**ATIVIDADE 9.1 – TRABALHANDO COM ARRAYS E ARRAYS LISTS (MATRIZES)**

Nesta atividade, você irá começar a familiarizar com o seguinte:

* Criando e usando arrays (matrizes)
* Trabalhando com arrays (matrizes) multidimensional
* Trabalhando com arrays lists (matrizes listas)

**Criando e usando Arrays (Matrizes)**

Para criar e popular um array, siga estes passos:

1. Inicie o Visual Studio. Selecione *File* -> *New* -> *Project*.
2. Escolha o projeto *Console Application*. Nomeie o projeto para *Activity9\_1*. O *Console Application* contém uma classe chamada *Program* com um método *Main*. O método *Main* é o primeiro método que é acessado quando a aplicação é lançada.
3. Aviso que o método *Main* aceita um parâmetro de entrada de uma *string array* chamada *args*. O *array args* contém alguns linhas de comandos argumentados passos enquanto o *console application* é lançado. Os membros do *args array* são separados por um espaço quando este é passado.

*static void Main (string[] args)*

*{*

*}*

1. Adicionando o seguinte código para o método *Main* para mostrar os argumentos da linha de comando passado:

*Console.WriteLine(“parameter Count = {0} “, args.Length);*

*for (int i = 0; i < args.Length; i++)*

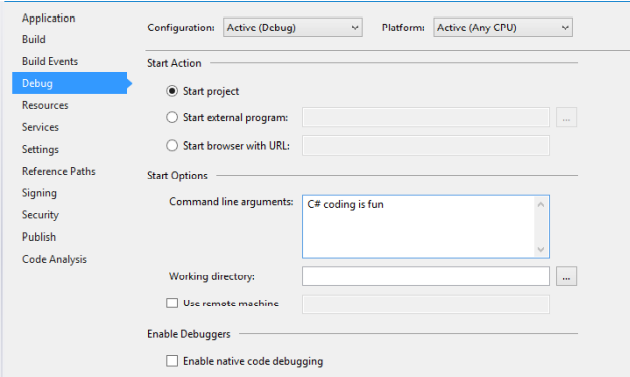
*{*

*Console.WriteLine(“Arg[{0}] = [{1}]”, i, args[i]);*

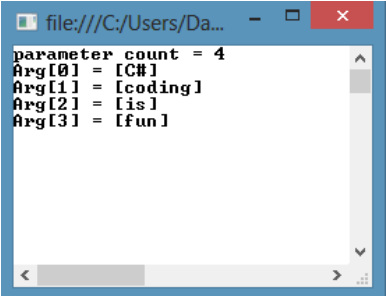
*}*

*Console.ReadLine();*

1. Na *Solution Explorer*, clique o botão direito no nó do projeto e escolha Properties. Na janela *Project Properties*, selecione a *tab Debug*. No campo *command line arguments*, digite “*C# coding is fun*”, veja na figura abaixo:



1. Selecione *Debug* -> *Start* para rodar o projeto. A janela *Console* deverá ser lançada com os valores de saída mostrando os valores abaixo. Depois de visualizar a saída, pare o depurador.



1. Adicionamos o seguinte código antes o método C*onsole.ReadLine*, no método *Main*. Este código limpa o valor do *array* no *index*(índice) 1 e seta o valor no *index* (índice) 3 para *“great”*.

*Array.Clear(args, 1,1);*

*args[3] = “great”;*

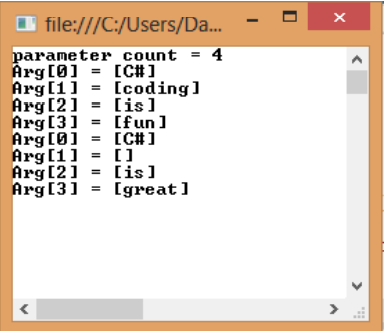
*for (int i = 0; i < args.Length; i++)*

*{*

*Console.WriteLine(“Args[{0}] = [{1}]”, i, args[i]);*

*}*

1. Selecione *Debug* -> *Start* para rodar o projeto. A janela *Console* deverá ser lançada com os valores de saída mostrando os valores abaixo. Depois de visualizar a saída, pare o depurador.

**

**Trabalhando com Arrays multidimensional**

Para criar e popular um array multidimensional, siga estes passos:

1. Comente o código no método *Main*.
2. Adicione o seguinte código para o método *Main* para criar e popular o array em duas dimensões:

*string[,] seatingChart = new string[2,2];*

*seatingChart[0, 0] = “Mary”;*

*seatingChart[0, 1] = “Jim”;*

*seatingChart[1, 0] = “Bob”;*

*seatingChart[1, 1] = “Jane”;*

1. Adicione o seguinte código para dar volta através do *array* e imprima os nomes na janela *Console*:

*for(int row = 0; row < 2; row++)*

*{*

*for( int seat = 0; seat < 2; seat++)*

*{*

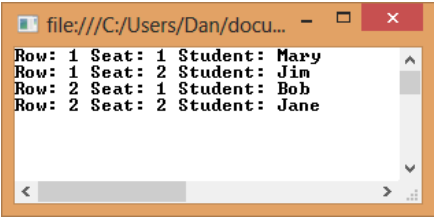
*Console.WriteLine(“Row: {0} Seat: {1} Student: [2]”, (row + 1), (seat + 1), seatingChart[row, seat]);*

*}*

*}*

*Console.ReadLine();*

1. Selecione *Debug* -> *Start* para rodar o projeto. A janela *Console* deverá lançar com a saída (*output*) que irá mostrar o mapa de assentos dos estudantes abaixo:



1. Depois de visualizar a saída (*output*), feche o depurador.

**Trabalhando com Array Lists**

Através de dois array de duas dimensões você irá ter criado (*works*) trabalhos, isto talvez seja mais intuitivo para armazenar a informação sobre cada assento dos estudantes na escola em uma classe com os assentos de escola e então para organizar estes objetos dentro de uma estrutura de array list (tipo lista). Para criar e popular um array list dos assentos na escola, siga estes passos:

1. Adicione um arquivo de classe para o projeto nomeando como *SeatingAssignment.cs*.
2. Adicione o seguinte código para criar a classe *SeatingAssignment*. Esta classe conterá as propriedades uma Row, Seat e Student. Isto também irá conter uma sobrecarga no construtor para esses conjuntos de propriedades.

*public class SeatingAssignment*

*{*

*int \_row;*

*int \_seat;*

*string \_student;*

*public int Row*

*{*

*get { return \_row; }*

*set { \_row = value; }*

*}*

*public int Seat*

*{*

*get { return \_seat; }*

*set { \_seat = value; }*

*}*

*public string Student*

*{*

*get { return \_student; }*

*set {\_student = value; }*

*}*

*public SeatingAssignment(int row, int seat, string student)*

*{*

*this.Row = row;*

*this.Seat = seat;*

*this.Student = student;*

*}*

*}*

1. No método *Main* da classe do *Program*, comente o código anterior e adicione a seguinte declaração *using* no topo do arquivo:

*using System.Collections;*

1. Adicione o seguinte código para criar um *ArrayList* dos assentos dos estudantes:

*ArrayList seatingChart = new ArrayList();*

*seatingChart.Add(new SeatingAssignment(0, 0, “Mary”));*

*seatingChart.Add(new SeatingAssignment(0, 1, “Jim”));*

*seatingChart.Add(new SeatingAssignment(1, 0, “Bob”));*

*seatingChart.Add(new SeatingAssignment(1, 1, “Jane”));*

1. Depois o *ArrayList* é populado, adicione o seguinte código para escrever a informação do *SeatingAssignment* para a janela *console*.

*foreach(SeatingAssignment sa in seatingChart)*

*{*

*Console.WriteLine(“Row: {0} Seat: {1} Student: {2}”, (sa.Row + 1),*

*(sa.Seat +1), sa.Student);*

*}*

*Console.ReadLine();*

1. Selecione *Debug* -> *Start* para rodar o projeto. A janela *console* deverá lançar com a saída (*output*) que irá mostrar o mapa de assentos dos estudantes abaixo:

Texto

Descrição gerada automaticamente

1. Uma das vantagens da classe *ArrayList* é habilitar para adicionar e remover itens dinamicamente. Adicione o seguinte código depois do passo 4 para adicionar mais dois estudantes para o seguinte mapa de assento:

*seatingChart.Add(new SeatingAssignment(2, 0, “Bill”));*

*seatingChart.Add(new SeatingAssignment(2, 1, “Judy”));*

1. Selecione *Debug* -> *Start* para rodar o projeto. A janela *console* poderá ser lançada com o *output* (saída) mostrando os novos estudantes.
2. Quando finalizar, pare o depurador, e feche o Visual Studio.