



UNIVERSIDADE FEDERAL DE PELOTAS
CENTRO DE DESENVOLVIMENTO TECNOLÓGICO
CURSOS DE CIÊNCIA E ENGENHARIA DE COMPUTAÇÃO
Disciplina de Programação Orientada a Objetos – Prova 2 - 2017/2
Prof.: Felipe de Souza Marques (felipem@inf.ufpel.edu.br)

Nome: _____
Usuário: / Senha: _____

Instruções para a prova:

- Utilize o nome “**nome_sobrenome.zip**” no arquivo a ser entregue.
- Coloque seu nome completo e turma em forma de comentário no arquivo fonte.
- **A prova deve ser entregue** através do **sistema BOCA**. O professor irá informar o endereço de acesso ao servidor. O arquivo compactado do projeto deve ser **submetido** ao **BOCA**, utilizando o modo **Runs**. Basta clicar na aba **Runs**, escolher a questão a qual o programa se refere, escolher o arquivo a ser enviado e depois clicar em **send**. **Utilize este sistema para guardar versões parciais do seu programa para evitar perdas de dados por eventuais quedas de energia**. Havendo dúvida para o envio, chame o professor!

1. (4 pontos) Escreva uma classe *Aviao* em que cada objeto representa um avião que viaja em determinada data. Cada avião possui no máximo 300 passageiros e a classe permite controlar a ocupação das vagas. Considere o uso do modificador **protected** na coleção que irá modelar os assentos do avião. Isso poder ser útil em função dos requisitos da questão seguinte. A classe deve ter os seguintes métodos:

Construtor	Configura os dados do avião (recebidos como parâmetro): número do avião (<i>int</i>) e data de partida (a data modelada pela classe <i>Data</i>).
proximoLivre	Retorna o número da próxima poltrona livre (buscas devem iniciar pelo início da lista de assentos).
verifica	Verifica se o número do assento recebido como parâmetro está ocupado. Retorna <i>true</i> se está ocupado e <i>false</i> caso contrário.
ocupa	Ocupa determinado assento do avião, cujo número é recebido como parâmetro, e retorna <i>true</i> se o assento ainda não estiver ocupado (operação foi bem sucedida) e <i>false</i> caso contrário.
vagas	Retorna o número de assentos vagos disponíveis (não ocupados) no avião.
getNumero	Retorna o número do avião.
getData	Retorna a data da viagem.

2. (3,5 pontos) Escreva uma classe *AviaoPremium* herdeira à *Aviao* criada na questão anterior, que permita definir quantos assentos existem no avião e dividi-lo em ala de poltronas premium e poltronas comuns. Para isto, esta classe deve acrescentar os atributos necessários e adicionar os seguintes métodos:

Construtor	Além dos parâmetros recebidos pelo construtor da superclasse, também receberá como parâmetro o número de assentos do avião e quantas poltronas são destinadas a área premium.
maxVagas	retorna o número máximo de poltronas do avião.
cadeirasPremium	retorna quantas poltronas estão destinadas à ala premium (as demais serão automaticamente destinadas aos assentos comuns); as poltronas premium serão sempre as primeiras poltronas do avião.

tipo	Recebe como parâmetro o número da poltrona e retorna 'P' se for uma poltrona premium e 'C' se for uma poltrona comum.
proximoLivre	Usando o conceito de polimorfismo, sobrescreva o método <i>proximoLivre</i> de forma que ele retorna próxima cadeira comum livre.
proximoPremiumLivre	Retorna o número da próxima poltrona premium livre.

3. (2,5 pontos) Dada as seguintes interfaces:

Pessoa – representa genericamente uma pessoa	
getCPF	Retorna o CPF da pessoa.
getNome	Retorna o nome da pessoa.
vip	Retorna <i>true</i> se a pessoa é VIP (viaja em poltronas premium). Caso contrário, retorna <i>false</i> .

Repositorio – representa genericamente um repositório	
guarda	Armazena uma instância de objeto Pessoa no repositório.
recupera	Recupera um objeto Pessoa a partir do objeto informado. Se o CPF não for encontrado, retorna null .
primeiro	Posiciona o apontador para o início do repositório e retorna a primeira pessoa do repositório.
proximo	Desloca o apontador para a próxima pessoa e a retorna. Caso não haja uma próxima pessoa, retorna null .

Escreva uma classe *Embarque* que contenha um método estático *processaEmbarque*, que receba como parâmetro um objeto da classe *AviaoPremium* e um objeto de uma classe que implementa a interface *Repositorio*. O método deve realizar o embarque de cada uma das pessoas do repositório, de forma que cada uma ocupe um acento no trem, considerando se a pessoa é VIP (indicando que deve ocupar uma cadeira premium) ou não (ocupa cadeira comum).

May the force be with you!