Engenharia de Telecomunicações
POO29004 – Programação Orientada a Objetos

13/03/2020

Lista 4: Construtor, modificadores de acesso e palavras reservadas

Professor: Emerson Ribeiro de Mello

http://docente.ifsc.edu.br/mello/poo

Nota:

- *Antes de iniciar, sincronize seu repositório local com o seu repositório remoto "Listas de exercícios" do Github. É possível que o professor tenha feito um commit por lá após a entrega da lista de exercício anterior;*
- *Crie um diretório com o nome* `lista-04` *na raiz do repositório e dentro deste, crie um projeto Java+gradle com o IntelliJ para cada um dos exercícios abaixo;*
- *Edite o arquivo* `Readme.md` *que está na raiz do repositório e adicione um novo item com o nome* `lista-04` *e este deverá ter um link o qual deverá apontar para o subdiretório com o nome* `lista-04`*.*

1 Retângulo em ASCII e UTF8

- Crie uma classe para representar um retângulo
  - Método construtor que permita definir a altura e largura
  - Método construtor padrão de forma que altura e largura sejam igual a 4
  - Métodos para alterar a largura e altura
  - A classe deverá garantir que em sua altura e largura sejam atribuídos somente valores que realmente possam representar um retângulo
  - Métodos chamados `desenharASCII()` e `desenharUTF8()`
    - Ex: Retângulo com 3 de altura e 4 de largura em ASCII

      ```
      +--+
      |  |
      +--+
      ```

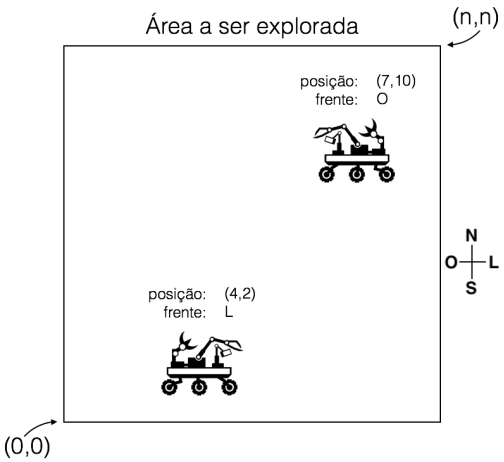
    - Códigos UTF8 para desenhar o mesmo retângulo que fora apresentado acima em ASCII:

      ```
      1 System.out.println("\u250c\u2500\u2500\u2510\n\u2502  \u2502\n\u2514\u2500\
            u2500\u2518");
      ```

- Faça um aplicativo Java que receba três argumentos de linha de comando, sendo: modo, altura e largura; e imprima na tela um retângulo de acordo com os argumentos fornecidos. O argumento **modo** poderá ser `ascii` ou `utf8`. A altura e largura deverão ser valores positivos e válidos para construção de retângulos. Se o usuário fornecer qualquer argumento inválido, ou mesmo fornecer um número diferente de 3 argumentos, deve-se apresentar uma mensagem de ajuda.

2 Robô de exploração com controle remoto

- Crie uma classe para representar um robô de exploração espacial. Ao instanciar um objeto dessa classe, deve-se obrigatoriamente indicar o tamanho da áre a a ser explorada (sempre será um quadrado), a coordenada inicial (x,y) do robô dentro dessa área e para onde está apontando sua frente (N, S, L ou O). Ou seja, todo robô, ao ser criado, já saberá a área de exploração e suas coordenadas dentro dessa área.
- Crie métodos para:
  - **Carregar o plano de exploração**, que deve ser uma `String` ou um vetor de `String`. O plano deverá ser composto somente pelos comandos: `E`, `D` ou `M` e deverá ter no mínimo 1 e no máximo 100 comandos.
    - E - girar no próprio eixo para esquerda
    - D - girar no próprio eixo para direita
    - M - mover 1 coordenada para frente
  - **Para iniciar a exploração**. Uma vez iniciada, deve-se registrar a movimentação do robô passo-a-passo. Ou seja, deve-se registrar as coordenadas atuais, o comando que foi processado e as novas coordenadas. Quando o robô terminar a exploração, esse deverá retornar esse registro de movimentações.
    - O robô só poderá explorar se houver plano carregado.
    - Uma vez que o plano tenha sido executado, o mesmo será excluído
    - O robô não poderá ultrapassar os limites da área de exploração. Ou seja, o robô deverá ignorar todo comando que o faria ultrapassar os limites da área de exploração
  - **Para indicar suas coordenadas atuais**.



- Entrada:
  - área, x, y, frente: 8 1 2 N
  - Plano: EMEMEMEMM
- Saída: 1 3 N

Faça um aplicativo Java que receba um arquivo texto como entrada (use o redirecionamento de entrada do terminal <), contendo todas informações para que o robô possa fazer a exploração da área e imprima o registro da movimentação feita pela robô. Abaixo o formato obrigatório para o arquivo:

```
area, x, y, frente
COMANDO
COMANDO
COMANDO
...
COMANDO
```

```
8, 1, 2, N
E
M
E
M
E
```

3 Soldados

- Implemente uma classe para representar um soldado em um jogo e esse soldado poderá se movimentar ou atacar
- Ao movimentar é possível indicar a distância do deslocamento, porém se o usuário não fornecer, então o soldado deverá se deslocar uma distância padrão
- Ao atacar deve-se informar com qual arma (i.e. fuzil, baioneta, punho), se não for informada, então atacar com o fuzil
- Ao invocar o método atacar, o soldado só atacará se houver pelo menos 10 soldados instanciados.
- Se o número de soldados for inferior a 3, o soldado dirá: ainda não, esperando o exército ficar maior. Ou seja, cada soldado deverá ter ciência do total de soldados que existem.
- **Obrigatório fazer uso de this, static e final**.
- **Não pode ter constantes literais espalhadas pelo código.**

Faça um aplicativo Java, com um menu de interação com o usuário, que permita ao usuário: criar soldado, verificar total de soldados, movimentar um soldado, movimentar todos soldados, atacar com todos os soldados e finalizar o aplicativo. A solução deverá permitir criar no máximo 10 soldados e deverá exibir mensagens de ajuda, caso o usuário tente executar uma operação não permitida, por exemplo, usuário indica que quer atacar, porém não foi criado nenhum soldado ainda.

Atenção:

- ***Para receber nota 10:*** *Os commits para essa entrega deverão ser encaminhados (push) para o Github Classroom até o dia 18/03/2020;*
- ***Para receber nota 5:*** *Os commits deverão ser encaminhados (push) para o Github Classroom até o dia 19/03/2020;*
- *Qualquer entrega após o dia **19/03/2020** não será contabilizada e receberá nota 0.*