FAB - Faculdade Barão do Rio Branco
FAC - Faculdade do Acre

Curso de Sistemas de Informação
Disciplina: Algoritmos – 1º e 2º Períodos – Turmas 216071A, 216071B e 216062
Notas de Aula – Aulas 07, 08 e 09.
Professor: Edkallenn

Aula 07

Variáveis

Talvez o conceito de variável seja o mais importante na construção de algoritmos. Dificilmente um algoritmo é feito sem que sejam utilizadas variáveis; é muito importante, pois, compreender o que são e como devemos usá-las.

Até o momento, vimos que a estrutura de um computador segue uma seqüência:

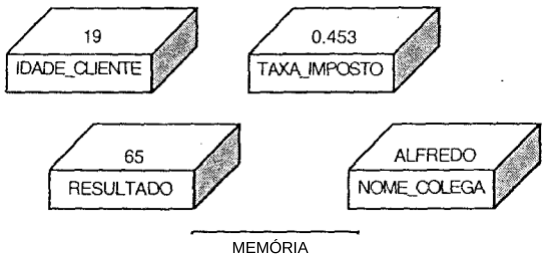
ENTRADA => PROCESSAMENTO => SAÍDA

Dados e programas ficam armazenados na memória. Quanto aos programas, cada comando é executado passo a passo pela UCP. Isso se chama execução seqüencial.

Quanto aos dados, vimos que existem tipos básicos, que dependem da sua natureza. Ocorre que os dados não podem ser armazenados na memória "soltos". Para ensinar ao compu¬tador como manipular um dado, o programador precisa nomear um lugar na memória onde ele será guardado - a variável.

Uma variável, portanto, é um local com um nome dentro da memória do computador, criado em um algoritmo para se armazenar um determinado dado.

Uma analogia muito comum é imaginar uma variável como uma caixinha dentro da memória do computador. Esta caixa tem a capacidade de guardar um dado de certo tipo; e o nome da variável é como uma etiqueta colada na frente da caixa. Veja a figura abaixo:



Uma representação abstrata das variáveis na memória do computador.

Esta noção de variável é um pouco diferente daquela a qual estamos acostumados em matemática. Ela possui este nome porque o dado nela armazenado pode se modificar (isto é, pode variar) durante a execução do algo ritmo pela máquina.

As variáveis existem na memória durante a execução do algoritmo. Quando ele termina, é como se todos os dados fossem apagados, e as caixas destruídas.

Nomes de Variáveis

Toda variável precisa ter um nome definido pelo programador, que deve ser **único** dentro de um mesmo algoritmo; com este nome o programador poderá fazer referências à variável no português estruturado.

Os nomes das variáveis devem começar por uma letra e depois conter letras, números ou *underline*, até um limite de 30 caracteres. Não pode haver duas variáveis com o mesmo nome.

Na nossa sintaxe, todo nome de variável deve começar por uma letra, e depois deve incluir apenas letras, dígitos ou ainda o sinal de sublinhado ("_"). Em programação, estes e outros nomes que são inventados pelo programador são chamados de identificadores. Nos nossos algoritmos os identificadores não poderão conter acentos, nem podem ter o mesmo nome de uma palavra usada como parte de um comando (veremos adiante que existe um comando LEIA, de modo que uma variável não pode se chamar LEIA).

Veja alguns exemplos de nomes de variável válidos:

```
valor_pago
soma
aa
k
num
precofinal
litros1
nome_da_funcionaria
idade
```

Veja agora exemplos de nomes de variáveis inválidos:

```
1nota
```
Não começa por uma letra

```
médiaarit
```
Possui um acento

```
quantidade inicial
```
Possui um espaço

```
valor_em_R$
```
Possui um caractere inválido ($)

A regra que estamos estabelecendo para a formação de um identificador é muito parecida com aquelas utilizadas em programação, por isto, você deve se acostumar a criar variáveis com os nomes neste formato.

Tipo de dado das variáveis

Toda variável deve ter um tipo a ela relacionado. Quando criamos uma variável, é preciso também definir qual o tipo de dado que ela vai armazenar.

O portugol prevê quatro tipos de dados: inteiro, real, cadeia de caracteres e lógico (ou booleano). As palavras-chave que os definem são as seguintes (observe que elas não têm acentuação):

- `inteiro`: define variáveis numéricas do tipo inteiro, ou seja, sem casas decimais.
- `real`: define variáveis numéricas do tipo real, ou seja, com casas decimais.
- `caractere`: define variáveis do tipo string, ou seja, cadeia de caracteres.
- `logico`: define variáveis do tipo booleano, ou seja, com valor VERDADEIRO ou FALSO.

Veja alguns exemplos:

- Queremos armazenar o preço de uma blusa em um algoritmo para uma loja de roupas. Podemos criar uma variável chamada PRECO_PRODUTO, e o tipo de dado que ela vai guardar é `real`;
- Uma variável que vai armazenar a quantidade de latas de cerveja no estoque de um bar. Podemos criar uma variável chamada QUANT_LATAS, e o tipo de dado que ela vai guardar é `inteiro`;
- Uma variável que vai armazenar o nome do cliente de um supermercado. Podemos definir uma variável chamada NOME_CLIENTE, e o seu tipo de dado será uma cadeia representada pelo tipo `caractere`

Aula 08

Declaração de Variáveis

Toda variável usada em um algoritmo precisa ser declarada, Isto deve ser feito no início do algoritmo. A seção de declaração de variáveis começa com a palavra-chave `var`, e continua da seguinte forma:

```
<variável1>, <variável2>, ... : <tipo>
```

Sendo do mesmo tipo, podemos declarar mais de uma variável na mesma linha; após o sinal de dois-pontos (separando cada uma com vírgulas). Os tipos básicos a serem declarados são Inteiro, Real, Lógico, ou Caractere (que reúne os tipos caractere e cadeia). Veja alguns exemplos:

```
quant_latas, quant_garrafas:inteiro
saldo, taxa, bonus:real
nome_funcionario:caractere
```

Há uma observação muito importante sobre o uso de variáveis: uma variável só pode armazenar um dado de cada vez! Qualquer valor armazenado em uma variável sempre vai apagar algum outro valor guardado anteriormente dentro dela, no mesmo algoritmo.

Suponha que você faça um algoritmo em que haja necessidade de guardar a idade de um cliente. Você pode guardar a idade 19 numa variável, digamos IDADE_CLIENTE, mas se outra idade de cliente for armazenada na variável do mesmo algoritmo, por exemplo 23, o valor 19 desaparece para dar lugar ao 23.

Agora que sabemos o que é e como declarar uma variável, vamos às definições das operações básicas de nossa linguagem algorítmica.

FAB - Faculdade Barão do Rio Branco
FAC - Faculdade do Acre

Comando de Atribuição

Chamamos de atribuição ao comando que permite armazenar o resultado de uma expressão dentro de uma variável, sendo considerada a instrução mais simples que podemos solicitar à máquina. A sintaxe no português estruturado é:

variável <- expressão

A expressão pode incluir números. Operadores, parênteses e nomes de variáveis, desde que respeitada a sintaxe de montagem das expressões vista anteriormente.

É comum entender o comando de atribuição como se a seta representasse a palavra "recebe". Deste modo, podemos ler uma atribuição como: "uma variável recebe o resultado de uma expressão".

Deve haver compatibilidade entre o tipo de dado resultante da avaliação da expressão e o tipo de dado da variável.

A variável deve sempre ter sido declarada anteriormente. **(muito importante – não esquecer)**

Exemplos:

```
valor <- preco_unitario * quant
```
A variável VALOR recebe o resultado da multiplicação da variável PRECO_UNITARIO por QUANT.

```
soma <- valor1 + valor2
```
A variável SOMA recebe a soma de VALOR1 com VALOR2.

```
salario_liquido <- (salario_bruto -100)*0.90
```
A variável SALARIO_LIQUIDO recebe o resultado da subtração de SALARIO_BRUTO por 100, multiplicada por 0.90.

```
preco_reais <- preco_dolar * 2.53
```
PRECO_REAIS recebe o valor de PRECO_DOLAR multiplicado por 2.53.

```
num <- num * 2
```
A variável NUM recebe o seu conteúdo atual multiplicado por 2.

Relembrando:

A atribuição de valores a variáveis é feita com o operador **<-**. Do seu lado esquerdo fica a variável à qual está sendo atribuído o valor, e à sua direita pode-se colocar qualquer expressão (constantes, variáveis, expressões numéricas), desde que seu resultado tenha tipo igual ao da variável.

Mais alguns exemplos de atribuições:

```
a <- 3
nome_do_aluno <- "José da Silva"
flag <- FALSO
```

FAB - Faculdade Barão do Rio Branco
FAC - Faculdade do Acre

Aula 09

Comando de Entrada e Saída de Dados

Quando ensinamos o computador a executar algo, na grande maioria das vezes a máquina precisa de dados de entrada a serem processados. Deste modo, nosso português estruturado precisa de um comando para ensinar ao computador quando ele deve solicitar dados do usuário. Esta solicitação é freqüentemente chamada de entrada ou leitura de dados, e é feita através de um dispositivo de entrada (um teclado, por exemplo).

A sintaxe do comando de entrada de dados é:

```
leia (<variável>)
```

Recebe valor digitado pelos usuário, atribuindo à variável cujo nomes está entre parênteses.

LEMBRANDO: OS PARÊNTESES SÃO OBRIGATÓRIOS.

Quando a máquina encontra um comando de entrada na seqüência de execução de um algoritmo, ela interrompe o processamento e aguarda que o usuário forneça um ou mais dados. Todo dado fornecido é armazenado em uma variável indicada pelo programador; se o programador indicou, por exemplo, três variáveis no comando de entrada; significa que o usuário deverá fornecer três dados para o algoritmo, compatíveis com o tipo de dado de cada. variável.

Veja no exemplo abaixo o resultado:

```
algoritmo "exemplo 1"
var x: inteiro;
inicio
leia (x)
escreva (x)
fimalgoritmo
```

O algoritmo acima declara uma variável inteira chamada x, solicita que o usuário insira um valor para a mesma e depois a “escreve”, a exibe na tela.

Quando há um comando de entrada no algoritmo, o computador pára e espera que o usuário forneça o dado, e só após esta leitura ele continua a executar o algoritmo.

Comando de Saída de Dados

Assim como o computador precisa de dados de entrada, não faria sentido fazer c processamento de dados sem que as respostas fossem exibidas, Portanto, usaremos um comando de saída de dados em nossos algoritmos, indicando que a máquina deve "escrever" (ou imprimir) resultados e mensagens em um dispositivo de saída (digamos um monitor de vídeo).

A sintaxe do comando de saída de dados é:

```
escreva (<lista-de-expressões>)
```

Os comandos de entrada e de saída de dados são a forma mais rudimentar de comunicação entre o usuário e a máquina. Em outros tempos, fornecíamos dados de entrada em cartões

perfurados e recebíamos respostas em papel. Hoje, teclado e monitor são os exemplos mais comuns de dispositivos de entrada e saída (E/S).

Podemos exibir três tipos de saída: um texto (uma cadeia entre aspas simples), o conteúdo de uma variável (referenciado pelo seu nome), ou o resultado de uma expressão. Para exibir mais de uma saída, separamos cada uma por vírgulas.

Exemplos de possíveis saídas de dados:

```
escreva (soma)    //Imprime o conteúdo da variável soma

escreva (“Fim do cálculo”)    //Exibe uma mensagem

escreva (“O saldo atual é ”, saldo)//Exibe mensagem e um conteúdo

escreva (a+b)        //Exibe resposta da expressão

escreva (7+3)        //Exibe resposta da expressão
```

Note que os nomes nas mensagens são mais flexíveis do que os nomes de variáveis. Podemos inclusive usar acentos e outros símbolos, desde que estejam entre aspas.

LEMBRANDO: OS PARÊNTESES SÃO OBRIGATÓRIOS.