

Clínica-De-Bugs(POO: Programação Orientada a Objetos).

Implementação contendo erros, fornecida pelo professor para ser corrigida.

Atividade 04: Nome_Do_Dia_Da_Semana.

```
CSharp
Console.Write("Digite um número de 1 a 7: ");
string diaa = Console.ReadLine();

switch (diaa)
{
    case 1:
        Console.WriteLine("Domingo")
        break
    case "2":
        Console.WriteLine("Segunda-feira");
    case 3:
        Console.WriteLine("Terça-feira");

    default:
        Console.WriteLine("Número inválido")
}
```

Implementação corrigida, contendo comentários para marcar erros e correções

```
CSharp
// Alterado a variável "dia" para int, pois o switch case funciona melhor
// com tipos inteiros ou strings.
// Utilizado o Convert.ToInt32 para converter a entrada do usuário de string
// para int.
Console.Write("Digite um número de 1 a 7: ");
int dia = Convert.ToInt32(Console.ReadLine());

//Corrigido a váriavel "dia" no switch case.
//Antes: switch (diaa) { ... }
//Agora:
switch (dia)
{
    case 1:
```

```

        Console.WriteLine("Domingo"); //Estava faltando ponto e virgula ";"
        break; //Estáva faltando ponto e virgula ";"
//Retirado aspas duplas desnecessárias do número.
//Antes: case "2": { ... }
//Agora:
case 2:
    Console.WriteLine("Segunda-feira");
    break; //Adicionado Break para evitar "fall through"
case 3:
    Console.WriteLine("Terça-feira");
    break;
case 4:
    Console.WriteLine("Quarta-feira");
    break;
case 5:
    Console.WriteLine("Quinta-feira");
    break;
case 6:
    Console.WriteLine("Sexta-feira");
    break;
case 7:
    Console.WriteLine("Sábado");
    break;
default:
    Console.WriteLine("Número inválido"); //Adicionado ponto e virgula
";"
    break; //adicionado Break para evitar "fall through"
}

```

Ficha de Erros

Problema nº	ERR - 0001	ERR - 002	ERR - 003	ERR - 004	ERR - 005
Projeto/Arquivo	Nome_Do_Dia_Da_Semana.cs	Nome_Do_Dia_Da_Semana.cs	Nome_Do_Dia_Da_Semana.cs	Nome_Do_Dia_Da_Semana.cs	Nome_Do_Dia_Da_Semana.cs
Linha(s) Afetada(s):	02	04	07 - 08 - 15	09 - 11	14
Tipo de Erro	Sintaxe/ Tipagem	Sintaxe	Sintaxe	Sintaxe	Sintaxe

Mensagem do compilador /exceção:	Conversão de literal nula ou possível valor nulo em tipo não anulável. (CS8600)	O nome "diaa" não existe no contexto atual (CS0103)	; esperado (CS1002)	Controle não pode passar através de um rótulo case ("case ":") para outro (CS0163)	O controle não pode ficar fora do switch do rótulo de caso final ('default:') (CS8070)
Linha(s) suspeitas:	Linha 02, Col 14	Linha 04, Col 09	Linha 07, Col 14 - Linha 08 - Col 09 - Linha 15 - Col 44	Linha 09, Col 05 - Linha 11, Col 05	Linha 14, Col 05
Hipótese (por que ocorre?):	Alerta que a variável atribuída é do tipo não anulável e pode receber uma expressão que possa ser NULL	O nome "identificador" não existe no contexto atual	O compilador detectou um ponto e vírgula ';' ausente.	O compilador não consegue passar de um rótulo de caso para outro	Construção errada do bloco switch
Experimento (o que testei?):	Utilizado a anotação "?" para transformar a variável de referência anulável	Comparado o nome identificador com a variável a qual se referia e corrigido	Adicionado um ponto e vírgula (;) após o final da declaração	Retirado as aspas duplas do valor numérico a ser testado e adicionado um break	Corrigido o bloco switch por completo
Descrição da correção:	O "?" é uma dica para o compilador saber sobre a expectativa de um valor nulo	Verificar sempre o nome e referências de montagem para garantir que o nome está certo	Um ponto e vírgula é necessário no final de cada instrução em C#	Após retirado as aspas ainda acusava que o compilador não conseguia seguir para a próxima instrução, até adicionado um break	Após o bloco ser corrigido, adicionando cada "break;" necessário, o erro parou de ocorrer
Teste de regressão (entradas/saídas esperadas):	Entrada: 1.64 → Saída Esperada: null	Entrada: {3} → Saída Esperada: Terça-Feira	Entrada: ; → Saída Esperada: Fim do bloco.	Entrada: {3} → Saída Esperada: Terça-Feira	Entrada: break. → Saída Esperada: Pula para o próximo Case
Regra/checklist:	A anotação utilizada fornece ao compilador informações para	Cuidar com o nome da variável, seguir sempre o	Cuidar com o ponto e vírgula no final de cada declaração	Sempre tomar cuidado com a sintaxe do switch.	Em um switch cada case ou default precisa terminar

	evitar construções que podem levar a exceções de referência nula	padrão do Clean Code			corretamente
Observações:	Tipos de referência anuláveis, incluindo os operadores '?' e '!' são permitidos somente quando o contexto anulável está definido como enable ou annotations.	Utilizar o padrão CamelCase para definir o nome das variáveis	Uma declaração pode abranger mais de uma linha.	Quando uma declaração de troca contém mais de uma seção de switch, você deve encerrar explicitamente cada seção, incluindo a última	Caso não encerrar o case o compilador tenta sair do bloco do switch, mas é impedido pelo último case (default)

MAPA RÁPIDO DE CÓDIGOS ÚTEIS (PARA CONSULTA)

CS1002 ; esperado - Sintaxe

CS0103 O nome "diaa" não existe no contexto atual - Sintaxe

CS8070 O controle não pode ficar fora do switch do rótulo de caso final ('default:')

TABELA DE TESTES

Caso	Entradas	Passos Relevantes	Saída Esperada	Saída Observada	OK?
1	int = 1	Executar o programa	Domingo	Domingo	OK
2	int = 8	Testar o Default	Número inválido	Número inválido	OK
3	int = a	Testar a variável dia	Error	CLR/System.Format Exception	OK

Resumo de Aprendizagem

Colocado em prática a utilização do Switch case, visto alguns possíveis erros que podem ocorrer ao utilizar o bloco switch case, utilizado conversor de tipo para conseguir converter um tipo string em int, adicionando linhas que faltavam no código.

Registro de uso de IA

Utilização do ChatGPT para transcrever a imagem contendo o código do problema fornecido pelo professor.

“Preciso que transcreva a imagem em anexo de maneira literal, sem correções ou comentários”.

No mais, apenas utilizado sites e documentação que seguem abaixo.

Site da Microsoft contendo referências em C#:

<https://learn.microsoft.com/en-us/dotnet/csharp/language-reference/compiler-messages/nullable-warnings>

Clean code: Comentários:

<https://danielwisky.com.br/2023-01-17-clean-code-comentarios/>

Além dos materiais fornecidos pelo professor.