



Lista de Exercícios II - Avaliativa

Utilizando os conceitos de classes e objetos, encapsulamento, métodos modificadores de acesso, construtores, herança, polimorfismo e exceções, resolva as questões a seguir:

Questão 1. Crie uma classe Elevador para armazenar as informações de um elevador de prédio. A classe deve armazenar o andar atual (térreo = 0), total de andares no prédio (desconsiderando o térreo), capacidade do elevador e quantas pessoas estão presentes nele. A classe deve também disponibilizar os seguintes métodos:

- **Inicializar:** que deve receber como parâmetros a capacidade do elevador e o total de andares no prédio (os elevadores sempre começam no térreo e vazio);
- **Entrar:** para acrescentar uma pessoa no elevador (só deve acrescentar se ainda houver espaço);
- **Sair:** para remover uma pessoa do elevador (só deve remover se houver alguém dentro dele);
- **Subir:** para subir um andar (não deve subir se já estiver no último andar);
- **Descer:** para descer um andar (não deve descer se já estiver no térreo);
- **Obs.:** Encapsular todos os atributos da classe (criar os métodos set e get).

Questão 2. Sistema de academia: Trata-se de um sistema que visa dar suporte ao funcionamento de academia de ginástica. O sistema é utilizado pela administração da academia, bem como pelos alunos (clientes) da mesma. Cada aluno é identificado através de um cartão magnético (CPF). O sistema inicia a atuar na porta da academia, onde o aluno passa o seu cartão e o sistema libera a sua entrada. Ao chegar à sala de musculação o aluno se dirige a um outro terminal e passa novamente o cartão, nesse caso o sistema exibe a lista de exercícios que ele deverá realizar, e quem será o professor a lhe acompanhar.

Questão 3. Clínica Veterinária: Os clientes primeiramente marcam consultas com a secretária, fornecendo suas informações pessoais e as dos animais que desejam tratar. A secretária deverá criar ou atualizar o cadastro. Em cada sessão de tratamento (uma sessão equivale a uma consulta), o cliente deve informar os sintomas aparentes do animal, os quais devem ser registrados, dependendo do diagnóstico do médico-veterinário, o veterinário pode marcar exames para o animal.

Questão 4. Crie um programa que receba n números e some esses numeros enquanto a soma não for superior a 100. O programa deverá imprimir o valor somado (antes de atingir o número maior que 100) e deverá informar quantos números foram somados e qual a média. Refaça seu programa utilizando as seguintes regras:

- Utilize os tratamentos de exceção para lidar com a entrada de dados.
- Quando a soma for superior a 100, o programa deverá gerar uma exceção criada pelo programador (veja transparência da aula pra ver como fazer), com nome `ExcecaoAcimaDeCem`.
- Lance essa exceção, com o uso de throws.



UNIVERSIDADE FEDERAL DO AMAZONAS
INSTITUTO DE CIÊNCIAS EXATAS E TECNOLOGIA
Algoritmos e Estruturas de Dados II
Prof. Alternei Brito



Questão 5. Escreva um programa para preencher valores de um vetor de inteiros com 10 posições. O usuário irá informar os valores a serem inseridos e suas respectivas posições no array. O programa deve tratar as exceções `ArrayIndexOutOfBoundsException` quando for informada uma posição inexistente do vetor e `NumberFormatException` quando o valor informado não for número.

Questão 6. Escreva uma classe `Data` cuja instância (objeto) represente uma data. Esta classe deverá dispor dos seguintes métodos:

construtor	define a data que determinado objeto (através de parâmetro), este método verifica se a data está correta, caso não esteja a data é configurada como 01/01/0001
compara	recebe como parâmetro um outro objeto da Classe <code>data</code> , compare com a data corrente e retorne: <ul style="list-style-type: none">▫ 0 se as datas forem iguais;▫ 1 se a data corrente for maior que a do parâmetro;▫ -1 se a data do parâmetro for maior que a corrente.
getDia	retorna o dia da data
getMes	retorna o mês da data
getMesExtenso	retorna o mês da data corrente por extenso
getAno	retorna o ano da data
isBissexto	retorna verdadeiro se o ano da data corrente for bissexto e falso caso contrário
clone	o objeto clona a si próprio, para isto, ele cria um novo objeto da classe <code>Data</code> com os mesmos valores de atributos e retorna sua referência pelo método

Questão 7. Escreva uma classe em que cada objeto representa um voo que acontece em determinada data e em determinado horário. Cada voo possui no máximo 100 passageiros, e a classe permite controlar a ocupação das vagas. A classe deve ter os seguintes métodos:

construtor	configura os dados do voo (recebidos como parâmetro): número do voo, data (para armazenar a data utilize um objeto da classe <code>Data</code> , criada na questão anterior);
proximoLivre	retorna o número da próxima cadeira livre
verifica	verifica se o número da cadeira recebido como parâmetro está ocupada
ocupa	ocupa determinada cadeira do voo, cujo número é recebido como parâmetro, e retorna verdadeiro se a cadeira ainda não estiver ocupada (operação foi bem sucedida) e falso caso contrário
vagas	retorna o número de cadeiras vagas disponíveis (não ocupadas) no voo
getVoo	retorna o número do voo



UNIVERSIDADE FEDERAL DO AMAZONAS
INSTITUTO DE CIÊNCIAS EXATAS E TECNOLOGIA
Algoritmos e Estruturas de Dados II
Prof. Alternei Brito



Questão 8. Considere a seguinte classe, cujo método `respostaQuestao` recebe como parâmetro o número de uma questão e retorna a sua resposta correta, proveniente de um gabarito.

```
public class Gabarito {  
    public char respostaQuestao(int numeroQuestao) {  
        :  
    }  
}
```

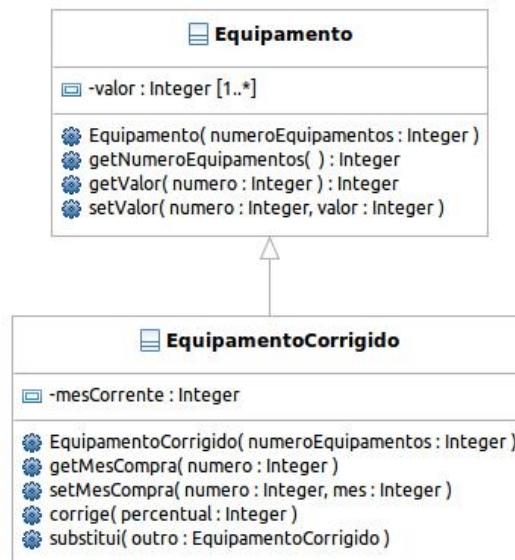
Escreva uma classe `Prova` em que cada objeto representa uma prova feita por um aluno. Esta prova possui 15 questões de múltipla escolha (letras A a E). As 10 primeiras questões valem 0,5 ponto e as 5 últimas questões valem 1 ponto. Esta classe deverá controlar as questões respondidas pelo aluno. Para isto, a classe deve implementar os métodos:

construtor	recebe como parâmetro um objeto da classe <code>Gabarito</code> contendo o gabarito da prova
respostaAluno	recebe como parâmetro a resposta dada pelo aluno a uma questão; este método não recebe entre os parâmetros o número da questão, ele mesmo deve estabelecer um controle interno de modo que as questões sejam inseridas sequencialmente, ou seja, a primeira vez que o método é chamado, insere a primeira questão, a segunda, insere a segunda questão, e assim por diante.
acertos	retorna a quantidade de questões que o aluno acertou
nota	retorna a nota que o aluno tirou na prova
maior	recebe como parâmetro um outro objeto da classe <code>Prova</code> e retorna a nota do aluno que acertou mais questões; se houver empate, retorna a maior nota; se houver empate novamente, retorna a nota do aluno representado no objeto corrente

Questão 9. Escreva uma classe herdeira à voo criada na lista de exercícios anterior, que permita definir quantas cadeiras existem no máximo no voo e se dividir o avião em ala de fumantes e não fumantes. Para isto esta classe deve acrescentar os atributos necessários e adicionar os seguintes métodos:

construtor	além dos parâmetros recebidos pelo construtor da superclasse, receberá também como parâmetros o número de vagas do voo e quantas cadeiras serão destinadas para fumantes
maxVagas	determina o número máximo de cadeiras no voo
cadeirasFumantes	determina quantas cadeiras estão destinadas aos fumantes (as demais serão automaticamente destinadas aos não fumantes); as cadeiras dos fumantes serão sempre as últimas do avião
tipo	recebe como parâmetro o número da cadeira e retorna 'F' se for uma cadeira para fumantes e 'N' se for para não fumantes

Questão 10. Dada uma classe Equipamento na qual cada objeto representa um conjunto de N equipamentos de uma empresa com seus respectivos valores, cujo diagrama UML está representado a seguir:



construtor	recebe como parâmetro o número de equipamentos e cria um vetor de valores do respectivo tamanho
getNumeroEquipamentos	retorna o número de equipamentos
getValor	recebe como parâmetro o número do equipamento (começando de zero) e retorna seu valor
setValor	recebe como parâmetro o número do equipamento e seu valor e o registra

Cada equipamento possui um código numérico sequencial, começando de zero, que corresponde a sua posição no vetor.

Escreva uma classe, herdeira da classe Equipamento, denominada EquipamentoCorrigido em que cada objeto representa os mesmos equipamentos com valor corrigido, conforme diagrama UML parcial representado anteriormente.

Todo equipamento só é corrigido anualmente no mês em que foi comprado, por este motivo a classe deve acrescentar para cada equipamento um registro do seu mês de compra. Além disto, deve possuir os métodos:

construtor	recebe como parâmetros o número de equipamentos e o mês corrente
getMesCompra	recebe como parâmetro o número do equipamento (começando de zero) e retorna seu mês de compra
setMesCompra	recebe como parâmetro o número do equipamento e seu mês de compra e o registra
corrige	este método recebe como parâmetro apenas o percentual de correção e corrige todos os equipamentos cujo mês de compra seja igual ao mês corrente; O objeto deve manter registrado em um atributo o mês corrente, que deve começar sempre em janeiro (quando o objeto é construído). Cada vez que este método é chamado, após a correção, o mês é incrementado de um e, se estiver em dezembro, retorna para janeiro
substitui	recebe como parâmetro um outro objeto da classe EquipamentoCorrigido e substitui o valor e o mês de compra de todos os equipamentos do objeto corrente pelos do objeto recebido como parâmetro; a operação só será realizada se ambos os objetos possuírem o mesmo número de equipamentos

Note que o atributo “valor” da classe Equipamento é privado, portanto, só poderá ser acessado indiretamente, até mesmo pela classe herdeira.



UNIVERSIDADE FEDERAL DO AMAZONAS
INSTITUTO DE CIÊNCIAS EXATAS E TECNOLOGIA
Algoritmos e Estruturas de Dados II
Prof. Alternei Brito



Questão 11. Um órgão de levantamento meteorológico possui equipamentos para medir a pluviosidade (pluviômetros), onde cada unidade é representada em um programa de computador por um objeto da classe:

```
public class Pluviometro {  
    protected String tipo;  
    public Pluviometro(String tipo) {  
        :  
    }  
    public String getTipo() {  
        :  
    }  
    public int getPeso() {  
        :  
    }  
    public int getCapacidade() {  
        :  
    }  
}
```

Construtor	Recebe como parâmetro o tipo de equipamento.
getTipo	Retorna o tipo do pluviômetro.
getPeso	Retorna o peso do pluviômetro em quilos. Este peso é calculado automaticamente pela classe a partir do tipo.
getCapacidade	Retorna a capacidade do pluviômetro em mililitros. Esta capacidade é calculada automaticamente pela classe a partir do tipo.

Os pluviômetros são carregados por caminhões que, no programa de computador, são representados genericamente por objetos da classe Caminhao (esta classe não deve ser implementada nesta questão). A classe define os seguintes métodos:

Construtor	Recebe como parâmetro a quantidade de equipamentos que o caminhão irá carregar.
inserePluviometro	Recebe como parâmetro um objeto da classe <code>Pluviometro</code> e o coloca dentro do caminhão se a capacidade do mesmo permitir.

Cada objeto da classe representa um caminhão. Esta classe não possui nenhum atributo e seus métodos não possuem implementação, pois serão implementados nas subclasses.

Escreva duas classes herdeiras da classe Caminhao que representam dois tipos deste veículo com capacidades diferentes:

Caminhão Alfa

Consegue carregar no máximo 5 toneladas de pluviômetros, independente da quantidade e tipo.

Caminhão Beta

Consegue carregar qualquer quantidade e peso de pluviômetros, no entanto, não é capaz de carregar mais do que um pluviômetro de cada tipo.

Ambas as classes devem sobrescrever o método `inserePluviometro`.