

Algoritmia e Estruturas de Dados

Módulo II **Arrays em C#**



Algoritmia

- **Arrays**

- ❑ É uma estrutura de dados **linear** e **estática**, composta por um número finito de elementos de determinado tipo de dados
- ❑ É uma Variável Composta Homogénea, de **acesso directo**
- ❑ É constituído por uma ou mais posições, acessíveis através da especificação de um **índice**, e cujo conteúdo é do mesmo tipo

Algoritmia

- **Arrays**

- ☐ Variável composta homogénea constituída por número finito de elementos de determinado tipo
- ☐ Acessível via índice indicando posição do elemento no array
- ☐ Dimensões:
 - ☐ Unidimensionais
 - ☐ Multidimensionais:
 - ☐ Bidimensional, Tridimensional,..., N dimensões
 - ☐ Arrays de Arrays (jagged arrays)

1	2	3
---	---	---

1	2	3
4	5	6
7	8	9

1	2	3
4	5	6
7	8	9

1	2				
3	4	5	6	7	8
9	10	11			

Algoritmia

- **Características Básicas de um Array**

- ☐ Número de dimensões – unidimensional, bidimensional, ...multi-dimensional
- ☐ Tipo do array – tipo de dados armazenados no array (inteiro, real, texto ...)
- ☐ Limites do array – limite máximo do índice do array, i.e., número máximo de elementos do array
- ☐ Acesso individual a cada posição do array, através do índice

Algoritmia

- Declarar um Array

Algoritmos Portugal

```

inicio

    inteiro numeros[5]
    texto alunos[3]

fim
    
```

C#

```

Int [] notas = new int[10];

String [] nome = new string[20];

Double [] temperatura = new double [24];
    
```

tipo de dados

Nome da estrutura de dados

Aloca espaço dinamicamente para o array

Dimensão



O índice de um array em C# começa em 0.

Logo, um array definido com 20 posições, compreende índices entre 0 e 19

Algoritmia

- **Declarar e inicializar um array em C#**

Array uni-dimensional

Define array de inteiros,
com 12 posições, **sem o inicializar**

```
int[] facturacao = new int[12];  
  
int[] vendas = new int[10] { 0,0,0,0,0,0,0,0,0,0 };  
  
string[] meses = new string[12] {"Janeiro", "Fevereiro", "Março", "Abril", "Maio", "Junho", "Julho",  
                                "Agosto", "Setembro", "Outubro", "Novembro", "Dezembro"};
```

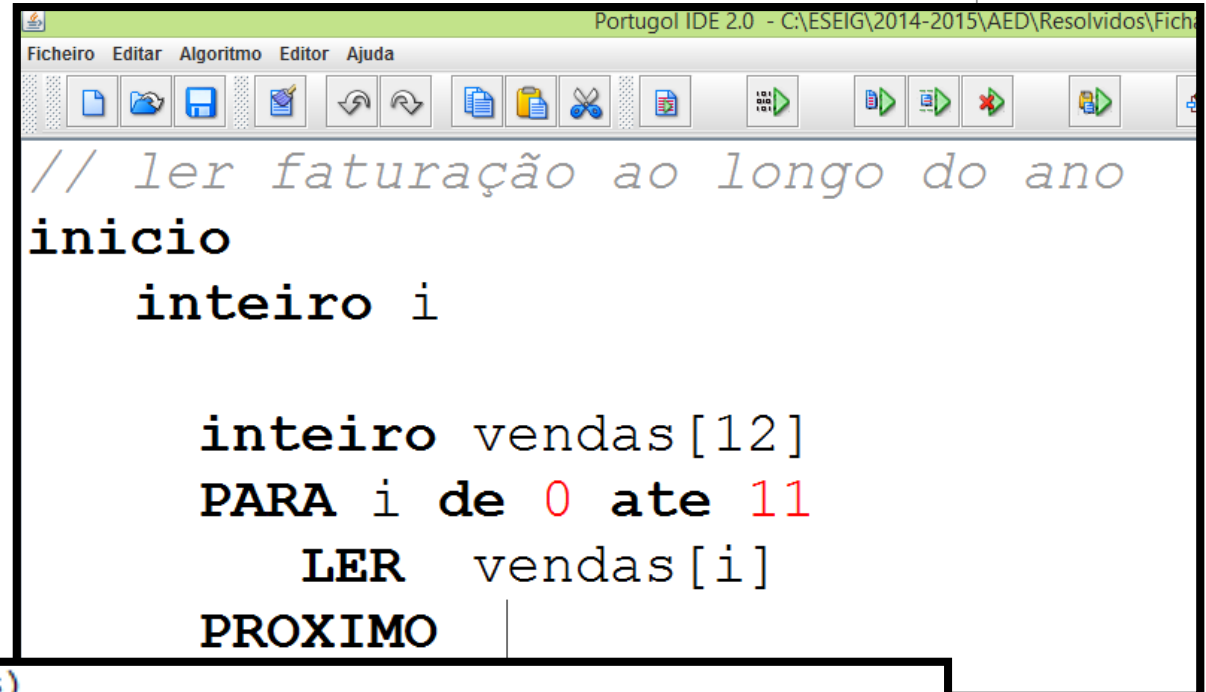
Define array de inteiros,
com 10 posições, e **inicializa** o
array

Define array de strings,
com 12 posições, e **inicializa** o
array

Algoritmia

Algoritmia

Declarar e ler array



The screenshot shows the Portugol IDE 2.0 interface. The title bar reads "Portugol IDE 2.0 - C:\ESEIG\2014-2015\AED\Resolvidos\Fich...". The menu bar includes "Ficheiro", "Editar", "Algoritmo", "Editor", and "Ajuda". The toolbar contains icons for file operations (new, open, save, print), editing (undo, redo, copy, paste, delete), and execution (run, step through, stop). The main editor area contains the following algorithmic code:

```
// ler faturação ao longo do ano
inicio
    inteiro i

    inteiro vendas[12]
    PARA i de 0 ate 11
        LER vendas[i]
    PROXIMO
```

```
static void Main(string[] args)
{
    // Definição do array para armazenar as vendas ao longo dos 12 meses
    int[] vendas = new int[12];

    // leitura dos dados de vendas ao longo dos 12 meses
    for (int i = 0; i < 12; i++)
    {
        Console.WriteLine("Volume de vendas do mês {0} : ", i + 1);
        vendas[i] = int.Parse(Console.ReadLine());
    }
}
```

C#

Declarar e ler array

Algoritmia

- Ciclos com Arrays

```
// Declara e lê array de vendas, com ciclo for
int[] vendas = new int[12];
for (int i=0; i<12; i++)
{
    vendas[i] = Convert.ToInt16(Console.ReadLine());
}
```

Ciclo for

Permite especificar os índices do array que pretendemos aceder/alterar

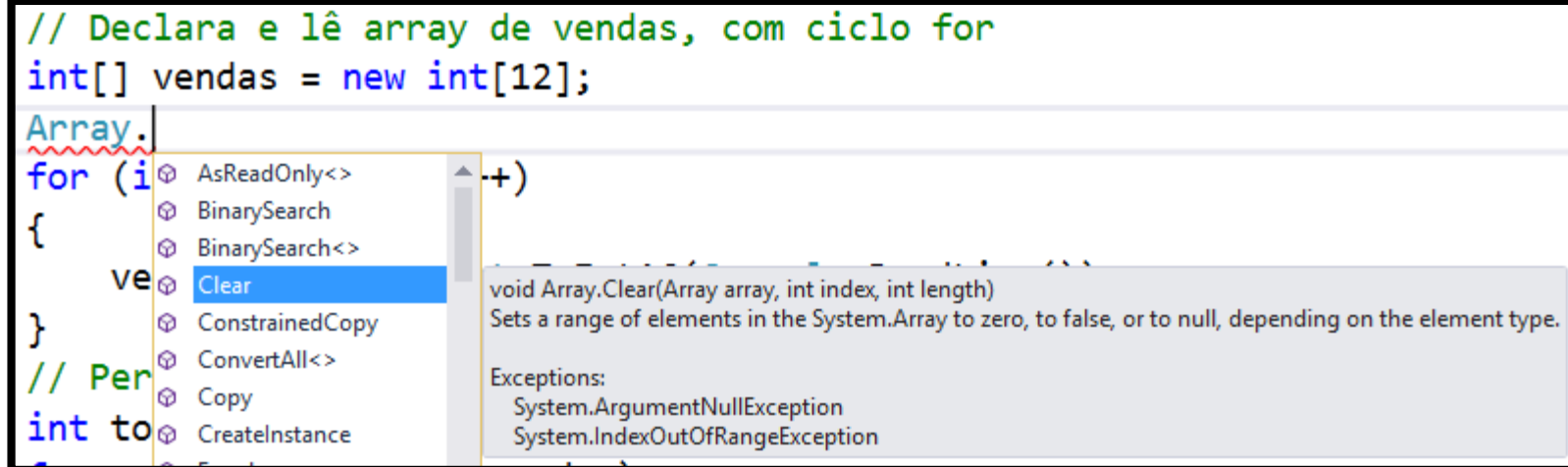
Ciclo foreach

Permite apenas aceder (não alterar!) a cada um dos elementos do array, necessariamente do primeiro ao último elemento do array

```
// Percorre todo o array vendas
int total = 0;
foreach (int valor in vendas)
{
    total = total + valor;
}
```


Algoritmia

- **C#: A Classe Array**



Classe Array

Podemos encontrar diversos métodos associados à classe array ou a objetos declarados sob a forma de um array

Algoritmia

- **C#: A Classe Array**

Métodos pré-definidos	Descrição
Copy	Copiar conjunto de elementos de um array
GetValue	Obter um determinado elemento do array
Resize	Redefinir a dimensão do array
SetValue	Atribui um valor a uma posição do array
Sort	Ordena os elementos de um array
IndexOf	Obter a posição(índice) da primeira ocorrência de um valor
Reverse	Inverte a ordem dos elementos do array
BinarySearch	Procura um valor num array, retornando a posição (índice) em que se encontra. O array TEM que estar ordenado.

Mais métodos da Classe Array:

http://msdn.microsoft.com/pt-br/library/system.array_methods%28v=vs.110%29.aspx

Algoritmia

- **C#: A Classe Array**

Lê nomes e preenche o array

```
// declarar array nomes com 10 posições
string[] nomes = new string[10];

// lê 10 nomes para as 10 posições do array
for (int i = 0; i < 10; i++)
{
    Console.WriteLine("Nome:");
    nomes[i] = Console.ReadLine();
}
```

```
// declarar array nomes com 10 posições
string[] nomes = new string[10];

// lê 10 nomes para as 10 posições do array
for (int i = 0; i < 10; i++)
{
    Console.WriteLine("Nome:");
    nomes[i] = Console.ReadLine();
}

// copia elementos do array nomes para array backup
string[] backup = new string[10];
Array.Copy(nomes, 0, backup, 0, 10);
```

Copia elementos do array nomes para array backup

Posição inicial

Posição inicial

Nº elementos a copiar

Algoritmia

- **C#: A Classe Array**

```
// copia elementos do array nomes para array backup
string[] backup = new string[10];
Array.Copy(nomes, 0, backup, 0, 10);
// ordena elementos do array por ordem ascendente
Array.Sort(nomes);
// inverte a ordenação do array - agora por ordem descendente
Array.Reverse(nomes);
```

Copiar e ordenar um array

Algoritmia

- **C#: A Classe Array**

```
// copia elementos do array nomes para array backup
string[] backup = new string[10];
Array.Copy(nomes, 0, backup, 0, 10);
// ordena elementos do array por ordem ascendente
Array.Sort(nomes);
// inverte a ordenação do array - agora por ordem descendente
Array.Reverse(nomes);
// Redefine a dimensão do array para 11 elementos
Array.Resize(ref nomes, 11);
// atribui nome à posição 10 (elemento 11 do array)
nomes.SetValue("João Moutinho", 10);

// Pesquisa um elementos no array - neste caso João Moutinho; devolve a posição onde
// encontra esse valor. NOTA: O ARRAY DEVE ESTAR ORDENADO
int pos = Array.BinarySearch(nomes, "João Moutinho");
Console.WriteLine(pos);

Console.ReadLine();|
```

Redefinir dimensão do array;
Atribuir dados ao array;
Pesquisa de dados num array

Algoritmia

C#:

A Classe Array

```
static void Main(string[] args)
{
    int soma = 0;
    // declarar array
    int [] notas = new int[10];
    // ordenar array (ascendente)
    Array.Sort(notas);
    // ordenar por ordem inversa (descendente)
    Array.Reverse(notas);
    // redimensiona array para 20 posições
    Array.Resize(ref notas, 20)
    // Obtem a primeira posição do array
    int nota1 = notas.GetValue(0);
    // atribui o valor 17 à posição 4 do array
    notas.SetValue(17, 3);
    // ontem índice da primeira ocorrência do 17 no array notas
    int pos= Array.IndexOf(notas, 17);
}
```

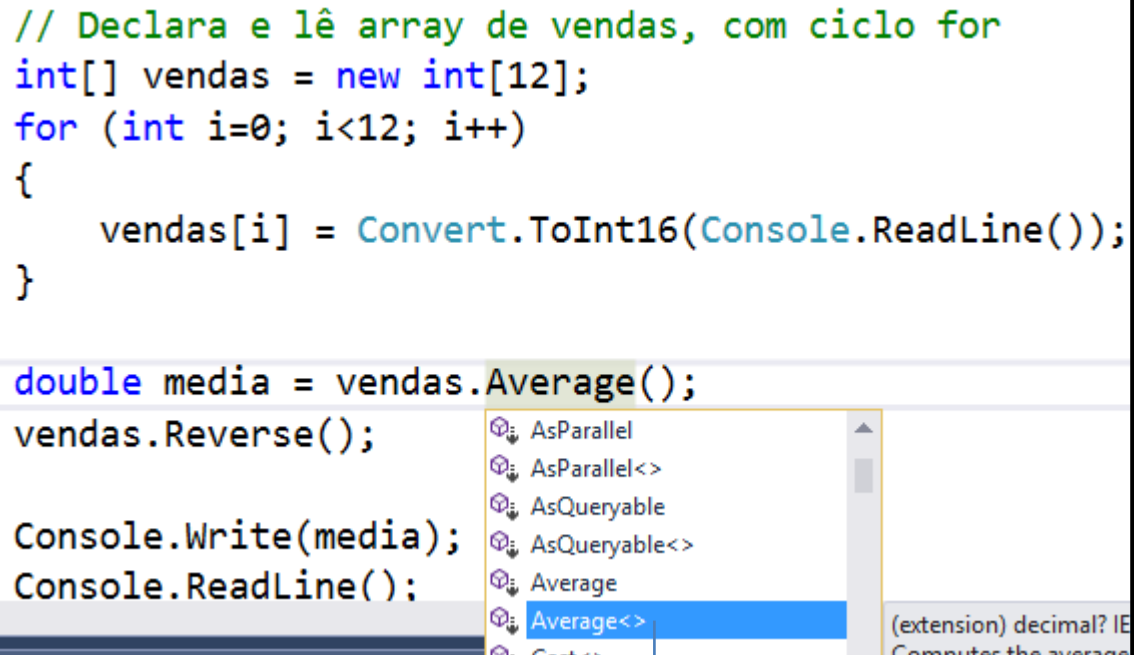
Algoritmia

- C#: A Classe Array

```
// Declara e lê array de vendas, com ciclo for
int[] vendas = new int[12];
for (int i=0; i<12; i++)
{
    vendas[i] = Convert.ToInt16(Console.ReadLine());
}

double media = vendas.Average();
vendas.Reverse();

Console.Write(media);
Console.ReadLine();
```



Inverter a ordem dos elementos do array

Classe Array

Podemos encontrar diversos métodos associados a objetos declarados sob a forma de um array!

Ex: Average, Clone, CopyTo

Obter média de um array

Algoritmia

- C#: A Classe Array

Mais alguns exemplos...

```
// Declara e lê array de vendas, com ciclo for
int[] vendas = new int[12];
for (int i=0; i<12; i++)
{
    vendas[i] = Convert.ToInt16(Console.ReadLine());
}
// valor médio dos elementos do array
double media = vendas.Average();
// maior valor do array
double maximo = vendas.Max();
// menor valor do array
double minimo = vendas.Min();
```

Classe Array

Podemos encontrar diversos métodos associados a objetos declarados sob a forma de um array!

Algoritmia

- C#: A Classe Array

Classe Array - Propriedades

Podemos encontrar diversas Propriedade associadas a objetos declarados sob a forma de um array!

```
// Declara e lê array de vendas, com ciclo for
int[] vendas = new int[12];
for (int i=0; i<12; i++)
{
    vendas[i] = Convert.ToInt16(Console.ReadLine());
}
// valor médio dos elementos do array
int tamanho = vendas.Length;
// Dimensão do array
int num = vendas.Rank;
```

Propriedades	Descrição
Length	Nº de elementos do array
Rank	Nº de dimensões do array (1, 2, 3...)