Departamento de Ciência da Computação – DCC – UFJF

Professor: Gleiph Ghiotto Lima de Menezes Disciplina: Orientação a objetos

Nome:_____ Matrícula:_____

Primeira prova

Orientações gerais

- A prova é individual e provas semelhantes receberão nota zero;
- Colocar nome e matrícula em todos os arquivos pedidos, a não observação desse requisito pode acarretar nota zero;
- As questões devem ter o código compilando e executando utilizando o Maven. Caso contrário, receberão nota zero;
- Sugiro utilizar os projetos em anexo no Classroom; e
- Leia atentamente a prova e boa prova!

Questão 1 (20 pontos)

Enunciado

Um hospital deseja criar um sistema para controlar os deslocamentos de suas ambulâncias e verificar se as rotas indicadas são as mais otimizadas. Neste sistema, as rotas são dadas através dos caracteres N, S, L e O que representam respectivamente os deslocamentos de uma unidade para o Norte, Sul, Leste e Oeste representada pela Figura 1.

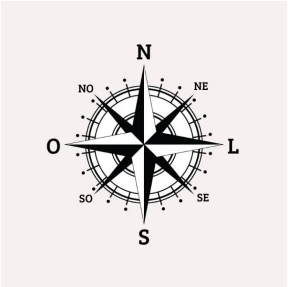

Figura 1 - Rosa dos ventos com as coordenadas N, S, L e O.

Todas as ambulâncias saem do hospital que é representado por um ponto HO na posição (0, 0) de um plano cartesiano e percorre por uma rota composta por um ou mais pontos de parada, representados pela letra P, antes de voltar para o hospital. Por exemplo, a rota "NNLLOPLLSP" faz uma ambulância sair do hospital, ponto HO (0, 0), e passar pelos pontos P1 (1, 2) e P2 (3, 1) conforme representados na Figura 2.

Um problema encontrado pelo hospital é que as rotas podem ser mais longas do que deveriam. Por exemplo, o trecho "NNLLOP" da rota poderia ser otimizado para "NNLP", "NLNP" ou "LNNP" dado que um deslocamento para leste(L) é anulado por outro deslocamento para oeste(O). Para auxiliar na gestão financeira do hospital, o sistema deve ser capaz de identificar a melhor rota e comparar a quantidade de deslocamentos das rotas. Por exemplo, para a rota original "NNLLOPLLSP" e a rota otimizada "NNLPLLSP" a quantidade de deslocamentos seriam 8 e 6, respectivamente, e o hospital teria uma economia de 2 deslocamentos que podem significar economia financeira e a redução de tempo para atender novos chamados do hospital.

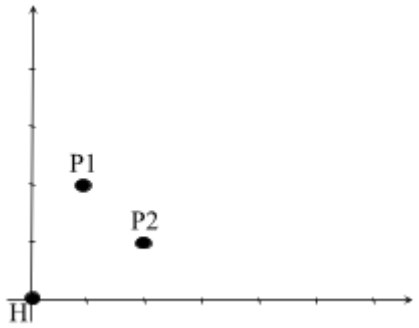

Figura 2 - Deslocamentos da rota "NNLLOPLLSP".

Implementação
Implemente um projeto em Java composto pelas seguintes classes:

Classe Ponto (4 pontos)
Crie uma classe ponto que representa um ponto no plano cartesiano (x, y) com os atributos x e y que não podem ser modificados após a inicialização, um construtor que receba os valores de x e y, e os métodos de acesso necessários.

Classe Rota
Classe que possui:

- um atributo *coordenadas* que armazena os deslocamentos de uma rota; **(1 ponto)**
- um método *ehValida* sem parâmetros que retorna *true* quando uma rota é válida e *false* quando a rota é inválida. Uma rota é inválida quando não possui nenhum deslocamento OU possui pelo menos um símbolo diferente de N, S, L, O, ou P OU possui dois P's seguidos; **(2 pontos)**
- um método *rotaOtimizada* sem parâmetros que retorna a rota com o menor deslocamento possível ou retorna uma rota com a coordenada vazia quando a rota é inválida; **(2 pontos)**

- um método *deslocamentoTotal* sem parâmetros que retorna o deslocamento total de uma rota. Por exemplo, para a rota "NNLLOPLLSP" o deslocamento é igual a 8 por ser válida e possuir 8 caracteres N, S, L ou O;**(2 pontos)**
- um método *extraiPontos* que retorna uma lista de Pontos de uma rota. Caso a rota for inválida, retorne uma lista vazia. Por exemplo, a rota "NNLLOPLLSP" retorna os pontos (1, 2) e (3, 1); **(2 pontos)** e
- os métodos de acesso necessários.**(1 ponto)**

Classe Mapa (3 pontos)
Classe que possui um método *imprime* que apresenta o hospital e os pontos visitados em um plano cartesiano. Considere que os pontos não podem se sobrepor e que os limites do mapa devem ser adaptados para cada conjunto de pontos.

Classe Prova1Questao1 (3 pontos)
Crie uma classe Prova1Questao1 com um método main que leia uma rota. Se a rota for válida imprima os pontos da rota, o mapa com as coordenadas e a rota original com o seu deslocamento e a rota otimizada com o seu deslocamento. Caso a rota for inválida, imprima a mensagem "Rota invalida!".

OBS1: considere que a rota pode ser escrita com letras maiúsculas ou minúsculas. Sugestão: utilize toUpperCase.

OBS2: o aluno pode criar métodos auxiliares durante as implementações.

Exemplo de Execução: Rota válida

```
Digite uma rota para a ambulancia
NLNLNLNLNNLNLPSSSPSSSNNLP
Pontos da rota:
P0 - (6, 7)
P1 - (6, 4)
P2 - (7, 3)
 **  **  **  **  **  **  P0  **
 **  **  **  **  **  **  **  **
 **  **  **  **  **  **  **  **
 **  **  **  **  **  **  P1  **
 **  **  **  **  **  **  **  P2
 **  **  **  **  **  **  **  **
 **  **  **  **  **  **  **  **
 HO  **  **  **  **  **  **  **
Rota original:   NLNLNLNLNNLNLPSSSPSSSNNLP - Deslocamento: 22
Rota otimizada:  NNNNNNNLLLLLLPSSSPSLP - Deslocamento: 18
```

Exemplo de Execução: Rota inválida

```
Digite uma rota para a ambulancia
NNNSSPAVP
Rota invalida!
```

Questão 2 (5 pontos)
Enunciado

Um escritor deseja utilizar um programa para contar a quantidade de cada caractere que ocorre no texto. Por coincidência, ele estudava Orientação a Objetos e resolveu utilizar Map para solucionar o seu problema.

Crie uma classe Prova1Questao2 com um método que leia um texto e imprima os caracteres presentes no texto com a quantidade de ocorrências de cada um deles. Diferencie maiúsculas e minúsculas.

OBS1: É obrigatório o uso de Map e HashMap.

Exemplo de execução

```
Digite um texto de apenas uma linha:
A prova de Orientacao a Objetos eh hoje.
  ===> 7
A ===> 1
a ===> 4
b ===> 1
c ===> 1
d ===> 1
e ===> 5
h ===> 2
i ===> 1
j ===> 2
n ===> 1
. ===> 1
o ===> 4
O ===> 2
p ===> 1
r ===> 2
s ===> 1
t ===> 2
v ===> 1
```