

Faculdade de Ciências e Tecnologias da Universidade de Coimbra

Manual do Programador do Projeto de PPP 2015/2016



TURMA TP3

Teresa Salazar 2015234237

João Mendes 2015233148

Introdução

Com este relatório pretendemos descrever a nossa aplicação na perspetiva da sua implementação. O programa permite um bom funcionamento de uma agência de viagens utilizando listas ligadas durante a execução e ficheiros para armazenar informação entre cada execução do programa.

Estruturas

Há dois tipos de estruturas (structs) no nosso projeto: struct Viagem e struct Cliente.

Cada viagem é uma estrutura constituída por um destino, uma data (dia, mês, ano), um numero que representa a soma da data ($\text{soma} = \text{ano} * 10000 + \text{mês} * 100 + \text{dia}$) que é utilizada para efeitos de ordenação, o número de clientes que possuem a viagem, o numero máximo de clientes para a viagem e, finalmente, uma lista de clientes em espera para a viagem.

Cada cliente é uma estrutura constituída pelo nome, número de cartão de cidadania e uma lista de viagens adquiridas por esse cliente.

Listas Ligadas

As viagens estão guardadas como elementos de uma lista ligada de viagens assim como os clientes estão guardados como elementos de uma lista ligada de clientes.

No início da execução programa, são logo criadas tanto a lista ligada de clientes como a lista ligada de viagens. São carregados os ficheiros e a sua informação é transferida para dentro das listas. É possível inserir viagens e clientes para além dos já existentes nos ficheiros, durante a execução do programa, se assim o utilizador o desejar. Estes, após inseridos, ficarão guardados nos ficheiros para posteriormente serem carregados numa futura utilização do programa.

Prints

As funções para listar clientes e viagens estão todas no ficheiro prints.c. A função `print_list_clientes` imprime todos os clientes da agência de viagens. A função `print_list_viagens` imprime as viagens para um dado destino. A função `print_list_todas_viagens` imprime todas as viagens disponíveis na agência de viagens. Por último, `print_viagens_adquiridas` imprime todas as viagens adquiridas de um dado cliente.

Ordenar

As funções de ordenamento encontram-se no ficheiro sort.c. A primeira função - `sort_data_recente_primeiro` – ordena viagens por data, sendo que as mais recentes vêm primeiro. Como já tinha sido afirmado, cada viagem possui na sua estrutura um número inteiro que representa a soma da data, sendo assim

mais fácil comparar datas. Em vez de compararmos anos, meses e dias basta compararmos a soma_data. A segunda função - sort_data_antigo_primeiro – ordena viagens por data, sendo que as mais antigas vêm primeiro, como o próprio nome o diz.

Adquirir

Cada cliente pode adquirir qualquer viagem se houver ainda lugar livre nessa viagem. A função de adquirir viagem chama uma função procura_cliente que permite ao cliente fazer o login. Posteriormente, através da função escolhe_viagem onde o cliente seleciona dentro de todas as viagens a que quer adquirir. Se o número de clientes da viagem não for igual ou superior ao número de clientes máximo da viagem, o cliente adquire a viagem o número de clientes da viagem é incrementado. Caso contrário, se não houver lugar livre na viagem (se o número de clientes for igual ao número de clientes máximo), o cliente é colocado em lista de espera.

Cancelar

É possível também cancelar viagens. Caso haja clientes em lista de espera, o cliente no topo da lista é promovido, passando a adquirir a viagem (segundo a disciplina FCFS - first come first served). O número de clientes da viagem só é decrementado caso não haja clientes para serem promovidos (clientes em fila de espera).

O cliente, se assim o desejar, também pode cancelar pedidos em fila de espera para viagens.

Para ambas as situações, a compra é removida do ficheiro ficheiro_compras.txt.

Ficheiros

Para armazenar informação utilizámos ficheiros que são carregados quando o programa é reiniciado.

No ficheiro ficheiro_clientes.txt cada linha representa um cliente e tem informação sobre o nome e o número do cartão de cidadania, separados por vírgulas, sem espaços.

No ficheiro ficheiro_viagens.txt cada linha representa uma viagem e tem informação sobre o destino, o dia, o mês, o ano e o número máximo de clientes da viagem.

No ficheiro ficheiro_compras.txt cada linha representa uma viagem adquirida por um cliente e tem informação sobre o nome, o número do cartão de cidadania, o destino e a soma_data da respetiva viagem.

Sempre que o utilizador adiciona um cliente ou uma viagem ou adquire uma viagem, é escrita informação nos ficheiros ficheiro_clientes.txt, ficheiro_viagens.txt e ficheiro_compras.txt, respetivamente.

Proteções

Para que o programa funcione sem problemas, criámos proteções para cada input do utilizador. Há proteções para diferentes tipos de dados como inteiros e strings. Enquanto o utilizador não inserir corretamente os inputs, a função chama-se a ela mesma para pedir de novo os dados.

Conclusão

Para terminar, pensamos que no geral o trabalho ficou bem feito o corresponde ao que foi pedido.