Programação Orientada a Objetos

Aula 09 – Manipulando Exceções

Prof. Luiz Mário Lustosa Pascoal

Exceções

- As exceções oferecem uma forma clara de verificar e tratar erros deixando o código bem legível

- O termo ***exception*** significa que uma condição excepcional, fora do normal, ocorreu

- Em Java, quando isto ocorre, diz-se que uma exceção é lançada (*thrown*)

- Basicamente, em Java, utilizamos um bloco **try/catch** para tratar exceções

Exceções

- O código que é responsável por executar alguma ação quando uma exceção ocorre é denominado de **exception handler**.

- Este código “pega” (catches) a exceção lançada e a trata.

- A parte **try** do código é utilizada para definir um bloco de código onde as exceções podem ocorrer.

Exceções

- Seguindo a região **try** podemos utilizar quantas cláusulas **catch** quisermos

- Os blocos catch seguem o bloco **try** imediatamente

- Não pode existir nenhum outro código entre os blocos **catch**

Try, catch e finally

```
try {
  // bloco de código
} catch (ExceçãoTipo1 e) {
    // manipulador de exceção para ExceçãoTipo1
} catch (ExceçãoTipo2 e) {
   // manipulador de exceção para ExceçãoTipo2
   throw(e); // lançar a exceção novamente ...
}finally {
}
```

Exceções não-capturadas

```
class Excecao01 {
  public static void main(String args[ ]){
    int d = 0;
    int a = 42 / d;
  }
}
```

```
java.lang.ArithmeticException: / by zero
  at Excecao01.main(Excecao01.java:4)
```

try e catch

```
class Excecao01 {
  public static void main{String args[ ]){
    try{
      int d = 0;
      int a = 42 / d;
    }catch (ArithmeticException e) {
        System.out.println(“divisão por zero”);
    }
  }
}
```

throw

- As exceções são lançadas utilizando-se a intrução throw, a qual toma um objeto como seu parâmetro
- Uma exceção também pode ser gerada pela chamada de um método que lança (por si) uma exceção

throw - exemplo

```
class ThrowDemo{
  static void demoproc( ) {
    try {
      throw new NullPointerException(“demo”);
    }catch(NullPointerException e) {
      System.out.println(“capturada dentro de demoproc”);
      throw e;
    }
  }
  public static void main(String args[ ]){
     try {
         demoproc( );
     } catch(NullPointerException e) {
        System.out.println(“recapturada: “ + e );
     }
  }
}
```

Cláusula throws

- Se um método é capaz de causar uma exceção com a qual ele mesmo não consegue lidar, ele deve especificar esse comportamento para que os chamadores possam proteger-se contra essa exceção.
- A palavra-chave throws é usada para identificar a lista de possíveis exceções que um método pode lançar.
- type nome-metodo(lista-arg) throws lista-excecoes{ }

throws - exemplo

```
class ThrowsDemo{
  static void proced( ) throws IllegalAccessException {
    System.out.println(“dentro do procedimento”);
     throw new IllegalAccessException(“demo”);
  }
  public static void main(String args[ ]){
     try {
         proced( );
     } catch(IllegalAccessException e) {
        System.out.println(“capturada: “ + e );
     }
  }
}
```

finally

- Às vezes é necessário que determinado bloco de código seja executado independentemente das exceções que são causadas e capturadas

- Isto é útil para liberar todos os recursos que possam ter sido alocados no início de um método

- Como a execução é transferida automaticamente para o bloco catch, necessitamos de um mecanismo para realizar o *cleanup* de recursos utilizados pelo programa

- Este mecanismo é representado pelo bloco finally

Tipos de exceção

- As exceções em Java são objetos

- Novos tipos de exceção estendem a classe Exception

Exemplo

```
class MinhaExcecao extends Exception{
 private int detalhe;
 MinhaExcecao(int a) {
  detalhe = a;
 }
 public String toString( ) {
  return "MinhaExcecao[" + detalhe + "]";
 }
}
```

```
class DemoExcecao{
 static void calcule (int a) throws MinhaExcecao{
  System.out.println("chamado calcule(" + a + ").");
  if (a > 10)
   throw new  MinhaExcecao(a);
  System.out.println("encerramento normal.");
 }
 public static void main(String args[ ]){
  try {
   calcule(1);
   calcule(20);
  }catch (MinhaExcecao e){
   System.out.println("capturada " + e);
 }
}
}
```