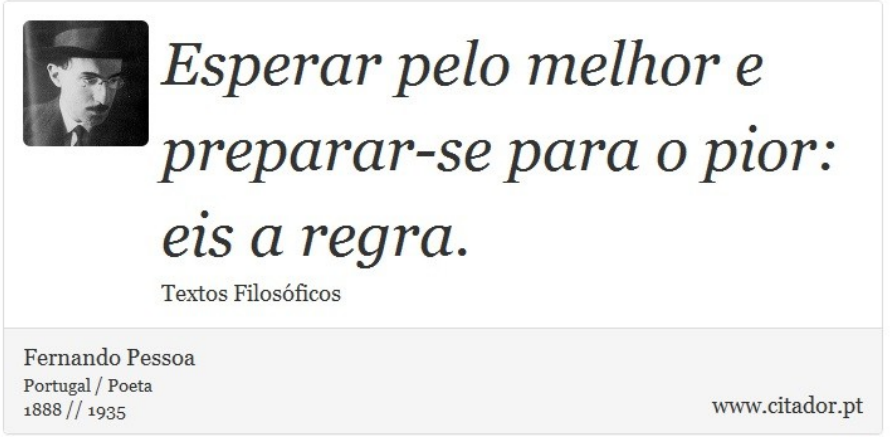
05-TratamentoDeErros-Exceções-Parte-1

Conteúdo: Tratamento de Erros. Exceções.

Introdução (1)



"Eu posso lembrar do exato momento em que percebi que uma boa parte da minha vida seria gasta procurando erros nos meus próprios programas.". Maurice Wilkes, Memoirs of a Computer Pioneer, 1985.

Introdução (2)

- Codificamos por natureza de forma otimista, como se nada fosse dar errado(caminho feliz)
- Tratamento de erro tradicional
  - Código de retorno

```
int main() {
    FILE *file = fopen("non_existent_file.txt", "r");
    if (file == NULL) {
        fprintf(stderr, "Error opening file: %s\n", strerror(errno));
        exit(EXIT_FAILURE);
    } else {
        printf("File opened successfully!\n");
        fclose(file);
    }
    return 0;
}
```

Tratamento de Exceções (1)

- Exceção: situação que foge ao cenário típico durante a execução de um código
- A documentação da API Java cita as exceções que podem ser disparadas por um método. Ver exemplo da classe Integer.parseInt
- Sintaxe de bloco de tratamento de erro

```
try { <instrução>* }
catch(T v) { <instrução>* }
finally { <instrução>* }
```

É obrigatório haver ao menos um bloco catch ou um finally.

Bloco protegido

Bloco de tratamento de erro. Podem haver vários blocos catch, um para cada tipo de erro (**T**). **v** é o nome da variável que recebe a referência ao objeto que representa a exceção.

Executado ao final de forma incondicional, ocorrendo ou não exceção no código protegido.

Tratamento de Exceções (2)

- Exemplo:

```
public class Somat {
 public static void main(String[] args){
  int soma=0;
  for( String arg : args ) {
   soma+=Integer.parseInt(arg);
  }
  System.out.println(soma);
 }
}
```

- `java Somat 1 2 3 4`

```
10
```

Forma simplificada do uso do **for**. A cada passo, a variável arg assume um dos valores contigos no vetor args. args recebe o vetor de strings produzido pela invocação do programa através da linha de comando.

- `java Somat 1 dois 3 4`

```
Exception in thread "main"
   java.lang.NumberFormatException: For input string:
   "dois"
at java.lang.NumberFormatException.forInputString(
   NumberFormatException.java:48)
at java.lang.Integer.parseInt(Integer.java:447)
at java.lang.Integer.parseInt(Integer.java:497)
at Somat.main(Somat.java:5)
```

Tratamento de Exceções (3)

- Bloco try-catch

```
public class Somat {
  public static void main(String args[]){
    int soma=0;
    try {
      for( String arg : args ) {
        soma+=Integer.parseInt(arg);
      }
      System.out.println(soma);
    }
    catch(NumberFormatException nfe) {
      System.out.println(“Parâmetro incorreto!”);
    }
  }
}
```

Bloco protegido. Ao ocorrer a exceção, o fluxo de execução é desviado para um dos blocos catch.

Captura um tipo de exceção não-verificada. Herda de RuntimeException

Tratamento de Exceções (4)


```
public class Somat {
  public static void main(String args[]){
    int soma=0;
    for( String arg : args ) {
      try {
        soma+=Integer.parseInt(arg);
      }
      catch(NumberFormatException nfe) {
        System.out.println(“Argumento “ + arg + “ ignorado!”);
      }
    }
    System.out.println(soma);
  }
}
```

Responda rápido:
qual a diferença entre este
e o código anterior

Tratamento de Exceções (5)

- Tratando várias classes de erro

```
try {
  /* código que pode disparar uma ou mais exceções */
}
catch (MinhaExcecao e1) {
  /* Código a ser executado se a exceção for da classe
  MinhaExcecao */
}
catch (MinhaOutraExcecao e2) {
  /* Código a ser executado se a exceção for da classe
  MinhaOutraExcecao */
}
catch (Exception e3) {
   /* Qualquer outra exceção é tratada aqui, uma vez que
     todas as outras exceções são herdadas de Exception */
}
```

Tratamento de Exceções (6)

- Mecanismo da pilha de execução
  - Se uma exceção é disparada e não existe tratamento (try..catch), a exceção é jogada para o método chamador, e assim sucessivamente
  - Se a exceção voltar ao método main e não for tratada também nele, o programa aborta.
  - A pilha de execução (stackTrace) é mostrada no console na ocorrência de erro fatal.

Tratamento de Exceções (7)

- Cláusula finally
  - Opcional e apenas um por bloco try..catch
  - É sempre executada, ocorrendo ou não exceção

```
try {
  abrirTorneira();
  iniciarAquecimento();
} catch(SuperAquecimentoException e) {
  exibirProblema(e);
} finally {
  fecharTorneira();
}
```

Exercício de Fixação

Faça o questionário
05-TratamentoDeErros-Exceções-Parte-1