

Melhorando um Gerenciador de Tarefas em C#

Introdução

Nesta conversa, exploramos como melhorar um gerenciador de tarefas escrito em C#, com foco em boas práticas, melhorias de código e explicações detalhadas sobre estruturas e métodos utilizados. Este PDF é uma compilação organizada das explicações e do código, para facilitar o estudo.

Melhorias Aplicadas ao Código

O código inicial do gerenciador de tarefas foi reescrito e aprimorado em várias partes:

1. Estrutura Inicial Melhorada

O código foi reorganizado em métodos bem definidos para melhorar a modularidade e a reutilização. Por exemplo:

- `AdicionarTarefa`: Para adicionar novas tarefas à lista.
- `RemoverTarefa`: Para remover tarefas, agora com suporte a busca parcial usando LINQ.
- `ListarTarefas`: Para exibir todas as tarefas de forma clara.

2. Uso de LINQ na Remoção de Tarefas

O método `RemoverTarefa` foi aprimorado para permitir que o usuário busque por parte do nome de uma tarefa. Isso torna o código mais flexível e amigável para o usuário.

Código Final Melhorado

Aqui está o código completo, com melhorias e explicações.

```
```csharp
```

```
using System;
```

```
using System.Collections.Generic;

using System.Linq;

namespace ExercicioListaDeTarefas
{
 internal class Program
 {
 static void Main(string[] args)
 {
 List<string> listaDeTarefas = new List<string>();

 int opcao;

 do
 {
 Console.Clear();

 Console.WriteLine("Escolha uma das opções: ");

 Console.WriteLine("1: Criar tarefa");

 Console.WriteLine("2: Remover tarefa");

 Console.WriteLine("3: Listar todas as tarefas");

 Console.WriteLine("4: Sair");

 Console.Write("Opção: ");

 // Validação da entrada

 if (!int.TryParse(Console.ReadLine(), out opcao))
 {
 Console.WriteLine("Opção inválida. Pressione qualquer tecla para tentar novamente.");

 Console.ReadKey();
 }
 }
 }
 }
}
```

```
continue;
```

```
}
```

```
switch (opcao)
```

```
{
```

```
case 1:
```

```
AdicionarTarefa(listaDeTarefas);
```

```
break;
```

```
case 2:
```

```
RemoverTarefa(listaDeTarefas);
```

```
break;
```

```
case 3:
```

```
ListarTarefas(listaDeTarefas);
```

```
break;
```

```
case 4:
```

```
Console.WriteLine("Saindo... Obrigado por usar o gerenciador de tarefas!");
```

```
break;
```

```
default:
```

```
Console.WriteLine("Opção inválida. Pressione qualquer tecla para tentar novamente.");
```

```
Console.ReadKey();
```

```
break;
```

```
}
```

```
} while (opcao != 4);
```

```
}
```

```
static void AdicionarTarefa(List<string> lista)
```

```
{
 Console.Clear();

 Console.Write("Digite a nova tarefa: ");

 string novaTarefa = Console.ReadLine();

 if (!string.IsNullOrEmpty(novaTarefa))
 {
 lista.Add(novaTarefa);

 Console.WriteLine("Tarefa adicionada com sucesso!");
 }
 else
 {
 Console.WriteLine("A tarefa não pode ser vazia.");
 }

 Console.WriteLine("Pressione qualquer tecla para continuar.");

 Console.ReadKey();
}
```

```
static void RemoverTarefa(List<string> lista)
{
 Console.Clear();

 if (lista.Count == 0)
 {
 Console.WriteLine("Nenhuma tarefa para remover.");
 }

 else
 {

```

```
Console.WriteLine("Tarefas atuais:");
```

```
foreach (var tarefa in lista)
```

```
{
```

```
Console.WriteLine($"{tarefa}");
```

```
}
```

```
Console.Write("Digite parte ou o nome completo da tarefa que deseja remover: ");
```

```
string busca = Console.ReadLine();
```

```
var tarefasEncontradas = lista
```

```
.Where(t => t.Contains(busca, StringComparison.OrdinalIgnoreCase))
```

```
.ToList();
```

```
if (tarefasEncontradas.Count == 0)
```

```
{
```

```
Console.WriteLine("Nenhuma tarefa encontrada com esse nome.");
```

```
}
```

```
else if (tarefasEncontradas.Count == 1)
```

```
{
```

```
lista.Remove(tarefasEncontradas[0]);
```

```
Console.WriteLine("Tarefa removida com sucesso!");
```

```
}
```

```
else
```

```
{
```

```
Console.WriteLine("Múltiplas tarefas encontradas:");
```

```
for (int i = 0; i < tarefasEncontradas.Count; i++)
```

```
{
```

```
Console.WriteLine($"{i + 1}: {tarefasEncontradas[i]}");

}
```

```
Console.Write("Escolha o número da tarefa que deseja remover: ");

if (int.TryParse(Console.ReadLine(), out int escolha) && escolha > 0 && escolha <= tarefasEncontradas.Count)
{

 lista.Remove(tarefasEncontradas[escolha - 1]);

 Console.WriteLine("Tarefa removida com sucesso!");

}

else
{

 Console.WriteLine("Escolha inválida.");

}

}

}

Console.WriteLine("Pressione qualquer tecla para continuar.");

Console.ReadKey();

}
```

```
static void ListarTarefas(List<string> lista)

{

 Console.Clear();

 if (lista.Count == 0)

 {

 Console.WriteLine("Nenhuma tarefa cadastrada.");

 }

}
```

```
else

{

Console.WriteLine("Lista de tarefas:");

foreach (var tarefa in lista)

{

Console.WriteLine($"{tarefa}");

}

}

Console.WriteLine("Pressione qualquer tecla para continuar.");

Console.ReadKey();

}

}

}

...

```

#### #### Tratamento de Duplicatas

O método `RemoveTarefa` suporta duplicatas e permite ao usuário escolher qual instância de uma tarefa remover. Se desejar evitar duplicatas na adição, inclua uma verificação no método `AdicionarTarefa`.

#### ### Conclusão

Este guia detalha como melhorar um gerenciador de tarefas, incorporando boas práticas e LINQ para flexibilidade. Estude os métodos e explore os conceitos para aprimorar suas habilidades em C#!