

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

Implementação contendo erros, fornecida pelo professor para ser corrigida.

Atividade 09: Matriz_3_3_Diagonais_Maiores_Transposta.

```
CSharp
int[][] matriz = new int[3,3];
int somaTotal = 0;
int somaDiagonal = 0;
int[] maiores = new int[3];

for (int i = 0; i <= 3; i++)
{
    for (int j = 0; j <= 3; j++)
    {
        Console.WriteLine($"Valor [{i},{j}]: ");
        int valor = int.Parse(Console.ReadLine());

        matriz[j, i] = valor;
        somaTotal = somaTotal + valor;

        if (i == j)
        {
            somaDiagonal += matriz[i, i + 1];
        }

        if (valor > maiores[i])
        {
            maiores[i] = valor;
        }
    }
}

Console.WriteLine("Soma total: {somaTotal}");
Console.WriteLine("Soma diagonal principal: " + somaDiagonal);

Console.WriteLine("Maiores por linha:");
for (int i = 0; i < 3; i++)
{
    Console.WriteLine(maiores[i]);
}

int[,] transposta = new int[3][3];

for (int i = 0; i < 3; i++)
```

```

{
    for (int j = 0; j < 3; j++)
    {
        transposta[i, j] = matriz[i, j];
    }
}

Console.WriteLine("Matriz transposta:");
for (int i = 0; i < 3; i++)
{
    for (int j = 0; j < 3; j++)
    {
        Console.Write(transposta[i, j] + " ");
    }
    Console.WriteLine()
}

```

Implementação corrigida, contendo comentários para marcar erros e correções

```

CSharp
int[,] matriz = new int[3,3]; // Corrigindo a declaração da matriz
int somaTotal = 0; // Adicionando ;
int somaDiagonal = 0;
int[] maiores = new int[3];

for (int i = 0; i < 3; i++) // Corrigindo a condição do loop para evitar
estouro de índice.
{
    for (int j = 0; j < 3; j++) // Corrigindo a condição do loop para evitar
estouro de índice.
    {
        Console.WriteLine($"Valor [{i},{j}]: ");
        string? input = Console.ReadLine();
        if (input is null) // Verifica se a entrada não é nula antes de
tentar usá-la.
        {
            Console.WriteLine("Entrada inválida.");
            return;
        }
        int valor = int.Parse(input);
    }
}

```

```

        matriz[j, i] = valor;
        somaTotal = somaTotal + valor;

        if (i == j)
        {
            somaDiagonal += matriz[i, j]; // Corrigindo o índice para
            acessar a diagonal principal
        }

        if (valor > maiores[i])
        {
            maiores[i] = valor;
        }
    }
}

Console.WriteLine($"Soma total: {somaTotal}"); // Adicionando $ para
interpolação de string
Console.WriteLine("Soma diagonal principal: " + somaDiagonal);

Console.WriteLine("Maiores por linha:");
for (int i = 0; i < 3; i++)
{
    Console.WriteLine(maiores[i]); // Corrigindo o índice para acessar o
    array corretamente
}

int[,] transposta = new int[3,3]; // Corrigindo a declaração da matriz
transposta

for (int i = 0; i < 3; i++)
{
    for (int j = 0; j < 3; j++)
    {
        transposta[i, j] = matriz[i, j];
    }
}

Console.WriteLine("Matriz transposta:");
for (int i = 0; i < 3; i++)
{
    for (int j = 0; j < 3; j++)
    {
        Console.Write(transposta[i,j] + " ");
    }
    Console.WriteLine(); // Adicionando ;
}

```

Ficha de Erros					
Problema nº	ERR - 001	ERR - 002	ERR - 003	ERR - 004	ERR - 005
Projeto/Arquivo:	Program.cs	Program.cs	Program.cs	Program.cs	Program.cs
Linha(s) Afetada(s):	Linha 02, Col 18	Linha 54, Col 24	Linha 01, Col 18	Linha 34, Col 31	Linha 37, Col 32
Tipo:	Sintático	Sintático	Sintático	Sintático	Sintático
Mensagem do compilador /exceção:	error CS1003: Erro de sintaxe, "," esperado	error CS1002: ; esperado	error CS0029: Não é possível converter implicitamente tipo "int[,*]" em "int[][]"	error CS0103: O nome "j" não existe no contexto atual	error CS0178: Especificador de classificação inválido: era esperado ",", ou "]"
Hipótese (por que ocorre?):	O compilador detectou um ponto e vírgula ';' ausente.	O compilador detectou um ponto e vírgula ';' ausente.	A declaração do vetor está incorreta.	Foi utilizado a variável j quando deveria ser a variável i.	A declaração do vetor está incorreta.
Experimento (o que testei?):	Adicionado um ponto e vírgula (;) após o final da declaração	Adicionado um ponto e vírgula (;) após o final da declaração	Trocado int[][] por int[,]	Trocado a variável j por i.	Trocado int[3][3] por int[3,3].

Descrição da correção:	Um ponto e vírgula é necessário no final de cada instrução em C#	Um ponto e vírgula é necessário no final de cada instrução em C#	Deve-se declarar a matriz como um vetor de vetores, declarando corretamente.	Trocado a variável j por i.	Deve-se declarar a matriz como um vetor de vetores, declarando corretamente.
Teste de regressão (entradas/saídas esperadas):	Entrada: 50 → Saída Esperada: "Maiores por linha: 50 [...]"	Entrada: 50 → Saída Esperada: "Maiores por linha: 50 [...]"	Entrada: 50 → Saída Esperada: "Maiores por linha: 50 [...]"	Entrada: 50 → Saída Esperada: "Maiores por linha: 50 [...]"	Entrada: 50 → Saída Esperada: "Maiores por linha: 50 [...]"
Regra/checlist:	Cuidar com o ponto e vírgula no final de cada declaração	Cuidar com o ponto e vírgula no final de cada declaração	Verificar se as variáveis estão sendo corretamente declaradas.	Verificar se as variáveis corretas estão sendo utilizadas	Verificar se as variáveis estão sendo corretamente declaradas.

Ficha de Erros					
Problema nº	ERR - 006	ERR - 007	ERR - 008	ERR - 009	ERR - 010
Projeto/Arquivo:	Program.cs	Program.cs	Program.cs	Program.cs	Program.cs
Linha(s) Afetada(s):	Linha 11, Col 31	Linha 6, Col 20	Linha 8, Col 24	Linha 24, Col 39	Linha 34, Col 19

Tipo:	Sintático	Execução	Execução	Execução/Lógica	Semântico
Mensagem do compilador /exceção:	warning CS8604: Possível argumento de referência nula para o parâmetro 's' em 'int.Parse(string s)'.				
Hipótese (por que ocorre?):	Porque existe a possibilidade de a variável ser nula em tempo de execução.	A condição do loop estoura o tamanho do vetor	A condição do loop estoura o tamanho do vetor	O índice i+1 tenta acessar um valor acima do limite do vetor.	É necessário o símbolo "\$" para apresentar o valor da variável <i>somaTotal</i> .
Experimento (o que testei?):	Existem várias formas de solucionar este erro. Foi adicionado uma condição if-else para verificação do estado da variável.	Modificado <= para <.	Modificado <= para <.	Modificado i+1 por j.	Adição do símbolo "\$".

Descrição da correção:	Verifica o estado da variável inserida pelo usuário e apenas executa a lógica do código caso a entrada não seja nula.	Modificado <= para <, assim interrompendo o loop antes de estourar o tamanho do vetor.	Modificado <= para <, assim interrompendo o loop antes de estourar o tamanho do vetor.	Corrigido o índice para acessar a diagonal principal	Garante que o valor apresentado ao usuário será o correto.
Teste de regressão (entradas/saídas esperadas):	Entrada: 50 → Saída Esperada: "Maiores por linha: 50 [...]"	Entrada: 50 → Saída Esperada: "Maiores por linha: 50 [...]"	Entrada: 50 → Saída Esperada: "Maiores por linha: 50 [...]"	Entrada: 50 → Saída Esperada: "Maiores por linha: 50 [...]"	Entrada: 50 → Saída Esperada: "Maiores por linha: 50 [...]"
Regra/checlist:	Sempre verificar o estado de variáveis possivelmente nulas.	Sempre se atentar ao tamanho do vetor e à lógica do loop.	Sempre se atentar ao tamanho do vetor e à lógica do loop.	Verificar se as variáveis corretas estão sendo utilizadas	Garantir que o código está funcionando da maneira e lógica correta.

MAPA RÁPIDO DE CÓDIGOS ÚTEIS (PARA CONSULTA)

CS1002 faltando ;

CS0103 variável inexistente

CS0029 tipagem errada

CS8602 variável pode ser nula

TABELA DE TESTES			
Caso	Entradas simuladas	O que observar	Saída esperada
1	50	Comparação correta de maior valor	Maior da primeira linha deve ser 50
2	3	Limite estabelecido no loop while.	Primeira coluna da matriz transposta deve ser 50, 3, 51
3	51	Teste do limite de comparação do maior valor	Maior da primeira linha deve ser 51

Resumo de Aprendizagem

A atividade permitiu exercitar atenção e cuidado com erros facilmente evitáveis, como a falta de ponto e vírgula no final de declarações e a lógica de tamanho de vetores, sendo assim possível visualizar com clareza os problemas que podem advir da declaração incorreta de loops condicionais e o acesso indevido fora dos limites de um vetor. Também permitiu o reforço do conhecimento em relação à declaração de variáveis possivelmente nulas.

Registro de uso de IA

A I.A. foi utilizada principalmente para transcrever o texto do pdf com os exercícios, pois não é possível copiar diretamente do arquivo disponibilizado no GitHub. Ex. de prompt: “copia e cola exatamente, sem mudar nada e sem corrigir”.