

TIPOS DE DADOS E VARIÁVEIS

3.1 Tipos de dados

Os dados manipulados por um algoritmo podem possuir natureza distinta, isto é, podem ser números, letras, frases, etc. Dependendo da natureza de um dado, algumas operações podem ou não fazer sentido quando aplicadas a eles. Por exemplo, não faz sentido falar em somar duas letras. Para poder distinguir dados de naturezas distintas e saber quais operações podem ser realizadas com eles, os algoritmos utilizam o conceito de tipo de dados.

O **tipo de um dado** define o conjunto de valores ao qual o dado pertence, bem como o conjunto de todas as operações que podem atuar sobre qualquer valor daquele conjunto de valores. Por exemplo, como veremos mais adiante, a linguagem que utilizaremos para descrever nossos algoritmos possui o tipo de dado inteiro, que consiste no conjunto de todos os números inteiros, denotado por \mathbb{Z} , e todas as operações que podem ser aplicadas aos números inteiros (isto é, adição, subtração, multiplicação, divisão inteira e resto).

A seguir, descrevemos os tipos de dados oferecidos pela linguagem Portugol do VISUALG. Na nossa descrição, o nome de um tipo é escrito no formato tipo, assim como as demais *palavras reservadas* da linguagem Portugol. Além disso, ao definirmos um dado tipo de dados, não fornecemos uma descrição detalhada das operações que atuam sobre seus valores, pois tais operações serão objetos de estudo das próximas aulas.

- inteiro: consiste dos números inteiros e das operações de adição, subtração, multiplicação, divisão inteira e resto. Na linguagem Portugol, os números inteiros são escritos apenas como a concatenação dos dígitos 0, 1, 2, 3, 4, 5, 6, 7, 8 e 9, tal como em 5, 100 e 1678. Números negativos são representados com o sinal “-” na frente do número, tal como -23.
- real: consiste dos números reais e das operações de adição, subtração, multiplicação, divisão. Na linguagem Portugol, os números reais são caracterizados por possuírem uma *parte inteira* e uma *parte fracionária*. Por exemplo, as partes inteira e fracionária do número real 3.141596 são 3 e 141596, respectivamente. Note que um “ponto” e não uma vírgula é usado para separar as partes inteira e fracionária.

Como sabemos, os números reais incluem os números inteiros. No entanto, para evitar ambigüidades na escrita de algoritmos, assumimos que todo número escrito sem a parte fracionária é do tipo inteiro. Por exemplo, 5 e 5.0 se referem ao mesmo número (cinco), mas o primeiro é do tipo inteiro e o segundo, do tipo real. Assim como os números inteiros negativos, números reais negativos são representados com o sinal “-” na frente do número, tal como 3.141596.

caractere: consiste de um único símbolo ou de uma concatenação de símbolos do alfabeto usado pela linguagem Portugol. Este alfabeto inclui todas as letras do alfabeto romano, todos os dígitos, 0, 1, ..., 9 e os caracteres de pontuação, tais como ?, ., ..., entre muitos outros símbolos. Os elementos do conjunto de valores do tipo caractere devem ser escritos, nos algoritmos, entre aspas duplas, como, por exemplo, "a", "Esta e uma frase formada por caracteres". Há um elemento especial, "", que é denominado de **palavra vazia**, pois não possui nenhum símbolo.

- logico: inclui apenas os valores lógicos falso e verdadeiro e as operações de negação, conjunção e disjunção. Nós estudaremos este tipo em maiores detalhes mais adiante.

3.2 Variáveis

Um algoritmo manipula dados, que podem ser dados variáveis ou constantes. Dados variáveis são representados por *variáveis*, enquanto dados constantes são representados por *constantes*¹.

Uma **variável** pode ser imaginada como um “caixa” para armazenar valores de dados. Esta caixa só pode armazenar um único valor por vez. No entanto, o valor armazenado na caixa pode mudar inúmeras vezes durante a execução do algoritmo. Em um ambiente computacional de verdade, a caixa correspondente a uma variável é uma posição da memória do computador.

Uma variável possui nome, tipo e conteúdo. O **nome de uma variável** deve ser único, isto é, identificar, de forma única, a variável no algoritmo. O **tipo de uma variável** define os valores que podem ser armazenados na variável. O **conteúdo de uma variável** é o valor que ela armazena. É importante lembrar que uma variável só pode armazenar um valor de cada vez. No entanto, ela pode assumir vários valores distintos do mesmo tipo durante a execução do algoritmo.

O ato de se criar uma variável é conhecido como **declaração de variável**.

Na linguagem Portugol, declaramos uma variável usando uma sentença da seguinte forma:

var nome : tipo

onde *nome* é o nome da variável e *tipo* é o tipo da variável.

Por exemplo, a sentença

var lado : real

declara uma variável de nome *lado* do tipo real.

Podemos declarar mais de uma variável do mesmo tipo em uma mesma linha. Por exemplo,

var lado, area : real

¹Não discutiremos constantes neste momento.

Note que nenhum conteúdo (isto é, valor) foi associado à variável durante a sua declaração. Esta associação é denominada **definição** e deve ser realizada após a declaração da variável usando uma **instrução de leitura** ou um **comando de atribuição**. Vamos detalhar essas duas formas.

A instrução de leitura tem a forma

leia (*nome*)

onde *nome* é o nome de uma variável. Por exemplo,

leia (lado)

é uma instrução de leitura que atribui um valor à variável *lado*. O valor atribuído pela instrução deve ser fornecido como entrada para o algoritmo durante a sua execução. Para você ter uma idéia mais concreta do que estamos falando, demonstraremos, em *sala de aula*, a execução da instrução de leitura do comprimento dos lados de um quadrado que escrevemos para o Algoritmo ??.

A instrução de atribuição possui a forma

nome <- *valor*

onde *nome* é o nome de uma variável e *valor* é um valor do mesmo tipo de dados da variável. Por exemplo,

lado <- 2.5

atribui o valor 2.5 à variável de nome *lado*. Para que uma instrução de atribuição faça sentido, a variável *lado* deve ser do tipo real e deve ter sido declarada antes da instrução de atribuição ser executada.

O símbolo <- é conhecido como **operador de atribuição**.

Muitas vezes, o valor atribuído a uma variável através de uma instrução de atribuição é resultante de uma *expressão aritmética* ou outro tipo de expressão que estudaremos mais adiante. Por exemplo,

area <- *lado* * *lado*

é uma instrução de atribuição que atribui o valor da variável *lado* ao quadrado à variável *area*. O que vemos no lado direito do operador de atribuição, *lado* * *lado*, é um exemplo de expressão aritmética.

Um valor atribuído a uma variável permanece associado a ela até que uma instrução de atribuição, que o substitua por outro, seja executada. Em qualquer dado instante de tempo durante a execução de um algoritmo, o valor armazenado em uma qualquer variável (se algum) é denominado **valor atual** (ou **valor corrente**) da variável. Enquanto não atribuirmos um valor a uma variável, a variável permanecerá com **valor desconhecido**. Finalmente, é importante lembrar que uma variável só poderá receber um valor através de uma instrução de leitura ou atribuição.

3.3 Exemplos

Seguem abaixo alguns exemplos de declaração de variáveis:

```
var fruta : caractere  
var letra : caractere  
var resultado : logico  
var altura : real  
var idade : inteiro
```

A seguir, temos exemplos de instruções de atribuição que atribuem valores a essas variáveis:

```
fruta <- "maçã"  
letra <- "c"  
resultado <- falso  
altura <- 1.83  
idade <- 5
```

As mesmas variáveis podem ter valores atribuídos a elas através de instruções de leitura como segue:

```
leia ( fruta )  
leia ( letra )  
leia ( altura )  
leia ( idade )
```

Note que não escrevemos uma instrução de leitura para a variável *resultado*. Isto se deve ao fato de instruções de leitura não poderem ser aplicadas a variáveis do tipo logico.

3.4 Nomes de variáveis

Na linguagem Portugol, usamos as seguintes regras para criar um nome de variável:

1. Nomes de variáveis devem possuir como primeiro caractere uma letra ou o símbolo ''' (sublinhado). Os demais caracteres, se algum, devem ser letras, números ou sublinhado.
2. Nomes de variáveis não podem ser iguais a palavras reservadas.

3. Nomes de variáveis podem ter, no máximo, 127 caracteres.
4. Não há diferença entre letras maiúsculas e minúsculas em nomes de variáveis.

De acordo com a regra 1, nomes de variáveis não podem conter espaços em branco. De acordo com a regra 2, nomes de variáveis não podem ser palavras reservadas da linguagem Portugol. Uma **palavra reservada** é uma palavra que possui um significado especial para a linguagem Portugol. Em geral, uma palavra reservada identifica uma instrução. Neste texto, tais palavras aparecem sublinhadas. O conjunto de palavras reservadas do Portugol é mostrado na Tabela 3.1.

aleatorio	e	grauprad	passo
abs	eco	inicio	pausa
algoritmo	enquanto	int	pi
arcos	entao	interrompa	pos
arcsen	escolha	leia	procedimento
arctan	escreva	literal	quad
arquivo	exp	log	radpgrau
asc	faca	logico	raizq
ate	falso	logn	rand
caractere	fimalgoritmo	maiusc	randi
caso	fimenquanto	mensagem	repita
compr	fimescolha	minusc	se
copia	fimfuncao	nao	sen
cos	fimpara	numerico	senao
cotan	fimprocedimento	numpcarac	timer
cronometro	fimrepita	ou	tan
debug	fimse	outrocaso	var
declare	funcao	para	verdadeiro
			xou

Tabela 3.1: Palavras reservadas da linguagem Portugol.

Por exemplo,

`_12234`, `fruta` e `x123`

são nomes válidos para variáveis, mas

`maria bonita`, `pi`, `fru?ta` e `1xed`

não são. O nome *maria bonita* contém um espaço em branco. O nome *pi* é uma palavra reservada. O nome *fru?ta* contém um caractere que não é letra, número nem sublinhado, ?. O nome *1xed* inicia com um número. Com exceção de *pi*, que viola a regra 2, os demais nomes violam a regra 1.

3.5 Exercícios Resolvidos

1. Escreva a declaração de uma variável do tipo real de nome *x*.
solução:
`var x : real`
2. Escreva a declaração de uma variável do tipo caractere de nome *carro*.
solução:
`var carro : caractere`
3. Escreva a instrução de atribuição que atribui o valor 2.3 à variável do problema 1.
solução:
`x <- 2.3`
4. Escreva a instrução de atribuição que atribui o valor "corsa" à variável do problema 2.
solução:
`carro <- "corsa"`
5. Quais dos seguintes nomes são válidos como nomes de variáveis?
 - (a) xyz_2
 - (b) _
 - (c) ____
 - (d) x123
 - (e) 123y
 - (f) 1_2solução:
(a), (b), (c) e (d).

3.6 Exercícios propostos

1. Escreva a declaração de uma variável do tipo caractere de nome *rua*.
2. Escreva a instrução de atribuição que atribui o nome de sua rua à variável do problema 1.
3. Escreva a declaração de uma variável do tipo logico de nome *achou*.
4. Escreva a instrução de atribuição que atribui a palavra reservada verdadeiro à variável do problema 3.
5. Quais dos seguintes nomes são válidos como nomes de variáveis?
 - (a) meucarro
 - (b) salute!

- (c) x_y_1
(d) ____
(e) 34
6. Escreva o tipo de dado ideal para se representar cada uma das seguintes informações:
- (a) O nome de uma rua
(b) A data de nascimento de uma pessoa
(c) Se uma pessoa é diabética ou não
(d) O saldo de uma conta bancária
(e) O resultado de uma operação de raiz quadrada
7. Identifique o tipo de dados dos seguintes valores:
- (a) "9 de agosto de 1968"
(b) 1.3
(c) falso
(d) -31
(e) "?"