

UPskill – Java+.NET

Laboratório de Programação

A empresa TravelPlus (TP) dedica-se ao planeamento e reserva de viagens com o compromisso de oferecer aos seus clientes os melhores produtos ao melhor preço, escolhendo parceiros de negócio de excelência.

Atualmente, a TP é parceira de um conjunto vasto de hotéis, mas não exclui novas entradas. A TP disponibiliza vários tipos de reservas aos seus clientes, mais especificamente, reservas de:

- Só voo;
- Voo de ida e voo de regresso (sem hotel);
- Só hotel;
- Hotel e voo(s) (de ida e/ou regresso).

A empresa pretende fazer a gestão das suas reservas, nomeadamente, calcular o custo das reservas. Para auxiliar a informatização deste processo, pretende-se implementar em JAVA, um projeto com as classes necessárias para representar as reservas possíveis de serem feitas na empresa.

A TP deve conhecer toda a informação necessária ao seu negócio, a qual é descrita a seguir.

Um hotel caracteriza-se por um código de reserva gerado automaticamente (número inteiro), um nome, uma categoria (i.e., número de estrelas), uma localidade, *transfer* (sim/não) e o preço por quarto.

Um aeroporto deve ter associado um código, um endereço e uma página web.

Um voo deve possuir um código, o nome da companhia aérea, a quantidade de lugares, a quantidade de lugares disponíveis, o aeroporto de saída, o aeroporto de chegada, o número de km entre os aeroportos, o custo do bilhete e a data e hora de partida.

Cada cliente é caracterizado por um código gerado automaticamente (número inteiro), um nome, uma data de nascimento, um género, um número de identificação fiscal, um número de passaporte, um email, número de reservas realizadas que se concretizaram e uma percentagem de desconto atribuída a uma nova reserva realizada pelo cliente quando o número de reservas confirmadas por um número múltiplo de 5.

Qualquer tipo de reserva tem de ter um identificador gerado automaticamente, a data da reserva, o número de pessoas que vão viajar, o cliente que fez a reserva e a informação se a reserva já se concretizou ou não.

Uma reserva só de um voo precisa de conhecer, para além dos dados referidos no parágrafo anterior, o próprio voo. No caso de uma reserva ter também voo de regresso, necessita de adicionar o voo de regresso. Qualquer voo tem de existir na TP.

Sempre que se efetue uma reserva de um voo, deve ser atualizado o número de lugares disponíveis nesse voo.

As reservas só de hotel necessitam de guardar, para além dos dados comuns a todas as reservas, um hotel (entre aqueles que a TP é parceira), uma data de chegada ao hotel e o número de noites da estadia. Se para além do hotel, houver lugar a voo (ida e/ou regresso) é necessário também guardar a(s) reserva(s) do(s) voo(s).

Os identificadores dos diferentes tipos de reserva gerados automaticamente, devem ser constituídos por um prefixo que identifique inequivocamente o tipo de reserva e um número sequencial. O identificador deverá ser do tipo PREFIXO-X.

Para efeitos de cálculo do custo de uma reserva, deverá tomar em consideração que:

- A taxa de reserva (valor de 20 euros) é aplicada uma única vez independentemente do tipo de reserva;
- Em todos os hotéis, cada quarto tem capacidade máxima para 2 pessoas sendo o preço por noite igual para uma ou duas pessoas. Por exemplo, se a reserva de hotel contemplar 3 pessoas, irá ser contabilizado o preço de dois quartos. No que se refere aos voos, o custo da reserva de um voo deverá ser igual ao preço do bilhete multiplicado pelo número de pessoas associadas à reserva.
- Qualquer reserva que inclua hotel, terá um desconto de 30% nos dias que se encontrarem dentro de um dos seguintes períodos: 15 de janeiro e 15 de março, entre 15 de maio e 15 de junho, entre 15 de setembro e 30 de novembro.
- Sempre que for realizada uma reserva, será necessário verificar se o número de reservas confirmadas por esse cliente corresponde a um número múltiplo de 5. Sempre que essa situação se verificar, o custo total da reserva terá um desconto acordado com o cliente.
- Numa reserva que inclua um voo de regresso (com ou sem reserva de hotel), o custo total da reserva terá um desconto de 1% por cada 1000 km do voo, se a reserva incluir um voo de ida.
- O custo dos voos efetuados entre 15 de janeiro e 15 de março, entre 15 de maio e 15 de junho, entre 15 de setembro e 30 de novembro têm um desconto de 30%.

Assuma que os valores que aparecem sublinhados no texto poderão ser atualizados.

As classes criadas (com exceção da classe principal) devem obedecer ao seguinte conjunto de especificações:

- implementação de construtores (pelo menos o construtor completo e o construtor sem parâmetros);
- implementação de métodos que sejam relevantes para aceder e modificar o valor dos atributos;
- reescrita do método toString;
- reescrita do método equals.

Adicionalmente devem ser implementadas, nas respetivas classes, funcionalidades para o cálculo do custo de uma reserva.

Deverá existir também uma classe instanciável de nome Empresa que tem como atributos, o nome da empresa, a morada, um contentor de objetos do tipo ArrayList para armazenar os clientes da empresa, outro para armazenar os hotéis, outro para armazenar os aeroportos, outro para os voos e um para as reservas efetuadas.

Esta classe deverá permitir disponibilizar as seguintes funcionalidades:

- Adicionar informação base à empresa, nomeadamente, clientes, hotéis, aeroportos e voos;
- Adicionar uma reserva (sugestão: um método por cada tipo de reserva);
- Listar clientes, hotéis, aeroportos, voos e reservas;
- Listar reservas de um cliente a partir do código do cliente;
- Listar hotéis com *transfer*;
- Eliminar uma reserva a partir do código da reserva;
- Pesquisar e retornar um cliente, um hotel, um aeroporto, um voo ou uma reserva a partir do respetivo código;
- Atualizar o número de reservas realizadas por um cliente que se concretizaram (quando determinada reserva se concretizou);
- Retornar o custo de todas as reservas;
- Retornar uma lista das reservas efetuadas, ordenada por ordem crescente de data, usando a interface nativa do Java Comparable;
- Retornar uma lista de reservas, por ordem decrescente do seu custo;
- Retornar uma lista de clientes, por ordem decrescente da sua idade e caso existam clientes com a mesma idade, os clientes deverão ser apresentados por ordem decrescente do número de reservas realizadas que se concretizaram. Caso existam ainda clientes com o mesmo número de reservas realizadas que se concretizaram, os clientes deverão ser apresentados, por ordem alfabética dos seus nomes. Para tal, use a interface nativa do Java Comparator.

Deve ser apresentado um diagrama UML para representar o descrito em cima.

Pretende-se que sejam tidas em conta regras de validação de dados com recurso a exceções, nomeadamente:

- Não devem ser admitidos nomes de clientes que tenham outros carateres para além de letras e espaços;
- O número de identificação fiscal formado por 9 carateres numéricos;
- O preço do quarto de um hotel não pode ser negativo;
- As categorias dos hotéis só poderão ser: uma estrela, duas estrelas, três estrelas, quatro estrelas ou cinco estrelas;

Na classe principal, o código a implementar, de forma a testar as funcionalidades implementadas, deve preencher os seguintes requisitos:

- Criar uma instância da classe Empresa;
- Carregar, de forma automática, informação da empresa: 4 aeroportos, 6 voos, 4 hotéis, 3 clientes e 2 reservas de cada tipo;
- Carregar uma reserva de cada tipo, com informação introduzida pelo utilizador (a validação dos dados introduzidos pelo utilizador deverá ser feita com recurso ao uso de exceções);
- Listar hotéis de uma determinada categoria introduzida pelo utilizador;
- Calcular e apresentar o custo total das reservas de hotel e voos de ida e regresso;
- Executar as funcionalidades implementadas na classe Empresa e visualizar o seu resultado.
- Deverá ser possível gravar toda a informação da empresa num ficheiro binário, bem como carregar para a memória do computador, a partir de um ficheiro binário, toda a informação da empresa nele gravado;
- Antes da aplicação terminar, deverá ser perguntado ao utilizador se deseja que a informação da empresa seja gravada num ficheiro binário. No caso afirmativo, deverá ser requerido o nome do ficheiro e, em seguida, guardada a informação no disco.

Sempre que for apropriado, deverão ser usadas Interfaces Java.

Sempre que se justificar, deverão ser usados Tipos Enumerados para limitar o valor que algumas variáveis podem assumir.

Na implementação de algumas classes deverá ser utilizada a relação de associação apropriada com outras classes: Agregação e/ou Composição.

Crie testes unitários para testar os métodos referentes ao cálculo do custo de uma reserva.

Todo o código produzido deve ter sempre em consideração os principais princípios da programação orientada por objetos: abstração, encapsulamento, herança e polimorfismo.

O código deverá conter os comentários necessários para que possa ser gerada a documentação usando a ferramenta Javadoc.

A implementação do software deve adotar a norma de codificação CamelCase.

O trabalho deverá ser realizado por um grupo de dois formandos.

Deverá ser submetido no Moodle do UPskill, um ficheiro ZIP com: o projeto Maven com o seguinte formato: PrimeiroUltimoNome_PrimeiroUltimoNome, como por exemplo, IsabelBras_ArturSilva; e um ficheiro com o diagrama UML.

O trabalho final deverá ser submetido no Moodle até às 09:00 do dia 18 de novembro (segunda-feira).

Diariamente, ao final do dia de trabalho, um dos elementos de cada grupo de alunos, deverá submeter no Moodle um ficheiro ZIP com as mesmas regras acima mencionadas, com o estado do projeto desenvolvido até ao momento.