UNIP

UNIVERSIDADE PAULISTA

Paradigmas de Linguagens

Aula 2: Tipos de dados

Professora Sheila Cáceres

Tipos de dados

- Dados são a matéria prima da computação junto com os programas.
- LPs precisam manipular dados.
- LPS utilizam os conceitos de tipos de dados para permitir a representação de valores (dados) em programas.
- **Valor:** entidade que pode ser avaliada, armazenada, etc. Ex: 3 2,5 'a' ”Paulo” 026
- **Tipo de dado:** conjunto cujos valores exibem comportamento univeforme nas operações associadas com o tipo.

Tipos de Dados

- Exemplo
  - {true 25 b, azul} não corresponde a um tipo
  - {true, false} corresponde a um tipo
- Quando um valor (e/ou expressão) é de um determinado tipo = esse valor pertence ao conjunto de valores definido por aquele tipo.
- **Cardinalidade:** número de valores distintos que fazem parte do tipo.
  - Ex: cardinalidade de boolean = 2.

Tipos de Dados

- Os tipos de dados podem se classificar em dois grupos:
  - Tipos de Dados Primitivos
  - Tipos de Dados Compostos

Tipos Primitivos

Tipos Primitivos

- São aqueles cujos valores não podem ser decompostos em outros valores de tipos mais simples.
- São a base de todo sistema de tipos de uma linguagem, pois a partir deles são construídos os demais.
- Os tipos primitivos são.
  - Tipos numéricos (inclui inteiro, ponto flutuante, decimal)
  - Tipos booleanos
  - Tipos caractere

Tipos Numéricos: Inteiro

- É o tipo numérico mais comum.
- Intervalo do conjunto dos números inteiros.
- Atualmente muitas LPs suportam diversos tamanhos de inteiros.
- Exemplo: Inteiros na linguagem C em uma máquina de 16 bits

Int	16	-32.768	32.767
unsigned int	16	0	65.535
signed int	16	-32.768	32.767
short int	16	-32.768	32.767
unsigned short int	16	0	65.535
signed short int	16	-32.768	32.767

Tipos Numéricos: Inteiro

- Inteiros em Java

Tipo Primitivo	Faixa de Valores	Quantidade de Bits
byte	-128 até 127	8 bits
short	-32 768 até 32 767	16 bits
int	-2,147,483,648 até 2,147,483,647	32 bits
long	9,223,372,036,854,775,808 até 9,223,372,036,854,775,807	64 bits

Tipos Numéricos: Ponto Flutuante

- Modela os números Reais.
- Como esta representação é finita, os números como Pi, somente podem ser aproximados (arredondado).
- A maioria das LPs inclui dois tipos de ponto flutuante frequentemente chamados **float** e **double** (+presição).
- Internamente combina representações binárias de frações e expoentes, como se fosse uma notação científica.
- Ex em java:

`float`	32-bit IEEE 754 ponto flutuante	32 bits
`double`	64-bit IEEE 754 ponto flutuante	64 bits

Tipos Numéricos: Decimal

- Este tipo primitivo armazena um número fixo de dígitos decimais.
- Cardinalidade: em função do número total de dígitos inteiros e fracionários n = 2* 10^n
- É um tipo de ponto fixo, e não flutuante.
- Sabe-se exatamente o número n de dígitos (parte decimal fixa).
- Essa representações são chamadas de binary coded decimal.

Tipo Booleano

- É o mais simples de todos
- Possui dois valores: verdadeiro e falso (**true** - **false**).
- Tem sido incluídos na maioria de LPs (exceto C)
- Pode usar só um bit, porém tipicamente usam um byte pois a menor célula de memoria disponível num computador é um byte.
- Cardinalidade = 2.

Tipo Caractere

- Os dados de caracteres são armazenados nos computadores como codificações numéricas.
- A codificação mais comum é ASCII que permite codificar 128 diferentes caracteres.

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040		Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	(	(	72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051	)	)	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[	[	123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174			|
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135	]	]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177	&#127;	DEL

Source: www.LookupTables.com

Tipos Compostos

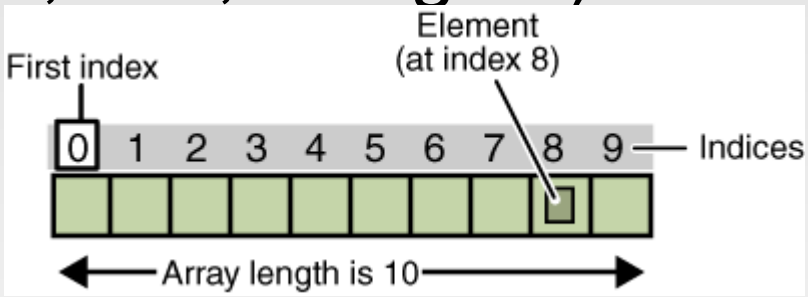
São construídos a partir dos tipos primitivos

Tipo String de Caracteres

- É aquele cujos valores correspondem a sequências de caracteres.
- Algumas LPs as tratam como tipos primitivos, outras como novos tipos, outras como uma lista, etc.
- Operações comuns são atribuição e comparação, concatenação e seleção.

Array

- É um tipo de dado estruturado unidimensional que consiste de um número fixo de elementos, sendo que todos os elementos devem ser do mesmo tipo (char, integer, real, string ....)



- Definição na linguagem Pascal:

```
nome_do_array: ARRAY [inicio..fim] of tipo;
```

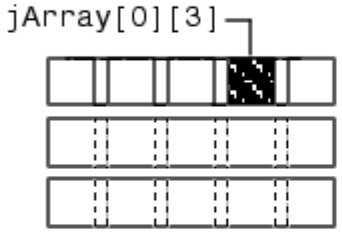
- Em Java

```
int[] myIntArray = {1,2,3};
```

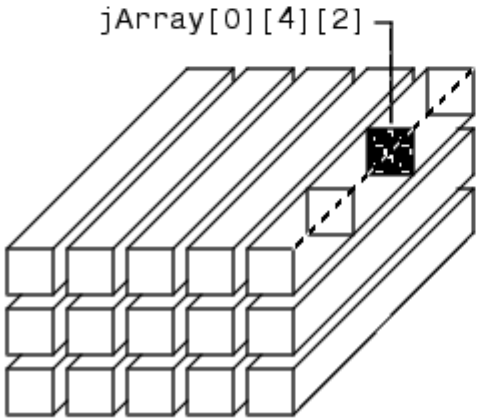
Outros tipos de arrays

Array Access from Java

Simple Array

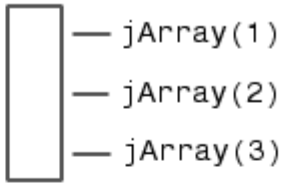


Array of Arrays

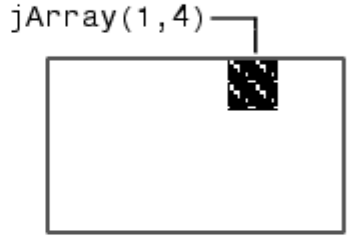


Array of Arrays of Arrays

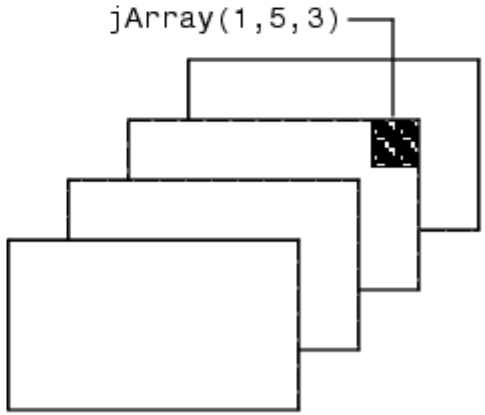
Array Access from MATLAB



One-dimensional Array



Two-Dimensional Array



Three-Dimensional Array

Registro

- É uma estrutura de dados composta e heterogênea, que permite armazenar valores, onde esses valores podem ser de diferentes tipos.
- Usualmente usa-se struct em C ou classes em c++ e java (linguagens orientadas a objetos)

Um registro

nome: João da Silva
e_mail: joao@gmail.com
telefone: 3344-5566
comissao: 150.00

Um array de registros

Registro de vendedores	
vendedor[0]	**nome**: João da Silva **e_mail**: joao@gmail.com **telefone**: 3344-5566 **comissao**: 150.00
vendedor[1]	**nome**: Ana Lúcia **e_mail**: analucia@gmail.com **telefone**: 3232-6868 **comissao**: 165.00
vendedor[2]	**nome**: Robert **e_mail**: robert@gmail.com **telefone**: 3838-9090 **comissao**: 120.00

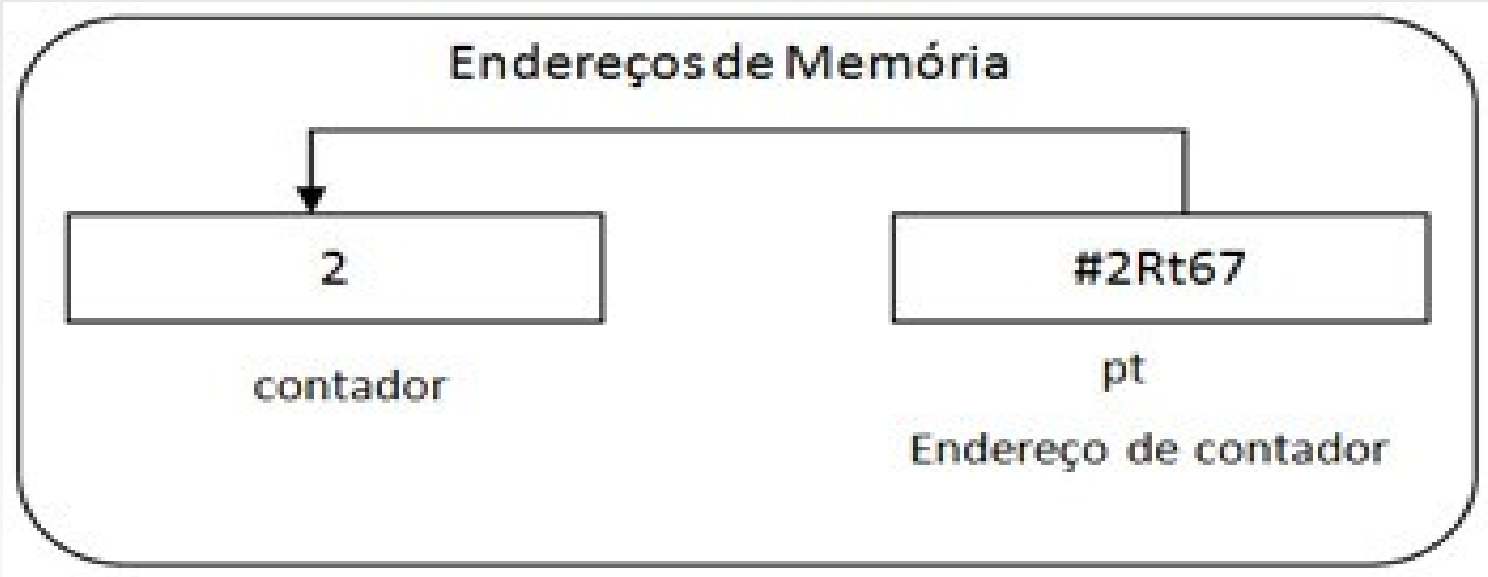
Ponteiro

- Um ponteiro é um valor que representa uma referência ou um endereço de memória.
- São usados comumente em C, C++, Ada e Perl.
- C fornece duas operações com ponteiros:
  - Operador Unário ”Endereço de ” (&): Recebe uma variável como argumento e retorna o endereço dessa variável
  - Operador Unário de ”Desreferenciação” (*): Recebe uma referência e produz o valor dessa referencia

Ponteiro

- Exemplo:

```
int contador = 2;
int *pt;
pt = &contador;
```



Referências Bibliográficas

* Sebesta, Linguagens de programação

* Tucker, Linguagens de programação – Princípios e Paradigmas

* VAREJÃO, Flavio. Linguagens de Programação. Campus, 2004.

* http://www.vidageek.net/2008/08/11/linguagens-de-programacao/