

Tratamento de erros

1- Defina como verdadeiro ou falso as declarações a seguir

Uma exceção é uma indicação de que ocorreu um erro durante a execução de um programa. (Verdadeiro)

A classe System.Exception é a classe base de todas as exceções em C#. **verdade**

A pilha de execução (stack trace) mostra a sequência de chamadas de métodos que levaram à exceção. **verdade**

As exceções em C# são organizadas em uma hierarquia de classes, com a classe System.Exception no topo da hierarquia. **verdadeiro**

Todas as exceções em C# são subclasses da classe System.Exception. **verdade**

2- Defina como verdadeiro ou falso as declarações a seguir

O bloco try é usado para envolver o código que pode gerar uma exceção. **verdade**

O bloco catch é usado para capturar a exceção e lidar com ela de alguma forma. **verdade**

O bloco finally é opcional e é usado para executar código, independentemente de ter ocorrido ou não uma exceção. **verdade**

O bloco finally é sempre executado antes do bloco catch. **falso**

É possível ter vários blocos catch para capturar diferentes tipos de exceções. **verdade**

3 - Escreva um program onde o usuário é solicitado a informar um valor via teclado e armazenar o valor na variável entrada do tipo string onde tem que verificar 3 condições diferentes:

a- Se a entrada é nula, uma exceção NullReferenceException é lançada com a mensagem "A entrada não pode ser nula."

b- Se a entrada está vazia, uma exceção ArgumentException é lançada com a mensagem "A entrada não pode estar vazia."

c- Se a entrada passar nas verificações anteriores, a entrada é exibida na tela.

Realize o tratamento de erro para essas condições usando o bloco try-cath-finally

4- Escreva um programa que solicite ao usuário a informação da idade e do nome via teclado que deverão ser armazenados nas variáveis idade do tipo int e nome do tipo string.

A seguir realize o tratamento de erro e lançando as exceções considerando as seguintes condições:

a- Se a idade é negativa, uma exceção ArgumentException é lançada com a mensagem "A idade não pode ser negativa."

b - Se a idade é zero, uma exceção NotImplementedException é lançada com a mensagem "A idade ainda não foi definida."

c- Se o nome é nulo, uma exceção NullReferenceException é lançada com a mensagem "O nome não pode ser nulo."

5- Dado um array de inteiros expresso da seguinte forma

```
int[] numeros = new int[] { 109, 211, 313, 405, 519, 617, 711, 891, 951, 1001 };
```

Exiba na janela do console os numeros do array e solicite via teclado ao usuário para informar o valor de um índice do array para obter o seu respectivo valor.

Realize o tratamento de exceções filtrando as exceções IndexOutOfRangeException e ArgumentNullException

6- Dado o seguinte código:

Tratamento de erros

```
try
{
    int saldo = 0;
    int valorSaque = 100;
    if (valorSaque > saldo)
    {
        throw new SaldoInsuficienteException("O saldo é insuficiente para este saque.");
    }
    saldo -= valorSaque;
    Console.WriteLine("Saque efetuado com sucesso. Novo saldo: " + saldo);
}
catch (SaldoInsuficienteException e)
{
    Console.WriteLine("Erro: " + e.Message);
}
```

Implemente a exceção personalizada SaldoInsuficienteException.

7- Considere o seguinte código C#:

```
static void MeuMetodo(int valor)
{
    try
    {
        if (valor < 0)
        {
            throw new MinhaException("Valor negativo não permitido.");
        }
        else if (valor > 100)
        {
            throw new ArgumentException("O valor não pode ser maior que 100.");
        }
        Console.WriteLine("O valor é válido.");
    }
    catch (MinhaException e) when (valor < 0)
    {
        Console.WriteLine("Erro: " + e.Message);
    }
    catch (ArgumentException e) when (valor > 100)
    {
        Console.WriteLine("Erro: " + e.Message);
    }
}
```

Tratamento de erros

```
catch (Exception e)
{
    Console.WriteLine("Erro genérico: " + e.Message);
}
finally
{
    Console.WriteLine("Método concluído.");
}
}

public class MinhaException : Exception
{
    public MinhaException() { }
    public MinhaException(string message) : base(message) { }
}
```

Qual é o resultado esperado da execução do método **MeuMetodo(-5)**?

- A) A mensagem "Erro genérico: Valor negativo não permitido." é exibida na tela.
- B) A mensagem "Erro genérico: O valor não pode ser maior que 100." é exibida na tela.
- ☒ C) A mensagem "Erro: Valor negativo não permitido." é exibida na tela.
- D) A mensagem "Erro: O valor não pode ser maior que 100." é exibida na tela.
- E) Nenhuma mensagem de erro é exibida na tela.