

Tratamento de Exceções

```
ostream>
using namespace std;

void division(double a, double b) {
    if (b == 0) {
        cout << "Division by zero.\n";
        return;
    }
    double result = a / b;
    cout << "operator, denominator, result;\n";
    cout << "type a numerator followed by denominator:\n";
    cout << "operator >> denominator;\n";
    double numerator, denominator;
    while (true) {
        if (cin.get() != '\n') {
            numerator = division(numerator, denominator);
            cout << "precision(2);\n";
            cout << fixed << numerator << "/" << denominator << " = " << result << endl;
        }
        const char* msg = "Enter a numerator followed by denominator:\n";
        cout << msg << endl;
    }
}
```

Definição

Exceção, como próprio nome sugere, é um evento que ocorre com pouca frequência e que, no contexto da programação, acontece em tempo de execução, ou seja, quando o programa está rodando no computador.

Conceito

Para tudo na vida há uma exceção! Com a lógica de programação não é diferente. Exceções podem ocorrer a qualquer momento: tentar acessar a área de memória de um ponteiro nulo, dividir um valor por zero, tentar acessar um arquivo sem possuir as devidas permissões, dentre outras eventualidades. Para lidar com esses acontecimentos trabalharemos com o tratamento de exceções.

Contextualização

Um programa que calcula a média aritmética de N valores pode receber nenhum valor para ser calculado. Dessa forma seu cálculo seria $0/0$ e, não havendo como dividir um número real por zero, esse evento representaria uma exceção. Tratar exceções significa retomar o fluxo normal do programa.

Formato de uma exceção em C

```
try {  
    // bloco de comandos  
} catch (tipo_da_exceção variável_de_referência) {  
    // bloco de comandos  
}
```

- *try* (tentar): bloco de código que tentará ser executado.
- *catch* (capturar): havendo uma exceção, a mesma será capturada e esse bloco executará suas instruções.
- tipo da exceção: a exceção pode ser de tipos predefinidos, como classes ou a própria chamada de uma função pode lançar (*throw*) uma exceção.

Tratamento de Exceção em C++

```
1  #include <iostream>
2
3  using namespace std;
4
5  double division(double a, double b) {
6      if(!b) {
7          throw "Division by zero.\n";
8      }
9      return a / b;
10 }
11
12 int main() {
13     double numerator, denominator, result;
14     cout << "Type a numerator followed by denominator:\n";
15     cin >> numerator >> denominator;
16     try {
17         result = division(numerator, denominator);
18         cout.precision(2);
19         cout << fixed << numerator << "/" << denominator << " = " << result << endl;
20     } catch(const char* msg) {
21         cout << msg << endl;
22     }
23 }
```

Referências

- SCHILDT, H. C: **Completo e Total**. 3 ed. São Paulo: Makron Books, 1996. 720 p.
- DEITEL, H. M.; DEITEL, P. J. Como Programar em C. 6 ed. São Paulo: Pearson, 2011. 818 p.
- DAMAS, L. Linguagem C. 10 ed. Rio de Janeiro: LTC 2007. 410 p.

Dúvidas



Obrigado...

**Boa semana
para
todos!!!**