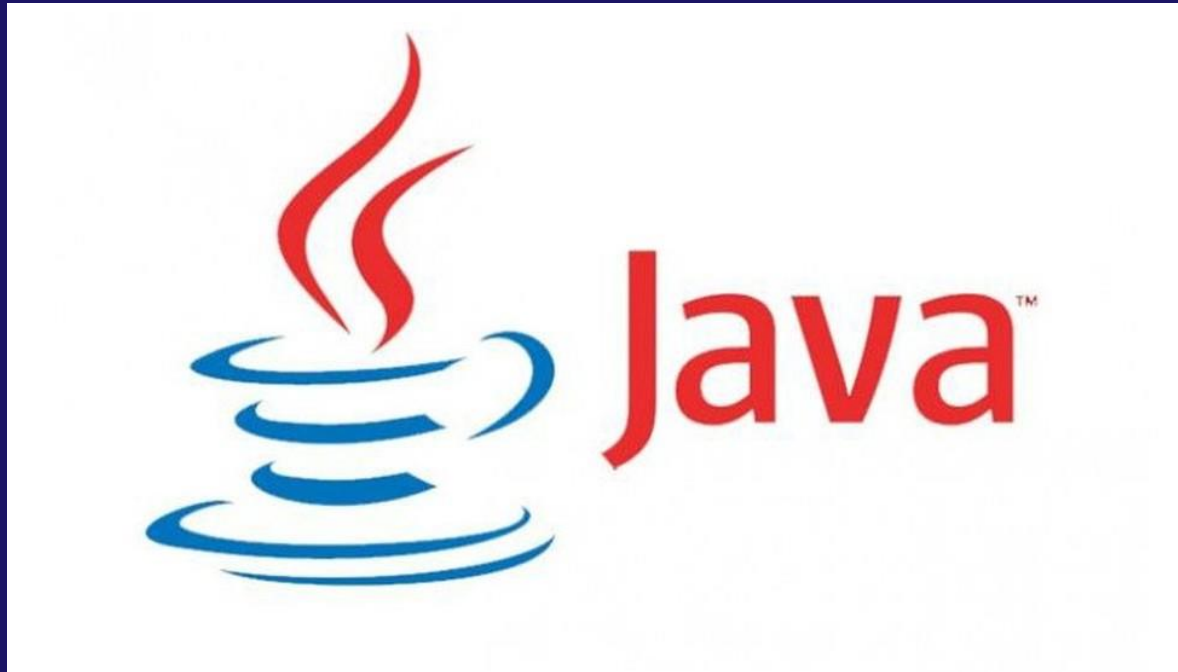


Curso Técnico em Desenvolvimento de Sistemas

UCR8 – Desenvolver e organizar interface gráfica para aplicações desktop

Aula 08 – Tratamento de Exceções



Exceções

- É um evento que interrompe o fluxo normal de processamento de uma classe, causando o término inesperado da classe.
- Problemas podem estar associados a: arrays (vetores) fora da faixa; acesso a recursos de hardware; divisão por zero; erros de aplicação; conexão a banco de dados; operações com ficheiros, entre outras.
- O uso de exceções torna o programais robusto e confiável.
- O mecanismo de exceções da linguagem Java fornece uma abordagem simples e organizada para tratamento de exceções.

Tratamento de Exceções

```
try {  
    //Codigo a ser executado pelo programa  
} catch (Exception e) {  
    //Tratamento de erro de execução  
}
```

- Para tratar exceções, podemos colocar as linhas de código que podem gerar uma exceção dentro de um bloco **try** e criar uma lista de blocos **catch** contíguos, um para cada exceção possível de ser lançada.
- As linhas de código de um bloco catch serão executadas se a exceção gerada for do mesmo tipo da listada nesse bloco de captura.

Tratamento de Exceções

```
try {  
    // Código a ser executado pelo programa  
} catch (ArithmeticException e) {  
    // Tratamento de erro de execução  
} catch (NullPointerException e) {  
    // Tratamento de erro de execução  
} catch (Exception e) {  
    // Tratamento de erro global  
} finally{  
    // Código sempre executado  
}
```

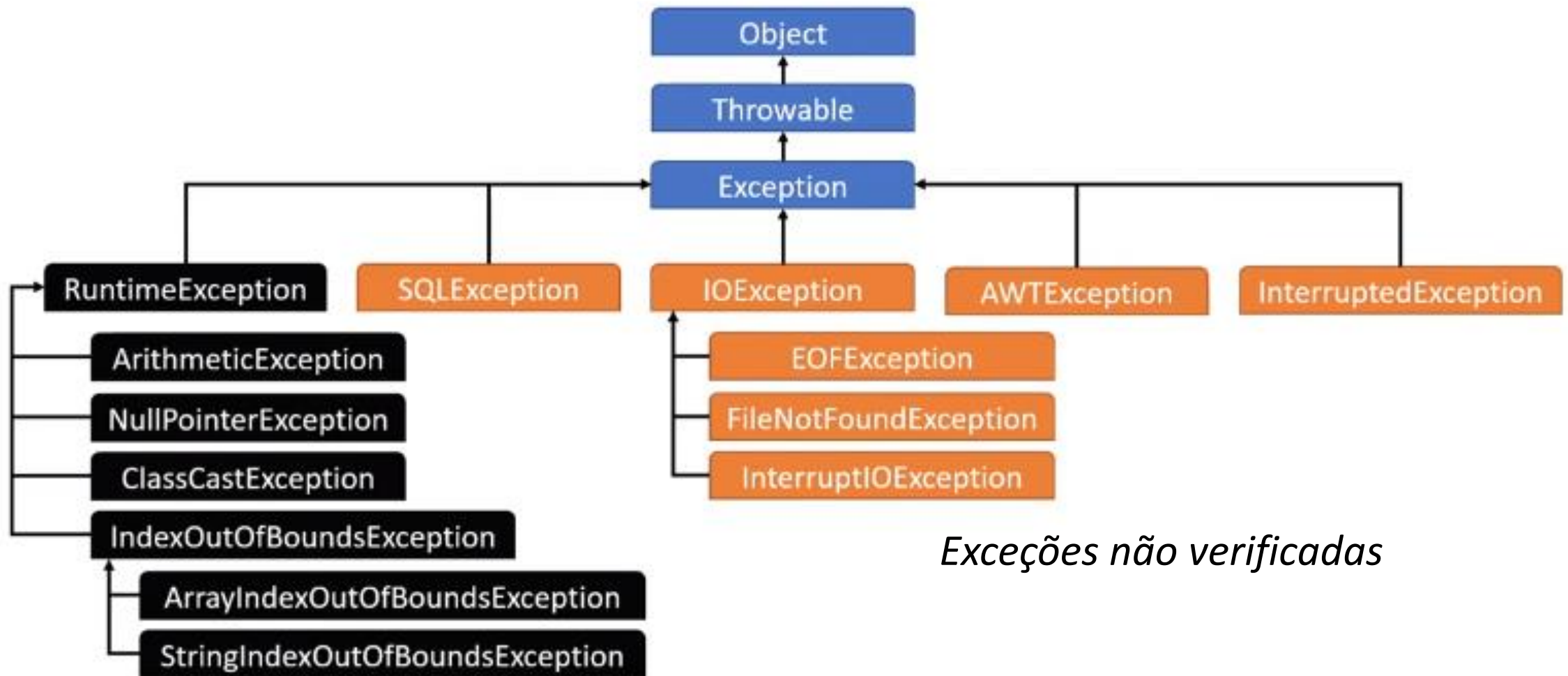
Tenta executar

Captura

Sempre executa

- Podem existir múltiplos blocos **catch** depois de um bloco **try**, cada um tratando uma exceção diferente.
- A instrução **finally** define um bloco de código que é sempre executado, quer tenha sido lançada uma exceção, ou não.

Hierarquia das Exceções



Exceções não verificadas

Exceções verificadas

Repassando Exceções entre Métodos

- Um método pode repassar uma exceção;
- Pilha de métodos

```
public static void main(  
    String[] args) {  
    metodoA();  
}  
  
public static void metodoA() {  
    try {  
        metodoB();  
    } catch (IOException e) {  
        //tratamento  
    }  
}
```

```
public static void metodoB() throws IOException {  
    metodoC();  
}  
  
public static void metodoC() throws IOException {  
    metodoD();  
}  
  
public static void metodoD() throws IOException {  
    //pode gerar uma exceção IOException  
}
```

Exercícios:

- a) Escreva um programa para preencher valores de um vetor de inteiros com 10 posições. O usuário irá informar os valores a serem inseridos e suas respectivas posições no array.

O programa deve tratar as exceções

ArrayIndexOutOfBoundsException quando for informada uma posição inexistente do vetor e *NumberFormatException* quando o valor informado não for numero.