



Tratamento de Exceções

Linguagem de Programação III

Prof. Edson Mota, PhD Student, MSc, PMP

O que é uma **Exceção**?



Mundo Ideal

- Os dados estão sempre nos formatos corretos , arquivos possuem tamanhos adequados, as entradas dos usuários são sempre precisas, etc..



Mundo Real

- As entradas não são sempre corretas, existem bugs e outros problemas que podem arruinar a experiência do usuário...

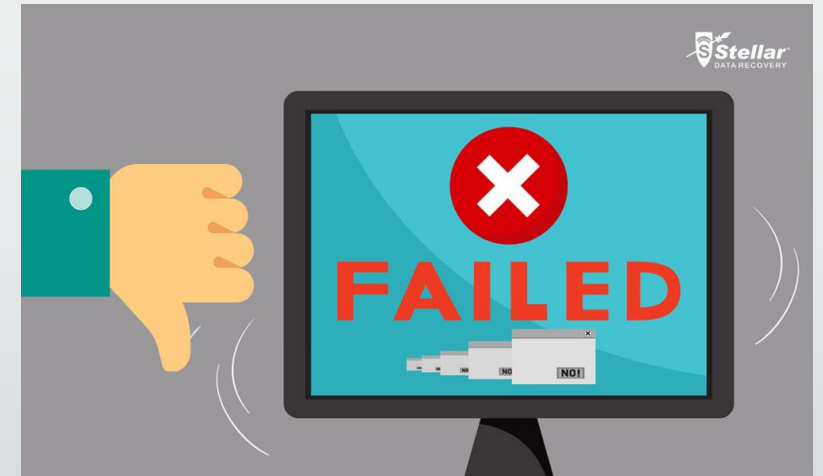


Exceção

- Condição provocada por uma situação excepcional que requer uma ação específica e imediata;
- Desvia o fluxo de controle do programa para um código de tratamento;
- Pode ser causada por diversas condições:
 - Erros sérios de hardware;
 - Erros simples de programação;
 - Condições esperadas (não são erros).

Programação Procedural

- Na era da programação procedural
 - Uma variável global inteira com valores de 0 até n permanecia disponível durante toda a execução da aplicação.
 - Na ocorrência de uma exceção:
 - Variável assumia um valor;
 - Remetia uma mensagem de erro;
 - Encerrava o programa.
- **Esse é um funcionamento ideal?**



Nos Tempos de POO

- Passamos ao conceito de **classes de erros**.
 - Adotando este mesmo paradigma no tratamento de exceções
- Nestas classes, possíveis tipos de erros e seus tratamentos são agrupados.
- Não há necessidade de interromper o programa.
- O mesmo erro pode ser tratado quantas vezes for necessário em uma mesma execução.

Alternativas para o Controle de Erros

- Deixar o programa abortar;
- Teste e tratamento imediato;
- Retorno de código de erro (*magic numbers*);
- Variável global (ex: C);
- Parâmetro de saída;
- API de tratamento de erros (raise(), signal() , setjmp(), longjmp() em C);
- **Mecanismo de exceções.**



Tratamento Imediato é Viável?

```
public int divideInteiros(int numerador, int denominador)
{
    if (denominador == 0)
        return trata_divisao_zero();
    else
        return numerador / denominador;
}
```

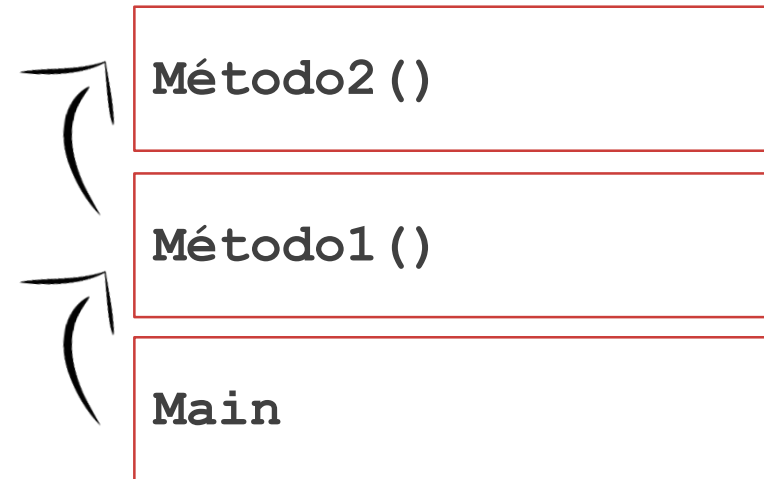
```
public int trata_divisao_zero()
{
    Console.WriteLine("Divisao por zero");
    return 0;
}
```



```

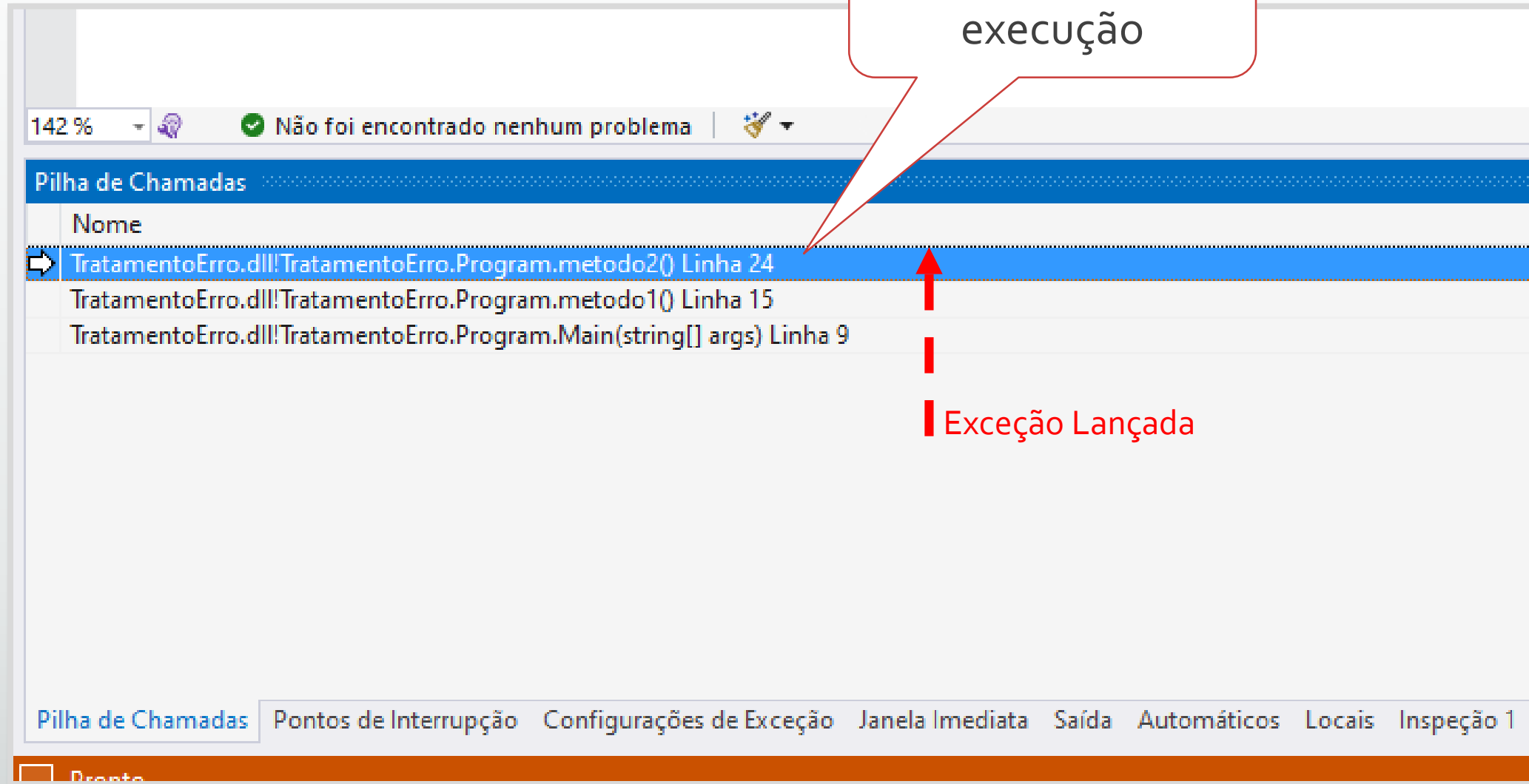
1 namespace TratamentoErro
2 {
3     0 referências
4     public class Program
5     {
6         0 referências
7         public static void Main(String[] args)
8         {
9             Console.WriteLine("inicio do main");
10            metodo1();
11            Console.WriteLine("fim do main");
12        }
13        1 referência
14        static void metodo1()
15        {
16            Console.WriteLine("inicio do metodo1");
17            metodo2();
18            Console.WriteLine("fim do metodo1");
19        }
20        1 referência
21        static void metodo2()
22        {
23            Console.WriteLine("inicio do metodo2");
24            int[] vet = { 1, 2, 3, 4, 5 };
25            for (int i = 0; i <= 10; i++)
26            {
27                Console.WriteLine(vet[i]);
28            }
29            Console.WriteLine("fim do metodo2");
30        }
31    }
32 }

```



Qual a saída?

Saída!



Exceções em Ação

```
public class Exceptions {  
    public static void main(String[] args) {  
        System.out.println("inicio do main");  
        metodo1();  
        System.out.println("fim do main");  
    }  
    static void metodo1() {  
        System.out.println("inicio do metodo1");  
        metodo2();  
        System.out.println("fim do metodo1");  
    }  
    static void metodo2() {  
        System.out.println("inicio do metodo2");  
        int[] vet = {1, 2, 3, 4, 5};  
        for (int i = 0; i <= 10; i++) {  
            System.out.println(vet[i]);  
        }  
        System.out.println("fim do metodo2");  
    }  
}
```

9: Exceção não tratada, relançada...
CLR (thread) morre!

15: Não tratada, relançada...

24: Exceção lançada, não tratada...

Tratando Exceções

```
static void metodo2()  
{  
    Console.WriteLine("inicio do metodo2");  
    int[] vet = { 1, 2, 3, 4, 5 };  
    try {  
        for (int i = 0; i <= 10; i++) {  
            Console.WriteLine(vet[i]);  
        }  
    } catch (IndexOutOfRangeException e) {  
        Console.WriteLine("erro: " + e);  
    }  
    Console.WriteLine("fim do metodo2");  
}
```

0 referencias

```
public static void Main(String[] args)
```

cs D:\Dropbox\research\academico\AULAS\ATIVIDADES\2023.1\cimatec\disciplinas\PROGRAMAÇÃO III\codigos\TratamentoErro\TratamentoErro\bin\De...

```
inicio do main  
inicio do metodo1  
inicio do metodo2
```

Falha tratada em
tempo de execução

```
1  
2  
3  
4  
5
```

```
erro: System.IndexOutOfRangeException: Index was outside the bounds of the array.  
    at TratamentoErro.Program.metodo2() in D:\Dropbox\research\academico\AULAS\ATIVIDADES\2023.1\cimatec\c  
RAMAÇÃO III\codigos\TratamentoErro\TratamentoErro\Program.cs:line 24
```

```
fim do metodo2  
fim do metodo1  
fim do main
```

```
static void metodo2 ()
```

```
{
```

```
    Console.WriteLine("inicio do metodo2");
```

```
    int[] vet = { 1, 2, 3, 4, 5 };
```

```
    try {
```

```
        for (int i = 0; i <= 10; i++) {
```

```
            Console.WriteLine(vet[i]);
```

```
        }
```

```
    } catch (IndexOutOfRangeException e) {
```

```
        Console.WriteLine("erro: " + e);
```

```
    }
```

```
    Console.WriteLine("fim do metodo2");
```

```
}
```

Bloco supervisionado!

Exceção é lançada e o tratamento é disparado

Captura (catch) da exceção, especificando a classe...

Bloco de tratamento. Exceção é disponibilizada como instância da classe (objeto!)

Tratando Exceções - Tem diferença?

```
static void metodo2()  
{  
    Console.WriteLine("inicio do metodo2");  
    int[] vet = { 1, 2, 3, 4, 5 };  
    for (int i = 0; i <= 10; i++) {  
        try {  
            Console.WriteLine(vet[i]);  
        } catch (IndexOutOfRangeException e) {  
            Console.WriteLine("erro: " + e);  
        }  
    }  
    Console.WriteLine("fim do metodo2");  
}
```

Qual a saída?


```
Console.ReadLine();
```

```
}
1 ref
sta inicio do main
inicio do metodo1
{
  inicio do metodo2
  1
  2
  3
  4
  5
}
erro: System.IndexOutOfRangeException: Index was outside the bounds of the array.
   at TratamentoErro.Program.metodo2() in D:\Dropbox\research\academico\AULAS\ATIVIDADES\2023.1\cimatec\disciplinas
1 ref
sta AMAÇÃO III\codigos\TratamentoErro\TratamentoErro\Program.cs:line 26
erro: System.IndexOutOfRangeException: Index was outside the bounds of the array.
   at TratamentoErro.Program.metodo2() in D:\Dropbox\research\academico\AULAS\ATIVIDADES\2023.1\cimatec\disciplinas
{
  AMAÇÃO III\codigos\TratamentoErro\TratamentoErro\Program.cs:line 26
  erro: System.IndexOutOfRangeException: Index was outside the bounds of the array.
   at TratamentoErro.Program.metodo2() in D:\Dropbox\research\academico\AULAS\ATIVIDADES\2023.1\cimatec\disciplinas
  AMAÇÃO III\codigos\TratamentoErro\TratamentoErro\Program.cs:line 26
  erro: System.IndexOutOfRangeException: Index was outside the bounds of the array.
   at TratamentoErro.Program.metodo2() in D:\Dropbox\research\academico\AULAS\ATIVIDADES\2023.1\cimatec\disciplinas
  AMAÇÃO III\codigos\TratamentoErro\TratamentoErro\Program.cs:line 26
  erro: System.IndexOutOfRangeException: Index was outside the bounds of the array.
   at TratamentoErro.Program.metodo2() in D:\Dropbox\research\academico\AULAS\ATIVIDADES\2023.1\cimatec\disciplinas
  AMAÇÃO III\codigos\TratamentoErro\TratamentoErro\Program.cs:line 26
  erro: System.IndexOutOfRangeException: Index was outside the bounds of the array.
   at TratamentoErro.Program.metodo2() in D:\Dropbox\research\academico\AULAS\ATIVIDADES\2023.1\cimatec\disciplinas
  AMAÇÃO III\codigos\TratamentoErro\TratamentoErro\Program.cs:line 26
  fim do metodo2
  fim do metodo1
  fim do main
```

Tratando Exceções – Qual será a saída?

```
static void metodo1() {  
    Console.WriteLine("inicio do metodo1");  
    try {  
        metodo2();  
    } catch (IndexOutOfRangeException e) {  
        Console.WriteLine("erro: " + e);  
    }  
    Console.WriteLine("fim do metodo1");  
}  
  
static void metodo2() {  
    Console.WriteLine("inicio do metodo2");  
    int[] vet = { 1, 2, 3, 4, 5 };  
    for (int i = 0; i <= 10; i++) {  
        Console.WriteLine(vet[i]);  
    }  
    Console.WriteLine("fim do metodo2");  
}
```



Qual a saída?

0 referências

`public class Program`

```
{  
    D:\Dropbox\research\academico\AULAS\ATIVIDADES\2023.1\cimaterc\disciplinas\PROGRAMAÇÃO III\codigos\TratamentoErro\Tratamen  
    inicio do main  
    inicio do metodo1  
    inicio do metodo2  
    1  
    2  
    3  
    4  
    5  
    erro: System.IndexOutOfRangeException: Index was outside the bounds of the array.  
        at TratamentoErro.Program.metodo2() in D:\Dropbox\research\academico\AULAS\ATIVIDADES\2023.  
    RAMAÇÃO III\codigos\TratamentoErro\TratamentoErro\Program.cs:line 26  
        at TratamentoErro.Program.metodo1() in D:\Dropbox\research\academico\AULAS\ATIVIDADES\2023.  
    RAMAÇÃO III\codigos\TratamentoErro\TratamentoErro\Program.cs:line 15  
    fim do metodo1  
    fim do main
```

O método2() não termina!

E se levarmos o tratamento de exceção para o **main()**?



0 referências

```
public class Program
```

```
{
```

0 referências

```
public static void Main(String[] args)
```

```
{
```

```
    Console.WriteLine("inicio do main");
```

```
    try
```

```
    {
```

```
        metodo1();
```

```
    }
```

```
    catch (IndexOutOfRangeException e)
```

```
    {
```

```
        Console.WriteLine("erro: " + e);
```

```
    }
```

```
    Console.WriteLine("fim do main");
```

```
    Console.ReadLine();
```

```
}
```

1 referência

```
static void metodo1()
```

```
{
```

```
    Console.WriteLine("inicio do metodo1");
```

```
    metodo2();
```

```
    Console.WriteLine("fim do metodo1");
```

```
}
```

1 referência

```
class Program
```

cs D:\Dropbox\research\academico\AULAS\ATIVIDADES\2023.1\cimatec\disciplinas\PROGRAMAÇÃO III\codigos\TratamentoErro\Trat

referências
inicio do main

```
public static void Main(String[] args)
{
    inicio do metodo1
    inicio do metodo2
```

```
    {
        Cons
```

```
        try
```

```
        {
```

```
        }
        catch
```

```
        {
```

```
        }
        Cons
```

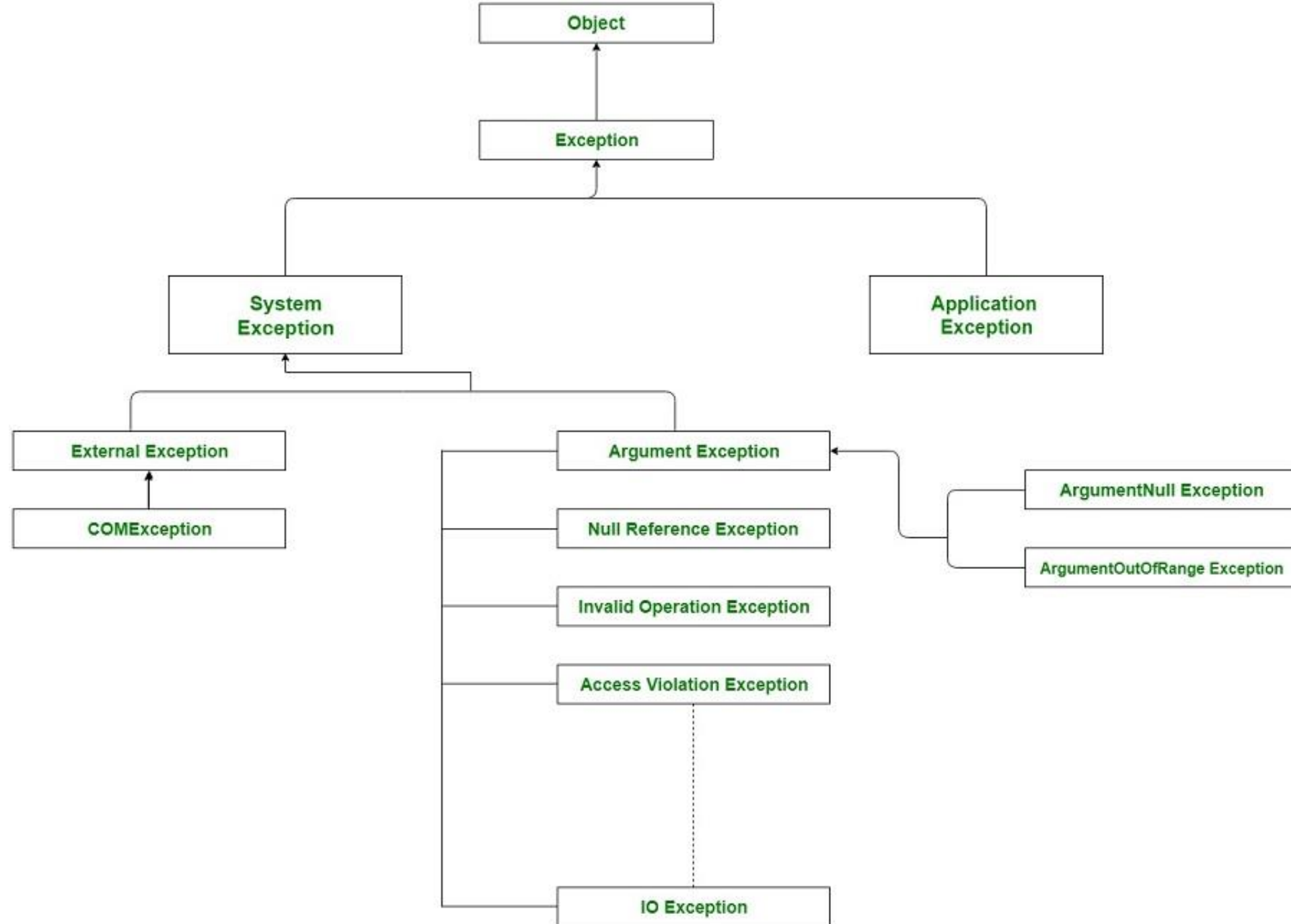
```
        Cons
```

```
        erro: System.IndexOutOfRangeException: Index was outside the bounds of the array.
            at TratamentoErro.Program.metodo2() in D:\Dropbox\research\academico\AULAS\ATIVIDADES\2023.1\cimatec\disciplinas\PROGRAMAÇÃO III\codigos\TratamentoErro\TratamentoErro\Program.cs:line 29
        }
        at TratamentoErro.Program.metodo1() in D:\Dropbox\research\academico\AULAS\ATIVIDADES\2023.1\cimatec\disciplinas\PROGRAMAÇÃO III\codigos\TratamentoErro\TratamentoErro\Program.cs:line 21
    }
    at TratamentoErro.Program.Main(String[] args) in D:\Dropbox\research\academico\AULAS\ATIVIDADES\2023.1\cimatec\disciplinas\PROGRAMAÇÃO III\codigos\TratamentoErro\TratamentoErro\Program.cs:line 10
}
fim do main
```

O método1() e método2() não terminaram!

Exceções

- Exceções, como tudo em C#, são objetos;
- Entretanto, são objetos especiais:
 - Podem ser manipulados pelo mecanismo de exceções.



Exceções de Sistema / Aplicações

- **Sistema**

- As exceções do sistema são derivadas da classe base *System.SystemException*
- Ocorre quando um erro fatal ou **irrecuperável** é encontrado, como uma falha no banco de dados, erros de vinculação, etc.

- **Aplicações**

- Derivadas da classe base *System.ApplicationException*
- Ocorre quando um erro **recuperável** é encontrado, por exemplo, o tipo errado de dados de entrada, exceções aritméticas, etc.
- Essas são exceções identificadas pelo usuário e lançadas pelo aplicativo.
- O programa pode retomar seu curso normal de instruções depois de resolver esse tipo de exceção

Lançando Exceções

- Exceções são lançadas pela palavra chave **Throw**
- Pode ser usada para notificar o usuário da função em que ocorreu uma situação não usual
- Quando uma exceção é lançada:
 - A execução do programa é parada
 - A exceção procura a instrução catch
 - Caso não encontre, ela será propagada para a interface e o CLR emitirá o erro.



CoolClips.com

Lançando Exceções

0 referências

```
public class Program
{
    0 referências
    public static void Main(String[] args)
    {
        try
        {
            Int32.Parse("texto");
        } catch (FormatException ex) {
            Console.WriteLine(ex.Message);
        }
    }
}
```

Console de Depuração do Microsoft Visual Studio

Input string was not in a correct format.

O D:\Dropbox\research\academico\AULAS\ATIVIDADES\2023.1\cimatec\disciplinas\PRO
mentoErro\bin\Debug\net6.0\TratamentoErro.exe (processo 11136) foi encerrado co
Para fechar o console automaticamente quando a depuração parar, habilite Ferram
console automaticamente quando a depuração parar.
Pressione qualquer tecla para fechar esta janela...

Usando Throw – Lançando Exceções

- Lançando uma exceção

0 referências
`public class Program`
{

0 referências
`public static void Main(String[] args)`
{

`try`

{

`Int32.Parse("texto");`

`} catch(FormatException ex) {`

`throw new ArgumentException("Aegumento inválido");`



`}`

`}`

`//public static void Main(String[] args)`
`/*`

Exceção Sem Tratamento

System.ArgumentException: 'Aegumento inválido'

[Mostrar Pilha de Chamadas](#) | [Exibir Detalhes](#) | [Copiar Detalhes](#) | [Inicie a Sessão](#)

▸ [Configurações de Exceção](#)

Escolhendo o Tipo de Exceção

- Quando um parâmetro inválido é passado para um método
 - **ArgumentException**, **ArgumentNullException**, **ArgumentOutOfRangeException**
- Quando a operação requisitada não é suportada
 - **NotSupportedException**
- Quando um método ainda não está implementado
 - **NotImplementedException**
- Quando não existe uma exceção padrão para o caso em questão
- Crie sua própria exceção herdando da classe **Exception**

0 referências

```
public class Program
```

```
{
```

0 referências

```
public static void Main(String[] args) {
```

```
    Console.WriteLine("inicio do main");
```

```
    try {
```

```
        metodo1();
```

```
    } catch (Exception ex) {
```

```
        Console.WriteLine(ex.ToString());
```

```
    }
```

```
    Console.WriteLine("fim do main");
```

```
    Console.ReadLine();
```

```
}
```

1 referência

```
static void metodo1() {
```

```
    Console.WriteLine("inicio do metodo1");
```

```
    metodo2();
```

```
    Console.WriteLine("fim do metodo1");
```

```
}
```

1 referência

```
static void metodo2() {
```

```
    Console.WriteLine("inicio do metodo2");
```

```
    try {
```

```
        int[] vet = { 1, 2, 3, 4, 5 };
```

```
        for (int i = 0; i <= 10; i++) {
```

```
            Console.WriteLine(vet[i]);
```

```
        }
```

```
    } catch (Exception ex) {
```

```
        throw new MyException("Erro no metodo2");
```

```
    }
```

```
    Console.WriteLine("fim do metodo2");
```

```
}
```

```
}
```



2 referências

```
public class MyException : Exception
```

```
{
```

1 referência

```
    public MyException(string message) : base(message) { }
```

```
}
```

Try-Finally

0 referências

```
public class Program
```

```
{
```

0 referências

```
public static void Main(String[] args)
```

```
{
```

```
    try
```

```
    {
```

```
        Int32.Parse("texto");
```

```
    } catch (FormatException ex) {
```

```
        Console.WriteLine(ex.Message);
```

```
    }
```

```
    finally
```

```
    {
```

```
        Console.WriteLine("Código sempre executado!");
```

```
    }
```

```
}
```

Console de Depuração do Microsoft Visual Studio

```
Input string was not in a correct format.  
Código sempre executado!
```

```
O D:\Dropbox\research\academico\AULAS\ATIVIDADES\2023.1\cimatec\disciplinas\PROGRAMAÇÃO III\c  
mentoErro\bin\Debug\net6.0\TratamentoErro.exe (processo 10228) foi encerrado com o código 0.  
Para fechar o console automaticamente quando a depuração parar, habilite Ferramentas -> Opção  
console automaticamente quando a depuração parar.  
Pressione qualquer tecla para fechar esta janela...
```

Bons Estudos!