



Universidade Federal do Agreste de Pernambuco
Curso: Bacharelado em Ciência da Computação.
Disciplina: Programação Orientada à Objetos.
Professora: Thaís Alves Burity Rocha.

6ª Lista de Exercícios

Parte I - Exercícios teóricos de fixação

- 1) Complete as frases a seguir:
 - a. Cada declaração de classe que começa com a palavra reservada _____ deve ser armazenada em um arquivo com exatamente o mesmo nome da classe e que termina com a extensão “.java”.
 - b. Por default, quando as classes não são organizadas em pacotes elas ficam no mesmo _____.
 - c. A palavra reservada public é um _____ e pode ser aplicada à _____, _____ e _____.
 - d. A declaração _____ não é necessária se uma classe é referenciada por seu nome completo.
 - e. A _____ deve ser a primeira coisa a aparecer em um arquivo .java.
 - f. Variáveis ou métodos declarados com o modificador de acesso _____ só podem ser acessados em classes no mesmo pacote da classe em que foram declarados.
- 2) A maioria das classes precisam ser importadas antes de ser possível utilizá-las em uma aplicação. Por que toda aplicação pode usar a classe System sem importá-la antes?
- 3) O que é **ocultação de informação** e como este conceito pode ser empregado em Java?
- 4) Discuta a diferença entre encapsulamento (*encapsulation*) e ocultação de informação (*information hiding*).

Parte II - Exercícios práticos

- 1) Crie uma classe que modele um Tamagushi (Bichinho Eletrônico), sabendo que todo ele possui nome, fome, saúde, idade e humor. O humor deve ser uma combinação de fome e saúde e, portanto, varia ao longo do tempo.
- 2) Considere o sistema bancário discutido em sala de aula:
 - Crie os pacotes entidade, repositório e main.
 - Mova as classes Conta, Cliente e Endereco para o pacote entidade.
 - Mova as classes de repositório para o pacote repositório.
 - Mova a classe Main para o pacote main.
 - Atualize a classe Conta utilizando modificadores de acesso para os atributos e métodos.
 - Atualize a classe Conta incluindo métodos *getters* e *setters* (que façam sentido).
 - Atualize a classe Cliente utilizando modificadores de acesso para os atributos e métodos.
 - Atualize a classe Cliente incluindo métodos *getters* e *setters* (que façam sentido).
 - Atualize a classe Endereco utilizando modificadores de acesso para os atributos e métodos.
 - Atualize a classe Endereco incluindo métodos *getters* e *setters* (que façam sentido).
 - Atualize as classes de repositório utilizando modificadores de acesso para os atributos e métodos.
 - Atualize a classe Conta para incluir a variável contador (static, int). Essa variável deverá ser inicializada com 0 e sempre que um objeto for instanciado ela deverá ser incrementada. A classe Conta deverá proteger o acesso a essa variável, de maneira que seu valor nunca possa ser alterado fora da classe.
 - Projete e implemente um método para consultar o valor da variável contador. Em seguida, teste na classe Main a funcionalidade de controle desenvolvida.
- 3) Atualize os exercícios das listas anteriores a fim de garantir que os atributos sejam private e que o acesso a estes seja feito apenas por métodos.
- 4) (Deitel) Crie a classe Retangulo. A classe possui atributos de altura e largura, ambos com valor default 1. Ela possui métodos para calcular o perímetro e a área do retângulo. Ela possui métodos get e set para altura e largura. Os métodos set devem verificar que altura e largura são números de ponto flutuante maiores que 0.0 e menores que 20.0. Escreva um programa para testar a classe Retangulo.

- 5) (Deitel) Crie uma classe Retangulo mais sofisticada que a da questão anterior. Essa classe deve armazenar apenas as coordenadas cartesianas das quatro pontas do retângulo. O construtor chama um método set que aceita quatro conjuntos de coordenadas e verifica que cada um deles está no primeiro quadrante e que não há coordenada x ou y com valor superior a 20.0. O método set também verifica que as coordenadas especificam um retângulo. Ofereça métodos para calcular a altura, largura, perímetro e área. A altura consiste na maior das duas dimensões. Inclua um método eQuadrado que determina se o retângulo é um quadrado. Escreva um programa para testar a classe Retangulo.
- 6) (Deitel - Adaptado) Crie uma classe chamada Racional para realizar operações com frações. Escreva um programa para testar sua classe. Use variáveis inteiras para representar os atributos privados numerador e denominador. Construa um construtor que permite um objeto desta classe ser inicializado quando for declarado. O construtor deve armazenar a fração na forma reduzida. Ex.: $\frac{2}{4}$ deve ser armazenado como $\frac{1}{2}$. 1 no numerador e 2 no denominador. Construa métodos públicos para as seguintes operações:
- Adição de dois números racionais.
 - Subtração de dois números racionais.
 - Multiplicação de dois números racionais.
 - Divisão de dois números racionais.
 - Print no formato a/b;
 - Print como ponto flutuante.
- 7) Implemente a classe ContaAPagar que contém os atributos código, descrição, data de vencimento, data de pagamento, valor e status. Status deve valer “Aberto” por default, mas também pode assumir os valores “Pago” ou “Vencida”. Além disso, considere que a classe deve ter:
- Dois construtores diferentes;
 - Getters e setters quando necessário;
 - Um método de efetuar pagamento, que considera a data corrente como a data de pagamento;
 - Um método para sinalizar que a conta está vencida.

Avalie a necessidade e viabilidade de restringir acesso aos métodos.

- 8) Implemente uma classe para representar um aparelho de ar-condicionado. Considere que o condicionador possui 10 potências diferentes. Cada unidade da potência do condicionador diminui a temperatura do ambiente em 1,8°C. A variação que o condicionador consegue causar está no intervalo [0°C - 18°C], ou seja, zero graus de variação quando desligado e dezoito graus de variação quando ligado na potência máxima. Através de um sensor, o condicionador é informado da temperatura externa. Dada essa temperatura e a potência selecionada, o condicionador calcula e retorna a temperatura do ambiente. No programa de teste, crie dois condicionadores, informe duas temperaturas externas diferentes para cada um (ex.: 25°C e 31°C), ajuste o primeiro em potência média (5) e o segundo em potência máxima (10), e exiba a temperatura resultante de cada ambiente.
- 9) Escreva uma classe para representar um voo que acontece em uma data e horário. Cada voo possui no máximo 100 passageiros, e a classe permite controlar a ocupação das vagas, além de outras informações básicas sobre o voo. A classe deve ter um construtor que recebe os dados do voo como parâmetro, e ao menos os seguintes métodos:
- Um método que informa o número da próxima cadeira livre;
 - Um método que verifica se uma cadeira está ocupada;
 - Um método que ocupa uma cadeira de interesse e retorna uma flag sinalizando se a operação foi bem sucedida;
 - Um método que informa o número de cadeiras não ocupadas.

Identifique e implemente os atributos da classe com base nesses dados, além de outros métodos que possam se aplicar nesse contexto. Utilize o conceito de ocultação de informação.

- 10) Modifique a classe Baralho (Lista 4) para que ela possa distribuir uma mão de cinco cartas de pôquer. Então inclua métodos que determinam se uma mão contém:
- Um par (2 cartas de mesmo valor);
 - Dois pares;
 - Trinca (3 cartas de mesmo valor; por exemplo, três valetes);
 - Quadra (4 cartas de mesmo valor; por exemplo, quatro ases);
 - Flush (5 cartas do mesmo naipe);
 - Straight (5 cartas de valores consecutivos);
 - Full house (2 cartas de um valor e 3 cartas de outro valor).

Não esqueça de atualizar a classe Carta (Lista 4) para proteger o acesso aos seus atributos. Por fim, crie um programa de teste que distribui duas mãos de pôquer de cinco cartas, avalia cada mão e determina qual é a melhor.

- 11) Padrões de projeto são soluções genéricas para os problemas mais comuns do desenvolvimento de software orientado a objetos, obtidas através de experiências de sucesso na indústria de software. **Dentre os padrões, temos o Singleton, cujo objetivo consiste em garantir que uma classe só tenha uma única instância, e prover um ponto de acesso global a ela. Ou seja, a classe deve controlar a criação do seu objeto, garantindo que o construtor seja chamado uma única vez, e deve oferecer algum meio para que outras classes tenham acesso a este objeto.** A utilidade nesse padrão é garantir que certos recursos, geralmente associados à funcionalidade de controle, sejam únicos em um sistema, evitando inconsistências. Com base nessa descrição, escreva uma classe que implementa esse padrão.
- 12) (Deitel adaptada) O uso de computadores na educação é chamado instrução assistida por computador (CAI). Escreva um programa que ajudará um aluno da escola elementar a aprender multiplicação. Utilize a API para selecionar aleatoriamente dois inteiros positivos de um algarismo. O programa deve então fazer ao usuário uma pergunta, como "Quanto é 6 vezes 7?". O aluno insere então a resposta. Em seguida, o programa verifica a resposta do aluno. Se estiver correta, exiba a mensagem "Muito bem!" e faça uma outra pergunta de multiplicação. Se a resposta estiver errada, exiba a mensagem "Não. Por favor, tente de novo." e deixe que o aluno tente a mesma pergunta várias vezes até que por fim ele acerte.
- 13) (Deitel) Um problema em ambientes CAI é a fadiga do aluno. Isso pode ser reduzido variando-se as respostas do computador para prender a atenção do aluno. Modifique o programa anterior para que vários comentários sejam exibidos para cada resposta como mostrado a seguir:
 - Possibilidades para uma resposta correta:
 - Muito bom!
 - Excelente!
 - Bom trabalho!
 - Mantenha um bom trabalho!
 - Possibilidades para uma resposta incorreta:
 - Não. Por favor, tente de novo.
 - Errado. Tente mais uma vez.
 - Não desista!
 - Não. Continue tentando.

Utilize a geração de números aleatórios para escolher um número de 1 a 4 que será utilizado para selecionar uma de quatro respostas adequadas a cada resposta correta ou incorreta.

Mais exercícios: Todos os exercícios práticos das listas anteriores podem ser atualizados para incorporar o conceito. Além disso, é esperado que o conceito seja empregado na resolução de listas futuras, ainda que não explicitamente solicitado nas questões.