

Exercício 1



André Souto/Isabel Nunes

Unidade Curricular de
Laboratórios de Programação

2021/2022

Exercício : *Debug* e Exceções

Objetivos

- Familiarização com o depurador de erros do Eclipse;
- Consolidação dos conhecimentos adquiridos sobre Exceções;

Antes de Começar

De modo a poder realizar este exercício deverá ter lidos os guiões referentes ao *debugger* e às exceções.

Crie um projeto java no Eclipse e “arraste” para a pasta `src` desse projeto os ficheiros contidos no ficheiro `studentsExercisel.zip`, acessível na página de LabP.

O que fazer

Deverá programar as soluções para as questões que se seguem.

Debugger

Usando como base o ficheiro `BugPermuta.java` fornecido e usando os conhecimentos adquiridos com o guião de Depuração no Eclipse, elimine todos os erros que encontrar no programa.

Espera-se que, com todas as correções feitas, no fim imprima uma permutação de um vetor que é lido.

Questões sobre Exceções

Para este exercício vai-se considerar a classe `ExercicioExcecoes` fornecida. O método `main` desta classe recebe o nome de um ficheiro de texto como argumento (em `args[0]`). Mais adiante neste texto explicamos como dar valores a esse parâmetro.

A ideia é criar vários ficheiros de texto de *input*, para testar este programa (ou ir alterando os valores num mesmo ficheiro de texto). Relembre no Tutorial do Eclipse como criar ficheiros de texto facilmente (Página 11, do 4º ao penúltimo parágrafo).

Para que o programa execute sem exceções, a primeira linha do ficheiro de texto de *input* deve conter o número de inteiros a ler (chamemos-lhe *n*). A segunda linha deve conter *n* inteiros separados por um espaço. A terceira linha deve conter um valor *double* que corresponde ao índice da potência com o qual o programa vai trabalhar.

Para indicar ao Eclipse qual o valor para o argumento `args[0]`, deve, na janela do *Package Explorer*, clicar com o botão direito do rato na classe `ExercicioExcecoes` e escolher *Run as* → *Run Configurations*. Na janela de diálogo que aparece, deve clicar 2 vezes na opção *Java Application*. À direita, deve escolher a aba (x)=*Arguments*, e escrever o nome do ficheiro de texto que quer usar na caixa de texto *Program arguments*. Finalmente deve clicar no botão *Run*.

Nas próximas alíneas **não (!) deve assumir que o input está sempre bem construído**. Deve testar a execução do programa **com diferentes ficheiros de input** de forma a que as várias exceções em causa sejam forçosamente lançadas.

1. Complete o método `lerTamanhoVetor` de forma a que lance exceções dos seguintes tipos, **mas não as trate (!)**:
 - a. `IOException`, quando não existem linhas no ficheiro (pode usar o método `hasNextLine` da classe `Scanner`);
 - b. `InputMismatchException`, quando a primeira linha do ficheiro não corresponde a um inteiro positivo.

Deve associar às exceções uma mensagem adequada.

2. Complete o método `lerInteirosVetor` de forma a que lance exceções dos seguintes tipos, **mas não as trate (!)**:
 - a. `IOException`, quando não existem mais linhas no ficheiro (pode usar o método `hasNextLine` da classe `Scanner`);
 - b. `InputMismatchException`, quando o valor do parâmetro `tamanho` não é igual ao número de inteiros na segunda linha.

Deve associar às exceções uma mensagem adequada.

3. Complete o método `calculaPotencias` de forma a que:
 - a. Construa e devolva um vetor cujos elementos correspondem aos elementos do vetor `inteiros` elevados ao valor `potencia`;
 - b. Lance exceção do tipo `ArithmeticException`, **mas não a trate (!)**, quando o `Double` resultante é NaN (Not a Number) que acontece por exemplo quando tenta calcular a raiz quadrada (isto é com `potencia = 0.5`) de um número negativo. Consulte a classe `Double` para saber como testar a condição. Deve associar uma mensagem adequada à exceção.

4. Complete o método `main` de forma que:

- a. Se o valor obtido como `potencia` for negativo este lance uma exceção do tipo `IllegalArgumentException`.
- b. Trate as possíveis exceções lançadas por ele próprio ou pelos métodos `lerTamanhoVetor`, `lerInteirosVetor` e `calculaPotencias` que foram mencionadas anteriormente.
- c. Deve tratar a exceção do tipo `NumberFormatException` que pode ser lançada pelo método `parseInt` quando a *String* lida não pode ser convertida para um valor numérico. Neste caso, deve imprimir uma mensagem de erro, bem como o traço da pilha que leva ao lançamento da exceção (veja o método `printStackTrace()` das exceções).
- d. Altere o método de forma a tratar a exceção `FileNotFoundException` caso o ficheiro de *input* não exista, associando uma mensagem adequada.

Entrega

Deve entregar o ficheiro `Elfcxxxxx.zip` (em que `xxxxx` é o seu número de aluno) contendo os ficheiros `ExercicioExcecoes.java` e `BugPermuta.java` já alterados.