

## 第2章 基本数据类型与数组

### 2.1 标识符与关键字

#### 1. 标识符

- (1) 可以使用字母、数字、美元号和下划线，但开头不能是数字。
- (2) 不要太长。
- (3) 不能与关键字（保留字）相同。
- (4) 不能取成 true, false 和 null。
- (5) 尽量按“见名知义”的原则命名，以增强程序的可读性。
- (6) 在一个程序中，应尽量避免出现几个标识符过于相似的情况。

例如，一个表示序号(serial number)的变量，可以取名为 serialno, serial\_no, serialNo (骆驼表示法) 或 iSerialNo (匈牙利表示法)。

#### 2. 关键字

p. 18

### 2.2 基本数据类型

#### 1. 逻辑型（布尔型，占1个字节，C无）

- (1) 常量  
true, false
- (2) 变量  
用 boolean 来声明。

#### 2. 整型

- (1) **基本整型**（占4个字节）

##### ①常量

##### 1) 十进制

0, 128, -128

##### 2) 八进制（以0开头）

0126, -0126

##### 3) 十六进制（以0x或0X开头）

0x1a, -0X1A

##### ②变量

用 int 来声明。

- (2) 字节整型（占1个字节，C无）

##### ①常量

无专门表示。

##### ②变量

用 byte 来声明。

- (3) 短整型（占2个字节）

##### ①常量

无专门表示。

## ②变量

用 short 来声明。

(4) 长整型 (占 8 个字节)

## ①常量 (以 **L** 或 **l** 结尾)

1) 十进制

128**L**, -128**L**

2) 八进制 (以 0 开头)

0126**L**, -0126**L**

3) 十六进制 (以 0x 或 0X 开头)

0x1a**L**, -0X1A**L**

## ②变量

用 long 来声明。

(5) 不能用 unsigned 来修饰整型 (C 能)。

(6) 要注意整型数据的溢出问题。

## 3. 字符型 (占 2 个字节)

(1) 常量

### ①可显示字符

'a'

### ②控制字符和特定功能字符

'\n', '\'

(2) 变量

用 char 来声明。

(3) 不能用 unsigned 来修饰字符型 (C 能)。

(4) 字符串常量

"This is a book."

## 4. 浮点型 (即实型)

(1) 单精度型 (占 4 个字节)

### ①常量 (以 f 或 F 结尾)

1) 十进制带小数形式 (定点数形式)

1.28f, -1.28F

2) 指数形式 (科学计数形式)

1.28e4f, -1.28E-4F

### ②变量

用 float 来声明。

(2) 双精度型 (占 8 个字节)

### ①常量 (可以以 d 或 D 结尾)

1) 十进制带小数形式 (即定点数形式)

1.28, 1.28d, -1.28D

2) 指数形式 (即科学计数形式)

1.28e4, 1.28e4d, -1.28E-4D

### ②变量

用 double 来声明。

### (3) 实型数据的舍入误差

不要直接进行两个实型数据的相等比较，可用它们的差的绝对值小于比如  $10^{-6}$  作为相等的判断条件。

## 2.3 类型转换运算

### 1. 隐式（自动）

p. 21 按精度从低到高排列的各种类型

各类数值数据间的混合运算是朝着数据表达能力更强的方向转换的。

### 2. 显式（强制）

（类型）常量或变量

## 2.4 输入、输出数据

### 1. 输入

(1) 先用 Scanner 类创建对象，再用 nextBoolean() 等方法输入数据。

(2) 输入前应给出提示信息。

### 2. 输出

(1) 用 println() 或 print() 方法输出数据。

(2) 可用并置符号 “+” 将若干表达式和字符串连接起来。

(3) 对于 **程序** 中较长的字符串，中间不能换行，可将该字符串拆成几个，之间用 “+” 连接，“+” 的前后能换行。

(4) 可用格式控制符：%d，%c，%f，%s。

## 2.5 数组

1. 之前用到的变量，不同的变量取不同的名字，名字不能反映变量之间的联系。

2. 同名的下标变量组成一个数组，下标变量又称数组元素。

### 3. 声明数组（与 C 有差异）

(1) 一维数组

① 元素类型 数组名 [], ...;

每个数组名后都要加 “[]”。

② 元素类型 [] 数组名, ...;

只要一个 “[]”。

(2) 二维数组

① 元素类型 数组名 [][] , ...;

②元素类型[][] 数组名, ...;

#### 4. 为数组分配元素空间

##### (1) 一维数组

数组名 = new 元素类型[个数];

##### (2) 二维数组

数组名 = new 元素类型[行数][列数];

#### 5. 声明数组和为数组分配元素空间可以合起来写。

#### 6. 数组元素的使用

##### (1) 一维数组

数组名[算术表达式]

算术表达式的值为下标, 取 0 至 (个数-1) 间的整数。

##### (2) 二维数组

数组名[算术表达式][算术表达式]

第一个算术表达式的值为行下标, 取 0 至 (行数-1) 间的整数; 第二个算术表达式的值为列下标, 取 0 至 (列数-1) 间的整数。

#### 7. length 的使用

“数组名.length”的值为该数组第一维的元素个数。

#### 8. 数组的初始化

(1) 为数组分配元素空间后, 系统给每个元素一个默认值。

(2) 若声明数组时主动给每个元素一个初始值, 则无需用 new。

允许使用不规则的二维数组。

#### 9. 数组名

数组名是引用型变量, 存储该数组的起始引用 (地址)。

## 2.6 应用举例

二分查找 (折半查找、对半查找) 算法 1946 年就写出来了, 但直到 1962 年才总算写对。

例 1. 在 010, 020, 030, 040, 050, 060, 070, 080, 090, 100, 110, 120, 130, 140, 150, 160 中查找 110。

010	020	030	040	050	060	070	080	090	100	110	120	130	140	150	160
↑						↑									↑
start=1						middle=8									end=16
						↑		↑							↑
						start=9		middle=12							end=16
						↑	↑	↑							
						start=9	middle=10	end=11							

↑  
start=middle=end=11

查找成功

例 2. 在 010, 020, 030, 040, 050, 060, 070, 080, 090, 100, 110, 120, 130, 140, 150, 160 中查找 55。

010	020	030	040	050	060	070	080	090	100	110	120	130	140	150	160
↑							↑							↑	
start=1							middle=8							end=16	
↑			↑			↑									
start=1		middle=4				end=7									
		↑		↑		↑									
		start=5		middle=6		end=7									
		↑													
		start=middle=end=5													
		↑		↑											
		end=5		start=6											

查找失败