

Empresa de Transportes “Semprarrolar” – afetação de condutores –

INTRODUÇÃO

A empresa de transportes **Semprarrolar** pretende desenvolver um programa informático para fazer o escalonamento interativo dos seus condutores. A empresa é obrigada a assegurar o transporte de passageiros num conjunto de **linhas** (percursos).

Uma **linha** é caracterizada pelos seguintes atributos: número identificador único; frequência de circulação dos autocarros (pode ser diferente de linha para linha mas não varia com o dia da semana); uma lista com a sequência das paragens que a constituem; uma lista com os tempos de viagem entre paragens; e um número de autocarros necessários para assegurar o serviço da linha. Cada autocarro de uma **linha** funciona de modo bidirecional (da paragem inicial para a paragem final e de volta à paragem inicial). Uma paragem é identificada por um nome único e pode pertencer a várias linhas. Todos os autocarros partem da paragem inicial da linha a que pertencem.

Cada **condutor** da empresa é caracterizado por um nome, um número identificador único, por um número máximo de horas consecutivas que pode conduzir (um turno), pelo número máximo de horas que pode conduzir por semana, pelo número mínimo de horas de descanso obrigatório entre turnos e pelo conjunto de turnos que lhe estão atribuídos.

Um **turno** é caracterizado pela linha onde é realizado, pelo identificador do autocarro em que vai ser realizado, pelo identificador do condutor alocado, pelo instante de início e pelo instante de fim do turno.

Um **autocarro** é caracterizado pelo identificador da linha onde circula, pelo número de ordem na linha, pelo identificador do seu condutor e pelos turnos que tem que realizar.

OBJETIVOS

O objetivo último do trabalho prático é desenvolver um programa que faz, **interativamente**, para uma semana, o escalonamento dos condutores dos autocarros da frota da empresa **Semprarrolar**, respeitando os limites máximos de horas de condução e horas de descanso obrigatórias.

Este segundo trabalho é uma continuação do Trabalho 1 pelo que deve reutilizar grande parte do código desenvolvido no Trabalho 1. Neste trabalho será dada mais ênfase a novos aspetos da linguagem C++ que não foram utilizados no trabalho anterior. Em particular será obrigatório a utilização de um conjunto de classes adequadas à aplicação.

Para além dos conhecimentos treinados no primeiro trabalho, ao realizar este segundo, os estudantes terão oportunidade de aplicar os conhecimentos mais avançados de programação em C++ adquiridos até ao momento, e que incluem:

- definição e uso de classes;
- *overload* de funções;
- definição e *overload* de operadores;
- utilização de funções *Template*;
- utilização de iteradores;
- algoritmos de ordenação;
- uso de diversos tipos de estruturas de dados (*maps*, *multimaps*, etc);

DESENVOLVIMENTO DO PROGRAMA

Definição de novos tipos de dados

Neste segundo trabalho os estudantes deverão substituir a utilização de estruturas (**struct**) associadas à definição de novos tipos de dados (**typedef**), utilizadas no trabalho anterior por um conjunto de classes. Os métodos que manipulavam os dados armazenados nas *struct* serão agora parte dos membros função das classes que contêm a informação das “*struct* correspondentes”. Será fornecido aos estudantes um ficheiro com um esboço inicial de classes adequadas a esta aplicação. O estudante tem toda a liberdade de rejeitar ou adaptar a estrutura de classes sugerida.

É importante reparar que a caracterização de algumas das entidades desta aplicação (linha, autocarro, turno, condutor, ...), mencionadas na introdução, têm características que podem tornar-se redundantes. Por exemplo: guardando os turnos de um condutor e o seu nome na classe **condutor**, não precisaria ter o campo nome do condutor na classe **turno**. Esta redundância permite, no entanto, simplificar a codificação, usando a mesma definição da classe **turno** tanto nos autocarros como nos condutores, por exemplo.

Funcionamento geral

O programa deve começar por ler o ficheiro de **linhas** e o ficheiro de **condutores**. O ficheiro de linhas tem, para cada linha, a seguinte informação: um identificador numérico único; a frequência dos autocarros na linha; a sequência de paragens; e uma lista com os tempos de viagem entre paragens. A frequência, expressa em minutos, indica o intervalo de passagem de um autocarro em cada paragem. Uma frequência com valor 15 (minutos) indica que em qualquer paragem daquela linha passa um autocarro de 15 em 15 minutos. Determinando o tempo de ida e volta de um autocarro (da paragem inicial até à final e de volta à inicial) e sabendo a frequência dos autocarros nessa linha pode calcular-se quantos autocarros são precisos na linha. Dependendo do tempo de “ida e volta” numa linha, os autocarros podem ter necessidade de esperar alguns minutos (sempre inferior à frequência de autocarros na linha) para garantir a cadência regular de autocarros antes de reiniciar um novo percurso. Depois de ler o ficheiro de linhas pode calcular a informação necessária para os autocarros. Pode calcular o número de autocarros necessários para, em cada linha, satisfazer a frequência de autocarros. O número de autocarros pode ser calculado com base na frequência de autocarros na linha e na duração de uma viagem de ida e volta (percurso completo), usando a fórmula $n = (\text{int}) ((\text{double}) \text{tempo_ida_e_volta} / \text{frequencia_dos_autocarros_na_linha} + 1.0)$. A informação sobre os autocarros pode ajudar na atribuição interactiva de turnos aos condutores. Por exemplo, pode construir uma “lista” com todos os “turnos” de todos os autocarros representando o serviço inicial a atribuir. À medida que vai atribuindo serviço aos condutores vai removendo os correspondentes turnos dos autocarros que vão sendo servidos. Quando um autocarro já não tem serviço por atribuir sai da “lista”. Para facilitar a interação na atribuição de serviço aos condutores pode mostrar qual o serviço que ainda falta atribuir, percorrendo a lista dos autocarros.

O número de ordem de um autocarro numa linha é um valor entre 1 e o número de autocarros dessa linha. O primeiro autocarro a sair (logo na hora de abertura do serviço) tem um número de ordem 1. O segundo autocarro que sai “frequência” minutos depois tem o número de ordem 2, e assim sucessivamente para os restantes autocarros dessa linha.

O ficheiro de condutores armazena, para cada condutor, a seguinte informação: o identificador numérico único do condutor; o nome do condutor; o número máximo de horas do seu turno; o número máximo de horas que pode conduzir por semana; e o número mínimo de horas de descanso entre turnos. O formato dos ficheiros será especificado adiante.

Ficheiro de linhas e de condutores

Qualquer um dos ficheiros utilizados no programa (linhas e condutores) são ficheiros de texto que podem ser criados com um editor de texto simples como, por exemplo, o *Notepad* do *Windows*. O programa deve começar por perguntar ao utilizador qual o nome dos 2 ficheiros a usar, terminando com um erro caso algum dos ficheiros indicados não exista.

Apresenta-se a seguir um possível conteúdo de um ficheiro de linhas.

| |
|---|
| 201 ; 20 ; Paranhos, Monte dos Burgos, ISEP, Hospital de Sao Joao; 10, 20, 45 |
| 207 ; 10 ; Aliados, Carmo, Galiza, Rotunda da Boavista; 5, 20, 20 |
| 505 ; 30 ; Matosinhos, 7 Bicas, Rotunda dos Produtos Estrela, Senhora da Hora; 40, 30, 45 |
| 506 ; 15 ; Ribeira, Alfandega, Fluvial, Foz; 10, 25, 55 |

Exemplo de um ficheiro de linhas (inventadas).

Cada uma das linhas do ficheiro contém a seguinte informação acerca de cada linha: o número identificador único; frequência de circulação dos autocarros na linha (em minutos); uma lista com a sequência das paragens que a constituem (elementos da lista separados por vírgula); uma lista com os tempos (em minutos) de viagem entre paragens (elementos da lista separados por vírgula). Os diferentes tipos de dados estão separados entre si por ";" e os elementos que fazem parte de uma lista estão separados entre si por ",".

Um conteúdo possível de um ficheiro de condutores é de seguida mostrado.

| |
|---|
| 1 ; Antonio Manuel ; 4 ; 36 ; 2 3 ; Joaquim Belmiro ; 8 ; 48 ; 4 |
|---|

Exemplo de um ficheiro de condutores.

Cada uma das linhas do ficheiro de condutores tem os seguintes campos: um identificador numérico único; um nome; o número máximo de horas consecutivas que pode conduzir (um turno); o número máximo de horas que pode conduzir por semana; e o número mínimo de horas de descanso obrigatório entre turnos. Os itens estão separados entre si por ";" .

Assuma que o transporte se faz apenas durante o dia e defina como constantes as horas de início e de fim dos serviços diários. Todos os autocarros começam o serviço sempre no mesmo extremo da linha. Cada autocarro faz o percurso até ao outro extremo e volta para a paragem inicial. Este percurso é feito desde a hora de início até à hora de encerramento do serviço no final do dia. Um percurso é sempre iniciado desde que a hora de partida seja inferior à hora de encerramento do serviço (mesmo que no final da viagem já tenha sido ultrapassada a hora de encerramento).

Sugestão de implementação

Uma vez que a aplicação respeita apenas ao escalonamento de uma semana de trabalho, sugere-se que use uma escala linear para o tempo, em minutos, para representar o tempo no período de uma semana. O dia de segunda-feira seria representado por [0, 1439] ($1430 = 24 \times 60 - 1$ minutos), terça-feira = [1440, 2879], etc. O período de serviço de segunda-feira (por exemplo 8h até às 23h) seria representado por [480, 1379] ($8 \times 60 = 480$ minutos e $23 \times 60 - 1 = 1379$ minutos).

Funcionalidades a implementar

As funcionalidades que a aplicação deve ter no final deste trabalho inclui as seguintes:

1. Ler e guardar a informação de linhas e condutores armazenada em ficheiros;
2. Gerir os condutores: criar, alterar e remover um condutor;
3. Gerar e visualizar de modo formatado tabelas com horários de uma paragem;
4. Visualizar o trabalho atribuído a um condutor;
5. Visualizar a informação de um autocarro;
6. Visualizar a informação de uma linha, visualizando de modo formatado a tabela com o seu horário ;
7. Inquirir sobre quais as linhas que incluem determinada paragem;
8. Calcular e visualizar um percurso e tempo de viagem entre duas quaisquer paragens indicadas pelo utilizador. Deve procurar a solução: na linha que contém a paragem inicial ou; na linha que contém a paragem inicial e numa das linhas que tenha paragens comuns com a "linha inicial". Havendo mais do que uma solução o programa deve apresentar todas as soluções encontradas mas por ordem crescente de distância. Se não encontrar uma solução dentro das restrições indicadas, o programa deve indicar que não encontrou uma ligação entre as duas paragens.
9. Listar todos os períodos de autocarros sem condutor atribuído.
10. Listar todos os períodos de condutores sem o serviço completo atribuído (que não tenham atingido o limite máximo semanal).

11. Efetuar interactivamente a atribuição de serviço a um condutor, permitindo ao utilizador ver as disponibilidades de serviço dos autocarros, pedindo ao utilizador informação sobre um novo turno e verificando a consistência dessa informação com as restrições de horário do condutor e restantes dados existentes.

MELHORIAS OPCIONAIS

Apresentam-se a seguir alguns desafios para desenvolvimentos adicionais:

- Melhorar as estruturas de dados de modo a tornar a aplicação mais eficiente.

NOTAS SOBRE O DESENVOLVIMENTO

Funcionamento geral do programa

O funcionamento geral do programa é o seguinte:

- Ler o conteúdo dos ficheiros de linhas, condutores e parâmetros da aplicação e guardar numa estrutura de dados adequada.
- Começar por exibir um menu onde aparecem as opções que implementam as funcionalidades: gestão de linhas; gestão de condutores; visualização de informação (horários, linhas e percursos); escalonamento de condutores.
- No final, gravar (se necessário) a informação das linhas e dos condutores.

Devem ser tomadas as precauções necessárias para evitar que o programa deixe de funcionar corretamente devido a entradas inválidas do utilizador, nomeadamente, valores fora da gama admissível.

Escrita do código

- Na escrita do código devem ser respeitadas as indicações dadas nas aulas, nomeadamente, em relação aos seguintes aspetos:
- Escolha das estruturas de dados mais adequadas para representar os dados do programa.
- Escolha adequada dos identificadores de tipos, variáveis e funções.
- Estrutura modular do código.
- Escolha adequada das classes
- Separação em dois ficheiros (.h e .cpp) da definição de cada classe e da implementação dos seus membros função
- Comentários ao código.

ENTREGA DO TRABALHO

- Criar uma pasta com o nome **TxGyy**, em que **x** representa o número da turma e **yy** representa o número do grupo de trabalho – por exemplo, T2G05, para o grupo 5 da turma 2 - e copiar para lá o código-fonte do programa (apenas os ficheiros com a extensão .cpp e .h). Incluir também um ficheiro ReadMe.txt (em formato de texto simples) indicando o estado de desenvolvimento do trabalho, isto é, se foram cumpridos todos os objetivos ou, caso contrário, quais os que não foram cumpridos, ou ainda que melhorias foram implementadas, se for esse o caso. Tudo o que não estiver especificado pode ser decidido pelos elementos do grupo de trabalho, sendo as decisões adoptadas descritas no mesmo ficheiro ReadMe.txt.
- Compactar o conteúdo desta pasta num ficheiro **Gxx.zip** ou **Gxx.rar** e submeter este ficheiro na página da disciplina de Programação, no Moodle da FEUP. Não serão aceites entregas por outras vias.

- Os estudantes deverão informar o docente da turma TP respetiva qual a constituição dos grupos de trabalho, até **25/abril**; a lista dos grupos constituídos será oportunamente publicada no Moodle.
- Data limite para a entrega: **15/maio/2017 (às 23:55h)**.