
– Exceções em Java – Tratamento e Lançamento

O que é Exceção?

- São objetos que sinalizam que ocorreu algum problema no tempo de execução de um programa.
- Ao ocorrer uma exceção, a JVM cria uma instância de um objeto da classe ***Throwable***
 - Lançamento de exceção

Ocorrendo exceções

- Basicamente quando
 - Um erro no programa em execução
 - Uma condição inesperada durante a execução de um programa
- A JVM lança dois tipos de objetos
 - Subclasses de **Throwable**:
 - **Exception**
 - Indicam condições menos sérias
 - Mais facilmente tratada
 - **Error**
 - Indicam geralmente problemas irre recuperáveis

Alguns exemplos

○ **Exception**

- `ArrayIndexOutOfBoundsException`
- `ClassCastException`
- `NullPointerException`
- `IllegalArgumentException`
- `IOException`

○ **Error**

- `OutOfMemoryError`
- `StackOverflowError`
- `NoClassDefFoundError`

Recursos para Exceção

- No Java você pode:
 - Capturar e tratar o erro/exceção
 - Lançar (forçar) situações de exceções
 - Criar a sua própria exceção

Tratamento de Exceções

- Declaração **try** / **catch** / **finally**
 - **try**: define o bloco de código a executar
 - **catch**: especifica o código a ser executada quando ocorre uma determinada exceção
 - **finally**: define o código a ser executado, ocorrendo ou não a execução
 - É opcional, normalmente usada para fechamento, como encerrar conexões, fechar arquivos, etc.

Estrutura Básica:

```
try {  
    ...  
} catch (Exception ex) {  
    ...  
} finally {  
    ...  
}
```

Exemplo simples:

```
try {  
    a = b/c;  
    System.out.println("resultado:" + a);  
} catch (ArithmeticException e1) {  
    System.out.println("Erro div por 0");  
} finally {  
    System.out.println("acabou");  
}
```

Tratamento de Exceções

○ Mais exemplos

```
public class Divisao {
    public static void main(String[] args) {
        double x,y,res;
        Scanner in = new Scanner(System.in);
        try {
            System.out.print("Entre com X: ");
            x = in.nextDouble();
            System.out.print("Entre com Y: ");
            y = in.nextDouble();
            res = x/y;
            System.out.println("Resultado divisao X por Y: "+res);
        } catch (InputMismatchException ex) {
            System.out.println("O dado tem que ser numerico (double)");
        } catch (ArithmeticException ex) {
            System.out.println("Y tem que ser diferente de zero");
        } catch (Exception ex) {
            System.out.println("Erro inesperado");
            ex.printStackTrace();
        }
    }
}
```

Lançando Exceções

- Acontece através do comando **throw**
- O programador decide quando utilizar
- Exemplo:

...

```
public static long fatorial(int num){  
    long fat=1;  
    if(num<0)  
        throw new IllegalArgumentException("Não permitido numero negativo");  
    else {  
        for(int i=1;i<=num;fat*=i++);  
    }  
    return fat;  
}
```


Criando Exceções

- É possível criar suas próprias classes de exceções
 - Subclasses de classes de Exceção
- Veremos isso depois mais adiante.

Exercícios (1)

- Crie uma classe Matematica com dois métodos estáticos
 - Calculo de Operações – Recebe como parâmetros dois números reais, mais o símbolo da operação (+, -, *, /)
 - Calculo de Fatorial – Recebe como parâmetro o número inteiro que se quer calcular fatorial
- Veja quais as possíveis condições de erro e lance as devidas exceções
- Faça um programa em outra classe que acesse essas operações e faça o tratamento adequado, impedindo do programa cair e dando uma mensagem consistente ao usuário
 - Esse programa deve receber números pelo teclado para acessar os dois métodos.
 - Utilize o método *nextLine()* da classe *Scanner* para isso e faça tratamentos de possíveis erros na entrada de dados.

Exercícios (2)

- Crie uma classe `Circulo` com os seguintes atributos (números reais) com seus respectivos métodos acessores
 - Coordenada x, Coordenada y e Raio
- No método acessor ***set*** de cada um, impeça do objeto receber valores negativos para quaisquer atributos.
 - Não deixe o objeto ser instanciado – ou ter valores de atributos alterados – caso isso ocorra
- O método construtor deve receber os 3 atributos como parâmetros
- Faça um programa em outra classe que instancie objeto da classe `Circulo` e altere os seus atributos, impedindo do programa cair e dando uma mensagem consistente ao usuário
 - Esse programa deve receber números pelo teclado para instanciar objetos e alterar atributos
 - Utilize o método `nextLine()` da classe `Scanner` para isso e faça tratamentos de possíveis erros na entrada de dados.