



Programação Estruturada e Orientada a Objetos

Estruturas

2013

O que veremos hoje?

- Introdução
- Revisão Estruturas
- Exercícios

Transparências baseadas no material do
Prof. Gilbert Azevedo

Estrutura (struct)

- Tipo de dados utilizado para representar uma coleção de valores de tipos diferentes
- Uma variável do tipo estrutura armazena diversos valores que são referenciados por identificadores denominados de membros ou campos
- Diferentemente de vetores e matrizes, onde os elementos são de um mesmo tipo, cada campo de uma estrutura pode ser de um tipo diferente

Declaração de Estrutura

- A declaração de uma estrutura é realizada através da palavra chave *struct*
- Nesta declaração é informado um identificador para nomear a estrutura e todos os seus campos com os respectivos tipos
 - `struct Nome_da_Estrutura {`
 - `public Tipo1 Campo1, Campo2;`
 - `public TipoN CampoN;`
 - `}`

Exemplos de Estruturas

- Estrutura para representar as notas de uma disciplina: notas dos bimestres, nota da prova final e média
 - struct Disciplina {
 - public double Bim1, Bim2;
 - public double ProvaFinal;
 - public double Media;
 - };

Exemplos de Estruturas

- Estrutura para representar os dados de uma conta bancária: número da conta, titular e saldo
 - struct ContaCorrente {
 - public string Numero;
 - public string Titular;
 - public double Saldo;
 - };

Declaração de Variáveis

- Quando uma estrutura é declarada, nenhum espaço de memória é reservado, ou seja, nenhuma informação pode ser armazenada
- A estrutura apenas define o modelo que será utilizado quando as variáveis forem declaradas
- As variáveis são chamadas de registros
- A declaração de variáveis é realizada de forma análoga às variáveis de tipos primitivos
 - Nome_da_Estrutura1 Variável1, Variável2;
 - Nome_da_EstruturaN VariávelN;

Exemplos de Variáveis

- Declaração de variáveis da estrutura Disciplina
- Sem iniciar os campos
 - Disciplina Por;
- Iniciando os campos com valor padrão
 - Disciplina Mat = new Disciplina();

Por	Bim1	
	Bim2	
	ProvaFinal	
	Media	

Mat	Bim1	0
	Bim2	0
	ProvaFinal	0
	Media	0

Operador de Seleção "."

- Cada campo de um registro é referenciado individualmente através do nome da variável, do operador de seleção "." e do nome do campo, conforme abaixo:
 - Registro.Campo
- Quando um campo é referenciado, as operações de escrita e leitura podem ser realizadas de forma análoga à utilizada para as variáveis de tipos primitivos

Operações Básicas em Campos

- Leitura dos campos Bim1 e Bim2 do registro Por
 - `Por.Bim1 = double.Parse(Console.ReadLine());`
 - `Por.Bim2 = double.Parse(Console.ReadLine());`
- Atribuição do campo Media do registro Por
 - `Por.Media = (Por.Bim1 + Por.Bim2)/2;`
- Escrita do campo Media de Por
 - `Console.WriteLine(Por.Media);`

Por	Bim1	0
	Bim2	0
	ProvaFinal	0
	Media	0

Atribuições entre Registros

- O operador de atribuição pode ser utilizado para copiar todos os campos de um registro para outro, desde que estes sejam variáveis da mesma estrutura
 - Supondo que *Por* tenha os valores { 10, 6, 0, 8 };
 - Disciplina *Mat*;
 - *Mat* = *Por*;
 - *// Mat* é iniciado com os mesmos valores de *Por*

Por	Bim1	10
	Bim2	6
	ProvaFinal	0
	Media	8

Mat	Bim1	10
	Bim2	6
	ProvaFinal	0
	Media	8

Operadores e Registros

- Os operadores aritméticos, relacionais e lógicos não são pré-programados para realizar operações sobre registros
- Portanto, são inválidas as operações abaixo:
 - $\text{Por} = 2 * \text{Mat};$
 - $\text{Por} = \text{Por} + \text{Mat};$
 - $\text{Por} = !\text{Mat};$
 - $\text{Por} = \text{Por} \&\& \text{Mat};$
- É possível, entretanto, programar os operadores para estruturas utilizando uma metodologia conhecida por sobrecarga de operadores

Exemplo

- Algoritmo para calcular a média de uma disciplina, dados as notas dos bimestres e da prova final
 - struct Disciplina {
 - public double Bim1, Bim2;
 - public double ProvaFinal;
 - public double Media;
 - };

Exemplo

```
static void Main(string[] args)
{
    Disciplina por;
    Console.WriteLine("Entre as notas dos 2 bimestres e da
        Prova Final");
    por.Bim1 = double.Parse(Console.ReadLine());
    por.Bim2 = double.Parse(Console.ReadLine());
    por.ProvaFinal = double.Parse(Console.ReadLine());
    por.Media = (por.Bim1 + por.Bim2) / 2;
    if (por.Media < 6)
        por.Media = (por.Media + por.ProvaFinal) / 2;
    Console.WriteLine("Media = {0}", por.Media);
    Console.ReadKey();
}
```

Exercícios

- 1. Definir uma estrutura para representar os dados de massa corporal de uma pessoa: peso, altura e IMC (índice de massa corporal). Implementar uma aplicação que permita ao usuário definir o peso e a altura e calcular o IMC, através da fórmula $IMC = \text{peso} / \text{altura}^2$, armazenando esses dados em uma variável desta estrutura.
- 2. Definir uma estrutura para representar um número racional: número com numerador e denominador inteiros. Implementar uma aplicação que permita ao usuário entrar dois números racionais e realizar as operações de adição, subtração, multiplicação e divisão entre eles, gerando um terceiro número racional.
- 3. Definir uma estrutura para representar os dados de uma data: número do dia, do mês e do ano. Implementar uma aplicação que permita ao usuário entrar uma data no formato "dd/mm/aaaa" e obter um registro desta estrutura.

Dúvidas

