Rafaela Yoko da Silva Takitani

Cybertron – Trilha DEV C# 1 Laboratório 1 – Criando Aplicativos de Console

Exercícios

**1. O que acontece se você remover o comando using System;? O que acontece se você também remover a referência System na chamada System.Console.WriteLine?**

Ao remover o comando using System o programa dá um erro e diz que o tipo ou o namespace ‘String’ não foi encontrado, está faltando uma diretiva using ou uma referencia de assembly.

A diretiva using cria um alias para um namespace ou importa tipos definidos em outros namespaces, por esse motivo o programa dá um erro quando é retirado o using Sytem, pois os tipos dos dados não estão mais definidos.

Interface gráfica do usuário, Texto

Descrição gerada automaticamente

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Ao remover a referência System na chamada System.Console.WriteLine dá um erro e diz que o nome “Console” não existe no contexto atual.

Interface gráfica do usuário, Texto, Aplicativo, Email

Descrição gerada automaticamente

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

**2. Mova o método Main para fora da classe Program. (Isto é, remova as palavras class Program e chaves associadas). O quê acontece**?

O programa dá um erro dizendo que tem mais de um ponto de entrada definido e pede-se para compilar com /main para especificar o tipo que contém o ponto de entrada

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Interface gráfica do usuário, Aplicativo, Teams

Descrição gerada automaticamente

3. O que acontece se você clonar a classe Program (copie, cole, e renomeie para Program2)?

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

**3. O que acontece se você clonar a classe Program (copie, cole, e renomeie para Program2)?**

**4. Pesquise os novos métodos e propriedades da classe System.Console.**

**Propriedades:**

**Console.BackgroundColor** : Obtém ou define a cor da tela de fundo do console.

**Console.ForegroundColor**: Um ConsoleColor que especifica a cor de primeiro plano do console, ou seja, a cor de cada caractere exibido. O padrão é cinza.

Interface gráfica do usuário, Texto

Descrição gerada automaticamente com confiança média

Interface gráfica do usuário, Texto, Aplicativo, chat ou mensagem de texto

Descrição gerada automaticamente

**Console.BufferHeight** : Obtém ou define a altura da área de buffer. Essa propriedade define o número de linhas (ou linhas) armazenadas no buffer que é acessado por uma janela de modo de console.

**Console.BufferWidth**: Obtém ou define a largura da área de buffer.

Texto

Descrição gerada automaticamente



**Console.CapsLock:** Obtém um valor que indica se a alternância de teclado CAPS LOCK está ativada ou desativada.

true se a tecla CAPS LOCK está ativada; false se CAPS LOCK está desativada.

Texto

Descrição gerada automaticamente

Texto

Descrição gerada automaticamente

**Console.CursorLeft:** Obtém ou define a posição da coluna do cursor na área de buffer.

**Console.CursorSize:** Obtém ou define a altura do cursor em uma célula do caractere.

**Console.CursorTop:** Obtém ou define a posição da linha do cursor dentro da área de buffer.

**Console.CursorVisible:** Obtém ou define um valor que indica se o cursor está visível.

**Console.NumberLock:** Obtém um valor que indica se a alternância de teclado NUM LOCK está ativada ou desativada.

**Texto

Descrição gerada automaticamente com confiança média**

Interface gráfica do usuário, Texto

Descrição gerada automaticamente

**Métodos**

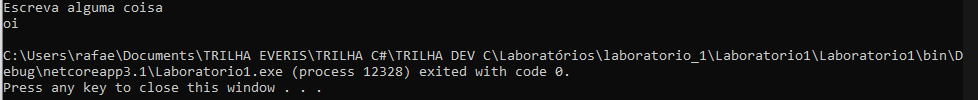
**Console.Clear** : Limpa o buffer do console e a janela do console correspondente de informações de exibição.

Texto

Descrição gerada automaticamente

**Console.Read**: Lê o próximo caractere do fluxo de entrada padrão.

**Console.ReadLine**: Lê a próxima linha de caracteres do fluxo de entrada padrão.



**Console.WriteLine**: Grava os dados especificados, seguido pelo terminador de linha atual para o fluxo de saída padrão.





**Console.Write**: Grava a representação de texto do valor ou valores especificados no fluxo de saída padrão.



