



UNIVERSIDADE ESTADUAL DE SANTA CRUZ - UESC

PRÓ-REITORIA DE GRADUAÇÃO – PROGRAD

DEPARTAMENTO DE CIÊNCIAS EXATAS - DCET

COLEGIADO DE CIÊNCIA DA COMPUTAÇÃO - COLCIC

Lista de Exercícios

Curso	Ciência da Computação
Disciplina	Linguagem de Programação III
Professor(es)	Otacílio José Pereira
Assunto	- Capítulo 2: Conceitos básicos de Orientação a Objetos - Lista 3 – Ênfase em pequenos cenários contextualizados

Foco

O foco desta lista é programar as classes e objetos para pequenos cenários contextualizados visando tanto ampliar a habilidade de programa Orientado a Objetos com os conceitos básicos que vimos, como vislumbrar a aplicação em cenários curiosos.

Perceba que é uma lista com ênfase diferente das outras, em outra por exemplo, as habilidades exploradas eram de conceitos e definições.

Exercício 1 – Candidato em concurso

Imagine que um concurso foi realizado com base em 3 provas cada uma com uma nota de 0 a 100. O candidato está classificado, isto é, ele só poderá ser convocado se obter nota superior a 50,0 em qualquer prova e que a média total seja superior a 60,0.

- Crie a classe Candidato com os atributos Nome e os valores das notas.
- Crie os construtores da classe, apenas com o Nome e daí as notas ficam zeradas ou com Nome e todas as três notas das provas.
- Crie um método VerificarClassificado que verifica se determinado candidato foi classificado no concurso.
- Crie um método getMedia que apresenta qual a média final do candidato.
- Na classe principal crie pelo menos três objetos candidatos com notas que permite ser aprovado, ser reprovado por conta de uma das notas ser abaixo de 50,0 e outro em que foi reprovado pela média total inferior a 60,0.

Exercício 2 – Itens em pedido de compra

Faça um programa que implemente a classe ItemCompra, ela tem a descrição do item, o valor unitário do item, a quantidade e o percentual de desconto. Com base nestes elementos é possível obter o valor total do item (qtde x valor unitário) e o desconto no item.

- Crie a classe ItemCompra com seus atributos, os métodos getters e setters e construtores (sem e com parâmetros).
- Crie o método ValorTotalItem que retorna o valor total do item
- Crie o método ValorDesconto que recupera qual o valor de desconto no item
- Na classe principal crie um item de compra, por exemplo, "TV 40\"", 2300, 2 e 20% e experimente os métodos.
- Ainda na classe principal, crie um vetor com 5 itens de compra e depois calcule o total da compra e o total do desconto.

Exercício 3 – Jogo da Velha

Crie um programa que codifique a classe Jogo da Velha.

- a) Crie a classe JogoDaVelha que representa o tabuleiro e as opções marcadas pelo jogador.
- b) Crie o método limpar tabuleiro que coloca o caractere “.” em todas as casas do tabuleiro.
- c) Crie o método Jogar que recebe a posição e o jogador e marca a jogada do jogador. Perceba que você precisa checar se a jogada é válida, se a casa está de fato desocupada.
- d) Crie o método ProximoJogador que indica quem é o próximo jogador a jogar. Perceba que você pode avaliar o próximo jogador olhando apenas para o tabuleiro ou pode criar um atributo para auxiliar neste controle e alternância.
- e) Crie o método AvaliarVencedor que é computador logo após realizar uma Jogada. O método verifica se a jogada culminou em um vencedor conforme as regras do Jogo da Velha.
- f) Na classe principal, crie o jogo da velha e realize algumas jogadas visualizando o tabuleiro em seguida.

Exercício 4: Passagens de ônibus.

Quando vamos comprar passagem em geral dizemos a origem, destino, data, hora e qual poltrona. Faça um programa que avalia as passagens vendidas de uma viagem.

- a) Crie a classe Viagem que possui Origem, Destino, Data, Hora e um mapa com todas as 44 poltronas do ônibus.
- b) Crie o método VenderPassagem para esta viagem que solicita o número da poltrona e a marca como vendida. Vale perceber que só pode ser vendida poltrona desocupada.
- c) Crie o método LiberarPassagem que libera uma poltrona vendida.
- d) Crie um método MostrarPoltronas que mostra o mapa das poltronas, sinalizando quais foram vendidas e quais estão vagas.
- e) Crie o método MostrarJanelasDireita que mostra todas as poltronas de janela à direita do ônibus. Faça o mesmo para janelas à esquerda e para corredores. Se quiser, podem pensar em um arranjo de métodos que facilite este tipo de observação das poltronas.
- f) Faça um método que calcule a taxa de ocupação do ônibus, isto é, qual percentual de poltronas vendidas.
- g) Na classe principal crie a classe Viagem e realize alguns testes para checar se os métodos estão funcionando adequadamente.

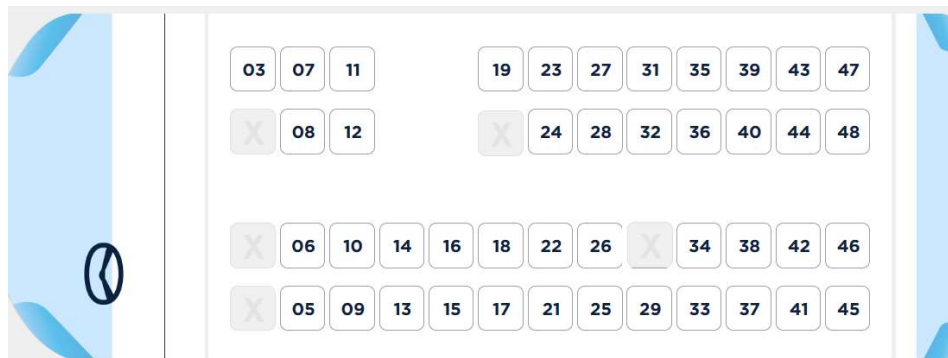
Dica: Veja figura ilustrativa para inspirar sua solução. Vale destacar, a figura está em formato Web/Gráfico mas para seu programa basta que seja um mapa na interface em texto.

09:30 → 17:40
previsão 8h 10m de viagem

Itabuna, BA - Rodoviária
Salvador, BA - Rodoviária

SEMILEITO

R\$ 186,87



Exercício 5 – Dados de operação de equipamentos na Agricultura 4.0

Em nossas primeiras aulas, quando revisamos a linguagem C, criamos um programa para um cenário de Agricultura 4.0 que acionava um equipamento, uma bomba. Imagine que no cenário gostaríamos de controlar quantas vezes o equipamento foi ligada e quanto tempo no total ele ficou ligado.

- Crie a classe equipamento que contém um número de identificação (ID), a descrição e o seu status (Ligado ou desligado). Crie todos os outros componentes da classe: construtores (com e sem parâmetros), getters e setters, impressão e outros que achar necessário.
- Crie o método ligar o equipamento que muda o status para Ligado e registra o instante em que o equipamento foi ligado. Perceba que para isso você precisará criar um atributo para guardar o instante em que o equipamento foi ligado.
- O método ligar também precisará contar quantas vezes o equipamento foi ligado, portanto um atributo de número de vezes em que foi ligado também deverá ser criado. E ele deve ser incrementado a cada operação de ligar.
- Crie o método desligar equipamento que muda o status do equipamento para desligado e além disso contabiliza quanto tempo o equipamento ficou ligado e ainda totaliza quanto tempo no total o equipamento foi ligado (para isso um atributo de tempo total ligado) precisa ser criado na classe.
- Na classe principal crie um equipamento, ligue e desligue por algumas vezes e depois imprima os valores verificando a contabilização de número de vezes ligado e o tempo total em que ficou ligado.

Dica: Para trabalhar com Data e Hora, você pode olhar artigos como este.

<https://mmarcosab.medium.com/usando-a-api-de-datas-do-java-da8b54a0dda2>

```
LocalDate hoje = LocalDate.now();  
System.out.println(hoje);
```

Exercício 6 – Perguntas em um questionário ou prova

Em um sistema de aplicação de provas, as questões objetivas podem ter o seguinte formato com número da questão, enunciado, alternativas e o gabarito da resposta certa.

Questão 10 – Na orientação a objetos os métodos responsáveis pela instanciação e inicialização de objetos são os:

- Criadores
- Getters

- c) Setters
- d) Construtores
- e) Impressores

Resposta dada: a)
Resposta certa: d)
Avaliação: ERRADA

Codifique as classes necessárias para implementar as questões conforme as orientações a seguir:

- a) Crie a classe Questao com os atributos necessários (número da questão, resposta dada, resposta certa, avaliação e a lista de alternativas). A lista de alternativas pode ser um vetor de Strings.
- b) Crie o método AlterarGabarito que altera a resposta correta.
- c) Crie o método responder que recebe a alternativa e indica se a avaliação é Correta ou Errada.
- d) Crie o método Embaralhar as alternativas que recebe um número e rotaciona as alternativas com base neste número. Por exemplo, para as alternativas a) Item 1, b) Item 2, c) Item 3, ... ao solicitar Embaralhar(1), o sistema mostra a) Item 2, b) Item 3, e) Item 1.
- e) Na classe principal crie uma questão, adicione as alternativas e realize outras operações para testa sua classe.

Exercício 7 – Transformações geométricas em Figuras 2D

Na computação gráfica é comum realizar algumas operações sobre figuras, por exemplo pode-se deslocar (translação) uma figura em um plano 2D ou rotacionar a figura. Faça um programa que implementa um triângulo com suas três coordenadas (X,Y) em um plano 2D e que permita realizar as operações em questão.

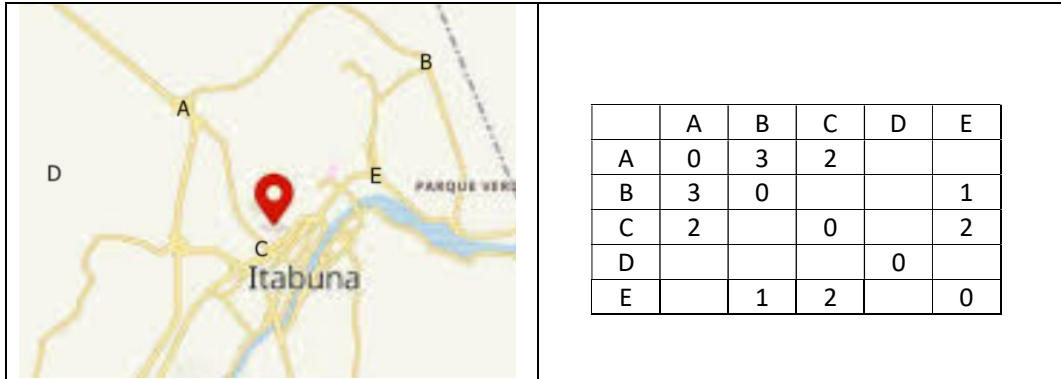
- a) Crie a classe ponto com suas coordenadas (X, Y). Crie todos os elementos da classe, os atributos, construtores, getters e setters e impressão.
- b) Crie a classe triângulo com suas coordenadas. Faça a opção que preferir para armazenar os três pontos (três atributos diferentes, um vetor de pontos ou outros).
- c) Crie os construtores de triângulo que recebe as coordenadas dos três pontos ou então recebe três objetos da classe ponto.
- d) Crie o método DeslocarX que altera as coordenadas X dos pontos de um deslocamento D. Por exemplo, de os pontos do triângulo são (2, 3), (5, 4) e (4, 6), com um deslocamento em X de D=4, eles passam para (6,3), (9,4) e (8,6).
- e) Crie um método que realize o deslocamento no eixo Y, de forma semelhante ao que foi feito no eixo X.
- f) Crie um método que faça o deslocamento conjunto, ele recebe os valores dos deslocamentos Dx e Dy e realiza dos dois deslocamentos.
- g) (Opcional) Caso tenha gostado do cenário, pesquise como é rotacionar uma figura em um plano 2D. Daí implemente um método que recebe um ângulo e rotaciona a figura com este ângulo e tendo como base/apoio o primeiro ponto.
- h) Na classe principal crie um triângulo e experimente ou teste as transformações no triângulo.

Dica: Se quiser saber mais sobre estas transformações em figuras, seguem alguns links (lembrando que nosso foco é na programação orientada a objetos, usamos o cenário contextualizado só para tornar o exercício mais curioso):

<https://www.todamateria.com.br/transformacoes-geometricas/>
<https://www.youtube.com/watch?v=OWMPVzuAv28>

Exercício 8 – Analisando rotas em GPS

A navegação por determinado mapa da cidade pode ser mapeada em pontos e representada por uma estrutura chamada de grafos que podem ser implementados por matriz. Basicamente o que ele indica é se há um caminho que liga um ponto A a um ponto B e em alguns casos com qual distância ou custo. Veja exemplo a seguir. De cada ponto para ele mesmo a distância é zero, de A para C ou de C para A, a distância é de 2. O ponto D não está conectado a ninguém e as casas vazias podem estar com -1 indicando que não há conexão direta.



- Crie a classe mapa que apresenta os 5 pontos e inicialmente ninguém está conectado com ninguém (células preenchidas com -1).
- Crie o método conectar, que recebe um número de origem, um número de destino e uma distância e atualiza as informações no mapa/gráfo. As vias são de mão dupla, por isso deve-se marcar tanto o caminho da origem para o destino como do destino para a origem.
- Crie o método HáCaminhoDireto que recebe origem e destino e retorna se há conexão e com qual distância.
- Crie o método que verifica se HáCaminho2Passos que verifica se há caminho de uma origem até um destino passando por apenas um local intermediário.
- Na classe principal crie o mapa e realize algumas experiências com ele, adicionando caminhos e consultando.
- (Opcional) Se quiser, realize uma pesquisa por algoritmo de encontrar caminhos em grafos e tente implementar sua versão (mas atenção, esta é uma tarefa que creio será bem difícil para o nível em que estamos, fica só como um desafio para aquele aluno mais curioso).