

 <p>UNIVERSIDADE DE COIMBRA FACULDADE DE CIÊNCIAS E TECNOLOGIA Departamento de Engenharia Informática</p>	<p>Programação Orientada aos Objetos</p> <p>Projeto 2015/16 Agência de Viagens</p> <p>Data de Entrega: 13 de Dezembro</p>
---	--

Descrição do Problema

Uma agência de viagens de autocarro está a planear desenvolver uma nova aplicação para a gestão de clientes, de viagens, de autocarros e das operações que são usualmente efetuadas (consultas, reservas, pagamentos, etc.).

Os utilizadores podem ser de dois tipos: administradores e clientes. Os utilizadores são caracterizados pelo nome, NIF, morada, telefone e e-mail. Todos os utilizadores devem autenticar-se através de uma palavra-chave quando acedem à aplicação. Os clientes podem ser regulares ou *premium*. Os primeiros têm uma penalização de 50% quando cancelam uma reserva até uma semana da data de partida. Os clientes *premium* podem cancelar as reservas sem qualquer custo até dois dias antes da partida. Após estes prazos não são feitos reembolsos. Os clientes *premium* têm também um desconto de 10% no preço base das viagens.

Os utilizadores têm privilégios diferentes no acesso à aplicação. Os clientes podem consultar viagens disponíveis, reservar viagens, consultar/cancelar reservas e inserir/ler comentários+pontuação(1-5) sobre as viagens realizadas. Os administradores podem gerir clientes, viagens e autocarros (criar, eliminar, alterar e listar).

Uma viagem é caracterizada por um código, origem, destino, preço, autocarros utilizados, data e duração.

Uma reserva, para além do cliente e da viagem, tem também informação sobre o autocarro, o nº do lugar e o valor do pagamento. No caso de uma reserva que foi cancelada, o valor do pagamento será o valor que não foi devolvido ao cliente no ato do cancelamento (da reserva).

Os autocarros são descritos pela matrícula e pela capacidade de passageiros (lotação).

Ainda que todos os lugares de uma viagem se encontrem reservados, os clientes podem continuar a fazer uma reserva ficando em lista de espera. Nestes casos, não são aceites os respetivos pagamentos.

Quando uma reserva é cancelada, os clientes em lista de espera recebem uma notificação (mensagem no ecrã). A partir desse momento, qualquer cliente em espera poderá proceder ao pagamento da sua reserva. O primeiro cliente que o fizer, ocupa o lugar vago.

O administrador poderá ainda efetuar as seguintes operações:

1. Identificar a viagem mais vendida num determinado mês;
2. Identificar o cliente que mais viagens comprou num determinado mês;
3. Listar todas as viagens que não tiveram reservas num determinado mês;
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;

8. Obter os dados estatísticos relativamente à venda de viagens durante um determinado ano: volume de vendas em cada mês e o dia do ano em que houve mais vendas.

A interação com o utilizador deverá ser realizada através de uma consola em modo de texto para permitir a entrada de dados e a apresentação de resultados. Os resultados deverão, também, poder ser armazenados em ficheiros de texto.

Implementação

A aplicação deve ser implementada na linguagem Java e deverá ter em conta os seguintes aspetos:

1. Ao entrar na aplicação, devem persistir todos os dados que se encontravam no fim da última utilização.
2. Cada classe deve gerir internamente os seus dados, pelo que deverá cuidar da proteção das suas variáveis e métodos;
3. Cada objeto deverá ser responsável por uma tarefa ou objetivo específico, não lhe devendo ser atribuídas funções indevidas;
4. Utilize a keyword **static** apenas quando tal se justifique e não para contornar erros do compilador.

Elabore um diagrama com as suas classes e objetos (em UML), antes de iniciar a implementação, para prever a estrutura do projeto.

Tenha ainda em conta os seguintes pontos que serão importantes na avaliação:

1. Comentar as classes, métodos e variáveis públicas segundo o formato Javadoc. Isto permitir-lhe-á gerar automaticamente uma estrutura de ficheiros HTML, descritivos do seu código, que deve incluir no seu relatório;
2. Comentar o restante código sempre que a leitura dos algoritmos não seja óbvia;
3. Tal como sugerido acima, evitar o uso abusivo de **static** e de variáveis e métodos **public**;
4. Na escolha de nomes para variáveis, classes e métodos, seguir as convenções adotadas na linguagem **Java**;
5. Na organização das classes deverá ser evitada a redundância dos dados.

Prazos de entrega

A entrega do trabalho compreende duas metas distintas:

Meta 1 (1 valor) – Análise do projeto e diagrama de classes (entrega até 9 de Novembro);

Meta 2 (4 valores) – Versão final (entrega até 13 de Dezembro).

Os trabalhos serão comparados (tanto entre os trabalhos da disciplina como com código disponível na Internet), no sentido de detetar eventuais fraudes por cópia. Nos casos em que se verifique que houve cópia de trabalho total ou parcial, os grupos envolvidos terão os projetos anulados, reprovando à disciplina.

Material a entregar

Cada grupo deve entregar obrigatoriamente:

Meta 1: Diagrama de classes em UML

1. Upload em pdf do diagrama de classes no InforEstudante;
2. Entrega de uma cópia impressa do diagrama no momento da defesa na aula teórico-prática.

Meta 2: Aplicação

1. Upload de zipFile no InforEstudante com:

- todas as classes .java;
- executável;
- ficheiros de dados para teste;
- JavaDoc;
- Relatório (em formato pdf).

2. Relatório em papel, entregue no cacifo do docente da turma teórico-prática a que os alunos pertencem, descrevendo o programa do ponto de vista técnico e que deve incluir:

- Estrutura geral do programa;
- Diagramas de classes inicial e final;
- Descrição das principais estruturas de dados e de ficheiros usados;
- Breve explicação de como o programa se executa.

Avaliação do trabalho

Para a avaliação do trabalho contam fatores de dois tipos:

- Caixa preta (tal como é percecionado pelo utilizador):
 - o Conjunto de funcionalidades implementadas;
 - o Robustez do programa;
 - o Qualidade da interface.
- Caixa branca (a forma como está construído):
 - o Qualidade das soluções técnicas encontradas para os problemas em causa;
 - o Estruturação do código;
 - o Qualidade dos comentários.

A avaliação em cada uma das metas é feita individualmente, independentemente dos grupos serem constituídos por 2 alunos. Aos 5 valores atribuídos ao projeto estão associados mínimos de 47.5%. Os alunos que tenham um resultado de avaliação inferior a 47.5% não serão admitidos ao exame da disciplina.

Nota: Não se aceitam trabalhos que apresentem erros de compilação no momento da defesa e que não estejam corretamente estruturados do ponto de vista da Programação Orientada aos Objetos.

Composição dos grupos

O trabalho deve ser realizado em grupos de 2 elementos, tendo os elementos do grupo que pertencer à mesma turma teórico-prática. A defesa dos trabalhos é feita individualmente, bem como a respetiva avaliação.

Defesa final do trabalho

O trabalho deve ser defendido através de uma discussão presencial e individual. Para isso, cada grupo deve inscrever-se num horário que esteja disponível para essa defesa no InforEstudante.