

Exceções

Lista 06

1 de junho de 2021

1. (2.5 ponto) Crie uma classe Primeira exceção, com um método `geraPau()` que sempre gera uma exceção `ArrayIndexOutOfBoundsException`. Fazer um outro programa `pegaPau()` que captura de forma adequada (`catch`) essa exceção. Fazer uma classe de testes que testa esses dois métodos.
2. (2.5 pontos) Modifique o Exercício 01 da Lista 03, de forma que um depósito só possa ser feito se o valor depositado é estritamente positivo, e os saques só podem ser feitos se o valor sacado for positivo e o saldo existente permitir; além disso, um depósito só pode ser feito se o valor a ser depositado é positivo. Crie uma classe de exceções `NegativeException` derivada da classe `RuntimeException` e faça com que seu programa gere uma exceção para cada um dos casos de erros mencionados acima. Use o `finally` para garantir que o saldo nunca é negativo; caso contrário, uma mensagem de erro deve ser impressa.

Gere testes automáticos que verifique que exceções são geradas em casos de depósito ou saldo de valores negativos.
3. (1.5 pontos) Modifique o exemplo anterior para que a classe `NegativeException2` seja derivada da classe `Exception`. Gere testes automáticos que verifique que exceções são geradas em casos de depósito ou saldo de valores negativos e que mostre a diferença entre estes dois tipos de Exceções.
4. (3.5 pontos sofridos) Suponha que `v` esteja escrevendo um método que recebe um string:

```
void recebeCodigo(String codigo){ ... }
```

onde `codigo` deve ter o seguinte formato

```
matriz <tipo> <num linhas> <num colunas>
```

onde `matriz` é uma palavra obrigatória no início do código, o tipo pode ser um de `0-1`, `int`, ou `double`, e `num linhas` e `num colunas` são inteiros positivos não nulos.

Crie exceções para os diversos erros que podem ocorrer na decodificação do código no formato acima. Crie um programa JUnit que verifique que as exceções são jogadas nos casos em que o método recebe um string que não obedece ao formato acima.