste e as saídas esperadas para o mesmo. Você deve, no entanto, criar seus próprios casos de teste, tanto interativos como por

arquivo. Novos testes por arquivo serão utilizados no dia da entrevista, a ser realizada como parte da avaliação deste trabalho.

Em particular, atente para:

- O programa deve ser organizado em módulos, conforme estudado em sala de aula. O módulo do programa principal deve estar separado dos módulos que compõem os TADs.
- O programa deve estar bem indentado e comentado
- Caso apareçam números fixos no código, estes devem ser definidos como constantes
- **Trabalhos copiados serão penalizados.**

ATENÇÃO: Soluções que não correspondam à implementação de Tipos Abstratos de Dados serão duramente penalizadas por não atenderem à especificação. A especificação pode conter erros ou problemas não intencionais que podem e devem ser reportados, caso encontrados. Além disso, diversas decisões de projeto e desenvolvimento precisam e devem ser feitas por cada trio. Implementações extras são possíveis e serão bem vindas, desde que tudo o que foi explicitamente pedido nesta especificação tenha sido contemplado e que, o objetivo central do tema do trabalho não seja comprometido.

O que deve ser entregue:

- Todo código fonte produzido, incluindo os arquivos de cabeçalho
- Uma pequena porém completa documentação, descrevendo o objetivo do trabalho, o projeto do sistema implementado, as principais decisões de projeto, os módulos desenvolvidos, os métodos implementados e a conclusão. O formato para a entrega da documentação é, preferencialmente, pdf
- Fazer um zip ou tar.gz de todos os arquivos, nomeá-lo com o nome e número de matrícula da dupla, bem como número do TP, e submetê-lo apenas uma vez pelo PVANet (NÃO UTILIZAR COMPACTAÇÃO RAR!).

Como será a avaliação:

- Entrevista
- Avaliação do professor