



INSTITUTO SUPERIOR POLITECNICO DE TECNOLOGIAS E CIENCIAS

**1º Semestre**

2025/26

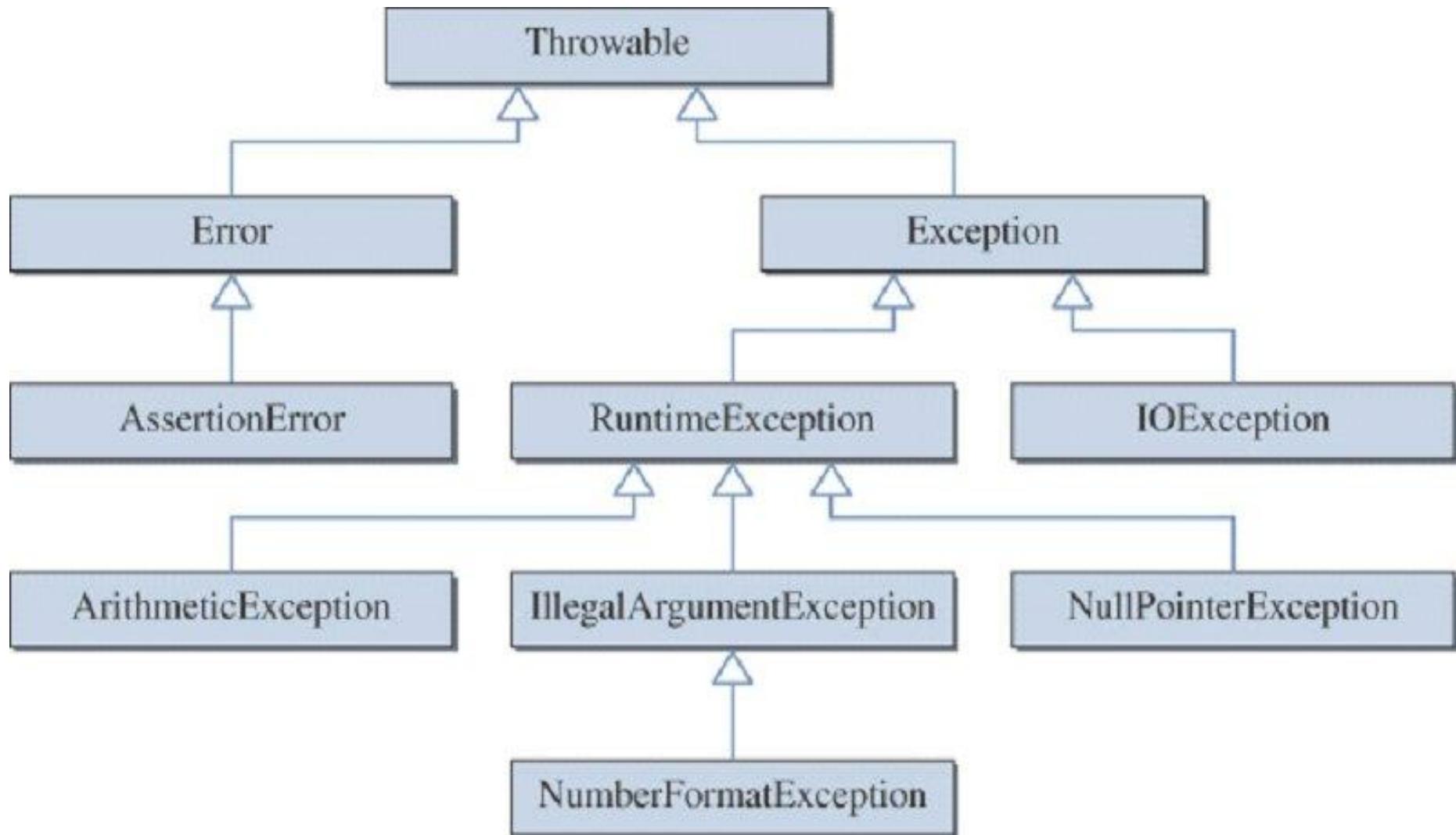


➤ Excepções

- O que é uma excepção
- Erro e excepção em java
- O que origina uma excepção
- Vantagens do uso de excepções
- Tratamento de excepções
- Classificação das Excepções
- Excepções mais comuns
- Exercícios de Fixação

## O que é uma excepção

- É um evento que ocorre durante a execução de um programa que interfere no fluxo normal das instruções deste programa.
- Em Java, a ocorrência de erros durante a execução de uma programa não significa necessariamente que o programa termine.
- A linguagem java possui um mecanismo para sinalizar partes críticas num programa e recuperar eventuais erros ocorridos nestas partes, sem parar a execução do programa. Este mecanismo é designado por **Excepção(Exception)**
- Uma Excepção é um sinal gerado pela máquina virtual de Java em tempo de execução do programa, que é comunicado ao programa indicando a ocorrência de **um erro recuperável**.



- Excepções

- O que é uma excepção
- Erro e excepção em java
- O que origina uma excepção
- Vantagens do uso de excepções
- Tratamento de excepções
- Classificação das Excepções
- Excepções mais comuns
- Exercícios de Fixação

### Erro

- Um Erro(**Error**) em Java corresponde a uma situação para a qual nenhuma recuperação é possível.
- Descreve erros internos e a exaustão de recursos durante a execução do programa
- Pouco se pode fazer se um erro interno desses ocorrer, além de notificar o utilizador e tentar finalizar o programa **adequadamente**.
- Essas situações são bastante raras.

### Excepção

- Uma Excepção(**Exception**) corresponde a uma situação para a qual a recuperação é possível.
- É um sinal gerado(lançado) pela máquina virtual de Java em **runtime**, indicando a ocorrência de um erro recuperável.
- A captura e o tratamento de Excepções contribui para proclamada **robustez do código dos programas Java**.

- Excepções

- O que é uma excepção
- Erro e excepção em java
- O que origina uma excepção
- Vantagens do uso de excepções
- Tratamento de excepções
- Classificação das Excepções
- Excepções mais comuns
- Exercícios de Fixação

## O que origina uma excepção?

- ❖ Tentar aceder a uma posição de um **array** fora de seus limites
- ❖ Tentar abrir um ficheiro inexistente
- ❖ Tentar ler um ficheiro para além do fim deste
- ❖ Tentar abrir um URL inexistente
- ❖ Tentar dividir por zero
- ❖ Tentar calcular a raiz quadrada de um número negativo

### ➤ Excepções

- O que é uma excepção
- Erro e excepção em java
- O que origina uma excepção
- Vantagens do uso de excepções
- Tratamento de excepções
- Classificação das Excepções
- Excepções mais comuns
- Exercícios de Fixação

- Não diminui o esforço necessário para se detectar, reportar e manipular erros.
- Permite a separação do código fonte regular, do código responsável pelo tratamento das situações anómalas que podem acontecer no programa.
- Relativamente a utilização da **instrução if** possui as seguintes vantagens:
  - ❑ Separação entre o tratamento de erros e o algoritmo
  - ❑ Propagação de erros através da pilha de execução dos métodos.
  - ❑ Divisão por grupos de erros organizados em hierquia.

### ➤ Excepções

- O que é uma excepção
- Erro e excepção em java
- O que origina uma excepção
- Vantagens do uso de excepções
- Tratamento de excepções
- Classificação das Excepções
- Excepções mais comuns
- Exercícios de Fixação

➤ A linguagem Java permite a descrição de situações de excepção de uma forma normalizada através da utilização de **5 palavras reservadas**, correspondentes a cláusulas especiais, a saber:

- try
- catch
- finally
- throw
- throws

```
try{
    // código que pode lançar a exceção.
}
catch(Excepção1 e1)
{
    // Tratamento da e1 do tipo Excepção1
}
catch(Excepção1 e2)
{
    // Tratamento da e1 do tipo Excepção2
}
...
finally
{
    // o bloco opcional
    // se existe é executado sempre
}
```

### ➤ Excepções

- O que é uma excepção
- Erro e excepção em java
- O que origina uma excepção
- Vantagens do uso de excepções
- Tratamento de excepções
- **Classificação das Excepções**
- Excepções mais comuns
- Exercícios de Fixação

### **RuntimeException**

- Ocorrem quando há um erro de programação
- Conversão explícita de tipo (cast).
- Acesso a uma posição inexistente de um array.
- Acesso de ponteiro nulo.

### **IOException**

- Tentar ler para além do final de um ficheiro
- Tentar abrir um URL incorrecto
- Tentar encontrar um objecto Class através de uma string que não denota uma classe existente

### ➤ Excepções

- O que é uma excepção
- Erro e excepção em java
- O que origina uma excepção
- Vantagens do uso de excepções
- Tratamento de excepções
- Classificação das Excepções
- Excepções mais comuns
- Exercícios de Fixação

### ➤ **ArithmeticException**

- Indica situações de erros em processamento aritmético, tal como uma divisão inteira por 0

### ➤ **ArrayIndexOutOfBoundsException**

- Indica a tentativa de acesso a um elemento de um array fora de seus limites.

### ➤ **NullPointerException**

- Indica que a aplicação tentou usar null onde uma referencia a um objecto era necessária.

### ➤ **IOException**

- Indica a ocorrência de algum tipo de erro em operações de entrada e saída

### ➤ **NumerFormatException**

- Indica que tentou-se a conversão de uma string para um formato numérico, mas o seu conteúdo não representava adequadamente um número para aquele formato

### ➤ **StringIndexOutOfBoundsException**

- Indica tentativa de utilizar um índice numa string fora dos limites destas

### ➤ Excepções

- O que é uma excepção
- Erro e excepção em java
- O que origina uma excepção
- Vantagens do uso de excepções
- Tratamento de excepções
- Classificação das Excepções
- Excepções mais comuns
- Exercícios de Fixação

- Escreva um programa em Java que recebe dois números inteiros ( $n1, n2$ ), calcule o seu quociente e apresente o resultado no ecrã. Para tal deverá implementar divide() que vai ser responsável por efectuar a operação e devolver o resultado.
  
- Implemente o devido tratamento das excepções que pode serem geradas no programa acima.

