



INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
RIO GRANDE DO NORTE
Campus Natal - Central

Programação Estruturada e Orientada a Objetos

Estruturas Aninhadas

2013

O que veremos hoje?

- Introdução
- Estruturas Aninhadas
- Exercícios

Transparências baseadas no material do
Prof. Gilbert Azevedo

Estrutura Aninhada

- Uma estrutura é dita aninhada quando possui campos que são registros

```
struct Data {
```

```
    public int Dia, Mes, Ano;
```

```
};
```

```
struct Pessoa {
```

```
    public string Nome, Endereco, Email, Fone;
```

```
    public Data Nascimento;
```

```
};
```

- A estrutura Pessoa é uma estrutura aninhada

Declaração de Variáveis

- Declaração sem iniciar campos
 - Pessoa A;
- Declaração iniciando os campos com valor padrão
 - Pessoa B = new Pessoa();
- É reservada memória para todos os campos de Pessoa e Data

A	Nome	""	
	Endereco	""	
	Email	""	
	Fone	""	
	Nascimento	Dia	0
Mes		0	
Ano		0	

Operador de Seleção "."

- Para referenciar Dia, Mes e Ano do campo Nascimento é necessário utilizar o operador de seleção "." em cascata:
 - Registro.Campo.Campo
- Leitura da data de nascimento da pessoa A
 - A.Nascimento.Dia = int.Parse(Console.ReadLine());
 - A.Nascimento.Mes = int.Parse(Console.ReadLine());
 - A.Nascimento.Ano = int.Parse(Console.ReadLine());

Vetor de Registros

- Em muitas situações práticas surge a necessidade de representar uma coleção de dados onde cada informação é um registro
- Nestes casos, a utilização de um vetor de registros torna-se necessária
- A declaração deste tipo de vetor é idêntica à declaração de vetores de tipos primitivos
- Declaração de um vetor com 100 pessoas
 - `Pessoa[] Agenda = new Pessoa[100];`

Acesso aos Campos no Vetor de Registros

- O acesso aos campos no vetor de registros é realizado da seguinte forma:
 - O operador de indexação "[]" acessa cada registro
 - O operador de seleção "." acessa cada campo
- Leitura da data de nascimento da primeira pessoa do vetor

```
Agenda[0].Nascimento.Dia = int.Parse(Console.ReadLine());
Agenda[0].Nascimento.Mes = int.Parse(Console.ReadLine());
Agenda[0].Nascimento.Ano = int.Parse(Console.ReadLine());
```

Exemplo

- Aplicativo para cadastrar uma agenda de 100 pessoas

```
struct Data {
```

```
    public int Dia, Mes, Ano;
```

```
};
```

```
struct Pessoa {
```

```
    public string Nome, Endereco, Email, Fone;
```

```
    public Data Nascimento;
```

```
};
```

```
Pessoa[] Agenda = new Pessoa[100];
```

Exemplo

```
static void Main(string[] args)
{
    Pessoa[] Agenda = new Pessoa[100];
    int op = 1, i, k = 0;
    while (op != 0)
    {
        // Menu
        Console.Write("Opções: 1-Inserir, 2-Listar, 0-Fim: ");
        op = int.Parse(Console.ReadLine());
        // Inserir
        // Listar
    }
    Console.ReadKey();
}
```

Exemplo

```
// Inserir
if (op == 1 && k < 100)
{
    Console.Write("Nome: ");    Agenda[k].Nome = Console.ReadLine();
    Console.Write("Endereço: "); Agenda[k].Endereco = Console.ReadLine();
    Console.Write("Email: ");   Agenda[k].Email = Console.ReadLine();
    Console.Write("Fone: ");    Agenda[k].Fone = Console.ReadLine();
    Console.Write("Dia: ");
    Agenda[k].Nascimento.Dia = int.Parse(Console.ReadLine());
    Console.Write("Mês: ");
    Agenda[k].Nascimento.Mes = int.Parse(Console.ReadLine());
    Console.Write("Ano: ");
    Agenda[k].Nascimento.Ano = int.Parse(Console.ReadLine());
    k++;
}
```

Exemplo

```
// Listar
if (op == 2)
{
    for (i = 0; i < k; i++)
    {
        Console.WriteLine("Nome : {0}", Agenda[i].Nome);
        Console.WriteLine("Endereço : {0}", Agenda[i].Endereco);
        Console.WriteLine("Email : {0}", Agenda[i].Email);
        Console.WriteLine("Fone : {0}", Agenda[i].Fone);
        Console.WriteLine("Nascimento : {0}/{1}/{2}",
            Agenda[i].Nascimento.Dia, Agenda[i].Nascimento.Mes,
            Agenda[i].Nascimento.Ano);
    }
}
```

Exercícios

- 1. Definir uma estrutura para representar os dados dos componentes de um computador, armazenando o nome do componente e o preço. Implementar uma aplicação que permita cadastrar até 20 componentes em um micro, listar os componentes cadastrados e totalizar o custo total do equipamento.
- 2. Definir uma estrutura para representar os dados das disciplinas do seu curso, armazenando semestre, nome da disciplina, média final e situação (aprovado ou reprovado). Implementar uma aplicação que permita cadastrar um vetor com até 50 disciplinas, listar todas as disciplinas cadastradas, obter as disciplinas com a maior e a menor média e calcular a média das médias de todas as disciplinas em que tenha sido aprovado.

Dúvidas

