



PROGRAMAÇÃO O.O. (C#)



Tratamento de exceções
Professor: João Luiz Lagôas



Observe o código abaixo



```
class Program
{
    static void Main(string[] args)
    {
        Console.WriteLine("Por favor entre com um numerador:");
        int numerador = Convert.ToDecimal(Console.ReadLine());

        Console.WriteLine("Por favor entre com um denominador:");
        int denominador = Convert.ToDecimal(Console.ReadLine());

        decimal resultado = numerador / denominador;
        Console.WriteLine("\nResultado: {0} / {1} = {2}",
                        numerador, denominador, resultado);

        Console.ReadLine();
    }
}
```

Observe o código abaixo



```
class Program
{
    static void Main(string[] args)
    {
        Console.WriteLine("Por favor entre com um numerador:");
        int numerador = Convert.ToDecimal(Console.ReadLine());

        Console.WriteLine("Por favor entre com um denominador:");
        int denominador = Convert.ToDecimal(Console.ReadLine());

        decimal resultado = numerador / denominador;
        Console.WriteLine("\nResultado: {0} / {1} = {2}",
                        numerador, denominador, resultado);

        Console.ReadLine();
    }
}
```

Há algum erro no código?

Observe o código abaixo



Não é possível converter implicitamente o tipo "decimal" em "int".

```
class Program
{
    static void Main(string[] args)
    {
        Console.WriteLine("Por favor entre com um numerador:");
        int numerador = Convert.ToDecimal(Console.ReadLine());

        Console.WriteLine("Por favor entre com um denominador:");
        int denominador = Convert.ToDecimal(Console.ReadLine());

        decimal resultado = numerador / denominador;
        Console.WriteLine("\nResultado: {0} / {1} = {2}",
                           numerador, denominador, resultado);

        Console.ReadLine();
    }
}
```

O programa não irá nem rodar...

Erro de compilação



- Esses tipos de erros, como o próprio nome diz, são identificados pelo compilador antes da execução do programa.
- Em ambientes de desenvolvimento como o Visual Studio, os erros de compilação são destacados e sugestões de solução são apresentadas.
- Erros de compilação costumam ser erros sintáticos ou semânticos:
 - Esquecer “;”, construir estruturas de modo errado (**if** ou **while** sem operação *booleana*; **for** não estruturado), tentar usar variáveis não declaradas, etc.
 - Atribuição de tipos incompatíveis, operar variáveis de tipos incompatíveis, etc.

Observe o código abaixo



```
class Program
{
    static void Main(string[] args)
    {
        Console.WriteLine("Por favor entre com um numerador:");
        decimal numerador = Convert.ToDecimal(Console.ReadLine());

        Console.WriteLine("Por favor entre com um denominador:");
        decimal denominador = Convert.ToDecimal(Console.ReadLine());

        decimal resultado = numerador / denominador;
        Console.WriteLine("\nResultado: {0} / {1} = {2}",
                        numerador, denominador, resultado);

        Console.ReadLine();
    }
}
```

Observe o código abaixo



```
class Program
{
    static void Main(string[] args)
    {
        Console.WriteLine("Por favor entre com um numerador:");
        decimal numerador = Convert.ToDecimal(Console.ReadLine());

        Console.WriteLine("Por favor entre com um denominador:");
        decimal denominador = Convert.ToDecimal(Console.ReadLine());

        decimal resultado = numerador / denominador;
        Console.WriteLine("\nResultado: {0} / {1} = {2}",
                          numerador, denominador, resultado);

        Console.ReadLine();
    }
}
```

Agora há algum erro no código?

Observe o código abaixo



Não há erro de compilação! O programa irá sim rodar.
Mas o código pode gerar um erro **durante a sua execução.**

```
class Program
{
    static void Main(string[] args)
    {
        Console.WriteLine("Por favor entre com um numerador:");
        decimal numerador = Convert.ToDecimal(Console.ReadLine());

        Console.WriteLine("Por favor entre com um denominador:");
        decimal denominador = Convert.ToDecimal(Console.ReadLine());

        decimal resultado = numerador / denominador;
        Console.WriteLine("\nResultado: {0} / {1} = {2}",
                          numerador, denominador, resultado);

        Console.ReadLine();
    }
}
```


Exemplo 1



Console

Por favor entre com um numerador

```
class Program
{
    static void Main(string[] args)
    {
        Console.WriteLine("Por favor entre com um numerador:");
        decimal numerador = Convert.ToDecimal(Console.ReadLine());

        Console.WriteLine("Por favor entre com um denominador:");
        decimal denominador = Convert.ToDecimal(Console.ReadLine());

        decimal resultado = numerador / denominador;
        Console.WriteLine("\nResultado: {0} / {1} = {2}",
                          numerador, denominador, resultado);

        Console.ReadLine();
    }
}
```

Exemplo 1



Console

Por favor entre com um numerador
40

```
class Program
{
    static void Main(string[] args)
    {
        Console.WriteLine("Por favor entre com um numerador:");
        decimal numerador = Convert.ToDecimal(Console.ReadLine());

        Console.WriteLine("Por favor entre com um denominador:");
        decimal denominador = Convert.ToDecimal(Console.ReadLine());

        decimal resultado = numerador / denominador;
        Console.WriteLine("\nResultado: {0} / {1} = {2}",
                          numerador, denominador, resultado);

        Console.ReadLine();
    }
}
```

Exemplo 1



Console

Por favor entre com um numerador
40
Por favor entre com um denominador

```
class Program
{
    static void Main(string[] args)
    {
        Console.WriteLine("Por favor entre com um numerador:");
        decimal numerador = Convert.ToDecimal(Console.ReadLine());

        ➡ Console.WriteLine("Por favor entre com um denominador:");
        decimal denominador = Convert.ToDecimal(Console.ReadLine());

        decimal resultado = numerador / denominador;
        Console.WriteLine("\nResultado: {0} / {1} = {2}",
                          numerador, denominador, resultado);

        Console.ReadLine();
    }
}
```

Exemplo 1



Console

Por favor entre com um numerador
40
Por favor entre com um denominador
hello

```
class Program
{
    static void Main(string[] args)
    {
        Console.WriteLine("Por favor entre com um numerador:");
        decimal numerador = Convert.ToDecimal(Console.ReadLine());

        Console.WriteLine("Por favor entre com um denominador:");
        decimal denominador = Convert.ToDecimal(Console.ReadLine());

        decimal resultado = numerador / denominador;
        Console.WriteLine("\nResultado: {0} / {1} = {2}",
                           numerador, denominador, resultado);

        Console.ReadLine();
    }
}
```

Exemplo 1



```
class Program
{
    static void Main(string
    {
        Console.WriteLine("
        decimal numerador =

        Console.WriteLine("Por favor entre com um denominador:");
        decimal denominador = Convert.ToDecimal(Console.ReadLine());

        decimal resultado = numerador / denominador;
        Console.WriteLine("\nResultado: {0} / {1} = {2}",
                           numerador, denominador, resultado);

        Console.ReadLine();
    }
}
```

Exceção Sem Tratamento

System.FormatException: 'A cadeia de caracteres de entrada não estava em um formato correto.'

[Exibir Detalhes](#) | [Copiar Detalhes](#)

► Configurações de Exceção

Exemplo 2



Console

Por favor entre com um numerador

```
class Program
{
    static void Main(string[] args)
    {
        Console.WriteLine("Por favor entre com um numerador:");
        decimal numerador = Convert.ToDecimal(Console.ReadLine());

        Console.WriteLine("Por favor entre com um denominador:");
        decimal denominador = Convert.ToDecimal(Console.ReadLine());

        decimal resultado = numerador / denominador;
        Console.WriteLine("\nResultado: {0} / {1} = {2}",
                           numerador, denominador, resultado);

        Console.ReadLine();
    }
}
```

Exemplo 2



Console

Por favor entre com um numerador
40

```
class Program
{
    static void Main(string[] args)
    {
        Console.WriteLine("Por favor entre com um numerador:");
        decimal numerador = Convert.ToDecimal(Console.ReadLine());

        Console.WriteLine("Por favor entre com um denominador:");
        decimal denominador = Convert.ToDecimal(Console.ReadLine());

        decimal resultado = numerador / denominador;
        Console.WriteLine("\nResultado: {0} / {1} = {2}",
                          numerador, denominador, resultado);

        Console.ReadLine();
    }
}
```

Exemplo 2



Console

Por favor entre com um numerador
40
Por favor entre com um denominador

```
class Program
{
    static void Main(string[] args)
    {
        Console.WriteLine("Por favor entre com um numerador:");
        decimal numerador = Convert.ToDecimal(Console.ReadLine());

        Console.WriteLine("Por favor entre com um denominador:");
        decimal denominador = Convert.ToDecimal(Console.ReadLine());

        decimal resultado = numerador / denominador;
        Console.WriteLine("\nResultado: {0} / {1} = {2}",
                          numerador, denominador, resultado);

        Console.ReadLine();
    }
}
```


Exemplo 2



Console

Por favor entre com um numerador
40
Por favor entre com um denominador
0

```
class Program
{
    static void Main(string[] args)
    {
        Console.WriteLine("Por favor entre com um numerador:");
        decimal numerador = Convert.ToDecimal(Console.ReadLine());

        Console.WriteLine("Por favor entre com um denominador:");
        decimal denominador = Convert.ToDecimal(Console.ReadLine());

        decimal resultado = numerador / denominador;
        Console.WriteLine("\nResultado: {0} / {1} = {2}",
                           numerador, denominador, resultado);

        Console.ReadLine();
    }
}
```

Exemplo 2



Console

Por favor entre com um numerador
40
Por favor entre com um denominador
0

```
class Program
{
    static void Main(string[] args)
    {
        Console.WriteLine("Por favor entre com um numerador:");
        decimal numerador = Convert.ToDecimal(Console.ReadLine());

        Console.WriteLine("Por favor entre com um denominador:");
        decimal denominador = Convert.ToDecimal(Console.ReadLine());

        decimal resultado = numerador / denominador;
        Console.WriteLine("\nResultado: {0} / {1} = {2}",
                           numerador, denominador, resultado);

        Console.ReadLine();
    }
}
```

Exemplo 2



```
class Program
{
    static void Main(string
    {
        Console.WriteLine("
        decimal numerador =

        Console.WriteLine("Por favor entre com um denominador:");
        decimal denominador = Convert.ToDecimal(Console.ReadLine());

        decimal resultado = numerador / denominador;
        Console.WriteLine("\nResultado: {0} / {1} = {2}",
                           numerador, denominador, resultado);

        Console.ReadLine();
    }
}
```

Exceção Sem Tratamento

System.DivideByZeroException: 'Tentativa de divisão por zero.'

[Exibir Detalhes](#) | [Copiar Detalhes](#)

▸ Configurações de Exceção

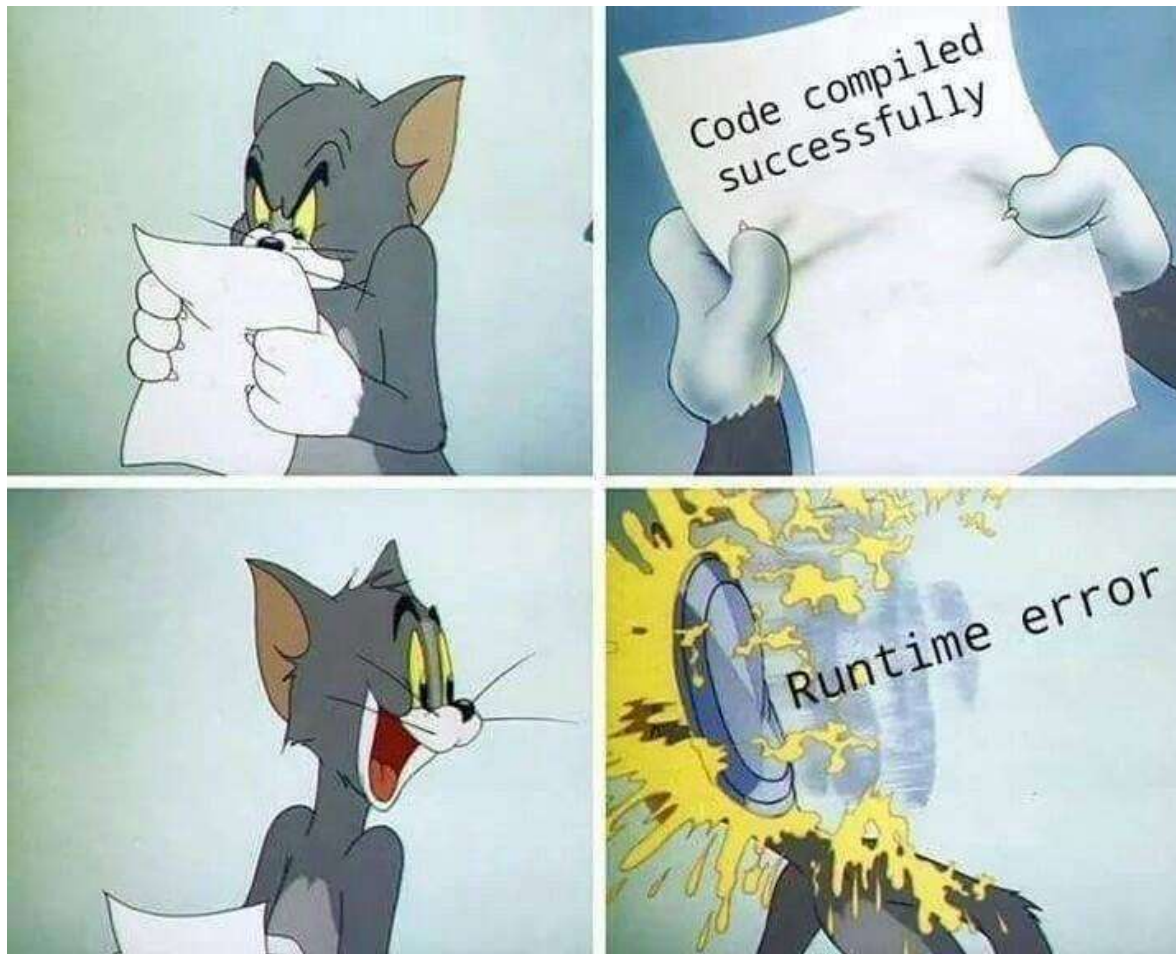
Erro de execução

(*runtime error*)

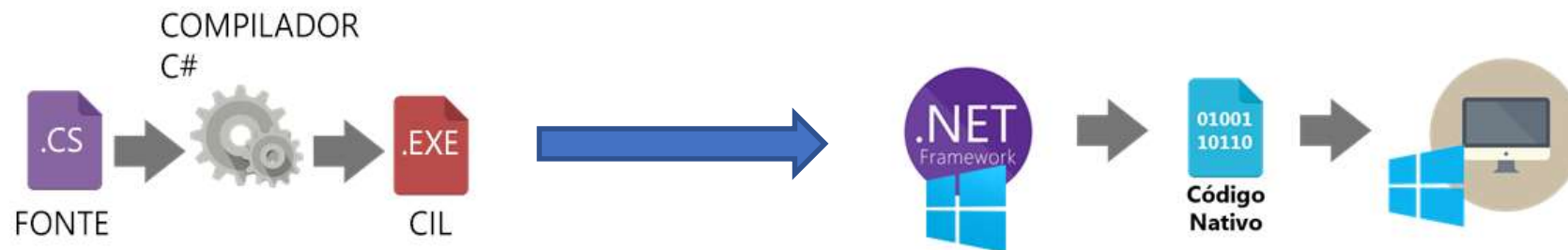


- Esses tipos de erros não são identificados pelo compilador. Um arquivo executável (.exe) será criado normalmente.
- Erros de *runtime* (erros de execução) acontecem enquanto o programa está sendo executado e podem interrompe-lo (“dar *crash*”).
- Erros de execução costumam estar relacionados com:
 - Entradas inválidas de usuários;
 - Uso de variáveis por referência que não apontam para nenhum objeto (apontam para null);
 - Tentar usar recursos (como arquivos) que não estão disponíveis;
 - Etc;

Tipos de Erros



Tipos de Erros



- **Erros de Compilação**

- Identificados pelo compilador;
- Acontecem antes da execução do programa;

- **Erros de Execução** (*runtime error*)

- Não identificados pelo compilador;
- Acontecem durante a execução do programa;
- Costumam gerar uma **Exceção** (**Exception**).

Exceções

Conceito teórico



- Uma exceção indica um problema que acontece durante a execução de um problema.
- O nome exceção vem do fato de que, apesar do problema poder acontecer, ele acontece infrequentemente.

Saber lidar com exceções permite criar aplicações mais tolerantes a falhas.

Exemplo 1

Ocorrência de uma exceção



```
class Program
{
    static void Main(string
    {
        Console.WriteLine("
        decimal numerador =

        Console.WriteLine("Por favor entre com um denominador:");
        decimal denominador = Convert.ToDecimal(Console.ReadLine());

        decimal resultado = numerador / denominador;
        Console.WriteLine("\nResultado: {0} / {1} = {2}",
                           numerador, denominador, resultado);

        Console.ReadLine();
    }
}
```

Exceção Sem Tratamento

System.FormatException: 'A cadeia de caracteres de entrada não estava em um formato correto.'

[Exibir Detalhes](#) | [Copiar Detalhes](#)

► Configurações de Exceção

Exemplo 2

Ocorrência de uma exceção



```
class Program
{
    static void Main(string
    {
        Console.WriteLine("
        decimal numerador =

        Console.WriteLine("Por favor entre com um denominador:");
        decimal denominador = Convert.ToDecimal(Console.ReadLine());

        decimal resultado = numerador / denominador;
        Console.WriteLine("\nResultado: {0} / {1} = {2}",
                           numerador, denominador, resultado);

        Console.ReadLine();
    }
}
```

Exceção Sem Tratamento

System.DivideByZeroException: 'Tentativa de divisão por zero.'

[Exibir Detalhes](#) | [Copiar Detalhes](#)

▸ Configurações de Exceção

Exceções

Conceito prático



- Uma exceção é um objeto que o .NET gera quando o seu programa encontra um erro de execução.
- Sim, uma exceção, na prática, é um objeto! Os seus atributos revelam informações a respeito do problema que ocorreu.
- Por exemplo, o atributo *Message* de um objeto *Exception* é uma string com a descrição do erro.

Quando uma exceção acontece, dizemos que o .NET lançou uma *Exception*.

Exemplo 1

Ocorrência de uma exceção



```
class Program
{
    static void Main(string[] args)
    {
        Console.WriteLine("Por favor entre com um denominador:");
        decimal denominador = Convert.ToDecimal(Console.ReadLine());

        decimal resultado = Convert.ToDecimal(Console.ReadLine());

        Console.WriteLine("O resultado da divisão é: " + resultado);

        Console.ReadLine();
    }
}
```

Exceção Sem Tratamento

System.FormatException: 'A cadeia de caracteres não estava em um formato correto.'

[Exibir Detalhes](#) | [Copiar Detalhes](#)

► Configurações de Exceção

Objeto do tipo (classe)
FormatException

```
decimal resultado =  
Console.WriteLine("O
```

Um objeto do tipo (classe) FormatException foi lançado durante a ocorrência do erro!

Exemplo 2

Ocorrência de uma exceção



```
class Program
{
    static void Main(string
    {
        Console.WriteLine("
        decimal numerador =

        Console.WriteLine("Por favor entre com um denominador:");
        decimal denominador = Convert.ToDecimal(Console.ReadLine());

        decimal resultado =
        Console.WriteLine("\n

        Console.ReadLine();
    }
}
```

Exceção Sem Tratamento

System.DivideByZeroException: 'Tentativa de di

Objeto do tipo (classe)
DivideByZeroException

[Exibir Detalhes](#) | [Copiar Detalhes](#)

▸ Configurações de Exceção

```
decimal resultado =
Console.WriteLine("\n
```

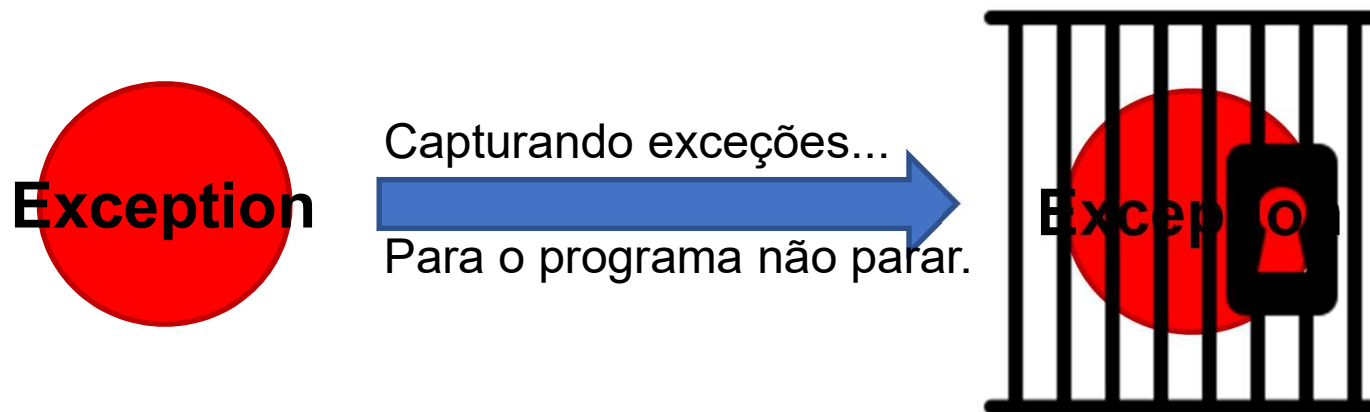
Um objeto do tipo (classe)
DivideByZeroException foi lançado durante a
ocorrência do erro!

Exceções

O que fazer com elas para não dar “**crash**” no programa?



- Quando uma exceção é lançada pelo .NET, ela pode ser capturada e tratada.



Se uma exceção não for capturada, o programa irá parar sua execução imediatamente na linha onde a Exception foi lançada.

Tratamento de exceções

blocos *try* e *catch*



- Sempre que no seu código você identificar que ele está sujeito a erros de execução, você pode basicamente dizer a ele:

“tente executar esse código e se uma exceção for lançada...”

try

“...capture-a e faça alguma coisa para dar continuidade ao seu programa”.

catch

Tratamento de exceções

blocos *try* e *catch*



- A parte do código que o programador acredita que pode dar erro e portanto quer **tentar** executar deve ser escrita dentro de um bloco **try**.
- A parte do código onde se **captura** e por sua vez lida com a exceção deve ser escrita dentro de um bloco **catch**.

No bloco de captura, você pode fazer coisas como apresentar uma mensagem de erro amigável, em vez de deixar seu programa ter um final desastroso.

Estruturando

o bloco try-catch



```
class Program
{
    static void Main(string[] args)
    {
        Console.WriteLine("Por favor entre com um numerador:");
        decimal numerador = Convert.ToDecimal(Console.ReadLine());

        Console.WriteLine("Por favor entre com um denominador:");
        decimal denominador = Convert.ToDecimal(Console.ReadLine());

        decimal resultado = numerador / denominador;
        Console.WriteLine("\nResultado: {0} / {1} = {2}",
                          numerador, denominador, resultado);

        Console.ReadLine();
    }
}
```

Qual trecho do código está
sujeito a lançar uma exceção?

Estruturando

o bloco try-catch



```
class Program
{
    static void Main(string[] args)
    {
        Console.WriteLine("Por favor entre com um numerador:");
        decimal numerador = Convert.ToDecimal(Console.ReadLine());

        Console.WriteLine("Por favor entre com um denominador:");
        decimal denominador = Convert.ToDecimal(Console.ReadLine());

        decimal resultado = numerador / denominador;
        Console.WriteLine("\nResultado: {0} / {1} = {2}",
                           numerador, denominador, resultado);

        Console.ReadLine();
    }
}
```

Qual trecho do código está
sujeito a lançar uma exceção?

Estruturando

o bloco try-catch



1º passo: criar o bloco try

```
class Program
{
    static void Main(string[] args)
    {
        try
        {
            Console.WriteLine("Por favor entre com um numerador:");
            decimal numerador = Convert.ToDecimal(Console.ReadLine());

            Console.WriteLine("Por favor entre com um denominador:");
            decimal denominador = Convert.ToDecimal(Console.ReadLine());

            decimal resultado = numerador / denominador;

            Console.WriteLine("\nResultado: {0} / {1} = {2}", numerador,
                               denominador, resultado);
        }

        Console.ReadLine();
    }
}
```

Estruturando

o bloco try-catch



Envolvemos o código sujeito a erros num bloco **try**.

1º passo: criar o bloco try

s Program

```
static void Main(string[] args)
{
    try
    {
        Console.WriteLine("Por favor entre com um numerador:");
        decimal numerador = Convert.ToDecimal(Console.ReadLine());

        Console.WriteLine("Por favor entre com um denominador:");
        decimal denominador = Convert.ToDecimal(Console.ReadLine());

        decimal resultado = numerador / denominador;

        Console.WriteLine("\nResultado: {0} / {1} = {2}", numerador,
            denominador, resultado);
    }

    Console.ReadLine();
}
```

Estruturando

o bloco try-catch



1º passo: criar o bloco try

```
class Program
{
    static void Main(string[] args)
    {
        try
        {
            Console.WriteLine("Por favor entre com um numerador:");
            decimal numerador = Convert.ToDecimal(Console.ReadLine());

            Console.WriteLine("Por favor entre com um denominador:");
            decimal denominador = Convert.ToDecimal(Console.ReadLine());

            decimal resultado = numerador / denominador;

            Console.WriteLine("\nResultado: {0} / {1} = {2}", numerador,
                               denominador, resultado);
        }

        Console.ReadLine();
    }
}
```

Precisamos do bloco catch para capturar qualquer exceção que ocorra dentro do bloco try.

Estruturando

o bloco try-catch



2º passo: criar o bloco catch

```
class Program
{
    static void Main(string[] args)
    {
        try
        {
            Console.WriteLine("Por favor entre com um numerador:");
            decimal numerador = Convert.ToDecimal(Console.ReadLine());

            Console.WriteLine("Por favor entre com um denominador:");
            decimal denominador = Convert.ToDecimal(Console.ReadLine());

            decimal resultado = numerador / denominador;

            Console.WriteLine("\nResultado: {0} / {1} = {2}", numerador,
                              denominador, resultado);
        }
        catch(Exception ex)
        {
            Console.WriteLine("Nome do erro: " + ex.Message);
            Console.WriteLine("Algum valor inválido foi passado... Tente novamente");
        }

        Console.ReadLine();
    }
}
```

Estruturando

o bloco try-catch



2º passo: criar o bloco catch

O bloco catch será executado apenas se uma exceção for lançada dentro do bloco try.

```
class Program
{
    static void Main(string[] args)
    {
        try
        {
            Console.WriteLine("Por favor entre com um numerador:");
            decimal numerador = Convert.ToDecimal(Console.ReadLine());

            Console.WriteLine("Por favor entre com um denominador:");
            decimal denominador = Convert.ToDecimal(Console.ReadLine());

            decimal resultado = numerador / denominador;

            Console.WriteLine("\nResultado: {0} / {1} = {2}", numerador,
                              denominador, resultado);
        }
        catch(Exception ex)
        {
            Console.WriteLine("Nome do erro: " + ex.Message);
            Console.WriteLine("Algum valor inválido foi passado... Tente novamente");
        }

        Console.ReadLine();
    }
}
```

Exercitando o Exemplo 1



Executando

o bloco try-catch

Console

Por favor entre com um numerador



```
class Program
{
    static void Main(string[] args)
    {
        try
        {
            Console.WriteLine("Por favor entre com um numerador:");
            decimal numerador = Convert.ToDecimal(Console.ReadLine());

            Console.WriteLine("Por favor entre com um denominador:");
            decimal denominador = Convert.ToDecimal(Console.ReadLine());

            decimal resultado = numerador / denominador;

            Console.WriteLine("\nResultado: {0} / {1} = {2}", numerador,
                              denominador, resultado);
        }
        catch(Exception ex)
        {
            Console.WriteLine("Nome do erro: " + ex.Message);
            Console.WriteLine("Algum valor inválido foi passado... Tente novamente");
        }

        Console.ReadLine();
    }
}
```


Executando

o bloco try-catch

Console

Por favor entre com um numerador
40



```
class Program
{
    static void Main(string[] args)
    {
        try
        {
            Console.WriteLine("Por favor entre com um numerador:");
            decimal numerador = Convert.ToDecimal(Console.ReadLine());

            Console.WriteLine("Por favor entre com um denominador:");
            decimal denominador = Convert.ToDecimal(Console.ReadLine());

            decimal resultado = numerador / denominador;

            Console.WriteLine("\nResultado: {0} / {1} = {2}", numerador,
                              denominador, resultado);
        }
        catch(Exception ex)
        {
            Console.WriteLine("Nome do erro: " + ex.Message);
            Console.WriteLine("Algum valor inválido foi passado... Tente novamente");
        }

        Console.ReadLine();
    }
}
```

Executando

o bloco try-catch

Console

Por favor entre com um numerador

40

Por favor entre com um denominador:



Colégio
Pedro II

```
class Program
{
    static void Main(string[] args)
    {
        try
        {
            Console.WriteLine("Por favor entre com um numerador:");
            decimal numerador = Convert.ToDecimal(Console.ReadLine());

            Console.WriteLine("Por favor entre com um denominador:");
            decimal denominador = Convert.ToDecimal(Console.ReadLine());

            decimal resultado = numerador / denominador;

            Console.WriteLine("\nResultado: {0} / {1} = {2}", numerador,
                               denominador, resultado);
        }
        catch (Exception ex)
        {
            Console.WriteLine("Nome do erro: " + ex.Message);
            Console.WriteLine("Algum valor inválido foi passado... Tente novamente");
        }

        Console.ReadLine();
    }
}
```

Executando

o bloco try-catch

Console

```
Por favor entre com um numerador
40
Por favor entre com um denominador:
hello
```



Colégio
Pedro II

```
class Program
{
    static void Main(string[] args)
    {
        try
        {
            Console.WriteLine("Por favor entre com um numerador:");
            decimal numerador = Convert.ToDecimal(Console.ReadLine());

            Console.WriteLine("Por favor entre com um denominador:");
            decimal denominador = Convert.ToDecimal(Console.ReadLine());

            decimal resultado = numerador / denominador;

            Console.WriteLine("\nResultado: {0} / {1} = {2}", numerador,
                              denominador, resultado);
        }
        catch (Exception ex)
        {
            Console.WriteLine("Nome do erro: " + ex.Message);
            Console.WriteLine("Algum valor inválido foi passado... Tente novamente");
        }

        Console.ReadLine();
    }
}
```

Executando

o bloco try-catch

Console

```
Por favor entre com um numerador
40
Por favor entre com um denominador:
hello
```



Colégio
Pedro II

```
class Program
{
    static void Main(string[] args)
    {
        try
        {
            Console.WriteLine("Por favor entre com um numerador:");
            decimal numerador = Convert.ToDecimal(Console.ReadLine());

            Console.WriteLine("Por favor entre com um denominador:");
            decimal denominador = Convert.ToDecimal(Console.ReadLine());

            decimal resultado = numerador / denominador;

            Console.WriteLine("\nResultado: {0} / {1} = {2}", numerador,
                              denominador, resultado);
        }
        catch (Exception ex)
        {
            Console.WriteLine("Nome do erro: " + ex.Message);
            Console.WriteLine("Algum valor inválido foi passado... Tente novamente");
        }

        Console.ReadLine();
    }
}
```

Objeto do tipo (classe)
FormatException



Executando o bloco try-catch

Console

Por favor entre com um numerador
40
Por favor entre com um denominador:
hello
Nome do erro: A cadeia de
caracteres de entrada não estava em
um formato correto.



Colégio
Pedro II

```
class Program
{
    static void Main(string[] args)
    {
        try
        {
            Console.WriteLine("Por favor entre com um numerador:");
            decimal numerador = Convert.ToDecimal(Console.ReadLine());

            Console.WriteLine("Por favor entre com um denominador:");
            decimal denominador = Convert.ToDecimal(Console.ReadLine());

            decimal resultado = numerador / denominador;

            Console.WriteLine("\nResultado: {0} / {1} = {2}", numerador,
                              denominador, resultado);
        }
        catch (Exception ex)
        {
            Console.WriteLine("Nome do erro: " + ex.Message);
            Console.WriteLine("Algum valor inválido foi passado... Tente novamente");
        }

        Console.ReadLine();
    }
}
```



Objeto **capturado** no bloco catch!
O programa não sofreu **crash**!

Executando

o bloco try-catch

Console

Por favor entre com um numerador
40
Por favor entre com um denominador:
hello
Nome do erro: A cadeia de
caracteres de entrada não estava em
um formato correto.
Algum valor inválido foi passado...
Tente novamente.



Colégio
Pedro II

```
class Program
{
    static void Main(string[] args)
    {
        try
        {
            Console.WriteLine("Por favor entre com um numerador:");
            decimal numerador = Convert.ToDecimal(Console.ReadLine());

            Console.WriteLine("Por favor entre com um denominador:");
            decimal denominador = Convert.ToDecimal(Console.ReadLine());

            decimal resultado = numerador / denominador;

            Console.WriteLine("\nResultado: {0} / {1} = {2}", numerador,
                               denominador, resultado);
        }
        catch (Exception ex)
        {
            Console.WriteLine("Nome do erro: " + ex.Message);
            Console.WriteLine("Algum valor inválido foi passado... Tente novamente");
        }

        Console.ReadLine();
    }
}
```



Executando

o bloco try-catch

Console

Por favor entre com um numerador
40
Por favor entre com um denominador:
Hello
Nome do erro: A cadeia de
caracteres de entrada não estava em
um formato correto.
Algum valor inválido foi passado...
Tente novamente.



Colégio
Pedro II

```
class Program
{
    static void Main(string[] args)
    {
        try
        {
            Console.WriteLine("Por favor entre com um numerador:");
            decimal numerador = Convert.ToDecimal(Console.ReadLine());

            Console.WriteLine("Por favor entre com um denominador:");
            decimal denominador = Convert.ToDecimal(Console.ReadLine());

            decimal resultado = numerador / denominador;

            Console.WriteLine("\nResultado: {0} / {1} = {2}", numerador,
                              denominador, resultado);
        }
        catch (Exception ex)
        {
            Console.WriteLine("Nome do erro: " + ex.Message);
            Console.WriteLine("Algum valor inválido foi passado... Tente novamente");
        }

        Console.ReadLine();
    }
}
```



Exercitando o Exemplo 2



EXAMPLE

2

Executando

o bloco try-catch

Console

Por favor entre com um numerador



Colégio
Pedro II

```
class Program
{
    static void Main(string[] args)
    {
        try
        {
            Console.WriteLine("Por favor entre com um numerador:");
            decimal numerador = Convert.ToDecimal(Console.ReadLine());

            Console.WriteLine("Por favor entre com um denominador:");
            decimal denominador = Convert.ToDecimal(Console.ReadLine());

            decimal resultado = numerador / denominador;

            Console.WriteLine("\nResultado: {0} / {1} = {2}", numerador,
                              denominador, resultado);
        }
        catch (Exception ex)
        {
            Console.WriteLine("Nome do erro: " + ex.Message);
            Console.WriteLine("Algum valor inválido foi passado... Tente novamente");
        }

        Console.ReadLine();
    }
}
```

Executando

o bloco try-catch

Console

Por favor entre com um numerador
40



```
class Program
{
    static void Main(string[] args)
    {
        try
        {
            Console.WriteLine("Por favor entre com um numerador:");
            decimal numerador = Convert.ToDecimal(Console.ReadLine());

            Console.WriteLine("Por favor entre com um denominador:");
            decimal denominador = Convert.ToDecimal(Console.ReadLine());

            decimal resultado = numerador / denominador;

            Console.WriteLine("\nResultado: {0} / {1} = {2}", numerador,
                              denominador, resultado);
        }
        catch(Exception ex)
        {
            Console.WriteLine("Nome do erro: " + ex.Message);
            Console.WriteLine("Algum valor inválido foi passado... Tente novamente");
        }

        Console.ReadLine();
    }
}
```

Executando

o bloco try-catch

Console

Por favor entre com um numerador

40

Por favor entre com um denominador:



Colégio
Pedro II

```
class Program
{
    static void Main(string[] args)
    {
        try
        {
            Console.WriteLine("Por favor entre com um numerador:");
            decimal numerador = Convert.ToDecimal(Console.ReadLine());

            Console.WriteLine("Por favor entre com um denominador:");
            decimal denominador = Convert.ToDecimal(Console.ReadLine());

            decimal resultado = numerador / denominador;

            Console.WriteLine("\nResultado: {0} / {1} = {2}", numerador,
                               denominador, resultado);
        }
        catch (Exception ex)
        {
            Console.WriteLine("Nome do erro: " + ex.Message);
            Console.WriteLine("Algum valor inválido foi passado... Tente novamente");
        }

        Console.ReadLine();
    }
}
```

Executando

o bloco try-catch

Console

```
Por favor entre com um numerador
40
Por favor entre com um denominador:
0
```



Colégio
Pedro II

```
class Program
{
    static void Main(string[] args)
    {
        try
        {
            Console.WriteLine("Por favor entre com um numerador:");
            decimal numerador = Convert.ToDecimal(Console.ReadLine());

            Console.WriteLine("Por favor entre com um denominador:");
            decimal denominador = Convert.ToDecimal(Console.ReadLine());

            decimal resultado = numerador / denominador;

            Console.WriteLine("\nResultado: {0} / {1} = {2}", numerador,
                              denominador, resultado);
        }
        catch (Exception ex)
        {
            Console.WriteLine("Nome do erro: " + ex.Message);
            Console.WriteLine("Algum valor inválido foi passado... Tente novamente");
        }

        Console.ReadLine();
    }
}
```

Executando

o bloco try-catch

Console

```
Por favor entre com um numerador
40
Por favor entre com um denominador:
0
```



Colégio
Pedro II

```
class Program
{
    static void Main(string[] args)
    {
        try
        {
            Console.WriteLine("Por favor entre com um numerador:");
            decimal numerador = Convert.ToDecimal(Console.ReadLine());

            Console.WriteLine("Por favor entre com um denominador:");
            decimal denominador = Convert.ToDecimal(Console.ReadLine());

            ➡ decimal resultado = numerador / denominador;

            Console.WriteLine("\nResultado: {0} / {1} = {2}", numerador,
                               denominador, resultado);
        }
        catch (Exception ex)
        {
            Console.WriteLine("Nome do erro: " + ex.Message);
            Console.WriteLine("Algum valor inválido foi passado... Tente novamente");
        }

        Console.ReadLine();
    }
}
```

Executando

o bloco try-catch

Console

```
Por favor entre com um numerador
40
Por favor entre com um denominador:
0
```



Colégio
Pedro II

```
class Program
{
    static void Main(string[] args)
    {
        try
        {
            Console.WriteLine("Por favor entre com um numerador:");
            decimal numerador = Convert.ToDecimal(Console.ReadLine());

            Console.WriteLine("Por favor entre com um denominador ");
            decimal denominador = Convert.ToDecimal(Console.ReadLine());

            decimal resultado = numerador / denominador;

            Console.WriteLine("\nResultado: {0} / {1} = {2}", numerador,
                              denominador, resultado);
        }
        catch (Exception ex)
        {
            Console.WriteLine("Nome do erro: " + ex.Message);
            Console.WriteLine("Algum valor inválido foi passado... Tente
                               novamente");
        }

        Console.ReadLine();
    }
}
```

Objeto do tipo (classe)
DivideByZeroException

Executando o bloco try-catch

Console

Por favor entre com um numerador
40
Por favor entre com um denominador:
0
Nome do erro: Tentativa de divisão
por zero.



Colégio
Pedro II

```
class Program
{
    static void Main(string[] args)
    {
        try
        {
            Console.WriteLine("Por favor entre com um numerador:");
            decimal numerador = Convert.ToDecimal(Console.ReadLine());

            Console.WriteLine("Por favor entre com um denominador:");
            decimal denominador = Convert.ToDecimal(Console.ReadLine());

            decimal resultado = numerador / denominador;

            Console.WriteLine("\nResultado: {0} / {1} = {2}", numerador,
                              denominador, resultado);
        }
        catch (Exception ex)
        {
            Console.WriteLine("Nome do erro: " + ex.Message);
            Console.WriteLine("Algum valor inválido foi passado... Tente novamente");
        }

        Console.ReadLine();
    }
}
```



Objeto **capturado** no bloco catch!
O programa não sofreu **crash**!

Executando o bloco try-catch

Console

Por favor entre com um numerador
40
Por favor entre com um denominador:
0
Nome do erro: Tentativa de divisão
por zero.
Algum valor inválido foi passado...
Tente novamente.



Colégio
Pedro II

```
class Program
{
    static void Main(string[] args)
    {
        try
        {
            Console.WriteLine("Por favor entre com um numerador:");
            decimal numerador = Convert.ToDecimal(Console.ReadLine());

            Console.WriteLine("Por favor entre com um denominador:");
            decimal denominador = Convert.ToDecimal(Console.ReadLine());

            decimal resultado = numerador / denominador;

            Console.WriteLine("\nResultado: {0} / {1} = {2}", numerador,
                               denominador, resultado);
        }
        catch (Exception ex)
        {
            Console.WriteLine("Nome do erro: " + ex.Message);
            Console.WriteLine("Algum valor inválido foi passado... Tente
                               novamente");
        }

        Console.ReadLine();
    }
}
```



Executando

o bloco try-catch

Console

Por favor entre com um numerador
40
Por favor entre com um denominador:
0
Nome do erro: Tentativa de divisão
por zero.
Algum valor inválido foi passado...
Tente novamente.



```
class Program
{
    static void Main(string[] args)
    {
        try
        {
            Console.WriteLine("Por favor entre com um numerador:");
            decimal numerador = Convert.ToDecimal(Console.ReadLine());

            Console.WriteLine("Por favor entre com um denominador:");
            decimal denominador = Convert.ToDecimal(Console.ReadLine());

            decimal resultado = numerador / denominador;

            Console.WriteLine("\nResultado: {0} / {1} = {2}", numerador,
                              denominador, resultado);
        }
        catch (Exception ex)
        {
            Console.WriteLine("Nome do erro: " + ex.Message);
            Console.WriteLine("Algum valor inválido foi passado... Tente novamente");
        }

        Console.ReadLine();
    }
}
```

Exceções diferentes



- No exemplo que fizemos, repare que estamos tratando tanto a Exceção `FormatException` quanto a `DivideByZeroException` da mesma maneira.
- Se for necessário tratar exceções de tipos diferentes de maneiras diferentes, basta destacar a captura adequada no bloco `catch`.
- Quando a exceção for lançada, o bloco `catch` que irá capturá-la será o primeiro que melhor se encaixa no seu tipo.

Tipos de Exceções

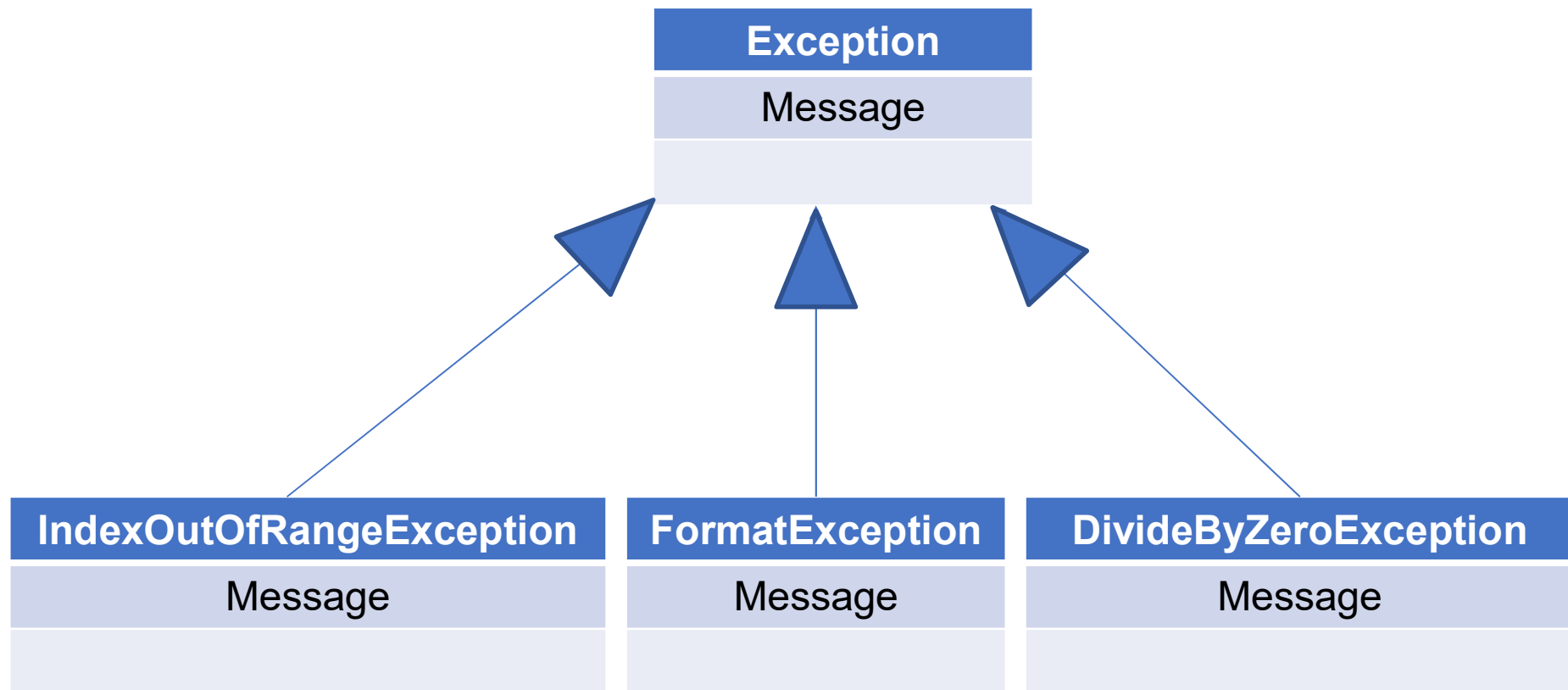
Hierarquia de Exceções .NET



- Há muitas formas de escrever-se código que o C# não sabe como lidar. Ou seja, muitos tipos de problemas diferentes podem acontecer em tempo de execução.
- Todos esses erros são Exceções mas cada um deles apresenta características específicas do seu problema.
- Por esse motivo, os diferentes tipos de exceções que podem vir a ocorrer são implementados utilizando o conceito de herança.

Tipos de Exceções

Hierarquia de Exceções .NET



Tipos de Exceções

Hierarquia de Exceções .NET



- Todos os objetos de exceção herdam de **Exception**.
- Em C#, o mecanismo de tratamento de exceção permite que apenas objetos da classe `Exception` (namespace *System*) e suas classes derivadas possam ser lançadas.
- A classe `Exception` tem alguns atributos e métodos úteis. A propriedade `Message`, por exemplo, armazena uma mensagem compreensível sobre o erro.

Console

Por favor entre com um numerador
40
Por favor entre com um denominador:
0



```
class Program
{
    static void Main(string[] args)
    {
        try
        {
            Console.WriteLine("Por favor entre com um numerador:");
            decimal numerador = Convert.ToDecimal(Console.ReadLine());

            Console.WriteLine("Por favor entre com um denominador:");
            decimal denominador = Convert.ToDecimal(Console.ReadLine());

            decimal resultado = numerador / denominador;

            Console.WriteLine("\nResultado: {0} / {1} = {2}", numerador, denominador, resultado);
        }
        catch (FormatException ex)
        {
            Console.WriteLine("Por favor, entre apenas com valores numéricos");
        }
        catch (DivideByZeroException ex)
        {
            Console.WriteLine("Quando se divide um número por zero, o resultado é um número indefinido!");
        }
        catch (Exception ex)
        {
            Console.WriteLine("Nome do erro: " + ex.Message);
            Console.WriteLine("Algum valor inválido foi passado... Tente novamente.");
        }
        Console.ReadLine();
    }
}
```

Console

Por favor entre com um numerador
40
Por favor entre com um denominador:
0



```
class Program
{
    static void Main(string[] args)
    {
        try
        {
            Console.WriteLine("Por favor entre com um numerador:");
            decimal numerador = Convert.ToDecimal(Console.ReadLine());

            Console.WriteLine("Por favor entre com um denominador:");
            decimal denominador = Convert.ToDecimal(Console.ReadLine());

            decimal resultado = numerador / denominador;

            Console.WriteLine("\nResultado: {0} / {1} = {2}", numerador, denominador, resultado);
        }
        catch (FormatException ex)
        {
            Console.WriteLine("Por favor, entre apenas com valores numéricos");
        }
        catch (DivideByZeroException ex)
        {
            Console.WriteLine("Quando se divide um número por zero, o resultado é um número indefinido!");
        }
        catch (Exception ex)
        {
            Console.WriteLine("Nome do erro: " + ex.Message);
            Console.WriteLine("Algum valor inválido foi passado... Tente novamente.");
        }
        Console.ReadLine();
    }
}
```

Objeto do tipo (classe)
DivideByZeroException

Console

Por favor entre com um numerador
40
Por favor entre com um denominador:
0
Quando se divide um número por
zero, o resultado é um número
indefinido!



```
class Program
{
    static void Main(string[] args)
    {
        try
        {
            Console.WriteLine("Por favor entre com um numerador:");
            decimal numerador = Convert.ToDecimal(Console.ReadLine());

            Console.WriteLine("Por favor entre com um denominador:");
            decimal denominador = Convert.ToDecimal(Console.ReadLine());

            decimal resultado = numerador / denominador;

            Console.WriteLine("\nResultado: {0} / {1} = {2}", numerador, denominador, resultado);
        }
        catch (FormatException ex)
        {
            Console.WriteLine("Por favor, entre apenas com valores numéricos");
        }
        catch (DivideByZeroException ex)
        {
            Console.WriteLine("Quando se divide um número por zero, o resultado é um número indefinido!");
        }
        catch (Exception ex)
        {
            Console.WriteLine("Nome do erro: " + ex.Message);
            Console.WriteLine("Algum valor inválido foi passado... Tente novamente.");
        }
        Console.ReadLine();
    }
}
```



Console

Por favor entre com um numerador
40
Por favor entre com um denominador:
0
Quando se divide um número por
zero, o resultado é um número
indefinido!



```
class Program
{
    static void Main(string[] args)
    {
        try
        {
            Console.WriteLine("Por favor entre com um numerador:");
            decimal numerador = Convert.ToDecimal(Console.ReadLine());

            Console.WriteLine("Por favor entre com um denominador:");
            decimal denominador = Convert.ToDecimal(Console.ReadLine());

            decimal resultado = numerador / denominador;

            Console.WriteLine("\nResultado: {0} / {1} = {2}", numerador, denominador, resultado);
        }
        catch (FormatException ex)
        {
            Console.WriteLine("Por favor, entre apenas com valores numéricos");
        }
        catch (DivideByZeroException ex)
        {
            Console.WriteLine("Quando se divide um número por zero, o resultado é um número indefinido!");
        }
        catch (Exception ex)
        {
            Console.WriteLine("Nome do erro: " + ex.Message);
            Console.WriteLine("Algum valor inválido foi passado... Tente novamente.");
        }
        Console.ReadLine();
    }
}
```

Console

Por favor entre com um numerador
40
Por favor entre com um denominador:
hello



```
class Program
{
    static void Main(string[] args)
    {
        try
        {
            Console.WriteLine("Por favor entre com um numerador:");
            decimal numerador = Convert.ToDecimal(Console.ReadLine());

            Console.WriteLine("Por favor entre com um denominador:");
            decimal denominador = Convert.ToDecimal(Console.ReadLine());

            decimal resultado = numerador / denominador;

            Console.WriteLine("\nResultado: {0} / {1} = {2}", numerador, denominador, resultado);
        }
        catch (FormatException ex)
        {
            Console.WriteLine("Por favor, entre apenas com valores numéricos");
        }
        catch (DivideByZeroException ex)
        {
            Console.WriteLine("Quando se divide um número por zero, o resultado é um número indefinido!");
        }
        catch (Exception ex)
        {
            Console.WriteLine("Nome do erro: " + ex.Message);
            Console.WriteLine("Algum valor inválido foi passado... Tente novamente.");
        }
        Console.ReadLine();
    }
}
```

Console

Por favor entre com um numerador
40
Por favor entre com um denominador:
hello



```
class Program
{
    static void Main(string[] args)
    {
        try
        {
            Console.WriteLine("Por favor entre com um numerador:");
            decimal numerador = Convert.ToDecimal(Console.ReadLine());

            Console.WriteLine("Por favor entre com um denominador:");
            decimal denominador = Convert.ToDecimal(Console.ReadLine());

            decimal resultado = numerador / denominador;

            Console.WriteLine("\nResultado: {0} / {1} = {2}", numerador, denominador, resultado);
        }
        catch (FormatException ex)
        {
            Console.WriteLine("Por favor, entre apenas com valores numéricos");
        }
        catch (DivideByZeroException ex)
        {
            Console.WriteLine("Quando se divide um número por zero, o resultado é um número indefinido!");
        }
        catch (Exception ex)
        {
            Console.WriteLine("Nome do erro: " + ex.Message);
            Console.WriteLine("Algum valor inválido foi passado... Tente novamente.");
        }
        Console.ReadLine();
    }
}
```

Objeto do tipo (classe)
FormatException

Console

Por favor entre com um numerador
40
Por favor entre com um denominador:
hello
Por favor, entre apenas com valores
numéricos



```
class Program
{
    static void Main(string[] args)
    {
        try
        {
            Console.WriteLine("Por favor entre com um numerador:");
            decimal numerador = Convert.ToDecimal(Console.ReadLine());

            Console.WriteLine("Por favor entre com um denominador:");
            decimal denominador = Convert.ToDecimal(Console.ReadLine());

            decimal resultado = numerador / denominador;

            Console.WriteLine("\nResultado: {0} / {1} = {2}", numerador, denominador, resultado);
        }
        catch (FormatException ex)
        {
            Console.WriteLine("Por favor, entre apenas com valores numéricos");
        }
        catch (DivideByZeroException ex)
        {
            Console.WriteLine("Quando se divide um número por zero, o resultado é um número indefinido!");
        }
        catch (Exception ex)
        {
            Console.WriteLine("Nome do erro: " + ex.Message);
            Console.WriteLine("Algum valor inválido foi passado... Tente novamente.");
        }
        Console.ReadLine();
    }
}
```



Console

Por favor entre com um numerador
40
Por favor entre com um denominador:
hello
Por favor, entre apenas com valores
numéricos



```
class Program
{
    static void Main(string[] args)
    {
        try
        {
            Console.WriteLine("Por favor entre com um numerador:");
            decimal numerador = Convert.ToDecimal(Console.ReadLine());

            Console.WriteLine("Por favor entre com um denominador:");
            decimal denominador = Convert.ToDecimal(Console.ReadLine());

            decimal resultado = numerador / denominador;

            Console.WriteLine("\nResultado: {0} / {1} = {2}", numerador, denominador, resultado);
        }
        catch (FormatException ex)
        {
            Console.WriteLine("Por favor, entre apenas com valores numéricos");
        }
        catch (DivideByZeroException ex)
        {
            Console.WriteLine("Quando se divide um número por zero, o resultado é um número indefinido!");
        }
        catch (Exception ex)
        {
            Console.WriteLine("Nome do erro: " + ex.Message);
            Console.WriteLine("Algum valor inválido foi passado... Tente novamente.");
        }
        Console.ReadLine();
    }
}
```

Quando devo usar try e catch?



- Quando estiver escrevendo código que apresente riscos, ou seja, que poderia lançar exceções. **O truque é saber quando o código é seguro e quando apresenta riscos.**
- Exemplo: você sabe que código que usa dados informados pelo usuário apresenta riscos. Usuários informam arquivos errados, palavras no lugar de números, nomes no lugar de datas e eles clicam em qualquer lugar possível e imaginável.

Quando devo usar try e catch?



- Um bom aplicativo deve saber lidar com todas essas situações de forma calma e previsível. Pode ser que ele não consiga apresentar aos usuários uma resposta útil, mas ele deve evitar um fechamento inesperado.

O seu programa parou de funcionar inesperadamente...



Foi mal...

Exemplo mais prático

Exemplo 3



nomes - Bloco de notas

Arquivo Editar Formatar Exibir Ajuda

João Lagôas
Henrique Araujo
André Almeida
Letícia Amaro
Flávio Costa
Natália Souza
Cláudio Passos
Gustavo Geraldo
Luiza Lagôas

profissoes - Bloco d...

Arquivo Editar Formatar Exibir Ajuda

Professor
Analista
Preparador
Engenheiro
Professor
Engenheiro
Professor
Empresário
Assistente Social

Profissão

Professor

Listar Nomes

João Lagôas
Flávio Costa
Cláudio Passos

Quantidade 3

Exceções costumam acontecer quando um aplicativo processa recursos externos.


```
public partial class Form1 : Form
{
    string caminhoNomes = @"C:\Users\João\OneDrive\Arquivos\nomes.txt";
    string caminhoProfissoes = @"C:\Users\João\OneDrive\Arquivos\profissoes.txt";

    public Form1()
    {
        InitializeComponent();
    }

    private void button1_Click(object sender, EventArgs e)
    {
        listBox1.Items.Clear();

        string profissaoEscolhida = textBox2.Text;

        StreamReader leitorDeNomes = null;
        StreamReader leitorDeProfissoes = null;

        leitorDeNomes = new StreamReader(caminhoNomes);
        leitorDeProfissoes = new StreamReader(caminhoProfissoes);

        while (!leitorDeNomes.EndOfStream)
        {
            string nome = leitorDeNomes.ReadLine();
            string profissao = leitorDeProfissoes.ReadLine();

            if (profissao == profissaoEscolhida)
                listBox1.Items.Add(nome);
        }

        leitorDeNomes.Close();
        leitorDeProfissoes.Close();

        textBox1.Text = Convert.ToString(listBox1.Items.Count);
    }
}
```

Em qual trecho do código há risco de uma exceção ser lançada?

```

public partial class Form1 : Form
{
    string caminhoNomes = @"C:\Users\João\OneDrive\Arquivos\nomes.txt";
    string caminhoProfissoes = @"C:\Users\João\OneDrive\Arquivos\profissoes.txt";

    public Form1()
    {
        InitializeComponent();
    }

    private void button1_Click(object sender, EventArgs e)
    {
        listBox1.Items.Clear();

        string profissaoEscolhida = textBox2.Text;

        StreamReader leitorDeNomes = null;
        StreamReader leitorDeProfissoes = null;

        leitorDeNomes = new StreamReader(caminhoNomes);
        leitorDeProfissoes = new StreamReader(caminhoProfissoes);

        while (!leitorDeNomes.EndOfStream)
        {
            string nome = leitorDeNomes.ReadLine();
            string profissao = leitorDeProfissoes.ReadLine();

            if (profissao == profissaoEscolhida)
                listBox1.Items.Add(nome);
        }

        leitorDeNomes.Close();
        leitorDeProfissoes.Close();

        textBox1.Text = Convert.ToString(listBox1.Items.Count);
    }
}

```

Dependendo da máquina onde esse programa seja executado, os arquivos nomes.txt e profissões.txt podem não existir!

O construtor StreamReader iria lançar uma exceção.

```

public partial class Form1 : Form
{
    string caminhoNomes = @"C:\Users\João\OneDrive\Arquivos\nomes.txt";
    string caminhoProfissoes = @"C:\Users\João\OneDrive\Arquivos\profissoes.txt";

    public Form1()
    {
        InitializeComponent();
    }

    private void button1_Click(object sender, EventArgs e)
    {
        listBox1.Items.Clear();

        string profissaoEscolhida = textBox2.Text;

        StreamReader leitorDeNomes = null;
        StreamReader leitorDeProfissoes = null;

        try
        {
            leitorDeNomes = new StreamReader(caminhoNomes);
            leitorDeProfissoes = new StreamReader(caminhoProfissoes);

            while (!leitorDeNomes.EndOfStream)
            {
                string nome = leitorDeNomes.ReadLine();
                string profissao = leitorDeProfissoes.ReadLine();

                if (profissao == profissaoEscolhida)
                    listBox1.Items.Add(nome);
            }
        }
        catch (Exception ex)
        {
            MessageBox.Show("Um erro aconteceu. Verifique a seguinte mensagem: " + ex.Message);
        }

        leitorDeNomes.Close();
        leitorDeProfissoes.Close();

        textBox1.Text = Convert.ToString(listBox1.Items.Count);
    }
}

```

Quase tudo resolvido!

No entanto, queremos ter a certeza de que os leitores sempre serão fechados independentemente se ocorrer exceções ou não!

Quero garantir que se o leitor foi criado com sucesso (seu construtor funcionar) então ele será fechado.

```

public partial class Form1 : Form
{
    string caminhoNomes = @"C:\Users\João\OneDrive\Arquivos\nomes.txt";
    string caminhoProfissoes = @"C:\Users\João\OneDrive\Arquivos\profissoes.txt";

    public Form1()
    {
        InitializeComponent();
    }

    private void button1_Click(object sender, EventArgs e)
    {
        listBox1.Items.Clear();

        string profissaoEscolhida = textBox2.Text;

        StreamReader leitorDeNomes = null;
        StreamReader leitorDeProfissoes = null;

        try
        {
            leitorDeNomes = new StreamReader(caminhoNomes);
            leitorDeProfissoes = new StreamReader(caminhoProfissoes);

            while (!leitorDeNomes.EndOfStream)
            {
                string nome = leitorDeNomes.ReadLine();
                string profissao = leitorDeProfissoes.ReadLine();

                if (profissao == profissaoEscolhida)
                    listBox1.Items.Add(nome);
            }
        }
        catch (Exception ex)
        {
            MessageBox.Show("Um erro aconteceu. Verifique a seguinte mensagem: " + ex.Message);
        }
        finally
        {
            leitorDeNomes.Close();
            leitorDeProfissoes.Close();
        }

        textBox1.Text = Convert.ToString(listBox1.Items.Count);
    }
}

```

```

public partial class Form1 : Form
{
    string caminhoNomes = @"C:\Users\João\OneDrive\Arquivos\nomes.txt";
    string caminhoProfissoes = @"C:\Users\João\OneDrive\Arquivos\profissoes.txt";

    public Form1()
    {
        InitializeComponent();
    }

    private void button1_Click(object sender, EventArgs e)
    {
        listBox1.Items.Clear();

        string profissaoEscolhida = textBox2.Text;

        StreamReader leitorDeNomes = null;
        StreamReader leitorDeProfissoes = null;

        try
        {
            leitorDeNomes = new StreamReader(caminhoNomes);
            leitorDeProfissoes = new StreamReader(caminhoProfissoes);

            while (!leitorDeNomes.EndOfStream)
            {
                string nome = leitorDeNomes.ReadLine();
                string profissao = leitorDeProfissoes.ReadLine();

                if (profissao == profissaoEscolhida)
                    listBox1.Items.Add(nome);
            }
        }
        catch (Exception ex)
        {
            MessageBox.Show("Um erro aconteceu. Verifique a seguinte mensagem: " + ex.Message);
        }
        finally
        {
            if (leitorDeNomes != null) leitorDeNomes.Close();
            if (leitorDeProfissoes != null) leitorDeProfissoes.Close();
        }

        textBox1.Text = Convert.ToString(listBox1.Items.Count);
    }
}

```

Pronto! Sabemos que um bloco finally SEMPRE será executado depois do try OU do catch.

Bloco finally

comumente usado com arquivos



- Código que processa arquivos normalmente aparecem dentro de blocos try.
- O bloco finally SEMPRE é executado independentemente se no bloco try uma exceção foi lançada ou não.
- Isso torna o bloco finally um local ideal para se colocar código de liberação de recursos que são adquiridos no bloco try.

Lançando Exceções

Exceções do programador



- Até agora entendemos como o sistema de exceções funciona. No entanto, estamos apenas capturando exceções que o próprio .NET lança ou que estão implementadas em outros métodos.
- E se na NOSSA classe, nós quiséssemos lançar exceções para indicar anomalias?
- Podemos lançar Exceptions em nossos próprios métodos de classe. Mas não esqueça de envolver o código desses métodos em blocos try/catch.

Lançando Exceções

Exceções do programador



- No exemplo abaixo, o construtor de FunçãoQuadrática não deve aceitar que o atributo A receba valores nulos.

```
class FuncaoQuadratica
{
    private decimal A = 1;
    public decimal B;
    public decimal C;

    public FuncaoQuadratica(decimal a, decimal b, decimal c)
    {
        SetA(a)
        B = b;
        C = c;
    }

    public decimal F(decimal x)
    {
        return A * x * x + B * x + C;
    }

    public decimal GetA() { return A; }
    public bool SetA(decimal a) { if (a == 0) return false; else { A = a; return true; } }
}
```

1

Solução: impor que o valor de A será 1 caso zero seja passado como parâmetro.

Lançando Exceções

Exceções do programador



- Podemos instanciar um objeto Exception e lança-lo através da palavra reservada **throw**.

```
class FuncaoQuadratica
{
    private decimal A = 1;
    public decimal B;
    public decimal C;

    public FuncaoQuadratica(decimal a, decimal b, decimal c)
    {
        if(a == 0) throw new Exception;
        B = b;
        C = c;
    }

    public decimal F(decimal x)
    {
        return A * x * x + B * x + C;
    }

    public decimal GetA() { return A; }
    public bool SetA(decimal a) { if (a == 0) return false; else { A = a; return true; } }
}
```

2

Solução: fazer com o que o método construtor instancie e lance uma exceção para que seja capturada. Atenção: o objeto NÃO SERÁ MAIS CRIADO.

Lançando Exceções

Exceções do programador



```
private void button1_Click(object sender, EventArgs e)
{
    decimal a = numericUpDown1.Value;
    decimal b = numericUpDown2.Value;
    decimal c = numericUpDown3.Value;
    decimal x = numericUpDown4.Value;

    FuncaoQuadratica fq = null;

    fq = new FuncaoQuadratica(a, b, c);

    decimal y = fq.F(x);

    numericUpDown5.Value = y;
}
```

Suponha que o botão foi clicado com A valendo 0 na numericUpDown1

Lançando Exceções

Exceções do programador



```
private void button1_Click(object sender, EventArgs e)
```

```
{
```

```
    decimal a = numericUpDown1.Value;
```

```
    decimal b = numericUpDown2.Value;
```

```
    decimal c = numericUpDown3.Value;
```

```
    decimal x = numericUpDown4.Value;
```

```
    FuncaoQuadratica fq = null;
```

```
    → fq = new FuncaoQuadratica(a, b, c);
```

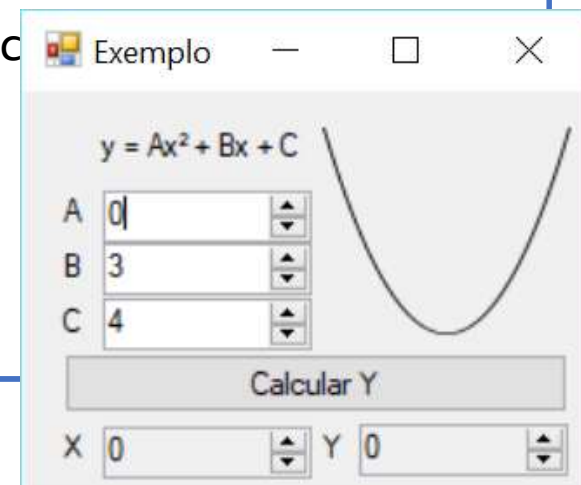
```
    decimal y = fq.F(x);
```

```
    numericUpDown5.Value = y;
```

```
}
```

Objeto do tipo (classe)
Exception

O programa irá parar já que uma exceção foi lançada e não capturada!



```
private void x(object sender, EventArgs e)
```

```
{
```

```
    decimal a = numericUpDown1.Value;
```

```
    decimal b = numericUpDown2.Value;
```

```
    decimal c = numericUpDown3.Value;
```

```
    decimal x = numericUpDown4.Value;
```

```
    FuncaoQuadratica fq = null;
```

```
    try
```

```
    {
```

```
        fq = new FuncaoQuadratica(a, b, c);
```

```
    }
```

```
    catch (Exception ex)
```

```
    {
```

```
        MessageBox.Show(ex.Message);
```

```
        return;
```

```
    }
```

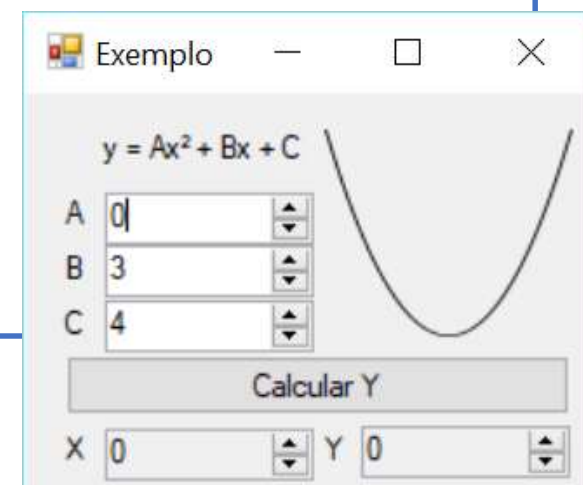
```
    decimal y = fq.F(x);
```

```
    numericUpDown5.Value = y;
```

```
}
```

Objeto do tipo (classe)

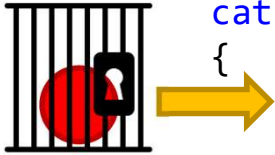
Exception



Com um bloco try catch poderíamos resolver esse problema!

```
private void x(object sender, EventArgs e)
{
    decimal a = numericUpDown1.Value;
    decimal b = numericUpDown2.Value;
    decimal c = numericUpDown3.Value;
    decimal x = numericUpDown4.Value;

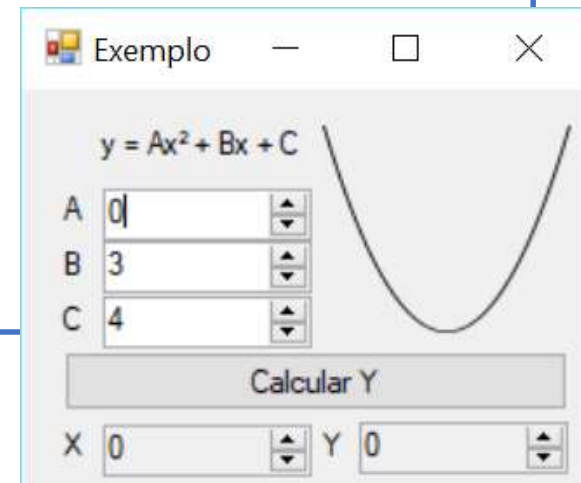
    FuncaoQuadratica fq = null;

    try
    {
        fq = new FuncaoQuadratica(a, b, c);
    }
    catch (Exception ex)
    {
        
        MessageBox.Show(ex.Message);
        return;
    }

    decimal y = fq.F(x);

    numericUpDown5.Value = y;
}
```

Com um bloco try catch poderíamos resolver esse problema!



Exemplo

$y = Ax^2 + Bx + C$

A 0

B 3

C 4

Calcular Y

X 0 Y 0

Conclusão



- Exceção tem tudo a ver com ajudar o programador a esperar o inesperado.
- Pense nas exceções como uma forma de ajuda-lo a encontrar e depurar seu programa, isso realmente ajudará quando você estiver tentando achar o que fez o código lançar o erro.