 <p>INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA MINAS GERAIS</p>	<p align="center">Ciência da Computação</p> <p align="center">Trabalho Prático 01</p> <p align="center">Criação de um protótipo de controle de voos.</p>	
Disciplina: Programação Orientada a Objetos	Data: ____/____/____	
Professor: Msc. Bruno Ferreira	Turma: Noturno	
Aluno	Valor: 15,0	Nota:

Objetivo: Reforçar conceitos de programação orientada a objetos (classes, objetos, pacotes, encapsulamento, variáveis), introdução as interfaces gráficas (JOptionPane).

Forma de Entrega: O código fonte e um diagrama de classe representando a implementação deverão ser enviados via email para bruno.ferreira@ifmg.edu.br em um arquivo zipado. A data limite para entrada é até o dia **18/04/2017**. A apresentação do trabalho será realizada em sala para o professor no dia **19/04/2017**.

OBS: Será observada e avaliada a legibilidade do código sendo fundamental uma boa indentação e a utilização de comentários.

O trabalho é em dupla, se for verificada cópia de trabalho os envolvidos receberão nota zero.

O consorcio “Vá-Com-Deus” que administra o aeroporto de Formiga - MG precisa de automatizar as informações dos voos que decolam a partir dele. O aeroporto é responsável por exibir todos os voos do dia em um painel e por exibir os dados dos voos do usuário via celular (iremos simular essas duas saídas utilizando JOptionPane). Para essas informações é necessário que cada voo seja mantido (incluído, alterado, excluído) com os seguintes dados: número do voo (inteiro), companhia aérea, avião, data, horário de voo, status do voo (confirmado, cancelado e atrasado) destino e origem. Esses dados são enviados pelas operadoras aéreas via arquivo texto seguindo um padrão de dados em classe chamado Json. Mas os dados podem também ser manipulados diretamente pelo programa sem o arquivo texto. Inclusive o status dos voos podem ser alterados se algum imprevisto acontecer antes da decolagem.

O aeroporto tem a capacidade de gerenciar até 100 voos por mês e todo final do mês os funcionários reiniciam os dados do programa, assim, novas informações podem ser cadastradas/importadas. Atenção o aeroporto só consegue atender a quatro aeronaves simultâneas a cada trinta minutos, então trate essa inconsistência ao inserir/importar voos.

Entre os dados que são recebidos dos voos têm-se os dados completos do avião: a autonomia de voo em quilômetros, a altura, o tamanho da envergadura da asa e seu comprimento - todos esses dados em metros, a capacidade de carga (toneladas) e, os dados de todos os passageiros que irão embarcar nesse voo (nome completo, telefone, e-mail (válido), CPF (válido) e data de nascimento).

A sua implementação deve incluir um modulo que é usado pela operadora aérea, o qual cadastra os voos e exporta os dados em um arquivo texto (Json). Um outro módulo de gerencia do aeroporto que mantém

esses dados dos voos e outros dois módulos responsáveis respectivamente por exibir os dados do dia no painel e por exibir os dados do voo no celular do cliente.

Veja algumas **operações gerenciais** que o aeroporto precisa ter e que devem ser atendidas pelo software:

- O aeroporto precisa saber quantos voos ela fez por mês de acordo com cada companhia aérea.
- O aeroporto precisa saber seu lucro, ela ganha 10 mil reais por decolagem e ganha um acréscimo de R\$ 1000,00 a cada tonelada excedente. O limite é de 30 toneladas.
- O aeroporto deve ser avisado se existe mais de quatro decolagens em um período de 30 minutos, assim o aeroporto pode avisar as companhias e pedir que elas remarquem os voos.
- Listar os passageiros de um voo de acordo com o número do voo em caso de acidente aéreo.

Dica: Expressões regulares podem validar um e-mail de forma completa.

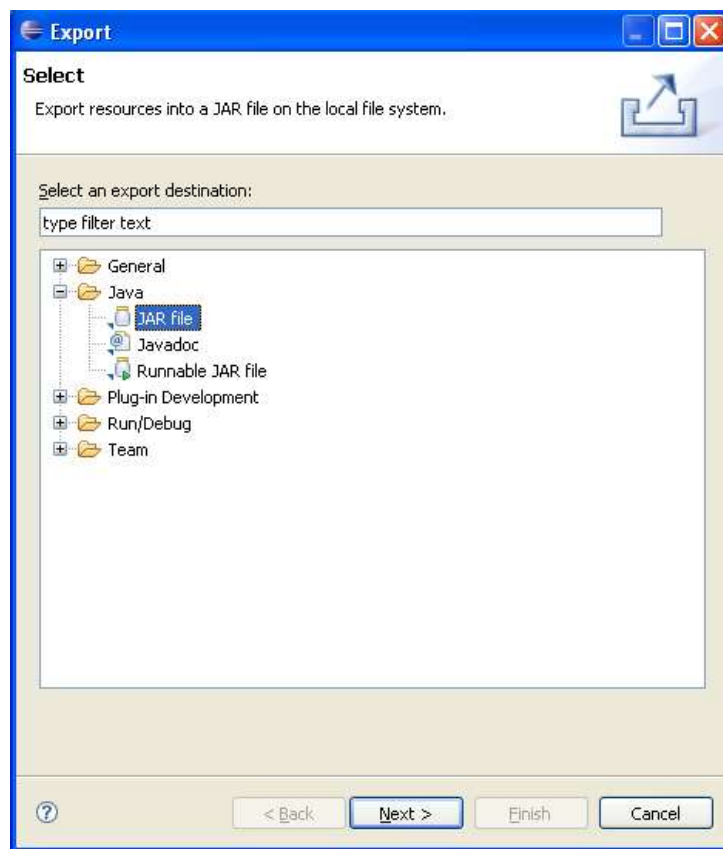
Se quiser total siga toda a especificação e faça algo diferente no seu sistema.

Trate os possíveis erros de entrada de dados pelo usuário ao utilizar a interface gráfica

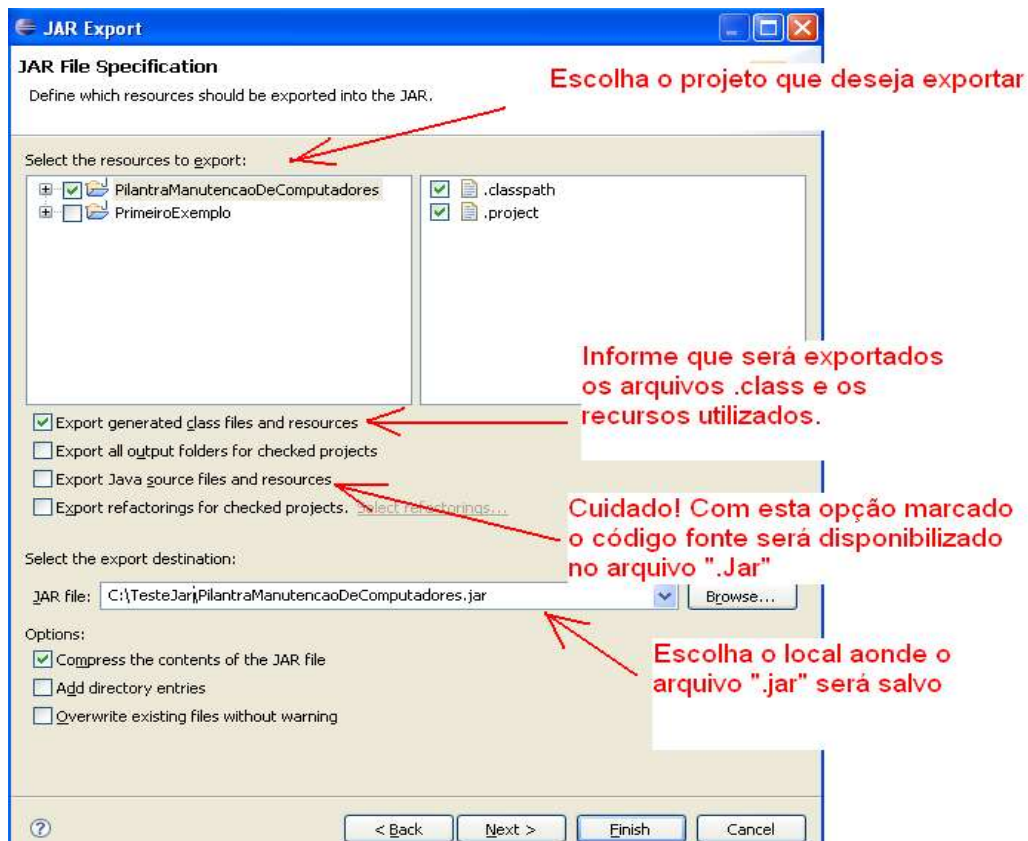
Último passo:

Gerar o executável: **(Atenção: lembre-se que o nome do seu projeto é: Vá-Com-Deus)**

- A primeira etapa é exportar o projeto. Para isso, clique com o botão direito do mouse sobre o projeto no "Package Explorer" e escolha a opção "Export...".
- Com a janela "Export" aberta, escolha a opção "Jar Files" e clique no botão "Next":



Na próxima tela será informado qual o projeto será exportado e qual o diretório que ele será salvo:



Na tela seguinte, deixe as configurações padrões e clique em “Next”:

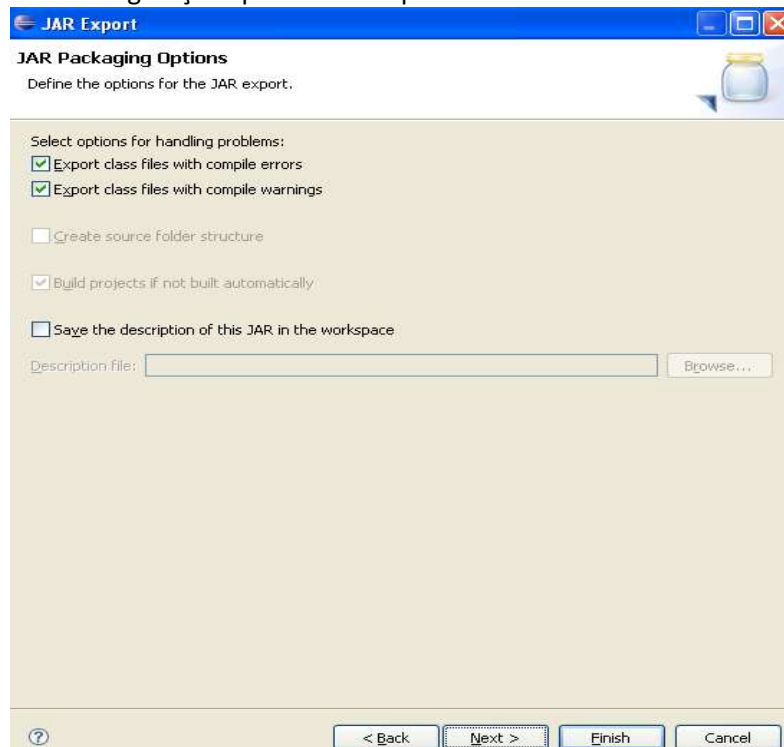


Figura 17

Para finalizar, informe qual a classe principal de sua aplicação e clique em “Finish”:

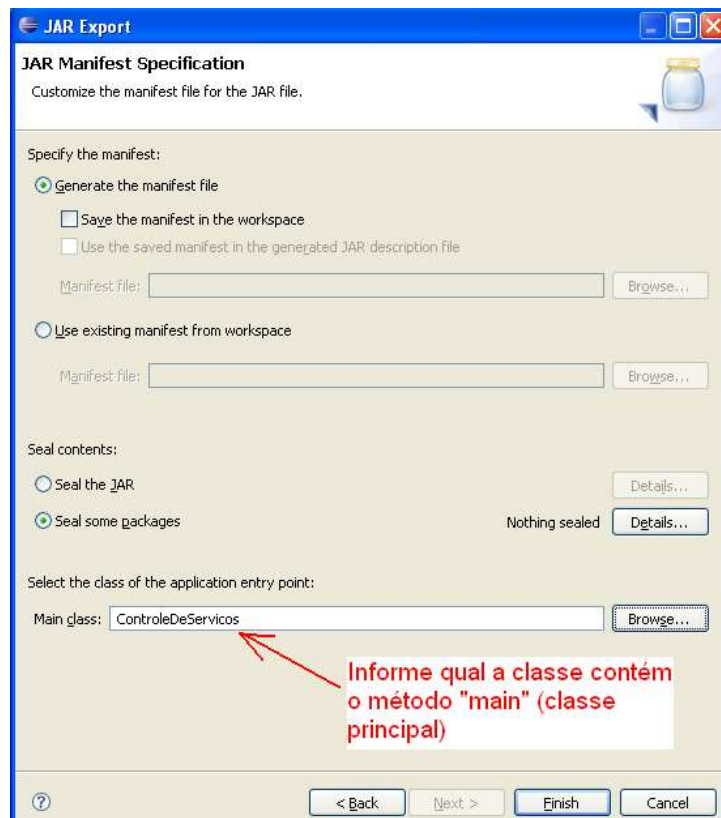


Figura 18

Pronto, o arquivo “Jar” está criado. Agora precisamos executá-lo. Para isso, devemos usar o comando: `java -jar ./NomeDoArquivoJar.jar`. Mas para facilitar vamos criar um arquivo “.bat”. Crie este “.bat” no mesmo diretório que o arquivo “.Jar” foi salvo.

Veja na Figura abaixo o que este arquivo deve conter.

```
@echo off

if exist "./PilantraManutencaoDeComputadores.jar" goto programaEncontrado
echo Nao foi possivel encontrar o programa especificado.
pause
goto fim

:programaEncontrado
java -jar ./PilantraManutencaoDeComputadores.jar
:fim
```

OBS: Pra criar um arquivo “.bat” basta abrir o Bloco de Notas e salvar o arquivo com a extensão “bat”.

Por fim, dê dois cliques no arquivo “bat” para executá-lo e verifique se sua aplicação está rodando normalmente.

Pontuação

Tópico	Valor
Diagrama de Classe	1,0
Manipulação do arquivo Json	3,0
Validação do CPF e do E-mail (usando expressões regulares)	3,0
Implementou as operações gerencias	3,0
Criou o .bat	1,0
Implementação geral dos requisitos entre as classes	3,0
Interface com usuário agradável e com tratamento de erros de entrada	3,0
Criou uma boa estrutura de classes	2,0
Sou bom aluno e implementei algo a mais para melhorar o sistema	1,0

Fim