C# Avançado
Aula 04

- Revisão
- Resolver exercício
- Trabalhando com datas
- Trabalhando com arquivos
- Exercícios

C# Avançado - DATAS

- O C# possui algumas classes para ajudar a tratar dados ou tempo. Entre elas estão:
  - TimeSpan
    - https://learn.microsoft.com/pt-br/dotnet/api/system.timespan?view=net-8.0
  - DateTime
    - https://learn.microsoft.com/pt-br/dotnet/api/system.datetime?view=net-8.0
  - DateTimeOffset
    - https://learn.microsoft.com/pt-br/dotnet/api/system.datetimeoffset?view=net-8.0

C# Avançado - DATAS

- TimeSpan
- Um TimeSpan representa um intervalo de tempo ou um momento do dia. No último caso, ele é simplesmente o horário do "relógio" (sem a data), equivalente ao tempo decorrido desde a meia-noite, assumindo que não houve transição para o horário de verão. Um TimeSpan tem uma resolução de 100 nanossegundos, um valor máximo de cerca de 10 milhões de dias e pode ser positivo ou negativo.
- O TimeSpan pode ser construído de 3 formas diferentes:
    - Pelo construtor:
        - public TimeSpan (int hours, int minutes, int seconds);
        - public TimeSpan (int days, int hours, int minutes, int seconds);
        - public TimeSpan (int days, int hours, int minutes, int seconds,
        - int milliseconds);
        - public TimeSpan (long ticks); // Each tick = 100ns
    - Pelo uso de método estático da classe TimeSpan
        - public static TimeSpan FromDays (double value);
        - public static TimeSpan FromHours (double value);
        - public static TimeSpan FromMinutes (double value);
        - public static TimeSpan FromSeconds (double value);
        - public static TimeSpan FromMilliseconds (double value);
        - Ex:
        - Console.WriteLine (new TimeSpan (2, 30, 0)); // 02:30:00
        - Console.WriteLine (TimeSpan.FromHours (2.5)); // 02:30:00
        - Console.WriteLine (TimeSpan.FromHours (-2.5)); // -02:30:00
    - E pela subtração de um “DateTime” por outro
        - TimeSpan nearlyTenDays = TimeSpan.FromDays(10) - TimeSpan.FromSeconds(1);
        - Console.WriteLine (nearlyTenDays.Days); // 9
        - Console.WriteLine (nearlyTenDays.Hours); // 23
        - Console.WriteLine (nearlyTenDays.Minutes); // 59
        - Console.WriteLine (nearlyTenDays.Seconds); // 59
        - Console.WriteLine (nearlyTenDays.Milliseconds); // 0

C# Avançado - DATAS

- DateTime e DateTimeOffset
- DateTime e DateTimeOffset são structs imutáveis para representar uma data e, opcionalmente, um horário. Elas têm uma resolução de 100 nanossegundos e uma faixa que cobre os anos de 0001 a 9999. `DateTimeOffset` é funcionalmente semelhante a `DateTime`. Sua característica distintiva é que também armazena um deslocamento de Tempo Universal Coordenado (UTC); isso permite resultados mais significativos ao comparar valores em diferentes fusos horários.
- Escolhendo entre um ou outro
- Construindo os objetos
    - DateTime - Existem 17 maneiras de fazer, veja o construtor
        - public DateTime (int year, int month, int day);
        - public DateTime (int year, int month, int day,int hour, int minute, int second, int millisecond);
        - public DateTime(long ticks)
    - DateTimeOffset
        - public DateTimeOffset (int year, int month, int day,int hour, int minute, int second,TimeSpan offset);
        - public DateTimeOffset (int year, int month, int day,int hour, int minute, int second, int millisecond, TimeSpan offset);
    - Acessando a data de onde está rodando
        - Console.WriteLine (DateTime.Now); Console.WriteLine (DateTime.UtcNow); // 04/12/2023 19:00:00 PM
        - Console.WriteLine (DateTimeOffset.Now); Console.WriteLine (DateTimeOffset.UtcNow); // 04/12/2023 1:23:45 PM -03:00

C# Avançado - DATAS

- Parse
  - https://www.c-sharpcorner.com/blogs/date-and-time-format-in-c-sharp-programming1

Format	Result
DateTime.Now.ToString("MM/dd/yyyy")	05/29/2015
DateTime.Now.ToString("dddd, dd MMMM yyyy")	Friday, 29 May 2015
DateTime.Now.ToString("dddd, dd MMMM yyyy")	Friday, 29 May 2015 05:50
DateTime.Now.ToString("dddd, dd MMMM yyyy")	Friday, 29 May 2015 05:50 AM
DateTime.Now.ToString("dddd, dd MMMM yyyy")	Friday, 29 May 2015 5:50
DateTime.Now.ToString("dddd, dd MMMM yyyy")	Friday, 29 May 2015 5:50 AM
DateTime.Now.ToString("dddd, dd MMMM yyyy HH:mm:ss")	Friday, 29 May 2015 05:50:06
DateTime.Now.ToString("MM/dd/yyyy HH:mm")	05/29/2015 05:50
DateTime.Now.ToString("MM/dd/yyyy hh:mm tt")	05/29/2015 05:50 AM
DateTime.Now.ToString("MM/dd/yyyy H:mm")	05/29/2015 5:50
DateTime.Now.ToString("MM/dd/yyyy h:mm tt")	05/29/2015 5:50 AM
DateTime.Now.ToString("MM/dd/yyyy HH:mm:ss")	05/29/2015 05:50:06
DateTime.Now.ToString("MMMM dd")	May 29
DateTime.Now.ToString("yyyy'-'MM'-'dd'T'HH':'mm':'ss.fffffffK")	2015-05-16T05:50:06.7199222-04:00
DateTime.Now.ToString("ddd, dd MMM yyy HH':'mm':'ss 'GMT'")	Fri, 16 May 2015 05:50:06 GMT
DateTime.Now.ToString("yyyy'-'MM'-'dd'T'HH':'mm':'ss")	2015-05-16T05:50:06
DateTime.Now.ToString("HH:mm")	05:50
DateTime.Now.ToString("hh:mm tt")	05:50 AM
DateTime.Now.ToString("H:mm")	5:50
DateTime.Now.ToString("h:mm tt")	5:50 AM
DateTime.Now.ToString("HH:mm:ss")	05:50:06
DateTime.Now.ToString("yyyy MMMM")	2015 May

C# Avançado - Arquivos

Para trabalhar com arquivos, o c# fornece uma classe estática para nós apoiar, ela é a “File”. Outra classe que também é muito usada é a “Directory”.

Elas permitem, criar, mover, deletar, encontrar arquivos, permissões e atributos.

- File
- https://learn.microsoft.com/pt-br/dotnet/api/system.io.file?view=net-8.0
    - bool Exists (string path); // Returns true if the file is present
    - void Delete (string path);
    - void Copy (string sourceFileName, string destFileName);
    - void Move (string sourceFileName, string destFileName);
    - void Replace (string sourceFileName, string destinationFileName,
    - string destinationBackupFileName);
    - FileAttributes GetAttributes (string path);
    - void SetAttributes (string path, FileAttributes fileAttributes);
    - void Decrypt (string path);
    - void Encrypt (string path);
    - DateTime GetCreationTime (string path); // UTC versions are
    - DateTime GetLastAccessTime (string path); // also provided.
    - DateTime GetLastWriteTime (string path);
    - void SetCreationTime (string path, DateTime creationTime);
    - void SetLastAccessTime (string path, DateTime lastAccessTime);
    - void SetLastWriteTime (string path, DateTime lastWriteTime);
    - FileSecurity GetAccessControl (string path);
    - FileSecurity GetAccessControl (string path,
    - AccessControlSections includeSections);
    - void SetAccessControl (string path, FileSecurity fileSecurity);

C# Avançado - Arquivos

- Directory
- https://learn.microsoft.com/pt-br/dotnet/api/system.io.directory?view=net-8.0
  - void SetCurrentDirectory (string path);
  - DirectoryInfo CreateDirectory (string path);
  - DirectoryInfo GetParent (string path);
  - string GetDirectoryRoot (string path);
  - string[] GetLogicalDrives(); // Gets mount points on Unix
  - // The following methods all return full paths:
  - string[] GetFiles (string path);
  - string[] GetDirectories (string path);
  - string[] GetFileSystemEntries (string path);
  - IEnumerable<string> EnumerateFiles (string path);
  - IEnumerable<string> EnumerateDirectories (string path);
  - IEnumerable<string> EnumerateFileSystemEntries (string path);

C# Avançado - Arquivos

- Json System.Text.Json
- https://learn.microsoft.com/pt-br/dotnet/standard/serialization/system-text-json/how-to?pivots=dotnet-8-0
  - public class WeatherForecast {

    public DateTimeOffset Date { get; set; }

    public int TemperatureCelsius { get; set; }

    public string? Summary { get; set; }

    }
  - Serializar
    - string jsonString = JsonSerializer.Serialize(weatherForecast);
  - Deserializar
    - string fileName = "WeatherForecast.json";

      string jsonString = JsonSerializer.Serialize(weatherForecast);

      File.WriteAllText(fileName, jsonString);

C# Avançado - Exercícios

Escreva um programa, para saber como as pessoas estão, com as seguintes regras:

1 Pergunte o nome de quem está digitando

2 Pergunte como ela está

3 Grave as respostas em arquivo dividindo em datas

4 O nome do arquivo deve ser no formato “como_esta_vc_DD_MM_YYYY_hh_mm_ss.txt”

Escreva outro programa que leia o arquivo feito no primeiro programa e tenha a seguintes regras:

1 Mostrar os nomes

2 Busque as respostas por data

3 Busque as respostas por nome

4 Gera uma versão do arquivo em json

C# Avançado - Exercícios

Vamos continuar com nossa pizzaria. Agora vamos aplicar novas funcionalidades com uso de data:

1. Adicionar cliente.

1.1 O cliente deve ter histórico de ações

2. Adicionar pizza.

2.1 Pizza deve ter dado de criação

3. Adicionar pedido.

3.1 O pedido deve ter horário de solicitação

3.2 O pedido deve ter horário de finalização da preparação

3.3 O pedido deve ter horário de saída para entrega

3.3 O pedido deve ter horário de finalização da entrega

4. Obter todos os clientes.

5. Obter todas as pizzas.

6. Obter todos os pedidos.

7. Gerar relatório em arquivo texto dos pedidos finalizados mostrando os dados de quem pediu, a pizza, horário que o pedido começou e horário de finalização da preparação, horário que saiu e finalizou a entrega.

7.1 Deve permitir gerar o relatório escolhendo a data inicial e final

7.2 Deve ter uma opção de escolher entre arquivo texto e json

SENAI

Serviço Nacional de Aprendizagem Industrial

PELO FUTURO DO TRABALHO

0800 048 1212 **sc.senai.br**

Rodovia Admar Gonzaga, 2765 - Itacorubi - 88034-001 - Florianópolis, SC