Programação III

Aula 6 – Exceções

Introdução

O que é uma exceção?

Uma condição anormal que ocorre durante a execução do programa e interrompe seu fluxo normal.

Diferença entre **erros de compilação** e **erros em tempo de execução**.

Erros de compilação são detectados pelo compilador antes da execução. Erros em tempo de execução ocorrem durante a execução do programa e podem ser tratados.

Importância do tratamento de exceções.

Evita falhas inesperadas, melhora a robustez do código e proporciona uma melhor experiência ao usuário.

Modelo do Tratamento de Exceções

- Uso do bloco try-catch:

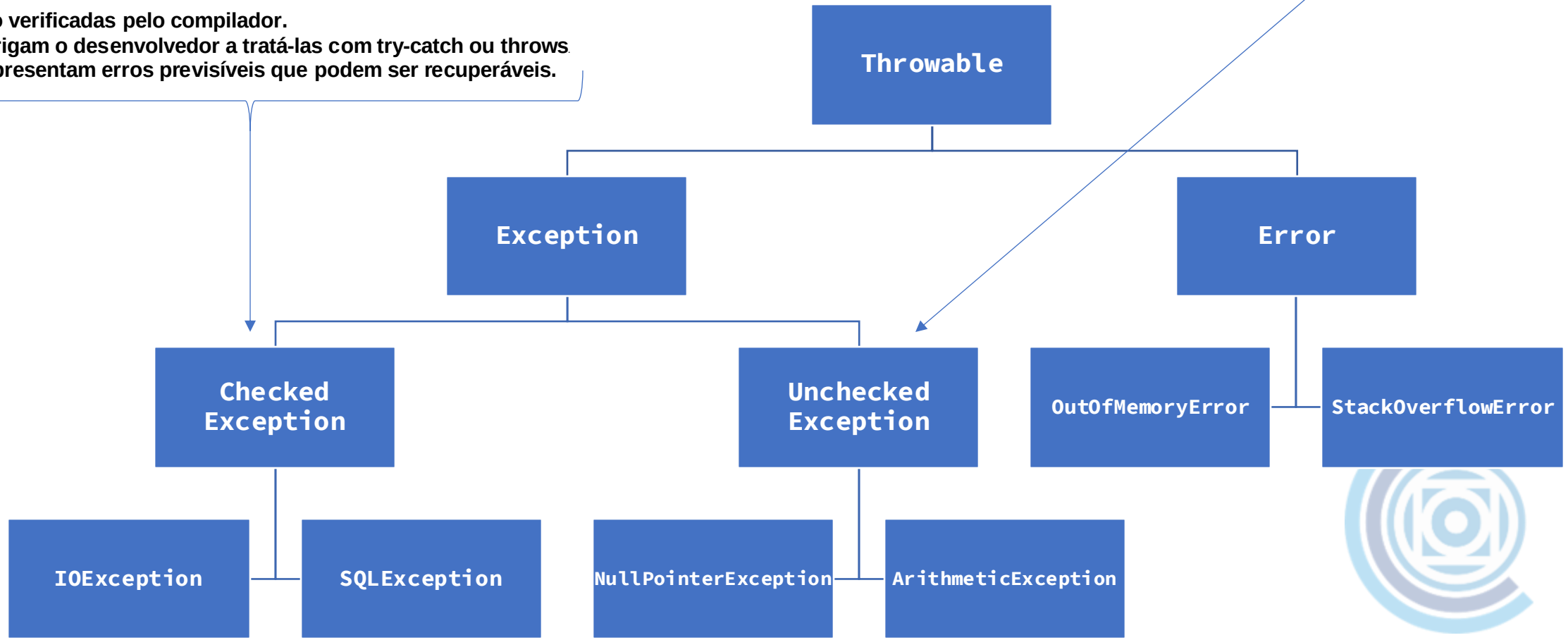
```
try
{
    int resultado = 10 / 0;
} catch (ArithmeticException e) {
    System.out.println("Erro: divisão por zero!");
}
```

- O que aconteceria ao programa se não houvesse try-catch?
- Existe alguma outra maneira de tratar a exceção?

Hierarquia de Exceções

- ✅ São verificadas pelo compilador.
- ✅ Obrigam o desenvolvedor a tratá-las com try-catch ou throws.
- ✅ Representam erros previsíveis que podem ser recuperáveis.

- ✅ Não são verificadas pelo compilador.
- ✅ O desenvolvedor não é obrigado a tratá-las.
- ✅ Ocorrem em tempo de execução (RuntimeException).
- ✅ Normalmente indicam erros de lógica que devem ser corrigidos no código.



Bloco `finally`

- Para garantir execução de código.
- Importante para **fechamento de recursos**.

```
try {
    FileReader arquivo = new FileReader("arquivo.txt");
    System.out.println("Primeira linha do arquivo: " + br.readLine());

} catch (IOException e) {
    System.out.println("Erro ao abrir o arquivo.");
} finally {
    System.out.println("Execução finalizada.");
}
```

- Em que situações o `finally` é útil?
- O que aconteceria se um erro ocorresse dentro do bloco finally?

Lançamento de Exceções (`throw` e `throws`)

- **`throw`**
  - Lançamento manual de exceções
- **`throws`**
  - Indica que um método pode gerar exceções e deve ser tratado ou propagado.

```
static void validarIdade(int idade) throws IllegalArgumentException {
    if (idade < 18) {
        throw new IllegalArgumentException("Idade inválida para acesso.");
    }
}
```

- Quando é melhor verificar condições antes de lançar uma exceção ao invés de tratá-la depois?
- O que acontece se um método lançar uma exceção e não houver tratamento?

Exemplos de métodos que lançam checked exceptions

Método	Classe	Exceção Lançada
FileReader(String fileName)	FileReader	FileNotFoundException
read()	FileReader, BufferedReader	IOException
write(String str)	FileWriter, BufferedWriter	IOException
close()	FileReader, FileWriter	IOException
getConnection(String url)	DriverManager	SQLException
createStatement()	Connection	SQLException
executeQuery(String sql)	Statement	SQLException
executeUpdate(String sql)	Statement	SQLException
close()	Connection, Statement, ResultSet	SQLException

Criando Exceções Personalizadas

- Quando usar exceções personalizadas?
- Como definir uma nova exceção?

```
class MinhaExcecao extends Exception {
    public MinhaExcecao(String mensagem) {
        super(mensagem);
    }
}
```