

Algoritmos II

Ricardo Ribeiro Assink
ricardo.assink@unisul.br
ricardo@equipedigital.com

<http://www.ricardoassink.com.br/>

Algoritmos II

Variáveis Compostas Heterogêneas

Os tipos de dados que são mais comumente usados são:

- 1.int (inteiro)
- 2.double (real)
- 3.String (Literal)
- 4.boolean (Lógico)

Uma outra forma de definir uma variável é através do tipo Registro. Esse tipo é diferente das demais formas de definir variáveis, porque permite que uma variável armazene valores de diversos tipos diferentes. Ao contrário dos vetores e matrizes onde todas as posições de memória são de um mesmo tipo de dado.

Algoritmos II

Variáveis Compostas Heterogêneas

Com este novo tipo de declaração você pode criar variáveis compostas heterogêneas definindo novos tipos de dados. Exemplo: Imagine que fosse desejado armazenar informações de uma pessoa, tais como: Nome, Idade, Altura, Sexo, Número de Dependentes, Profissão.

No JAVA um tipo Registro pode ser criado na forma de uma classe interna.

Algoritmos II

Variáveis Compostas Heterogêneas

Na forma tradicional, seria necessário definir uma variável para cada tipo de informação, ou seja:

```
public class Declaracao  
{  
    public static void main(String args[])  
    {  
        String nome;  
        double altura;  
        boolean sexo;  
        int idade, numerodependente;  
        String profissao;  
    }  
}
```

Algoritmos II

Variáveis Compostas Heterogêneas

Utilizando o tipo Registro, implementado como uma classe interna do JAVA, a definição seria a seguinte:

```
public class Declaracao{  
    static class Pessoa  {  
        String nome;  
        double altura;  
        boolean sexo;  
        int idade, numerodependente;  
        String profissao;  
    }  
    public static void main(String args[]) {  
        <comandos>  
    }  
}
```

Algoritmos II

Variáveis Compostas Heterogêneas

Ao definir uma variável como sendo do tipo Registro, devemos definir, também quais serão as partes componentes desta variável (nome, idade, altura, sexo, numerodependente e profissão), junto com o seu tipo.

Componentes de mesmo tipo são separados por vírgulas (,).

Componentes de tipos diferentes são separadas por ponto e vírgula (;).

Algoritmos II

Variáveis Compostas Heterogêneas

Pessoa	nome	peessoa.nome
	altura	peessoa.altura
	sexo	peessoa.sexo
	idade	peessoa.idade
	numero dependente	peessoa.numero dependente
	profissao	peessoa.profissao

Quando estamos trabalhando com **Registro**, as partes componentes do mesmo recebem um Nome próprio, o qual é conhecido como “campo” ou “membro”.

Algoritmos II

Variáveis Compostas Heterogêneas

Desta forma, um variável Registro pode ter campos de qualquer tipo válido, sendo permitido inclusive que um Registro seja definido dentro do outro, ou como parte de um Vetor ou Matriz.

peessoas	nome	peessoas[1].nome
	altura	peessoas[1].altura
	sexo	peessoas[1].sexo
	idade	peessoas[1].idade
	numerodependente	peessoas[1].numerodependente
	profissao	peessoas[1].profissao
	nome	peessoas[2].nome
	altura	peessoas[2].altura
	sexo	peessoas[2].sexo
	idade	peessoas[2].idade
	numerodependente	peessoas[2].numerodependente
	profissao	peessoas[2].profissao

Algoritmos II

Variáveis Compostas Heterogêneas

Com este novo recurso você pode abstrair dados complexos como Pessoa, Cliente, Fornecedor, Automóvel, Figura Geométrica, e assim por diante. Todas as entidades são compostas por diversos tipos de dados que precisam ser manuseados juntos durante a execução de um algoritmo. Estes tipos de dados complexos são chamados de **Tipo Abstrato de Dados ou simplesmente TAD**.

Os tipos abstratos de dados podem ser definidos antes da declaração de variáveis, com isto podem aparecer mais de uma vez na declaração de variáveis apenas utilizando a sua referência.

Algoritmos II

Variáveis Compostas Heterogêneas

```
public class DeclaracaoTipo{  
    static final int TAMANHO = 10;  
    static class Pessoa {  
        String nome;  
        double altura;  
        boolean sexo;  
        int idade, numerodependente;  
        String profissao;  
    }  
    public static void main(String args[]) {  
        Pessoa pessoas[] = new Pessoa[TAMANHO];  
        Pessoa joao;  
    }  
}
```

Algoritmos II

Variáveis Compostas Heterogêneas

Continuando o Exemplo, caso desejarmos atribuir um valor a variável Pessoa, devemos fazê-lo da seguinte forma:

```
public class DeclaracaoTipo{
    static final int TAMANHO = 10;
    static class Pessoa {
        String nome;
        double altura;
        boolean sexo;
        int idade, numerodependente;
        String profissao;
    }
    public static void main(String args[]) {
        Pessoa pessoas[] = new Pessoa[TAMANHO];
        Pessoa joao = new Pessoa();
        joao.idade = 19;
        pessoas[0] = new Pessoa();
        pessoas[0].idade = 20;
    }
}
```

Algoritmos II

Variáveis Compostas Heterogêneas

O uso do “.” indica que esta variável possui campos, e que “idade” é um deles. É importante lembrar que as operações realizadas sobre uma variável Registro, são as mesmas de uma variável “comum”, a única diferença que devemos indicar o Nome da variável, seguido de um ponto(.), seguido do Nome do campo.

Algoritmos II

Variáveis Compostas Heterogêneas

Exercícios:

1 - Faça um programa que possua uma constante para o tamanho do vetor que será utilizado e uma classe interna chamada Pessoa para manipular os seguintes dados de cada pessoa.

nome(String),
altura(Double),
idade (int),
profissao(String)

Construa um vetor do tipo Pessoa de tamanho 3, solicite todos os dados para o usuário e depois mostre o resultado de cada pessoa.

Algoritmos II

Variáveis Compostas Heterogêneas

Exercícios:

2 - Considere que uma figura geométrica possui área, cor e tipo(1-Triângulo, 2-Círculo e 3-Quadrado). Faça um algoritmo que armazene N figuras geométricas em um vetor. Após a leitura determine:

- a) O total da área das figuras geométricas;
- b) O tipo da maior figura geométrica;
- c) A quantidade de círculos;
- d) O percentual de quadrado em relação ao total.

Algoritmos II

Variáveis Compostas Heterogêneas

Exercícios:

3 -Faça um algoritmo que:

- a) Defina um Registro para aluno tendo os seguintes campos: Nome, Semestre, Sala, Curso, Notas(total de seis)
- b) Ler as informações, descritas acima.
- c) Ampliar a definição anterior, acrescentando a definição de um outro campo(Endereço) que será também um Registro, o qual terá os seguintes campos: Rua, Numero, Bairro, Cidade, Estado, CEP
- d) Ler as informações de um aluno, junto com o endereço descrito acima.
- e) Declare um Vetor de 100 posições de alunos, os campos serão os mesmos descritos nos itens anteriores.
- f) Ler as informações de N alunos
- g) Ordene crescentemente pelo Nome, os alunos. Mostre os dados antes e depois da ordenação.

Algoritmos II

FIM

Material Original: Osmar de Oliveira Braz Junior