

EXERCÍCIO 4

Escreva dois programas em Java que forcem de maneiras distintas a ocorrência do erro `java.lang.StackOverflowError`. Explique as estratégias utilizadas em cada um dos programas, incluindo uma descrição detalhada do que causou o erro.

Resposta:

As duas maneiras distintas que pensamos passíveis de gerar o erro de *Stack Overflow* foram:

- 1) Um método recursivo chama a si mesmo sem ter uma condição de parada, gerando, assim, um loop “infinito” e enchendo a *stack*. O método implementado no código referente a esse caso é “Forma1()”.
- 2) Vários métodos distintos são empilhados na *stack* até que esta fique cheia. Isso ocorreria, por exemplo, se tivéssemos n métodos m_i (com $i = 1, 2, 3, \dots, n$) com a seguinte lógica de funcionamento: m_1 chama m_2 que chama m_3 que chama m_4 que chama ... que chama m_{n-1} que chama m_n . Dessa forma, quando m_{n-1} chamasse m_n a *stack* já estaria cheia e teríamos o erro de *Stack Overflow*. Como é inviável criar manualmente uma quantidade suficiente de métodos para que consigamos encher a *stack*, utilizamos a simplificação de criar os métodos “Forma2()”, “a()” e “b()” para simularmos essa situação, sendo que “Forma2()” chama “a ()” que chama “b ()” que chama “Forma2()” novamente (isso ocorre para podermos gerar o erro solicitado).