

TRABALHO 1

PROGRAMAÇÃO ORIENTADA A OBJETOS

Descrição do sistema a ser desenvolvido.

O sistema automatiza reservas em uma companhia aérea. Os usuários do sistema são funcionários ou clientes. O sistema impõe as regras de negócio: cada cliente pode fazer até duas reservas; cada reserva pode relacionar até cinco vôos; um cliente só pode incluir um vôo na reserva se houver vaga. O sistema possibilita ao cliente: cadastrar nome, telefone e CPF; consultar e editar seus dados pessoais; efetuar e cancelar reservas; obter os códigos das suas reservas; a partir do código de uma reserva, obter os números dos vôos, as datas de partida e chegada, os horários de partida e chegada, os códigos dos aeroportos de partida e chegada; listar os vôos disponíveis a partir de uma cidade de origem para uma de destino em uma data (a resposta deve informar, para cada vôo, número do vôo, datas de partida e chegada, horários de partida e chegada, códigos dos aeroportos de partida e chegada). Os funcionários podem: cadastrar nome, telefone e matrícula; incluir, remover e editar os dados dos vôos.

Domínios a serem observados.

• Nome	20 letras
• Telefone	10 dígitos
• CPF	11 dígitos
• Número de vôo	03 letras seguidas por 03 dígitos
• Matrícula	05 dígitos
• Código de aeroporto	03 letras
• Horário	dígitos no formato HH:MM:SS
• Data	dígitos no formato DD:MM:AA

Requisitos do projeto.

- O sistema deve ser organizado em camadas de apresentação, negócio e persistência.
- A camada de apresentação é responsável pela interface com o usuário e pela validação de domínios.
- A camada de negócio deve impor as restrições de negócio.
- A camada de persistência deve ser responsável pela conexão ao banco de dados e execução de comandos SQL.
- O banco de dados deve ser o *HSQldb* e o acesso ao banco de dados deve ser via *API JDBC*.
- O sistema deve ser composto por um programa decomposto em módulos.
- Os módulos devem ser decompostos em classes.
- Cada módulo da interface com o usuário deve ter uma classe controladora de interação.
- Cada módulo da lógica de negócio deve ter uma classe controladora de negócio.
- As telas devem ser representadas por classes.
- As entidades devem ser representadas por classes.
- A validação dos campos de entrada deve ser viabilizada por classes que modelem os tipos básicos.

Requisitos do trabalho.

- Deve ser usada a linguagem *Java*.
- Devem ser usadas as ferramentas: *NetBeans*, *Javadoc*, *JUnit*, *Cobertura* e *Checkstyle*.
- Não pode ser gerado código automaticamente por ferramenta.
- Os documentos devem ser preenchidos com clareza e cuidados devem ser tomados com a ortografia.
- Devem ser seguidas as convenções de codificação da Sun: <http://java.sun.com/docs/codeconv/>.
- Forneça os artefatos produzidos em um CD adequadamente organizado (diretórios).
- Na raiz do CD deve haver um arquivo *LEIAME.PDF* onde é explicado o objetivo de cada diretório.
- Textos e diagramas devem ser fornecidos no formato PDF.

Questão 1 – 2,5 pontos

- Codifique as classes que representam os domínios.
- Documente as classes no formato *Javadoc* e produza relatório via *Checkstyle*.
- Em cada classe, codifique um método público e estático para validação do domínio.
- Em caso de falha na validação, deve ser lançada uma exceção.
- Use uma classe existente para a exceção, ou codifique uma classe apropriada.

Questão 2 – 2,5 pontos

- Identifique as classes que sejam entidades no sistema.
- Crie um diagrama com os relacionamentos entre as classes que sejam entidades no sistema.
- Documente as responsabilidades de cada classe com um cartão CRC.
- Identifique e documente os contratos entre as classes (*get*, *set*, etc).

Questão 3 – 2,5 pontos

- Codifique as classes que representam entidades.
- Documente as classes no formato *Javadoc* e produza relatório via *Checkstyle*.
- As validações devem ser realizadas vias as classes que validam os domínios.
- Em caso de falha na validação, deve ser lançada uma exceção.
- Não deve ser possível inicializar uma entidade com valores incorretos.
- A tentativa de inicialização com valores incorretos deve resultar no lançamento de exceção.

Questão 4 – 2,5 pontos

- Codifique um teste de unidade para cada entidade usando *JUnit* e *Cobertura*.
- Documente cada classe no formato *Javadoc* e produza relatório via *Checkstyle*.
- Cada teste de unidade deve atingir, no mínimo, a cobertura de 90% das linhas do código sendo testado.
- Forneça documento descrevendo como usar os testes codificados.
- Forneça cópias dos relatórios de cobertura produzidos pela ferramenta *Cobertura*.