

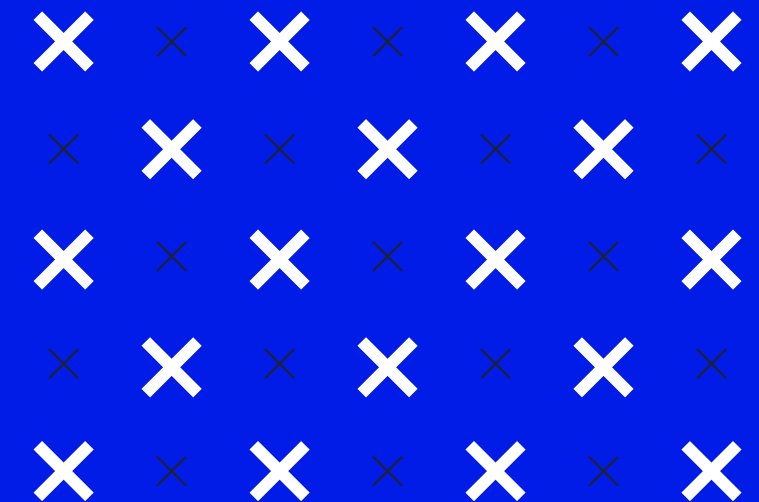


MÓDULO 02

# Java Avançado

Exceptions, debug e logs

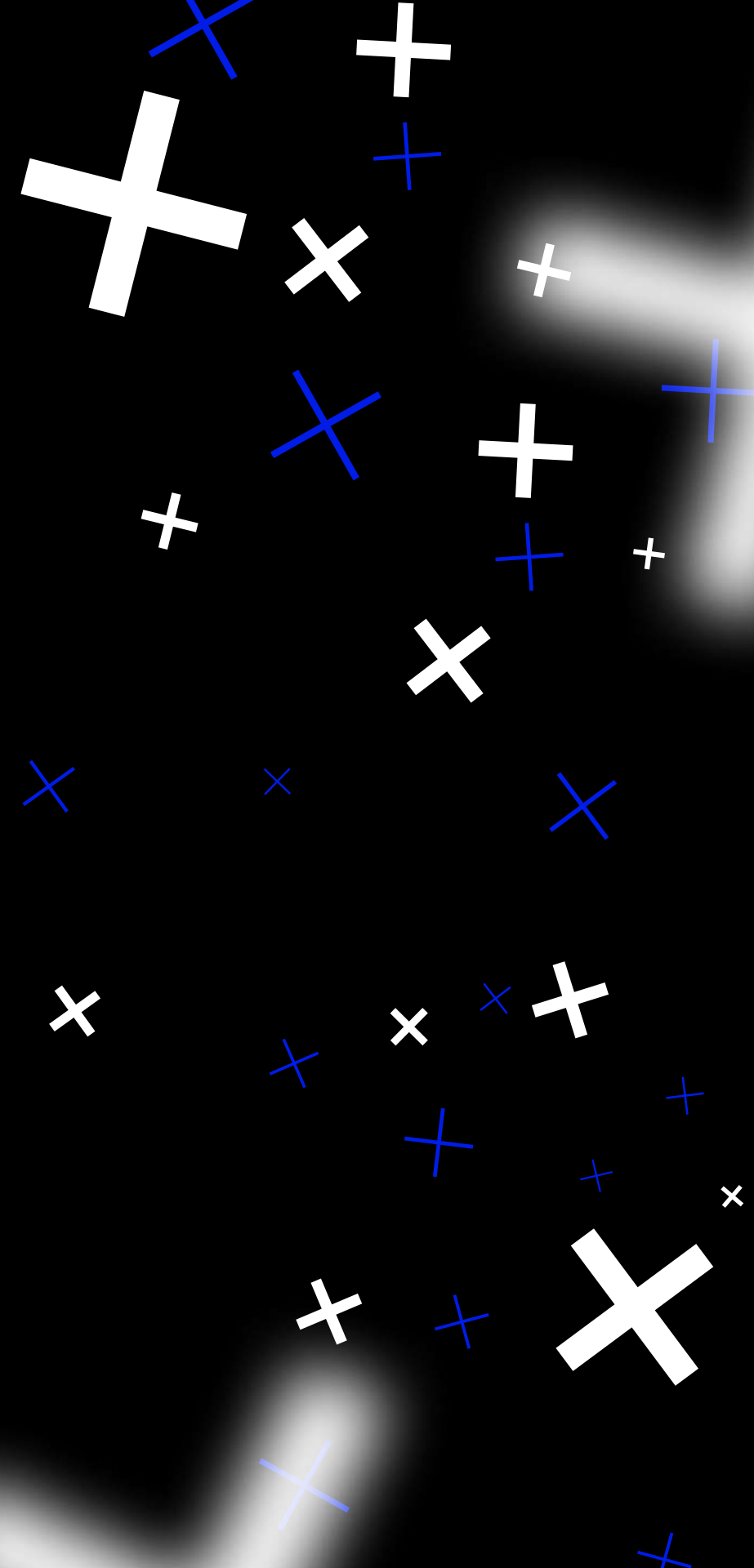
mentorama.



# Java Exceptions

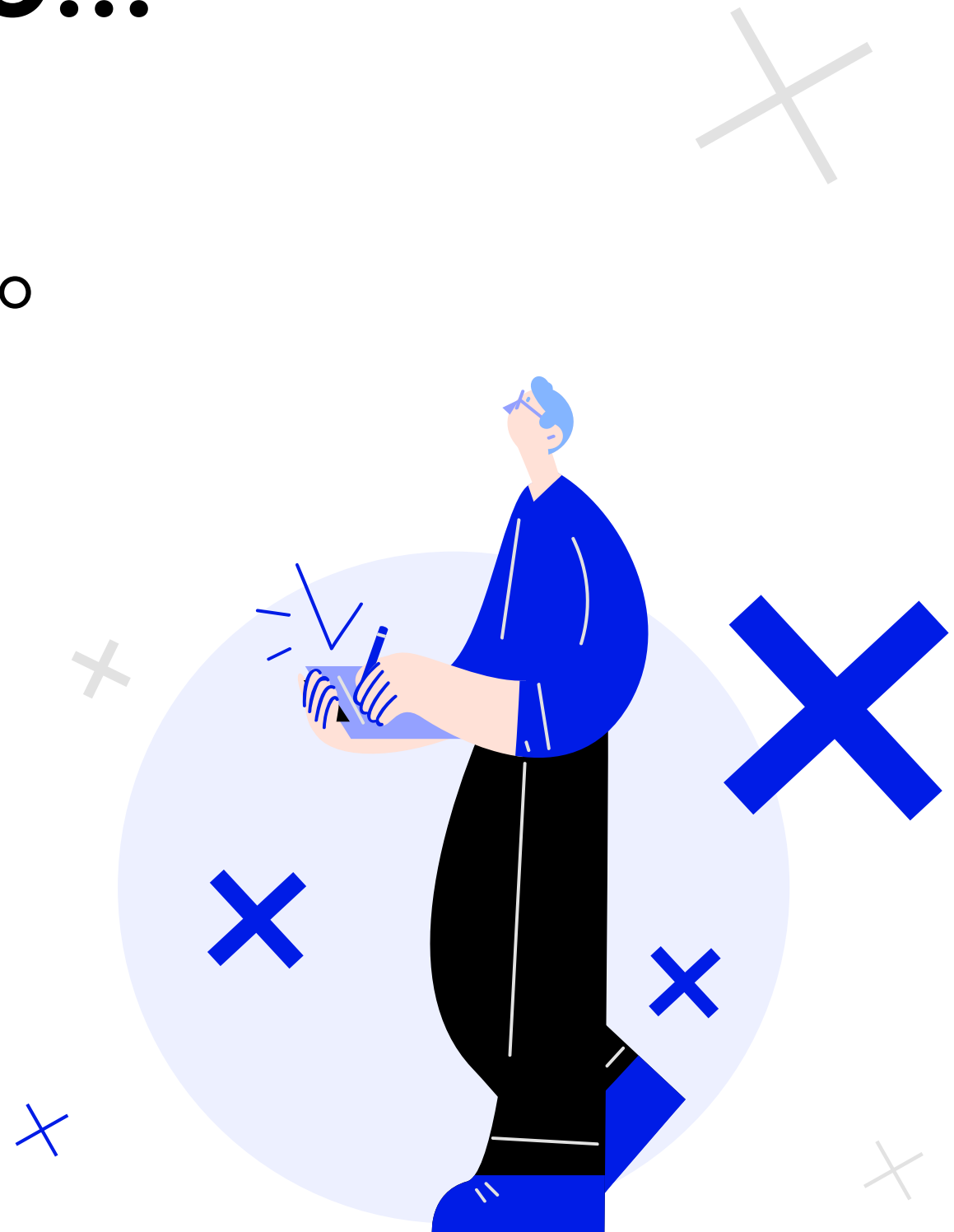
Exceptions, debug e logs

**mentorama.**



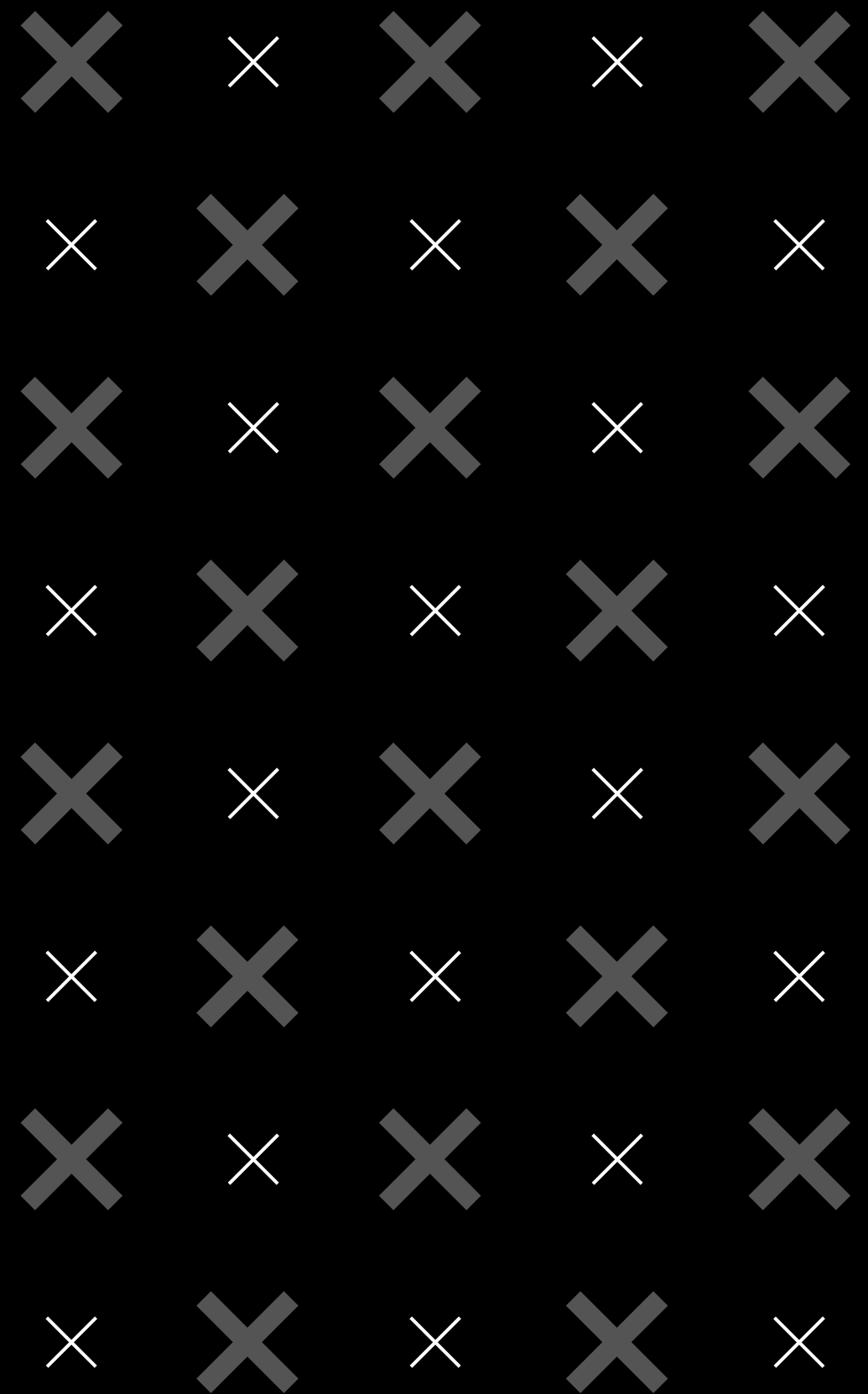
# Nem sempre tudo dá certo...

- ◆ Quando trabalhamos com projetos de software, uma coisa é certa, problemas inesperados sempre existirão
- ◆ Esses cenários inesperados, são conhecidos como **Exceções** ou **Exceptions**
- ◆ Nossas aplicações devem estar preparadas para “tratar” essas Exceptions



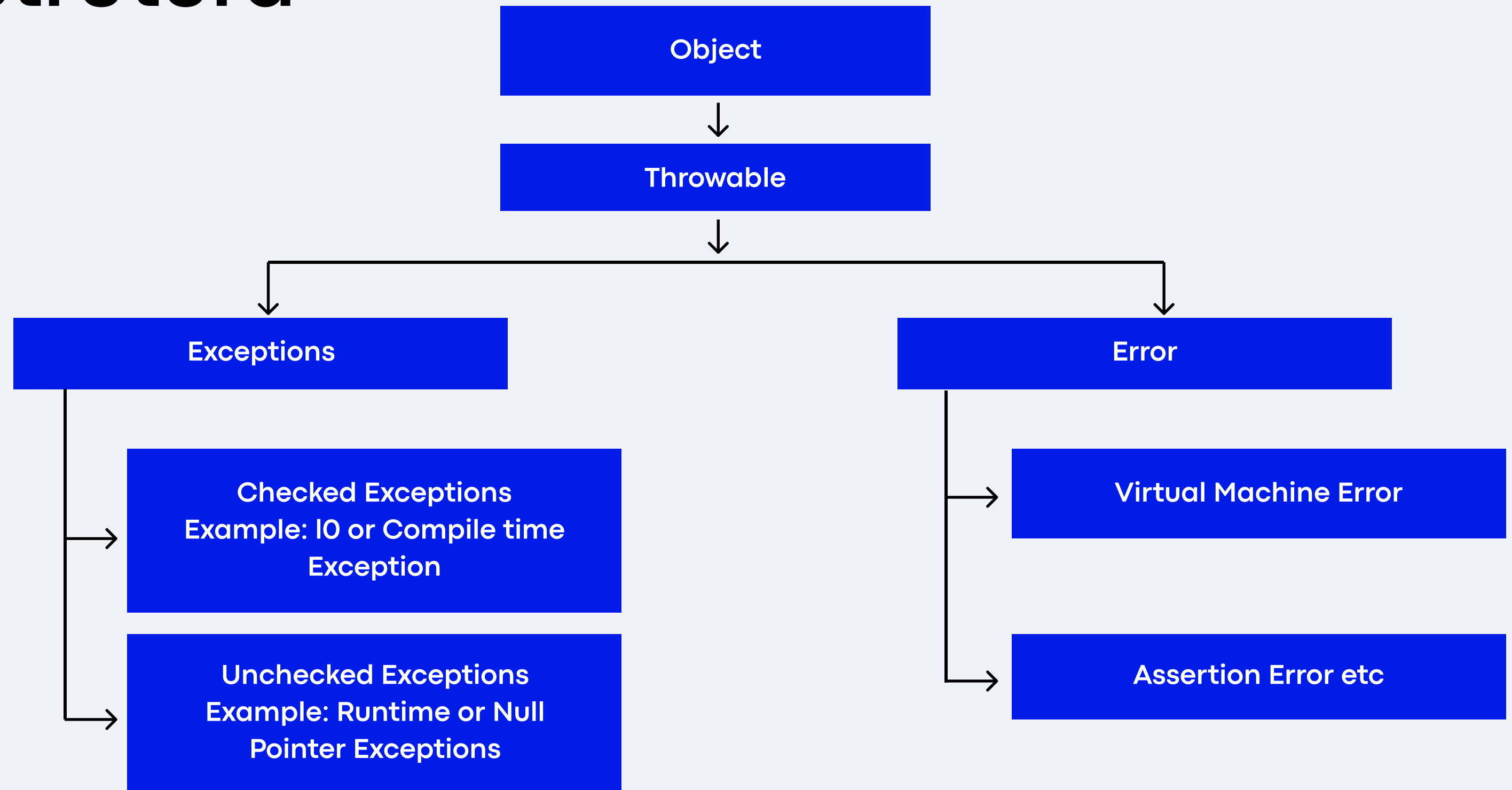
# Java Exceptions

- ◆ Em Java, assim como quase tudo, as Exceptions são representadas por classes que por sua vez possuem uma hierarquia para diferentes tipos de exceções
- ◆ Essas classes empacotam todos os detalhes referentes ao problema que ocorreu



# Estrutura

mentorama.



# Checked Exceptions

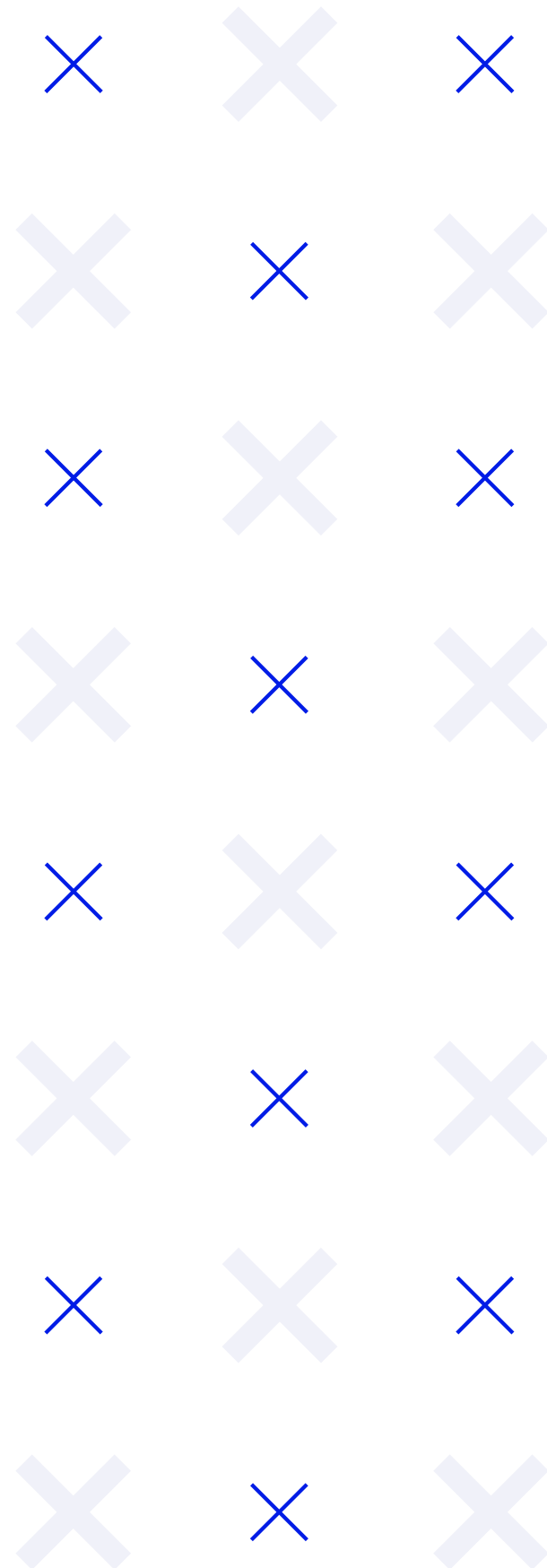
- ◆ São Exceptions que conseguimos identificar ainda em tempo de compilação a possibilidade de ocorrência

Exemplo:

- ◆ **IOException**

- ◇ Ao tentar ler um arquivo, o próprio compilador me indica que há uma possibilidade de falha pois N problemas podem acontecer ao tentar fazer isso
  - Caminho do arquivo inválido
  - Arquivo com lock, impossibilitando alteração

**mentorama.**



# Unchecked Exceptions

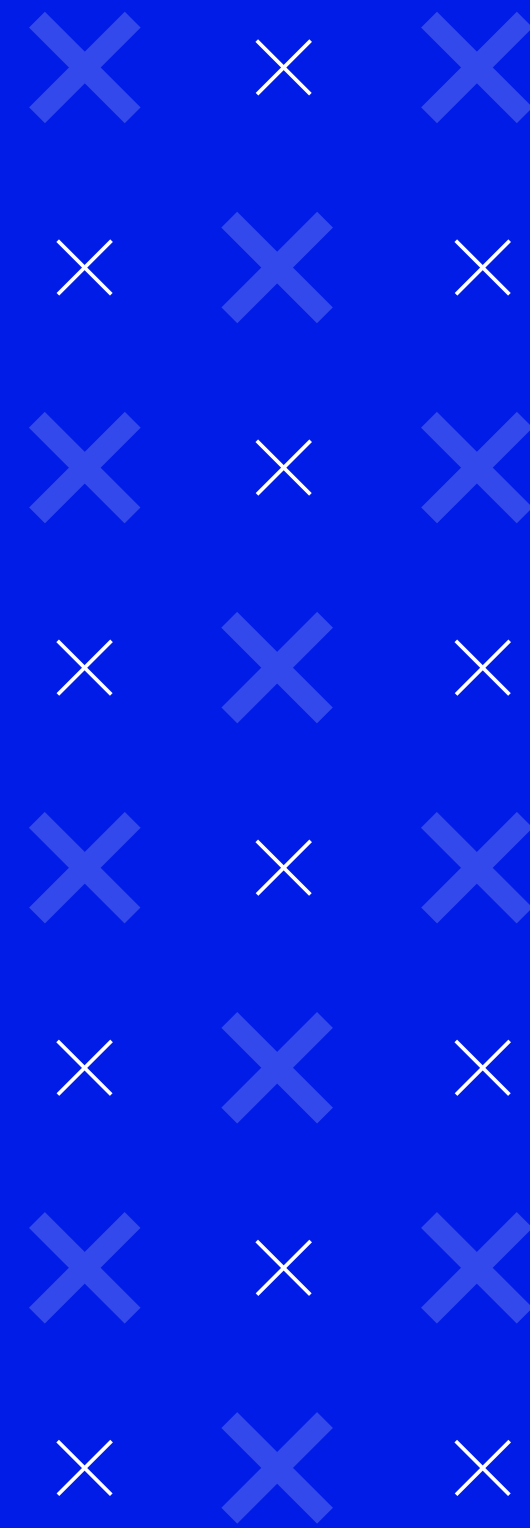
- ◆ Exception não checadas, são as mais imprevisíveis, são as que não conseguimos determinar se irão ocorrer ainda em tempo de compilação
- ◆ A maioria delas conseguimos evitar utilizando boas práticas de programação

## Exemplo:

### ◆ NullPointerException

- Ocorre quando tentamos acessar um objeto que está nulo
- Podemos evitar checando se o objeto é nulo antes de tentar acessa-lo

**mentorama.**



# Errors

- ◆ Errors indicam problemas mais sérios que aplicações razoáveis não deveriam tratar
- ◆ Se tratam de condições irrecuperáveis em que geralmente causam a finalização anormal das aplicações
- ◆ São considerados RuntimeExceptions, ou seja, exceções não checadas

## Exemplo:

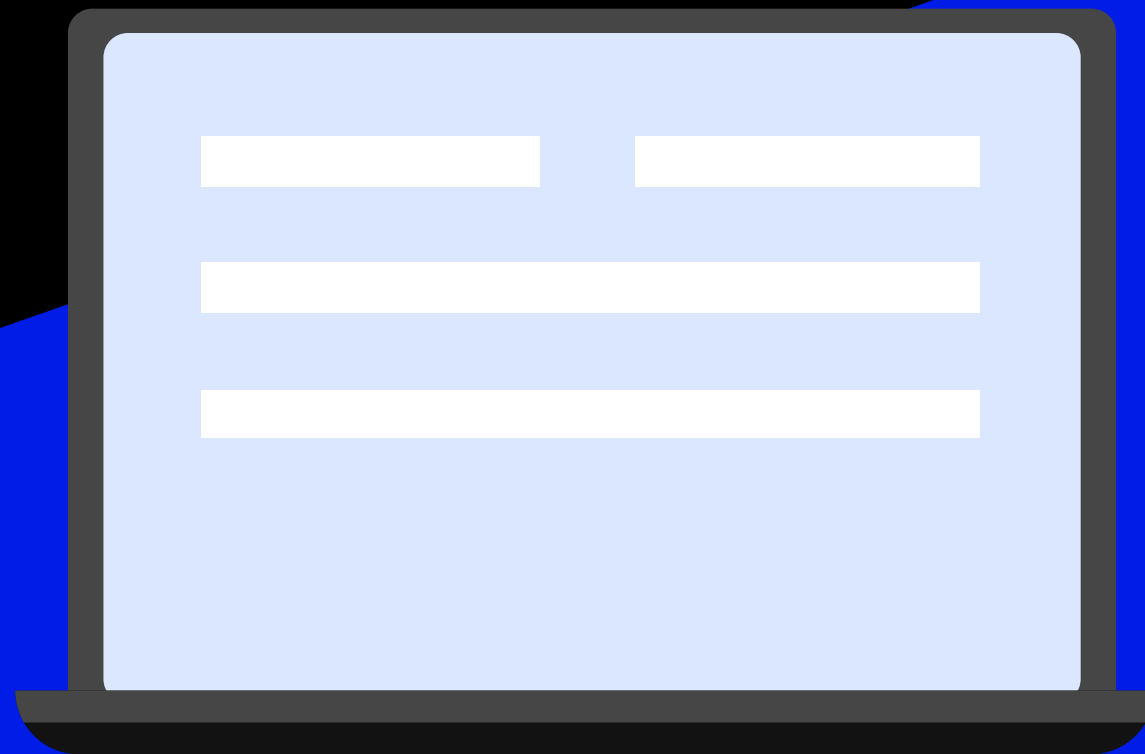
### ◇ OutOfMemoryException

- Usar mais memória do que há disponível para a JVM pode ocasionar essa exception e imediatamente o encerramento da aplicação
- Se não há mais memória disponível, não conseguimos tratar esse tipo de falha na aplicação



# Prática

- ◆ Vamos ver como tudo isso parece e funciona na prática



mentorama.