

Universidade de Coimbra
Faculdade de Ciências e Tecnologia de Coimbra
Departamento de Engenharia Informática

PROGRAMAÇÃO ORIENTADA AOS OBJETOS



Criar uma Agencia de viagens

Trabalho Realizado por:
Alexandre Ferreira Costa Nº 2014206463
Fábio Silva Antunes Nº 2014206491

Introdução:

Este trabalho foi realizada no âmbito da cadeira de Programação orientada a objetos e tem como objetivo familiarizar os alunos para a programação orientada a objetos.

O programa consiste na criação de uma agência de viagens onde existem vários administradores, dois tipos de cliente (regular,premium) e várias viagens.

Estrutura geral do programa:

Para que o programa corra normalmente é necessário possuir um Administrador, caso não exista ele é criado automaticamente. (username:abc abc Pass:abc).

O programa criado maioritariamente dividido em duas partes:

- A parte do menu do Administrador que tem as seguintes opções:
 1. Criar Viagens;
 2. Alterar Viagens;
 1. Origem;
 2. Destino;
 3. Data início;
 4. Data fim;
 5. Preço;
 6. Sair para o menu inicial;
 3. Listar viagens;
 4. Eliminar viagem;
 5. Eliminar Cliente;
 6. Criar cliente regular;
 7. Criar cliente premium;
 8. Comentários;
 9. Criar Administrador;
 10. Listar Clientes;
 11. Estatísticas:
 1. Viagem mais vendida num determinado mês;
 2. Cliente que mais viagens comprou num determinado mês;
 3. Listar todas as viagens que não tiveram reservas;
 4. Listar as reservas de uma viagem;
 5. Listar as reservas canceladas de uma viagem;
 6. Listar as reservas/clientes em espera;
 7. Identificar a viagem com melhor pontuação num determinado mês;
 12. Criar autocarros;
 13. Listar Autocarros;
 14. Sair
- A parte do menu Cliente que tem as seguintes opções:
 1. Listar viagens;
 2. Suas viagens executadas;
 3. Reservar viagens;

4. Cancelar Viagens;
5. Listar reservas;
6. Comentários;
 1. Adicionar comentários;
 2. Ver comentários;
 3. Voltar ao menu inicial;
7. Sair

Também ele é composto por 13 classes e 12 ArrayList que são essências para que todas estas opções corram devidamente.

Classes:

- **Public class Viagens;**
- **Public Class Autocarros;**
- **Abstract ID;**
- **Public class Data;**
- **Class Reservas;**
- **Class Valida extends Reservas;**
- **Class Espera extends Reservas;**
- **Class Cancelada extends reservas;**
- **Public class Adimin extends ID;**
- **Public class Cliente extends ID;**
- **Public class Comentarios;**
- **Class Agencia;**
- **Class Ficheiro;**

ArrayLisy:

- **List <Comentarios> lista_comentarios:** Cada viagem tem uma lista desta, nesta lista são guardados todos os comentários e todas as pontuações das viagens já realizadas.
- **List <Cliente> Clientes:** Lista onde são guardados todos os clientes regulares.
- **List<Cliente> Clientes_pr:** Lista onde são guardados todos os clientes premiun.
- **List<ID>:**Lista onde são guardados todos os administradores.
- **List<Viagens> lista_viagens:** Tem as viagens que foram criadas e ainda não foram realizadas.
- **List < Viagens> lista_viagens_feitas:** Contém todas as viagens que já foram realizadas.
- **List <Viagens> viagens_canceladas:** Lista onde são armazenadas todas as viagens que o administrador cancela.
- **List<valida> lista_reserva:** Contem todas as reservas validas de um cliente.
- **List <espera> lista_espera:** Contém todas as reservas em espera de um pessoas (reserva em espera: reserva que ainda não passou a valida por o autocarro da viagem se encontra cheio, aguarda alguma desistência para poder passar a valida).

- **List <cancelada> lista_cancelada:** Contem todas as reservas que o utilizador cancela.
- **List <reservas> lista_reserva_viajada:** Contém todas as viagens feitas pelo cliente.

Viagens

Atributos:

Os atributos desta classe são um inteiro código, onde cada viagem tem o seu código diferente, um lugar de Origem um de Fim, uma data de início e fim associadas, um preço, um autocarro, os lugares associados aos cliente dessa viagem e ainda uma lista de comentários.

Métodos

- **Public List <Comentarios> getlintaComentarios():** Este método devolve uma lista de todos os comentários associado a uma viagem.
- **Public void adicionaComentarios (Viagens viagem,Cliente cliente,String comentario,int pontuacao):** Este método adiciona um comentário e uma pontuação a uma viagem.
- **Public void verComentarios ():** Lista todos os comentários de uma viagem.
- **Public Cliente[]listarLugares():** Lista os lugares do autocarro.
- **Public int AssociaLugar (Cliente U):** Retorna o valor do lugar para onde vai o cliente.
- **Public void retiraLugar (Cliente U):** Desassocia o lugar de autocarro ao cliente.
- **Public boolean verificaAutoLugar():** Verifica se existe algum lugar livre no autocarro se existir retorna true se não retorna false.

Autocarros

Atributos:

Na classe dos autocarros só existem dois atributos a matricula que é dada na forma de string e a capacidade que é dada na forma de inteiro.

ID

Atributos:

O ID é a classe importante porque é a classe que guarda a identificação dos administradores como dos clientes.

Guarda o Nome a Pass a morada o telefone o email e o NIF.

Data

Atributos:

A classe data tem como atributos o Dia o mês e o ano, esses atributos vão ser muito uteis tanto para fazer o login para criar viagem com para reservar viagem e muitas outras coisas.

Construtor

Public Data (String Dia, String mês, String ano);

Métodos

- **DiferencaData(Data data_viagem, Data data_reserva):** Este método vai retornar o numero de dias que passaram de uma data para a outra.

Reservas

Atributos:

As reservas tem associado uma Viagem um Cliente um valor da viagem uma data da reserva e um lugar.

Construtor

Public reservas(Cliente cliente, Viagens v, double valor, Data data)

A class cancelada, valida e a espera são extend da class das reservas.

Agencia

Atributos:

A Agencia tem a maioria dos ArryList(), tem as lista de Administradores a de Cliente, clientes pr, lista viagens, lista autocarros, listas viagens feitas e ainda as viagens canceladas.

Construtor

Agencia()

Métodos

- **Public void VerificaData(Data data_geral):** Verifica a data e vê se as viagens já ocorreram. As que já foram efetuadas vão ser transferidas para a lista_viagens_feitas.
- **Public void Criar_admin(String nome , String pass, String morada, String telefone, String email, String NIF):** Percorre todas as litas dos Clientes e do administrador e se se existe algum cliente/admin com o nome que foi dado não vai ser criar um admin porque já existe ou um cliente ou
- **Public void Criar_cliente(String nome , String pass, String morada, String telefone, String email, String NIF):** Parecido ao criar_admin.

- **Public boolean ValidaData(Data data)** :Verifica se a data valida segundo o calendário universal
- **Public void IDlogin(String nome, String Pass):** Faz o login e vai para o menu admin se for estiver numa conta de administrador, se fizer o login com a conta do cliente o cliente vai para o menu Cliente
- **Public void Menu_administrador(Adimin U, Data Data_geral, int código):**Função que gere todos os processos do administrador.
- **Public void Menu_cliente(Cliente U, Data Data_geral):** função que gere todos os processos do cliente.
- **Public Viagens retorna_viagem(int código):**Esta função vai retornar uma viagem dado um código . vai percorrer a lista de viagens criadas e a viagem que tiver o mesmo código que o código inserido vai ser retornada.
- **Public void remover_cliente(String Nome):** Elimina um cliente e atualiza as listas de cliente.
- **Public boolean verificaAuto (Data Data_fim,Data data_nova_viagem):**se a Data_fim for maior ou igual a data_nova_viagem retorna false, se não retorna true.

Adimin

Atributos:

Não existe atributos, ele vai busca-los a classe ID.

Construtor

- **public Adimin (String Nome, String Pass, String morada, String telefone, String email, String NIF)**

Métodos

- **public void Criar_viagens(List <Viagens> lista_viagens, int Codigo,String Origem,String Destino, Data Data_inicio, Data Data_fim, double preco,Autocarros auto):**Cria_viagens.
- **public void Cancelar_viagens(List <Cliente> clientes,List <Cliente> clientes_pr, List <Viagens> lista_viagens, List<Viagens> viagens_canceladas,int Codigo):** Cancela uma viagem caso a viagem exista.
- **public void criar_autocarros(List <Autocarros> autocarros):**Cria autocarros com um numero de capacidade a definir pelo administrador.
- **Viagens ComparVendidas(List <Viagens> lista_viagens_feitas,List<Cliente> lista_clientes_r,List<Cliente> lista_clientes_pr, int mes):** Compara as viagens feitas com os utilizadores e vê qual é a viagem que teve mais volume de vendas num determinado mês. Retorna a viagem com mais volume de vendas daquele mês.

- **public void listar_autocarros(List <Autocarros> autocarros)**
- **public Viagens ComprarPontuação(List <Viagens> lista_vigens_executadas,int mes):** Compara as viagens todas efetuadas e vê qual é que obteve maior pontuação.
- **public boolean verReseva(List<Cliente> clientes, List <Cliente>cliente_pr,Viagens Viagem):** Percorre todos os clientes regulares e premiun e verifica se existe ou não reserva.

Cliente

Atributos:

Categoria: 0-cliente regular ou 1-cliente premiun

Construtor

- **public Cliente (int categoria,String Nome, String Pass, String morada, String telefone, String email, String NIF)**

Métodos

- **public int get_categoria ()**
- **public String getNome()**
- **public List getListaEspera()**
- **public List getListaReservas()**
- **public void reservar_viagens (Cliente cliente,Viagens Viagem, Data data):**Esta função vai verificar se há autocarros disponíveis se houver vai ainda verificar se há lugares livres nesse autocarro. Se houver lugares livres cria uma reserva valida e conclui a reserva.
- **public void Cancelar_viagens(Cliente cliente,Viagens Viagem,Data data):**Esta função cancela a reserva e desassocia o lugar no autocarro correspondente à sua reserva. Só para os regulares
- **public void Cancelar_viagensPr(Cliente cliente,Viagens Viagem,Data data):** Esta função cancela a reserva e desassocia o lugar no autocarro correspondente à sua reserva só para os clientes premiun.
- **public void listar_reservas():**Lista todas as reservas do cliente.
- **public void listar_reserva_viagem (Cliente U, Viagens V_1):** Esta função lista as reservas todas de uma determinada viagem.
- **public List<reservas> getListReservaEfetuada():** Retorna a lista de reservas viajadas.
- **public void verificaLugarAuto(Cliente U, Data Data_geral):**Verfica se existe algum lugar disponível.
- **public void removeReserva(Viagens viagem_removida,Cliente clien):**remove a reserva, passa de valida a cancelada .

- **public void viajaReverva (List<Viagens> viagens_viajadas):**Apaga das reservas as viagens já feitas.
- **public void viagensExecutadas():**Lista as viagens que já ocorreram.

Comentários

Atributos:

Viagens viagem, Cliente cliente, String Comentário, Int pontuação;

Construtor

public Comentarios (Viagens viagem,Cliente cliente,String Comentario,int pontuacao)

Métodos

- **public Viagens getViagem()**
- **public int getPontuação()**

Execução do programa

No início do programa o programa vai ler o ficheiro de objetos que guarda a informação ao longo da execução do programa, se esse mesmo ficheiro estiver vazio vai ser gerado um administrador (username:abc abc pass:abc).

Para o administrador: Para se criar uma viagem primeiro temos de criar um autocarro, ter em atenção que não podemos criar duas viagens no mesmo dia só com um autocarro, ao criar o segundo o programa entra num ciclo infinito e tem de ser fechado à força.

Para o Cliente: Só se pode comentar e dar uma pontuação no fim de o cliente a ter viajado

Tanto para o cliente como para o utilizador a opção sair não significa sair do programa, ao escolher essa opção vai ser gerada uma mensagem a perguntar se quero mesmo sair do programa se clicar 1-sai totalmente se clicar 0- faz so o logout.