\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Última atualização - 11/05 às 23:14

User : Dani

Comecei a tratar de passar packets para .txt mas foram encontrados problemas pelo caminho

Tratei da mal formatação de packets.txt e da sua leitura

Clients já guarda, falta agência e packets

Atualizei o zip

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Step a trabalhar → 1.5 (ver segunda página)**

!!!A FAZER:

* Em void clientsMenu(Agency & agency) -> na opção 4 (ver client info), possibilitar 2 escolhas, escrever VAT ou escolher de lista.
* Embelezar a parte de ver client info
* A opção de mudar packs deve dar para remover e adicionar packs em vez de obrigar a escrever tudo de novo

A CORRIGIR:

* Change packet, change availability, voltar atrás dá mal
* Buy Packet,verificar available seats e agregado (\*1) - se pode comprar o pack
* Buy Packet, atualizar o total purchased(\*2)
* Add Client, \*1 & \*2
* Change Client(está feito dentro do possível)
* Na validação de strings como o state/location e a morada, não estão a ser permitidas vírgulas e deviam estar
* Na validação da rua não é permitido usar números - ex: rua dos 4 caminhos e outras ruas com números não são permitidas e deviam ser

A aplicação que deve implementar neste trabalho inclui as seguintes funcionalidades:

0.Ctrl-Z para voltar para trás

1. Ler e guardar a informação da agência, dos clientes e dos pacotes turísticos

armazenada em ficheiros.

1.5 Guardar (Atualizar) a informação da ~~agência~~, clientes (Necessário guardar nome de ficheiro de clientes) e packets de memória para ficheiros

2. Clientes:

* Criar
* Alterar
* Remover

Packs:

* Criar
* Alterar
* Remover (colocar como indisponível)

3. Gerar e visualizar de modo formatado a informação de um cliente especificado.

4. Gerar e visualizar de modo formatado a informação de todos os clientes da agência.

5. Gerar e visualizar de modo formatado os pacotes turísticos disponíveis

todos,

todos relativos a um destino específico

todos entre duas datas,

todos os relativos a um destino específico e entre duas datas

6. Gerar e visualizar de modo formatado os pacotes turísticos vendidos

a um cliente específico,(Escolher por vat / escolher por lista)

a todos os clientes

7. Efetuar a compra de uma pacote turístico por um cliente.

8. Calcular e visualizar o número e o valor total de pacotes vendidos.

9. Obter o nome dos N locais mais visitados (um pacote pode incluir visitas a vários locais),

ordenados por ordem decrescente do número de visitas

( = número de pacotes vendidos (lugares reservados) que passam por esse local).

10. Gerar uma listagem de todos os clientes na qual se indica, para cada cliente, um dos pacotes em que seja visitado um dos N locais mais visitados (ver ponto 9) que ele ainda não visitou.

<https://stackoverflow.com/questions/982677/command-to-collapse-all-sections-of-code>

**8.** Necessário apresentar:

Número total de pacotes vendidos:

Pelo packets.txt -> soma dos lugares reservados em cada pacote

Pelo clients.txt -> somar o número de pacotes comprados por cada cliente

Valor total de pacotes vendidos:

Pelo packets.txt -> para cada pack multiplicar o numero de lugares reservados pelo preço por pessoa, somar os valores de cada pack

Pelo clients.txt -> Para cada client ver quanto gastou vendo o preço por pessoa de cada pack, multiplicar pelo agregado familiar, somar de todas as pessoas

**9.**

Fazer um map (<https://www.geeksforgeeks.org/map-associative-containers-the-c-standard-template-library-stl/>) com sitios e vezes visitados, quando pedir os N mais visitados, ver valor mais altos do map e apresentar ()

Determinar os sítios que ficarão como keys e ao mesmo tempo atualizar valores:

Tem de percorrer seguinte lista: (Por em Menus.cpp depois das opções do mainMenu para visualizar).

cout << agency.getPackets().at(0).getSites().at(0) << endl;

cout << agency.getPackets().at(0).getSites().at(1) << endl;

cout << agency.getPackets().at(0).getSites().at(2) << endl;

cout << agency.getPackets().at(1).getSites().at(0) << endl;

cout << agency.getPackets().at(2).getSites().at(0) << endl;

cout << agency.getPackets().at(2).getSites().at(1) << endl;

cout << agency.getPackets().at(2).getSites().at(2) << endl;

cout << agency.getPackets().at(2).getSites().at(3) << endl;

cout << agency.getPackets().at(2).getSites().at(4) << endl;

Esboço de pedaço de código:

Um “for” para getPacks() com .size() (variável i)

um “for” para getSites() com .size() (variável j)

Se não estiver no map, adiciona uma Key e põe value a 0, se já estiver no map não adiciona key

Soma ao valor que já está lá o valor de agency.getPackets().at(i).getReservedSeats()

Arranjar forma de organizar o map pelo value (hope it’s possible)

Mostrar os N mais visitados -> Mostrar os primeiros n do map

**10.**

Pedir o 9. e guardar os locais num vetor (vetor 1)

fazer um vector2 com ID dos packs que tenham pelo menos um dos locais de vetor 1

Ir cliente a cliente, comparar os packs de vetor2 com os packs de client, ao primeiro pack de vetor2 que não estiver nos packs de client, dar cout de Cliente - ID e Destino(s) de Pack.

(Cheira-me que vai dar coco em casos maiores, com packs com alguns destinos iguais isto não funciona a 100%, mas depois vê-se)