



# Java

极夜酱

# 目录

1	数组	1
1.1	一维数组 . . . . .	1
1.2	二维数组 . . . . .	4
1.3	字符 . . . . .	7

# Chapter 1 数组

## 1.1 一维数组

### 1.1.1 数组 (Array)

一个变量只能存储一个内容，如果需要存储更多数据，就需要使用数组解决问题。一个数组变量可以存放多个数据，数组是一个值的集合，它们共享同一个名字，数组中的每个变量都能被其下标所访问。

```
1 int[] number = new int[10];
2 float[] grade = new float[50];
```

a[0]	a[1]	a[2]	a[3]	a[4]
------	------	------	------	------

- 元素：数组中的每个变量
- 大小：数组的容量
- 下标 / 索引 (index)：元素的位置，下标从 0 开始，必须为非负整数

### 1.1.2 数组初始化

一维数组可以在声明时进行初始化：

```
1 int[] arr = {3, 6, 8, 2, 4, 0, 9, 7, 1, 5};
2 int[] arr = new int[] {3, 6, 8, 2, 4, 0, 9, 7, 1, 5};
```

很多时候在使用数组之前需要将数组的内容全部清空，这可以利用循环来实现。

#### 一维数组初始化

```
1 public class InitArr {
```

```

2     public static void main(String[] args) {
3         int[] arr = new int[100];
4         for(int i = 0; i < arr.length; i++) {
5             arr[i] = 0;
6         }
7     }
8 }

```

### 数组最大值和最小值

```

1 public class MaxMin {
2     public static void main(String[] args) {
3         int[] num = {7, 6, 2, 9, 3, 1, 4, 0, 5, 8};
4         int max = num[0];
5         int min = num[0];
6
7         for(int i = 1; i < num.length; i++) {
8             if(num[i] > max) {
9                 max = num[i];
10            } else if(num[i] < min) {
11                min = num[i];
12            }
13        }
14
15        System.out.println("max = " + max);
16        System.out.println("min = " + min);
17    }
18 }

```

### 运行结果

max = 9 min = 0

### 1.1.3 for-each

for-each 环是 for 循环的特殊简化版。

```
1 for(dataType var : set) {  
2     // code  
3 }
```

### 遍历数组

```
1 public class ForEach {  
2     public static void main(String[] args) {  
3         int[] arr = {7, 6, 2, 9, 3, 1, 4, 0, 5, 8};  
4         for(int elem : arr) {  
5             System.out.print(elem + " ");  
6         }  
7     }  
8 }
```

### 运行结果

7 6 2 9 3 1 4 0 5 8

## 1.2 二维数组

### 1.2.1 二维数组 (2D Array)

二维数组包括行和列两个维度，可以看成是由多个一维数组组成。

a[0][0]	a[0][1]	a[0][2]	a[0][3]
a[1][0]	a[1][1]	a[1][2]	a[1][3]
a[2][0]	a[2][1]	a[2][2]	a[2][3]

二维数组可以在声明时进行初始化：

```
1 int[][] arr = new int[2][3];
2 int[][] arr = {{1, 2, 3}, {4, 5, 6}};
```

#### 初始化二维数组

```
1 public class Init2dArr {
2     public static void main(String[] args) {
3         int[][] arr = new int[3][4];
4         for(int i = 0; i < arr.length; i++) {
5             for(int j = 0; j < arr[i].length; j++) {
6                 arr[i][j] = 0;
7             }
8         }
9     }
10 }
```

#### 矩阵运算

矩阵的加法/减法是指两个矩阵把其相对应元素进行加减的运算。

矩阵加法：两个  $m \times n$  矩阵 A 和 B 的和，标记为  $A + B$ ，结果为一个  $m \times n$  的矩阵，其内的各元素为其相对应元素相加后的值。

矩阵减法：两个  $m \times n$  矩阵 A 和 B 的差，标记为  $A - B$ ，结果为一个  $m \times n$  的矩阵，其内的各元素为其相对应元素相减后的值。

$$\begin{bmatrix} 1 & 3 \\ 1 & 0 \\ 1 & 2 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 7 & 5 \\ 2 & 1 \end{bmatrix} = \begin{bmatrix} 1+0 & 3+0 \\ 1+7 & 0+5 \\ 1+2 & 2+1 \end{bmatrix} = \begin{bmatrix} 1 & 3 \\ 8 & 5 \\ 3 & 3 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 3 \\ 1 & 0 \\ 1 & 2 \end{bmatrix} - \begin{bmatrix} 0 & 0 \\ 7 & 5 \\ 2 & 1 \end{bmatrix} = \begin{bmatrix} 1-0 & 3-0 \\ 1-7 & 0-5 \\ 1-2 & 2-1 \end{bmatrix} = \begin{bmatrix} 1 & 3 \\ -6 & -5 \\ -1 & 1 \end{bmatrix}$$

```
1 public class Matrix {
2     public static void main(String[] args) {
3         int[][] A = {
4             {1, 3},
5             {1, 0},
6             {1, 2}
7         };
8         int[][] B = {
9             {0, 0},
10            {7, 5},
11            {2, 1}
12        };
13        int[][] C = new int[3][2];
14
15        System.out.println("矩阵加法");
16        for(int i = 0; i < 3; i++) {
17            for(int j = 0; j < 2; j++) {
18                C[i][j] = A[i][j] + B[i][j];
19                System.out.print(String.format("%3d", C[i][j]));
20            }
21            System.out.println();
22        }
23    }
```

```
24     System.out.println("矩阵减法");
25     for(int i = 0; i < 3; i++) {
26         for(int j = 0; j < 2; j++) {
27             C[i][j] = A[i][j] - B[i][j];
28             System.out.print(String.format("%3d", C[i][j]));
29         }
30         System.out.println();
31     }
32 }
33 }
```

### 运行结果

矩阵加法

1 3

8 5

3 3

矩阵减法

1 3

-6 -5

-1 1



## 1.3 字符

### 1.3.1 字符 (Character)

单个的字符是一种特殊的类型，是用单引号表示字符字面量。每一个字符都有其对应的码值。

ASCII 全称 American Standard Code for Information Interchange (美国信息交换标准代码)，一共定义了 128 个字符。

ASCII	字符	ASCII	字符	ASCII	字符	ASCII	字符
0	NUL	32	(space)	64	@	96	`
1	SOH	33	!	65	A	97	a
2	STX	34	"	66	B	98	b
3	ETX	35	#	67	C	99	c
4	EOT	36	\$	68	D	100	d
5	ENQ	37	%	69	E	101	e
6	ACK	38	&	70	F	102	f
7	BEL	39	'	71	G	103	g
8	BS	40	(	72	H	104	h
9	HT	41	)	73	I	105	i
10	LF	42	*	74	J	106	j
11	VT	43	+	75	K	107	k
12	FF	44	,	76	L	108	l
13	CR	45	-	77	M	109	m
14	SO	46	.	78	N	110	n
15	SI	47	/	79	O	111	o
16	DLE	48	0	80	P	112	p
17	DC1	49	1	81	Q	113	q
18	DC2	50	2	82	R	114	r
19	DC3	51	3	83	S	115	s
20	DC4	52	4	84	T	116	t
21	NAK	53	5	85	U	117	u

22	SYN	54	6	86	V	118	v
23	TB	55	7	87	W	119	w
24	CAN	56	8	88	X	120	x
25	EM	57	9	89	Y	121	y
26	SUB	58	:	90	Z	122	z
27	ESC	59	;	91	[	123	{
28	FS	60	<	92	/	124	
29	GS	61	=	93	]	125	}
30	RS	62	>	94	^	126	~
31	US	63	?	95	_	127	DEL

表 1.1: ASCII 码表