

---

---

# Registros

## Aula 5

---

---

# Variáveis Compostas Heterogêneas

- Bom, sabemos que uma Variável Composta Homogênea “guarda” um conjunto de valores de um mesmo tipo em um vetor ou matriz.
- As Variáveis Compostas Heterogêneas são também um conjunto de valores, mas de tipos diferentes, agora armazenadas em um **registro**.



Um registro é capaz de guardar informações de tipos diferentes.

# Registros

- Registros, no VisuAlg, são um conjunto heterogêneos de dados.
- Imagine um formulário de inscrição que possui vários campos diferentes para serem preenchidos, alguns numéricos (idade, altura, peso) e outros de texto (nome, endereço, sexo), mas todos referentes às informações de apenas um indivíduo.
  - Se fossemos guardar todas essas informações, precisaríamos de várias variáveis de tipos diferentes.
  - O registro permite guardar todos esses valores dentro de uma única variável.
  - O registro define um tipo de dado que será o resultado da composição de outras variáveis.
  - Dentro do registro, há espaços para guardar diversos valores diferentes.

# Registros

- Partindo dessa idéia, podemos associar registros a qualquer coisa que possua diversas características específicas, mas que descrevem como conjunto apenas uma entidade.
- Por exemplo, uma pessoa possui várias características como: altura, peso, sexo, idade que fazem parte do registro pessoa.



## Registro Pessoa

nome: caractere

idade: inteiro

sexo: caractere

altura: real

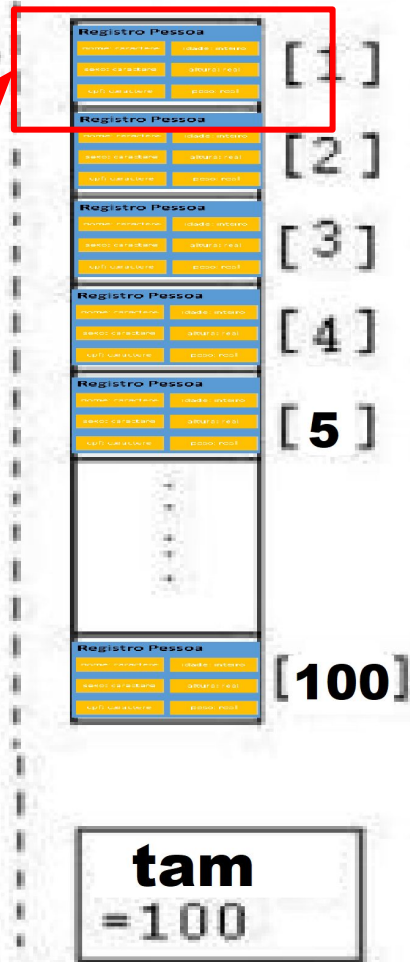
cpf: caractere

peso: real

# Registros

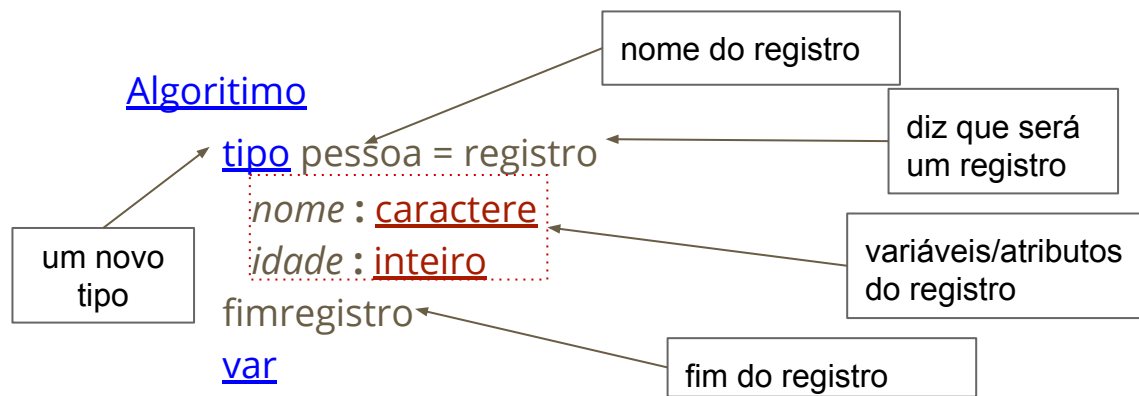
- Talvez, você pense que é mais simples usar várias variáveis para guardar as informações de uma única pessoa.
- Entretanto, imagine que você quer guardar as informações de centenas de pessoas. Você terá de usar diversos vetores, cada um com um tipo específico de informação. Provavelmente, você terá de assegurar que as informações de uma pessoa estejam nas mesmas posições dos diferentes vetores. **Gerenciar isso, pode não ser uma tarefa simples.**
- Ao optar por Registros, você pode criar um vetor do tipo do Registro e as informações de cada Pessoa estarão dentro de uma mesma posição, do seu vetor de Registros.

Cada um desses  
é uma pessoa  
diferente.



# Criando registros

- No VisuAlg, a declaração de registros é um pouco diferente e acontece antes mesmo da palavra-chave var, usando palavra-chave tipo:



- A palavra reservada tipo indica que estamos definindo um novo tipo de dado (além dos tipos caractere, inteiro, lógico, etc.)
- O registro permitirá criar variáveis desse tipo e atribuir valores aos seus “campos”, além de generalizar operações com os mesmos.

# Criando registros

- A seguir colocamos o nome do registro (pessoa), o operador = (igual) e a palavra **registro** (dependendo da versão do VisuAlg essa palavra tem um destaque de cor diferente).  
`tipo pessoa = registro`
- Depois declaramos os campos do registro, que são nada mais do que variáveis que estarão ligadas ao registro.  
`nome : caractere`  
`idade : inteiro`
- Assim, o registro pessoa tem um nome e uma idade que estão somente atrelados a este registro, ou seja não podemos usar essas variáveis, nem nome nem idade, diretamente sem referenciar o registro ao qual elas pertencem.
- Por último usamos, o fimregistro (dependendo da versão do VisuAlg essa palavra tem um destaque de cor diferente) para finalizar o registro, tudo que for declarado após essa palavra não pertencerá ao registro criado.  
`fimregistro`

# Declarando registros

- Para que possam ser usados, precisamos declarar variáveis do tipo do registro. A partir do momento em que criamos um registro, ele pode ser considerado como um novo tipo de variável e, assim, poderemos declará-lo da mesma forma que outros tipos de dados que já conhecemos:

```
fimregistro
```

```
var
```

```
    cliente : pessoa
```

```
    // outras variáveis...
```

- Então, criamos uma variável chamada cliente que é do tipo pessoa, esse cliente possui um nome e uma idade, aos quais podemos atribuir valores para eles.
- Note que, no VisuAlg, o tipo pessoa não fica nem sublinhado nem com a cor avermelhada. Isso é normal, já que não estamos referenciando uma palavra chave do VisuAlg.



# Atribuindo valores para os registros

- Não podemos atribuir valores para um registro diretamente. Logo, o código abaixo não funciona:  
    cliente <- "João".
- Devemos usar o operador . (ponto) para acessar o campo que queremos atribuir valor. Por exemplo, se quisermos dar o nome para o cliente que criamos, fazemos da seguinte maneira.

Início

```
cliente.nome <- "leonardo"
```

- Se quisermos atribuir o valor para idade agora:

```
cliente.nome <- "leonardo"
```

```
cliente.idade <- 29
```

# Usando valores de registros

- Assim como na atribuição, para usar valores de uma determinado campo do registro, temos que usar o operador `.` (ponto) para acessar o valor guardado nele. Por exemplo, se quisermos escrever o nome do cliente que criamos, fazemos da seguinte maneira:

Início

```
escreval(cliente.nome)
```

- Da mesma forma para a idade:

```
escreval(cliente.nome)
```

```
escreval(cliente.idade)
```

# Cadastro de pessoas

- Para exemplificar melhor, vamos criar um simples programa que o usuário possa cadastrar 4 pessoas, informando o nome e a idade de cada uma delas. Ao cadastrar todos, o programa imprime as informações de cada pessoa na tela.
- Primeiro vamos criar o registro pessoa:

Algoritmo

tipo pessoa = registro

nome : caractere

idade : inteiro

fimregistro

var

- Criamos um campo nome do tipo caractere e uma idade do tipo inteiro para o nosso registro chamado pessoa.

# Cadastro de pessoas

- Em seguida precisaremos declarar um vetor do tipo do nosso registro, isto é, do tipo pessoa. Esse vetor guardará as informações das 4 pessoas. Vamos criar um vetor com o tamanho de 0 a 3 (com 4 espaços para atribuir valores) e vamos dizer que esse vetor é do tipo pessoa (lembramos que pessoa agora é um novo tipo de dado):

var

    pessoas: vetor[0..3] de pessoa

    i: inteiro

- Dessa forma, temos um vetor de 0 a 3 chamado pessoas do tipo pessoa, ou seja podemos armazenar nesse vetor até quatro pessoas, sendo que cada pessoa possui duas informações: nome e idade.
- Além disso, criamos também uma variável “ i ” que servirá como contador para iteração do loop quando formos percorrer cada pessoa do vetor.

# Cadastro de pessoas

- Agora, precisamos criar um **para** para percorrer o vetor e adicionar uma pessoa juntamente com o nome e a idade em cada posição do nosso vetor.

Início

**para** i **de** 0 **ate** 3 **faca**

**escreva**("Digite o nome da pessoa: ")

- Assim temos um loop que vai rodar 4 vezes (o número que precisamos para informar os dados das pessoas). Também usamos a variável "i" que será incrementada a cada loop. Por último, usamos o comando **escreva** para pedir para o usuário escrever o nome da pessoa.
- Agora, precisaremos adicionar esse nome a uma pessoa e essa pessoa está na posição "i" do vetor (já que i vai ser incrementado a cada repetição do laço, assim a posição i vai mudar de 0 a 3).

# Cadastro de pessoas

- Vamos usar os [] (colchetes) para informar a posição do vetor que a pessoa deve ser adicionado no vetor, em seguida usamos normalmente o . (ponto) para referenciar o atributo nome da pessoa.

```
para i de 0 ate 3 faça  
    escreva("Digite o nome do pessoa: ")  
    leia(pessoas[i].nome)
```

- Dessa forma podemos ler uma nova pessoa na posição " i " do vetor pessoas com o nome que o usuário digitar, ou seja, a cada loop será adicionado uma nova pessoa com o nomes que o usuário informar. Ex: se o usuário digitar João no primeiro loop, vai ser adicionado na posição 0 do vetor o pessoas com o nome João e da mesma forma para as próximas iterações do loop.

# Cadastro de pessoas

- Além do nome precisamos informar a idade da pessoa, basta fazer da mesma forma que fizemos com o nome:

Início

```
para i de 0 ate 3 faca
    escreva("Digite o nome da pessoa: ")
    leia(pessoas[i].nome)
    escreva("Digite a idade da pessoa: ")
    leia(pessoas[i].idade)
fimpara
```

- Pronto agora podemos preencher todo o vetor de pessoas com nomes e idades informadas pelo o usuário.

# Cadastro de pessoas

- Agora só falta percorrer o vetor já preenchido e informar os atributos de cada pessoa. Vamos reutilizar a variável “ i ” para servir como contador, então precisamos “resetar” seu valor para 0:

```
fimpara
```

```
i <- 0
```

```
para i de 0 ate 3 faça
```

```
    escreval(pessoas[i].nome)
```

```
    escreval(pessoas[i].idade)
```

```
fimpara
```

- Pronto, dessa forma quando o usuário cadastra todas as pessoas, é impresso na tela todos os nomes e idades respectivamente de cada uma dessas pessoas.



# Código Completo

## VisuAlg

**Algoritmo** "CadastroPessoas"

**tipo** pessoa = registro

nome : caractere

idade : inteiro

fimregistro

**Var**

pessoas : vetor[0..3] de pessoa

i: inteiro

**Inicio**

para i de 0 ate 3 faca

    escreva("Digite o nome do pessoa: ")

    leia(pessoas[i].nome)

    escreva("Digite a idade do pessoa: ")

    leia(pessoas[i].idade)

fimpara

i <- 0

para i de 0 ate 3 faca

    escreval(pessoas[i].nome)

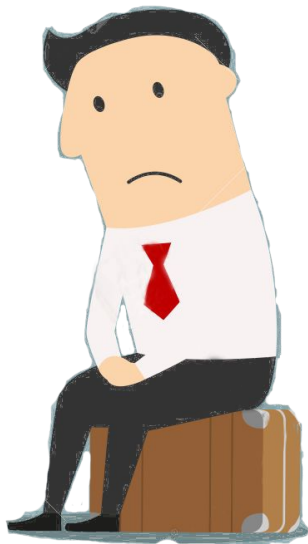
    escreval(pessoas[i].idade)

fimpara

**Fimalgoritmo**

# Código Completo

- Infelizmente, não é possível aplicar o conceito de registros no Portugol Studio, pois ele não trabalha com variáveis compostas heterogêneas.



# Aprenda Mais

- coditube - Registros no Visualg - <<https://youtu.be/Sr2uieF720k0>>
- Aula 6 ead - Elaine Cecília Gatto  
<<https://pt.slideshare.net/elainececiliagatto/aula-6-aed-registros>>
- Neri Neitzke - Aula 1212 Algorit e Log-Variáveis Heterogêneas Registro -  
<<https://youtu.be/4chxP3N3s-A>> \*
- Neri Neitzke - Aula 1213 Algorit e Log-Variáveis Heterogêneas Registro -  
<<https://youtu.be/suBerbZE9XM>> \*
- Aula 8 - Registros -  
<<https://sites.google.com/site/unipliconstrucaodealgoritmos/aulas/aula-8---registros>>
- Algoritmos - Manzano - Capítulo 9 -  
<<https://docs.google.com/file/d/0ByLDgIKIkMi2MTc0MDJmOWItNTQ1Yy00NTczLWI0OTQtNjA1NDI5YTllYzZm/edit?hl=en>> \*

\* essas recomendações não usam a sintaxe do Português do VisuAlg

# Exercício

1. Crie um registro chamado carro, e adicione atributos para esse registro como marca, modelo, placa. Depois peça para o usuário informar os valores de um carro e os imprima na tela.
2. Modifique a questão anterior criando um vetor que vai comportar 10 carros peça para o usuário informar os dados de carros e depois imprima os atributos de cada carro cadastrado na tela.