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

5ª Lista de Exercícios

Parte I- Exercícios teóricos

1) Complete as frases a seguir:
   a. Para um método ser executado é necessário _____.
   b. Uma variável conhecida apenas dentro do método em que foi declarada é denominada uma _____.
   c. _____ é a possibilidade de uma classe possuir mais de um método com mesmo nome, porém com parâmetros diferentes (tipo ou quantidade).
   d. O comando _____ em um método em execução pode ser usado para passar o valor de uma expressão de volta para o método que o chamou.
   e. A palavra reservada _____ indica que um método não retorna nenhum valor.
   f. As três maneiras de retornar o controle da execução do programa dada a execução de um método são: _____, _____ e _____.
   g. O _____ de uma declaração é a parte do programa que pode referenciar a entidade na declaração pelo nome.
   h. Quando há sobrecarga, a versão do método a ser executada é definida em tempo de _____.
   i. _____ na lista de parâmetros de um método indica que o método pode receber um número variável de argumentos de um dado tipo.
   j. Em Java, só existe passagem de parâmetros por _____.
   k. Em Java é possível ter diversos métodos com o mesmo nome em que cada um opera sobre diferentes tipos ou quantidades de argumentos. Esse recurso é chamado _____ de método.
   l. Comentários de bloco são delimitados por _____ e _____.
   m. Comentários de linha são iniciados com _____.
   n. Métodos que realizam tarefas comuns e que não precisam de objetos para serem invocados são denominados métodos _____.
2) Qual a diferença entre uma variável local e um atributo?
3) O que são var-args? Qual a utilidade desse recurso? Justifique sua resposta.
4) Qual é a utilidade de comentários em programas?
5) O que é Javadoc?
6) Explique a afirmativa: **O comportamento de um método static não depende do estado dos objetos de sua classe.**

Parte II- Exercícios práticos

1) Escreva uma classe chamada NumeroComplexo que represente um número complexo. A classe deverá ter um construtor para inicializar seus atributos e ainda conter os seguintes métodos:
   - Impressão de um número complexo no console, usando a notação **a+bi**, onde **a** é a parte real e **b** é a parte imaginária;
   - Verificação de igualdade entre dois números complexos;
   - Soma de dois números complexos;
   - Subtração de dois números complexos;
   - Multiplicação de dois números complexos;
   - Divisão de dois números complexos.

   Escreva uma classe de teste também.

   Observação: Fórmulas de operações básicas com números complexos

- (a+bi) + (c+di) = (a+c) + (b+d)i;
- (a+bi) - (c+di) = (a-c) + (b-d)i;
- (a+bi) x (c+di) = (ac-bd) + (ad+bc)i;
- (a+bi) / (c+di) = (ac+bd)/(c^2+d^2) + [(bc-ad)/(c^2+d^2)]i.

2) Gere a documentação da classe NumeroComplexo, utilizando as tags @author, @param e @return.

3) (Deitel) Escreva o método múltiplo que determina, para um par de inteiros, se o segundo inteiro é um múltiplo do primeiro. O método deve receber dois inteiros como argumento e retornar true se o segundo é múltiplo do primeiro ou false, caso contrário. Dica: Use o operador de resto da divisão. Incorpore esse método dentro de uma aplicação que recebe uma série de pares de inteiros (um par por vez) e determina se o segundo valor em cada par é um múltiplo do primeiro.

4) (Deitel) Escreva um método exibirQuadradoDeAsteriscos que exibe um quadro sólido (mesmo número de linhas e colunas) de asteriscos cujo lado é especificado como um parâmetro inteiro denominado lado. Por exemplo, se lado é 4, o método deve exibir:
   ****
   ****
   ****
   ****

   Incorpore esse método dentro de uma aplicação que lê um valor inteiro para lado do teclado e exibe asteriscos com o método exibirQuadradoDeAsteriscos.

5) (Deitel) Crie uma nova versão do método da questão anterior para exibir o caractere que for desejado. Para tanto, defina um novo parâmetro, denominado caractere. Dessa forma, se o valor de lado for 5 e de caractere for "#", o método deve exibir:
   #####
   #####
   #####
   #####
   #####

6) (Deitel adaptado) Escreva um programa que recebe um valor como parâmetro e retorna o valor inverso. Se o valor for um número inteiro, o programa retorna o número com seus dígitos invertidos. Por exemplo, o valor invertido do número 7631 é 1367. Se o valor for uma String, o comportamento deve ser o mesmo. Exemplo, o valor invertido da String "Garanhuns" é "snuhnaraG". O valor a ser invertido deve ser lido do teclado. O programa deve exibir o valor original e, em seguida, exibir o valor invertido. Dica: Utilize os métodos toCharArray e copyValueOf de String. O primeiro gera um vetor de char a partir de uma String. O segundo gera uma String a partir de um vetor de char.

7) Utilizando a classe Math (java.lang.Math) da API de Java, crie uma classe denominada TesteMath, adicione o método main e então, utilizando a saída padrão, exiba o resultado das seguintes operações:
   - Math.pow(x, y)
   - Math.sqrt(x)
   - Math.round(x)
   - Math.max(x,y)
   - Math.random()

   Para entender o que é cada método, consulte a documentação da classe Math.

8) (Deitel) Escreva uma aplicação que solicita que o usuário informe o raio do círculo e usa um método para calcular a área do círculo. Justifique o projeto do seu método.

9) (Deitel) Escreva um método que identifique o menor dentre 3 números de ponto flutuante. Use o método min da classe Math. Incorpore o método dentro de uma aplicação que solicita os 3 valores ao usuário, identifica o menor valor e exibe o resultado.

10) (Rafael Santos) Escreva a classe ConversaoDeUnidadesDeTempo com métodos para conversão aproximada das unidades de velocidade segundo a lista abaixo.
    - 1 minuto = 60 segundos
    - 1 hora = 60 minutos
    - 1 dia = 24 horas
    - 1 semana = 7 dias
    - 1 mês = 30 dias
    - 1 ano = 365.25 dias

11) (Rafael Santos) O tempo de gestação de um elefante indiano é de aproximadamente 624 dias. Usando a classe ConversaoDeUnidadesDeTempo, escreva um programa em Java que mostre qual é o tempo de gestação de um elefante indiano em dias, horas, minutos e segundos. Escreva métodos adicionais para a classe ConversaoDeUnidadesDeTempo se necessário.

12) (Rafael Santos) Crie o método calcularDistancia para a classe Ponto2D (Lista 3) que recebe uma outra instância da classe Ponto2D e retorna um valor do tipo double correspondente à distância euclidiana entre o Ponto2D encapsulado e o passado como argumento.

    Dica: A distância euclidiana d entre um ponto com coordenadas (x1,y1) e outro ponto com coordenadas (x2,y2) é calculada por $d = \sqrt{(x1 - x2)^2} + (y1 - y2)^2$. Para calcular a raiz quadrada, utilize o método Math.sqrt.

13) (Rafael Santos) Crie o método calcularDistanciaDaOrigem para a classe Ponto2D (Lista 3) que não recebe nenhum argumento, mas calcula a distância euclidiana entre as coordenadas encapsuladas e a origem do sistema de coordenadas. Para isso, reuse o método calcularDistancia da questão anterior. Dica: Dentro do novo método, crie um objeto Ponto2D correspondente à origem.

14) (Rafael Santos) Crie o método éProximo para a classe Ponto2D (Lista 3) que recebe uma outra instância da classe Ponto2D e um limiar (valor do tipo double) como argumentos, calculando a distância entre as coordenadas encapsuladas e as coordenadas da instância passada como argumento, retornando o valor booleano true se a distância for menor do que o limiar passado como argumento. Por exemplo, dado o ponto encapsulado (3,3), o ponto passado como argumento (4,1) e o limiar 3.0, o método deve retornar true, já que a distância entre os dois pontos (2.236) é menor do que o limiar. Já se o limiar fosse 2.0, o método deveria retornar false.

15) (Rafael Santos) Escreva uma classe SenhaAtendimento que encapsula um valor inteiro sequencial representando o lugar da senha em uma ordem de atendimento. Essa classe deve permitir que um programa crie um número limitado de instâncias dela, cada uma numerada com um valor sequencial. Por exemplo: Senhas criadas primeiro possuem um valor menor, sequencial. Em seguida, explique se foi necessário usar o modificador static.

16) Gere a documentação para o sistema bancário utilizando Javadoc e as tags @author, @param e @return.

17) Defina uma classe para representar uma máquina que produz café, chocolate quente e cappuccino. O usuário começa adicionando moedas. Em seguida ele aperta um botão para escolher uma das três bebidas e por fim solicita o preparo da bebida escolhida. Quando for solicitado o preparo, a máquina deverá verificar se o valor de moedas entrado é suficiente para comprar a bebida escolhida, e caso não seja suficiente, deverá avisar ao usuário. O usuário também terá a possibilidade de retirar seu dinheiro a qualquer instante e o valor retornado poderá ser o troco se uma bebida foi vendida. Assuma que os preços do café, chocolate e cappuccino são respectivamente 2, 5 e 3 reais. Considere que é possível inserir moeda de R$0,05, R$0,10, R$0,25, R$0,50 e R$1,00. Qualquer outro valor deve ser ignorado.

18) Implemente um jogo de Craps para um jogador. O jogador lança um par de dados, obtendo a soma entre 2 e 12. Se na primeira jogada ele tirar 7 ou 11, ele ganha. Se ele tirar 2, 3 ou 12 na primeira jogada, isto é chamado de "craps" e ele perdeu. Se na primeira jogada ele somou 4, 5, 6, 8, 9 ou 10, este é seu "Ponto". Seu objetivo agora é continuar jogando os dados até tirar este número novamente. Ele perde, no entanto, se tirar um 7 antes de tirar este “Ponto” novamente.

19) Atualize o jogo de Craps para suportar 2 jogadores. As jogadas entre os jogadores devem ser alternadas e estes devem ser informados dos resultados das jogadas.

20) Utilizando as classes Carta e Baralho (Lista 4), desenvolva o jogo de Tri-du, um jogo de cartas derivado do popular jogo de Truco. O jogo utiliza um baralho normal de 52 cartas, com treze cartas de cada naipe, mas os naipes são ignorados. Apenas o valor das cartas, considerados como inteiros de 1 a 13 (1 = Ás; 11 = J; 12 = Q; 13 = K), são utilizados. No jogo, cada jogador recebe três cartas. As regras são simples:
    - Um trio (três cartas de mesmo valor) ganha de uma dupla (duas cartas de mesmo valor).
    - Um trio formado por cartas de maior valor ganha de um trio formado por cartas de menor valor.
    - Uma dupla formada por cartas de maior valor ganha de uma dupla formada por cartas de menor valor.

    Note que o jogo pode não ter ganhador em muitas situações; nesses casos, as cartas distribuídas são devolvidas ao baralho, que é embaralhado e uma nova partida é iniciada. Observação: Cartas coringas devem ser ignoradas. Logo, caso uma carta coringa seja sorteada, deverá ser tirada uma nova carta em seu lugar.

21) Desenvolva a simulação de um jogo no qual o usuário deve encher um balde com o máximo volume possível de água. Inicialmente a capacidade do balde é desconhecida, e pode variar entre 10 e 50 unidades. A cada jogada, o jogo sorteia uma quantidade de água, mostra essa quantidade ao usuário e pergunta se o usuário deseja ou não utilizá-la. O jogo termina quando o balde estiver cheio ou o usuário desistir de colocar mais água no balde. Para facilitar, o jogo avisa quando o volume armazenado chega ou ultrapassa a metade da capacidade do balde.

Mais exercícios: Capítulo 6 do livro de Deitel (10ª Edição) e capítulos 2, 3 e 5 do livro de Rafael Santos.
Observação: Recomenda-se a revisão da solução das listas anteriores a fim de garantir métodos bem projetados.