

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

## Implementação contendo erros, fornecida pelo professor para ser corrigida.

Atividade 01: Coletor\_De\_Dados\_Pessoais.

```
CSharp
Console.WriteLine("Por favor, digite seu primeiro nome:")
string primeiroNome = Console.ReadLine();

Console.WriteLine("Agora, digite sua idade:");
int idade = Console.ReadLine();

Console.WriteLine("Por fim, digite sua altura em metros (ex: 1,80):");
float altura = float.Parse(Console.ReadLine());

Console.WriteLine($"Resumo do Cadastro: Nome: {primeiroNome}, Idade: {idade} anos, Altura: {altura}m.")
```

## Implementação corrigida, contendo comentários para marcar erros e correções

```
CSharp
//Console.WriteLine() sem ponto e virgula ";"
//Adicionado um ponto e virgula no final da linha
//Adicionado o operador de coalescência nula "??" para evitar erro de conversão
Console.WriteLine("Por favor, digite seu primeiro nome: ");
string? primeiroNome = Console.ReadLine();

//Utilizado o Convert.ToInt32 para converter a string em inteiro
Console.WriteLine("Agora, digite sua idade: ");
int idade = Convert.ToInt32(Console.ReadLine());

//Alterado exemplo de entrada de altura para o formato correto de "," para "."
//Adicionado o operador de coalescência nula "??" para evitar erro de conversão, e utilizado 0 como valor padrão
Console.WriteLine("Por fim, digite sua altura em metros (ex: 1.80)");
float altura = float.Parse(Console.ReadLine() ?? "0");
```

```
//Corrigito o nome da variável "PrimeiroNome" para "primeiroNome"
Console.WriteLine($"Resumo do Cadastro: Nome: {primeiroNome}, Idade: {idade}
anos, Altura: {altura}m.");
```

Ficha de Erros						
Problema nº	ERR - 0001	ERR - 002	ERR - 003	ERR - 004	ERR - 005	ERR - 006
Projeto/Arquivo	Coletor_De_Dados_Pessoais.cs	Coletor_De_Dados_Pessoais.cs	Coletor_De_Dados_Pessoais.cs	Coletor_De_Dados_Pessoais.cs	Coletor_De_Dados_Pessoais.cs	Coletor_De_Dados_Pessoais.cs
Linha(s) Afetada(s):	07 - 19	08	11	14	16	16
Tipo de Erro	Sintaxe	Semântico/Tipagem	Tipagem	Semântico/Tipagem	Sintaxe	Sintaxe
Mensagem do compilador/exceção:	; esperado (CS1002)	Conversão de literal nula ou possível valor nulo em tipo não anulável. (CS8600)	Não é possível converter implicitamente tipo "string" em "int" (CS0029)	Possível argumento de referência nula para o parâmetro 's' em 'float.Parse(string s)'. (CS8604)	"Console" não contém uma definição para "writeline" (CS0117)	O nome "PrimeiroNome" não existe no contexto atual (CS0103)
Linha(s) suspeitas:	Linha 07 Col 09; Linha 19, Col 112	Linha 08, Col 31	Linha 11, Col 21	Linha 14, Col 36	Linha 16, Col 17	Linha 16, Col 56
Hipótese (por que ocorre?):	O compilador detectou um ponto e	Alerta que você está atribuindo uma variável	Requer uma conversão explícita do	Alerta que você está atribuindo uma	Esse erro ocorre quando é feita uma	Ocorre quando uma tentativa de usar um

	vírgula ';' ausente.	cujo tipo não é anulável a uma expressão cujo estado nulo é talvez nulo	compilador.	variável cujo tipo não é anulável a uma expressão cujo estado nulo é talvez nulo	referência a um membro que não existe para o tipo de dados	nome que não existe na classe, namespace, ou escopo é feito
<b>Experimento (o que testei?):</b>	Adicionado um ponto e vírgula (;) após o final da declaração	A utilização do '?', anotação para tornar a variável um tipo de referência anulável.	Testado alguns tipos de métodos de conversão para transformar uma string em um dado do tipo inteiro. Utilizado o método 'Convert.ToInt32()'	Utilizado o operador '??' (coalescência nula) seguido de "0" para fornecer um valor padrão caso o valor fornecido seja null	Verificado que o método 'writeln' estava escrito errado. Corrigido para o método correto (WriteLine).	Corrigido o nome da variável que estava escrito errado no escopo. ERRADO: PrimeiroNome e CERTO: primeiroNome
<b>Descrição da correção:</b>	Um ponto e vírgula é necessário no final de cada instrução em C#	O objetivo dos avisos anuláveis é minimizar a chance de seu aplicativo lançar um Sistema.NullReferenceException quando executado.	Método de conversão, converte string para um tipo (int, double) mas pode gerar exceções	Fornecer um valor padrão caso a expressão à esquerda seja null	Corrigido o método WriteLine que pertence a classe Console	Corrigido o nome da variável no escopo

<b>Teste de regressão (entradas/saídas esperadas):</b>	Entrada: Eduardo → Saída Esperada: “Eduardo”	Entrada: Eduardo → Saída Esperada: “Eduardo”	Entrada: String→ Saída Esperada: inteiro com sinal de 32 bits	Caso expressao != null → o resultado é expressao.  Caso expressao == nula → o resultado é valorPadrao ∴	Entrada: Escreve algo no console → Saída Esperada: N/A	Entrada: Nome Informado pelo usuário → Saída Esperada: Valor da variável informada pelo usuário
<b>Regra/checklist:</b>	Cuidar com o ponto e vírgula no final de cada declaração	Fornecer ao compilador informações para evitar construções que podem levar a exceções de referência nulas.	Sempre cuidar para realizar as devidas conversões e sempre utilizar o melhor método ou operador para a conversão que deseja realizar	O operador ?? não valida se a entrada é numérica.	Sempre cuidar para vincular a classe certa ao método certo.	Cuidar com o nome da variável, seguir sempre o padrão do Clean Code
<b>Observações:</b>	Uma declaração pode abranger mais de uma linha.	Tipos de referência anuláveis, incluindo os operadores '?' e '!' são permitidos somente quando o contexto anulável está definido como enable ou annotations.	Se o número for maior que o limite do tipo de dado inteiro, irá lançar um alerta 'OverflowException'. Pode lidar com valores null sem quebrar o programa, retornando 0 como valor.	Se o usuário digitar "abc", ainda ocorrerá o aviso 'FormatException'.	Console é uma classe estática do C# enquanto WriteLine é um método dessa classe	Utilizar o padrão CamelCase para definir o nome das variáveis

## MAPA RÁPIDO DE CÓDIGOS ÚTEIS (PARA CONSULTA)

**CS1002** ;esperado - sintaxe

**CS0029** Não é possível converter implicitamente tipo "string" em "int"  
-Tipagem

**CS0117** "Console" não contém uma definição para "writeline" - Sintaxe

## TABELA DE TESTES

Caso	Entradas	Passos Relevantes	Saída Esperada	Saída Observada	OK?
1	"Eduardo", 25, 1.71f	Executar o programa	Eduardo -25 - 1.71	Eduardo -25 - 1.71	OK
2	"123", 22, 1.80	Testar a variável do tipo string	123 - 22 - 1.80	123 - 22 - 1.80	OK
3	"1,2", 2.5,	Testar a variável do tipo int	1,2, 0, 2	O programa foi encerrado	NO

## Resumo de Aprendizagem

Visto mais exemplos de problemas de tipagem, sintaxe e reforço na importância de escrever um código limpo e bem comentado, além de rever sobre conversão de tipo, e aprender sobre tipos anuláveis e coalescência nula sempre visando boas práticas de nomenclatura.

Notei que o exercício teve bastante erros de sintaxe e semântica.

## 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 abaxo.

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.