

## Guia Prático 4 Excepções e Ficheiros

João Paulo Barros

18 de abril de 2022

### 1 Excepções

As excepções em Java™ são uma forma alternativa de um método terminar e devolver dados. Tal com o nome indica, devem ser utilizadas para casos excepcionais, ou seja, para situações que não ocorrem normalmente. Tipicamente, tais situações correspondem à ocorrência de erros do tipo "falta um ficheiro", "não consigo contactar o servidor", etc.. Ou seja, situações que **podem e é acitável que ocorram**, mas que não são as mais prováveis; por outras palavras, não são as "normais", mas também não são para tratar erros no código. Estes devem ser corrigidos e não devem ser tratados como excepções. Devem apenas ser removidos.

Quando um método termina com uma excepção tal implica duas coisas:

1. *Lança* um objecto da classe `Exception` ou que dela herde. Para tal utiliza a seguinte sintaxe: `throw new OurException("something bad");`.
2. Executa o código especificado no bloco **finally**, se existir.

Note que a palavra reservada **throw** tem uma semântica parecida à da palavra **return**, pois também termina o método enviando para fora um valor. A grande diferença está do lado de quem recebe. O método que chamou aquele que lança a excepção, não é obrigado a recebê-la para que o valor não desapareça: pode apanhá-lo (**catch**) e fazer algo com ele, OU pode passá-lo para o método que o chamou adicionando no final do seu cabeçalho as palavras (**throws OurException**) um pouco como o tipo que o método retorna (com o **return**) e que se coloca antes do nome do método. Note o **s** no final do **throws**. Por isso, os métodos "lançam excepções". O método que chamou aquele que lança a excepção ou o método que chamou o que chamou aquele que lança a excepção que a apanhe. Na verdade, qualquer método que tenha directa ou indirectamente chamado o método que lança a excepção (ou seja, um que esteja na *call stack*) pode apanhá-la e tratá-la. Se ninguém a apanhar, surgem as famosas mensagens de erro na consola, as quais já deve ser encontrado.

## 2 Exercício — Signos e Horóscopos

Considere agora o programa em <https://cms.ipbeja.pt/mod/resource/view.php?id=222394>. Este código lê um ficheiro de texto linha a linha e mostra-o numa caixa de diálogo. Note a utilização de excepções. São uma presença constante quando processamos ficheiros com Java™.

Com base nesse programa crie um gerador de horóscopos no estilo dos que facilmente encontramos em muitos jornais e revistas. Com um pouco de imaginação e inspiração, ou simplesmente escolhendo bem as frases que programa irá utilizar, este poderá gerar previsões tão boas como outras para qualquer signo e semana! Basta seguir as seguintes instruções:

**Base de dados** A base de dados do programa é constituída por um simples ficheiro de texto com uma frase por linha. Quando necessitar de referir o nome do signo deve utilizar as letras "SIGNO" em maiúsculas. Por exemplo:

Tenha cuidado com as falsas amizades.  
Os nativos de SIGNO são tímidos, mas amigos dos seus amigos.  
Bom período para investir, mas cuidado com as falsas promessas.  
Aproveite as boas oportunidades. Não se irá arrepender.  
Sorte ao amor.  
Cuidado com as apostas arriscadas.

Ou num estilo diferente...

Os nativos de SIGNO são falsos e perigosos.  
Os nativos deste signo gostam de humilhar os amigos.  
Nunca deve pedir dinheiro emprestado a um nativo de SIGNO.  
Os nativos de SIGNO são pouco inteligentes e nada cativantes.  
Se tem algum amigo do signo SIGNO arranje um cão.  
Infelizmente, a verdade é que todas as pessoas deste signo são insuportáveis.  
Estas pessoas são simplesmente más e perversas.

Naturalmente pode criar vários ficheiros de texto, e cada um corresponderá a uma base de dados que permite gerar previsões para um dado signo.

**Classe Horoscope** Crie uma cópia do projecto *MinimalFileReader* e dê-lhe o nome de *HoroscopeCreator*. Mude também o nome da classe principal Reader para Horoscope.

**Atributos da classe Horoscope** Defina os seguintes atributos na classe Horoscope: (1) um quadro (da classe, ou seja, static) de Strings denominado SIGN\_NAMES contendo os nomes dos signos; (2) uma lista (List) de *strings*, denominada sentences, para guardar as frases lidas do ficheiro de texto.

**Construtor** O construtor recebe o nome do ficheiro com as frases e lê-as para a lista sentences.

**Leitura** A leitura das frases deve utilizar um objecto da classe Scanner para colocar cada uma das linhas numa nova posição da lista sentences, ou o método readAllLines da class java.nio.file.Files

**Funcionamento** O programa começa por perguntar ao utilizador qual o ficheiro (a base de dados) que quer utilizar. Depois cria um objecto da classe Horoscope passando esse nome de ficheiro por parâmetro. Finalmente, chama um método **public void askSigns()** que pergunta o signo cuja previsão o

utilizador quer ver. A pergunta é feita utilizando a lista de nomes de signos e uma caixa de diálogo. Pode consultar um pequeno exemplo em <http://code.makery.ch/blog/javafx-dialogs-official/>. Seguidamente, gera-se uma lista de comprimento aleatório entre 3 e 6 com base nas frases já lidas. Note que não podem existir frases repetidas.

**Repetição** Depois de mostrar a previsão para o signo especificado, o programa deve perguntar se o utilizador deseja ver outra. Se sim, repete a criação de uma previsão a partir da base de dados já indicada. Se não deseja ver outra previsão, o programa termina.

**Escrita** Adicione a possibilidade de escrever o conteúdo da caixa de diálogo (a previsão) para um ficheiro com o nome <signo>.txt em que <signo> é o nome do signo escolhido pelo utilizador. Para implementar este requisito, consulte o código que exemplifica operações sobre ficheiros, disponibilizado na página da unidade curricular.

**Completo** Adicione uma opção que permite escrever num ficheiro de texto, as previsões para todos os signos.

**Com menu** Nesta variante o programa tem uma janela. Esta janela tem um menu "Generate" com o um submenu "Sign". Este submenu contém um item para cada signo. Veja a lista dos signos ocidentais em [https://en.wikipedia.org/wiki/Astrological\\_sign](https://en.wikipedia.org/wiki/Astrological_sign) (Aries, Taurus, Gemini, ...). Quando o utilizador escolhe um dos signos, o texto gerado surge na própria janela, por baixo do menu, dentro de um `javafx.scene.control.TextArea`. Note que não necessita que o texto seja editável. Procure como tornar o texto não editável.

**Completo com menu** Adicione uma opção que permite escrever na janela, as previsões para todos os signos. Tal deve ser feito escolhendo um item de menu com o nome "All" dentro do menu "Generate".

Para saber mais sobre estas coisas consulte, por exemplo, a entrada sobre astrologia, <http://skepdic.com/astrology.html> no "The Skeptic's Dictionary" em <http://skepdic.com/>.

*Deve continuar a resolução deste guia fora das aulas. Traga as dúvidas para a próxima aulas ou coloque-as no fórum de dúvidas da disciplina. As sugestões para melhorar este texto também são bem-vindas.*