

Lista de Exercícios 3

Programação Orientada a Objetos

SCC-0504

Prof. Luiz Eduardo V. Silva / Estevam Arantes (Monitor)

Aconselha-se que o aluno faça o exercício de teste para a familiarização do run.codes para testes automatizados em java.

Exercício 2

Pense em uma classe Stack (pilha) de inteiros. Implemente a classe e suas três principais operações (push, pop e top).

Push é um método que deve lançar uma exceção do tipo “FullStackException” sempre que não couber elementos na pilha (considere uma pilha de tamanho máximo fixo igual a 5).

Pop e Top devem lançar uma exceção “EmptyStackException” caso a pilha não tenha mais elementos para serem retirados ou impressos respectivamente. Ambas exceções são herdeiras da exceção "StackException". Implemente todas as 3 classes de exceção e os métodos que as lançam. Implemente também uma classe Teste que mostra como criar a pilha e invocar os métodos push, pop e top com os respectivos tratamentos de exceções seguindo o padrão abaixo.

O **input** será dado da forma: Na primeira linha, um inteiro N indicando o número de operações feitas na pilha. Nas N linhas seguintes terão instruções do seguinte formato:

- I x - Insere o inteiro x no topo da pilha.
- R - Remove o elemento que está no topo da pilha.
- P - Imprime, em uma nova linha, o elemento que está no topo da pilha.

Caso alguma das exceções ocorra, imprima *Erro!* na mensagem da classe "StackException" e uma mensagem adicional caracterizando o tipo da exceção ocorrida:

- Caso seja do tipo uma remoção inválida, imprima *Remocao Invalida!*
- Caso seja do tipo a pilha esteja cheia imprima *Pilha cheia!*
- Caso a pilha esteja vazia e peça para imprimir um elemento, imprima *Operacao Invalida!*

E, no caso de Remoção ou Inserção inválidas, finalize a execução do programa logo após as mensagens de erro, ignorando instruções seguintes.

Obs: Para facilitar a leitura da entrada é aconselhável que se utilize o código disponibilizado no Exercício 2 da Lista 2 como base, com alterações necessárias, porém isso não é obrigatório.

Input 1

8
P
I 10
P
I 4
I 5
R
P
I 12

Output 1

Erro !
Operacao Invalida !
10
4

Input 2

6
I 3
P
R
R
I 5
P

Output 2

3
Erro !
Remocao invalida !

Explicação dos casos de teste:

No primeiro caso, tínhamos a pilha vazia na primeira instrução de impressão, então lançamos uma exceção com a mensagem "Operacao Invalida", que não interrompe a execução do programa. Logo, colocamos 10 na pilha e imprimimos o 10. Então, colocamos 4 e 5, porém retiramos o elemento do topo dela (que seria o 5), obtendo então 4 como o novo topo. Em seguida, colocamos 12 na pilha e finalizamos a execução do programa.

No segundo caso, inserimos 3 na pilha, o imprimimos e depois o removemos. Então, na quarta instrução tenta-se remover o topo da pilha, que não existe, lançando uma exceção do tipo `EmptyStackException`, imprimindo a mensagem de erro e finalizando a execução do programa.

Observações:

Data limite para entrega: 30/05/2018.

Local de entrega: Run Codes (Lista3 - Exercício 2).

Forma de entrega: Códigos diretamente no Run.Codes para a correção automática.

Caso existam dúvidas favor contactar o monitor.

A discussão de ideias entre colegas é estimulada, porém a solução das listas é individual. A identificação de plágio acarretará em nota zero para todos os envolvidos.