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

3ª Lista de Exercícios

Parte I- Exercícios teóricos

1) O que você entende por: Classe? Objeto? Programação orientada a objetos?
2) Qual é o propósito da palavra reservada `new`? Explique o que acontece quando ela é utilizada.
3) O que é um construtor default?
4) Explique o propósito de uma variável de referência.
5) Explique o uso do operador **`this`** tal como visto até o momento em sala de aula (em métodos e construtores).
6) Dado o trecho de código a seguir, responda:

```
1  Endereco end = new Endereco();
2  Cliente cliente = new Cliente(end);
3  Conta conta1 = new Conta(cliente);
4  Conta conta2 = new Conta();
5  cliente = null;
6  end = null;
7  conta2 = null;
```

   a. Quantos objetos foram criados após a execução da linha 4?
   b. Quantos objetos estão aptos a serem removidos da memória a partir da linha 7?

7) Objetos da classe `String` são considerados imutáveis, visto que eles não podem ter seus tamanhos ou conteúdos alterados. Entretanto, algumas instruções parecem contradizer esta afirmação. Por exemplo, a sequência de instruções a seguir é perfeitamente legal:

```
String umString = "Bom";
umString = umString + "Bom"; // Contradição?
System.out.print(umString);
```

   Explique como tais instruções não contradizem a afirmação feita com relação a objetos da classe `String`.
8) Explique o que é o relacionamento TEM-UM entre classes.

Parte II- Exercícios práticos

1) Crie uma classe com o construtor default (que não recebe argumento algum) que imprime uma mensagem no console. Crie um objeto dessa classe.
2) Na classe projetada na questão anterior, adicione um construtor sobrecarregado, que recebe uma String como argumento e a exibe juntamente com a mensagem.
3) Crie uma classe denominada Empregado que contém os atributos nome, sobrenome e salário mensal e um método que calcula o salário anual. A classe deve ter um construtor que inicializa esses atributos. Se o salário mensal for negativo, altere-o para zero. Crie uma classe de teste denominada EmpregadoTeste que cria um empregado, exibe seu salário anual, reajusta o salário mensal em 10% e exibe o salário anual novamente.
4) Considere as classes Conta e Cliente apresentadas em sala de aula:
   a. Crie a classe Conta. Ela deve conter dois atributos: numero (String) e saldo (double). Além disso, a classe deve prover um construtor para inicializá-los.

b. Crie a classe Main. Ela deve conter o método main que, por sua vez, deverá instanciar dois objetos Conta e exibir o saldo de cada conta.
c. Crie a classe Cliente. Ela deve conter dois atributos: nome (String) e cpf (String). Além disso, a classe deve prover um construtor para inicializá-los.
d. Atualize a classe Conta adicionando o atributo cliente (Cliente).
e. Atualize a classe Main. Ela deve conter o método main, que deverá instanciar dois objetos Conta para o mesmo cliente e deve exibir o saldo de cada conta.
f. Atualize a classe Conta, inserindo os métodos creditar, debitar e transferir. Em seguida, teste o funcionamento de cada método na classe Main. Observação: Ignore valores negativos.
g. Atualize a classe Conta adicionando três construtores: um construtor sem parâmetros, que inicializa numero como uma String vazia e saldo com 0; um construtor que inicializa numero com o valor passado como parâmetro e saldo com 0; um construtor que inicializa saldo com o valor passado como parâmetro e inicializa numero com uma String vazia. Os construtores deverão utilizar o operador this. Teste o funcionamento de cada construtor na classe Main.
h. Projete e crie uma classe Endereco e faça ela se relacionar com a classe Cliente, com base no que acontece no mundo real. Utilize a classe Main para testar o novo código.

5) Atualize os métodos creditar, debitar e transferir da classe Conta para impedir que estes aceitem valores negativos como argumento, quando forem chamados. Nessa situação, deverá ser exibida uma mensagem informando que o valor é inválido. Atualize a classe Main para testar os novos métodos.

6) Atualize os métodos debitar e transferir da classe Conta para impedir que o saldo da conta fique negativo. Além disso, quando o valor do débito/transferência extrapolar o saldo da conta, deverá ser exibida uma mensagem informando que não há saldo suficiente. Atualize a classe Main para testar os novos métodos.

7) Atualize o(s) construtor(es) da classe Conta para impedir que seja criada uma conta com saldo negativo. Nessa situação, deverá ser atribuído o valor 0 ao saldo. Atualize a classe Main para testar o(s) novo(s) construtore(s).

8) Reveja o projeto da classe Data (Lista 2), a fim de garantir que ela possua um construtor que inicializa os valores de dia, mês e ano corretamente, verificando antes se a data é válida. Em caso de data inválida, defina valores default de sua preferência. A classe também deve ter um método para imprimir a data no formato "dia/mês/ano". Em seguida, crie um programa para testar seu código.

9) (Deitel) Uma loja vende 5 produtos cujos preços de varejo são: Produto 1, R$2,98; produto 2, R$4,50; produto 3, R$9,98; produto 4, R$4,49 e produto 5, R$6,87. Escreva uma aplicação que lê uma série de pares de valores como a seguir:
   - Número do produto
   - Quantidade vendida

   Seu programa deve usar o comando switch para determinar o preço de varejo de cada produto. Ele deve calcular e exibir o total do valor de varejo de todos os produtos vendidos. Use um loop while para determinar quando o programa deve parar e exibir os resultados. Lembre-se que o método main deve ser usado apenas para demonstrar o funcionamento do programa.

10) (Deitel) Escreva um programa que recebe cinco números e determina e imprime a quantidade de número negativos, a quantidade de números positivos e a quantidade de zeros. Lembre-se de projetar um método para esse fim, usando o método main apenas para demonstrar o uso do método criado.

11) Crie um programa que representa uma calculadora (simples), conforme descrito a seguir. Lembre-se de projetar uma classe tal como visto em sala, usando o método main apenas para demonstrar o uso da calculadora.
   - O programa deve exibir no console um menu solicitando a operação a ser realizada:
     (1) soma
     (2) subtração
     (3) multiplicação
     (4) divisão
     (5) sair
   - Uma vez que o usuário digite uma opção, o programa deve solicitar para ele informar os operandos.
   - Em seguida, o programa deve exibir o resultado da operação e o menu, novamente.
   - O programa deve executar até que o usuário digite a opção de saída (5).

12) (Rafael Santos) Escreva a classe Contador que encapsule um valor usado para contagem de itens ou eventos e que contenha métodos para zerar, incrementar e consultar o valor do contador. A classe deve ter dois construtores, um que considere que o contador começa a contar a partir do zero, e outro que aceita um valor inicial para contagem.

13) Sobre a classe Lampada (Lista 2):
   - Escreva um programa que use várias instâncias da classe Lampada.

- Modifique a classe Lampada para que esta contenha também um campo que indique quantas vezes a lâmpada foi acesa, usando a classe Contador da questão 12.
- Escreva um programa para exibir quantas vezes uma lâmpada foi acesa.

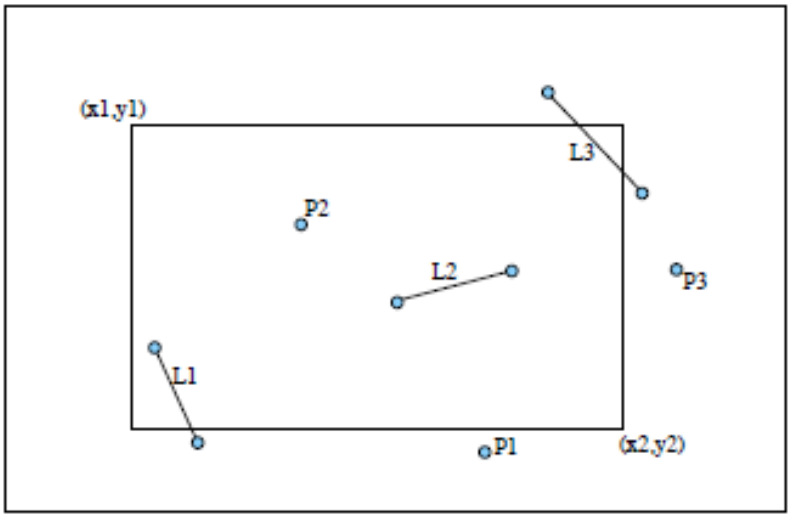
14) Usando a classe Contador da questão 12, atualize a classe Conta do nosso sistema bancário para registrar quantas transferências bancárias foram realizadas a partir de uma conta. A cada 5 transferências, deverá ser descontada uma taxa de serviço no valor de R$2,00, desde que haja saldo suficiente. Caso não haja saldo suficiente, será acrescido R$1,00 à taxa de serviço. Para testar seu código, utilize o teclado para receber o valor a ser transferido entre duas contas. Por fim, responda: Qual é o tipo de relacionamento entre as classes Conta e Contador?

15) Defina uma classe para representar o resultado de um jogo de futebol e uma classe para testá-la, que use a entrada padrão. Todo resultado deve ter os nomes dos dois times que jogaram e a quantidade de gols marcados por cada um. Além disso, todo resultado deve ter os seguintes métodos:
    - vencedor, que retorna o nome do time que ganhou o jogo ou a string vazia caso o jogo tenha sido empate;
    - gols, que recebe como parâmetro o nome de um time e retorna como resultado o número de gols marcados por aquele time

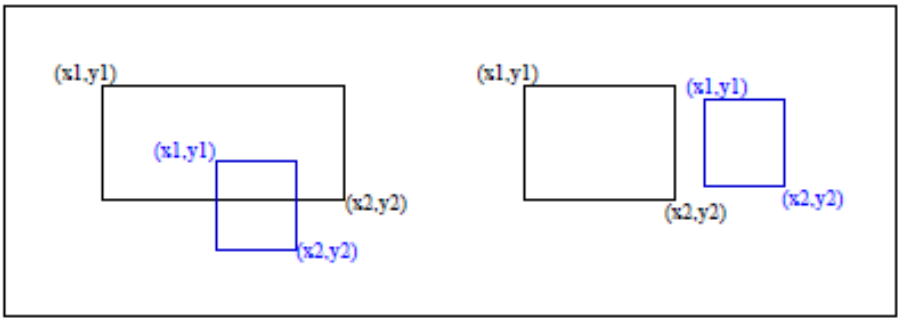
16) (Rafael Santos, adaptada) Escreva uma classe Ponto2D que representa um ponto no plano cartesiano. A classe, além de atributos condizentes, deve permitir que a criação de objetos se dê por qualquer um dos três meios listados a seguir. Utilize a entrada padrão para testar seu código.
    - Por default, o ponto deve ser criado na origem do espaço 2D;
    - O ponto pode ser criado num local indicado por dois parâmetros (coordenadas x e y);
    - O ponto pode ser criado num local indicado por apenas um parâmetro. Nesse caso, as coordenadas x e y serão iguais.
    - O ponto pode ser criado em um local indicado por outro ponto. Nesse caso, o novo ponto será criado em posição oposta ao ponto passado como argumento. Por exemplo, o oposto do ponto com coordenadas (5, 8) é o ponto com coordenadas (-5, -8).

17) (Rafael Santos, adaptada) Crie uma classe Linha para representar um segmento de reta unido por dois pontos no espaço cartesiano de duas dimensões, usando duas instâncias da classe Ponto2D. Considere que essas instâncias podem ser passadas por parâmetro ou criadas junto a um objeto do tipo Linha. Por fim, responda: Qual é o tipo de relacionamento entre as classes Ponto2D e Linha?

18) (Rafael Santos, adaptada) Crie uma classe Retangulo para representar um retângulo cujos pontos extremos sejam duas instâncias da classe Ponto2D. Por exemplo, na imagem a seguir (x1; y1) e (x2; y2) são os pontos extremos que definem o retângulo. Deve ser possível criar objetos Retangulo a partir de seus dois pontos extremos (dois objetos Ponto2D) ou das coordenadas destes (quatro valores inteiros). Além disso, a classe deve ter um método para verificar se um ponto passado como argumento está localizado dentro de um retângulo. O ponto deve ser representado por uma instância da classe Ponto2D. O método deverá retornar true se o ponto estiver contido no retângulo, e false se não estiver. Dica: Para verificar se um ponto está dentro do retângulo, verifique se as coordenadas do ponto estão dentro das coordenadas do retângulo. Por exemplo, de acordo com a figura, o ponto P1 estaria fora do retângulo, uma vez que a sua coordenada y é menor do que a menor coordenada y do retângulo. Já o ponto P2 estaria dentro do retângulo, e o ponto P3 também estaria fora do retângulo.



19) (Rafael Santos) Modifique a classe Retangulo para que esta contenha um método calculaIntersecção, que recebe como argumento uma outra instância da própria classe Retangulo e que calcula as coordenadas de um retângulo que é a intersecção do retângulo encapsulado com o passado como argumento, retornando uma nova instância da classe Retangulo correspondente à intersecção. Dicas: Os pontos do novo retângulo podem ser calculados com regras simples, implementadas através de ifs encadeados. Nem sempre existe intersecção entre dois retângulos. Considere a figura abaixo: no lado esquerdo existem dois retângulos (mostrados em cores diferentes) que têm intersecção, e, no lado direito, dois que não têm. No caso de não existir intersecção, o método deve retornar null.



2) Usando apenas o diagrama de classes da UML, projete as classes Pessoa, Endereço e PacoteEncomenda no contexto de um sistema de controle de envio de encomendas, considerando que:
   - Uma pessoa está atrelada a um endereço, mas um endereço pode pertencer a mais de uma pessoa;
   - Um PacoteEncomenda é único e não pode ser entregue em diferentes endereços;
   - Um PacoteEncomenda é direcionado a uma pessoa.

   Qual é o tipo de associação entre as classes: agregação ou composição? Explique.

20) Uma empresa quer verificar se um empregado está qualificado para a aposentadoria ou não. Para estar em condições, um dos seguintes requisitos deve ser satisfeito:
   - Ter no mínimo 65 anos de idade.
   - Ter trabalhado no mínimo 30 anos.
   - Ter no mínimo 60 anos e ter trabalhado no mínimo 25 anos.

   Com base nas informações acima, faça um programa que, dados o código identificador do empregado, o ano de seu nascimento e o ano de seu ingresso na empresa, avalie se ele está qualificado para se aposentar. A classe de empregado deve ter 2 construtores: Um construtor que permite configurar todos os atributos e outro que admite que o ano de ingresso na empresa é o ano corrente. Ambos os construtores devem garantir que sejam usados valores válidos para os atributos. Em caso de valores inválidos, defina valores default de sua preferência. O programa deve usar a entrada padrão (teclado) para receber do usuário os dados do empregado de interesse.

21) (Ana Cristina Melo) Desenhe o diagrama de classes completo para o cenário a seguir. Mariana prepara diversos exercícios para suas filhas que estão na primeira e na segunda séries. Ela gostaria de informatizar esses exercícios para gerar testes aleatórios. Cada teste gerado deve ser guardado (acompanhado de suas questões) com a indicação de sua data de geração. Na geração de um teste é preciso informar o número de questões desejadas e a qual disciplina pertence o teste. Para cada disciplina, cadastra-se uma lista de questões objetivas, identificando de que bimestre é cada questão e a que matéria pertence. O gabarito também é cadastrado a fim de facilitar a correção do teste. Cada matéria faz parte de uma única disciplina. A série está ligada a matéria. Por exemplo, para a disciplina matemática, Mariana prepara um teste com 20 questões. Cada questão corresponde à um bimestre (1º, 2º, 3º ou 4º) e a uma matéria (ex: adição, divisão, números pares e ímpares, números primos, sinônimos etc.). Cada matéria corresponde a uma disciplina (adição – matemática; sinônimos – português etc.).

22) (Ana Cristina Melo) Desenhe o diagrama de classes completo para o cenário a seguir. Rafaela possui vários temas de festas infantis para aluguel. Ela precisa controlar os aluguéis e para isso quer uma aplicação que permita cadastrar o nome e o telefone do cliente, o endereço completo da festa, o tema escolhido, a data da festa, a hora de início e término da festa. Além disso, para alguns clientes antigos, Rafaela oferece descontos. Sendo assim, é preciso saber o valor realmente cobrado num determinado aluguel. Para cada tema, é preciso controlar a lista de itens que compõe o tema (ex: castelo, boneca da cinderela, bruxa etc.), o valor do aluguel e a cor da toalha da mesa que deve ser usada com o tema.

23) (Ana Cristina Melo) Desenhe o diagrama de classes completo para o cenário a seguir. A empresa AProf oferece cursos diversos de aperfeiçoamento profissional. Inicialmente, o dono precisa controlar os cursos oferecidos organizados em turmas. As turmas possuem data de início e término, horário de início e término, e um professor. É necessário saber o nome do professor, o celular do professor e o valor da sua hora/aula. Para cada curso, deve-se controlar a carga horária, o conteúdo programático e o valor do curso.

24) (Ana Cristina Melo) Desenhe o diagrama de classes completo para o cenário a seguir. Cristina quer fazer um jogo da forca em computador para os seus filhos e sobrinhos. O jogo consiste em tentar acertar as letras de uma palavra escondida. Se o jogador errar a letra, surge mais um pedaço de um boneco que ao final irá para a forca. As palavras ou frases para cada rodada são obtidas de um banco de palavras, que o próprio jogador pode incrementar. Cada palavra ou frase pertence a um tema. A cada rodada, a aplicação sorteia se mostrará uma, duas ou três palavras ou uma frase, e escolhe aleatoriamente o tema e as palavras (ou frase). A palavra (ou frase) é exibida escondida e cada letra acertada desvenda sua posição correspondente. Letras erradas são colocadas num quadro e um dos pedaços do boneco aparece. O nome do jogador é guardado com seus pontos para fazer parte do quadro de maiores escores (pontuações). Quando ele acerta a palavra, ganha 100 pontos e para cada letra que ficou encoberta somam-se mais 15 pontos.

25) (Ana Cristina Melo) Gustavo tem uma coleção grande de revistas em quadrinhos. Por isso, resolveu emprestar para os amigos. Assim foi criado o Clube da Leitura. Mas para não perder nenhuma revista, seu pai lhe fez uma aplicação que cadastra e controla o empréstimo. Para cada revista, cadastram-se o tipo de coleção (ex: Cebolinha, Pato Donald, Batman etc.), o número da edição, o ano da revista e a caixa onde está guardada. Cada caixa tem uma cor, uma etiqueta e um número. Para cada empréstimo cadastram-se o amiguinho que pegou a revista, qual foi a revista, a data de empréstimo e a data de devolução. Cada criança só pode pegar uma revista por empréstimo. O cadastro do amiguinho consiste no nome do amiguinho, o nome da mãe, o celular e de onde é o amigo (do prédio ou da escola). Projete o diagrama de classes.

Mais exercícios: Capítulos 2, 3, 4 e 5 do livro de Deitel (10ª Edição) e capítulos 3 e 4 do livro de Rafael Santos. Observação: Vários exercícios dos livros consideram o uso de modificadores de acesso (métodos get e set), que ainda iremos abordar em sala de aula. Não se preocupem com isso agora.