

算法第四版 谢路云

第一章 基础

java基本结构

- 原始数据类型
 - 整数、浮点、字符、布尔
 - 布尔型bool(boolean)为两种取值 true false,在bool数据里, 0 为 false, 否则则为 true(除0之外的任何数)
 - false 看作0, true看作 1, 二者逻辑运算, 按0 1 走
 - 逻辑运算, ! >&&>||
 - int-32 bit
 - double-64 bit IEEE 754
 - char-16 bit (解释: int 32位, double IEEE 754 规则规定64位, char 16位)
- 语句
 - 声明 给定变量名数据类型 (int a, double b, 取整型数a、浮点型b)
 - 赋值 给变量赋值 - java中给数组定义空间, 使用new 语句 - 用法为 int[] a=new int[n];
 - 条件 简单改变执行流程
 - 循环 使系统不断执行某些语句 - break,continue - if,while 同c
 - 调用和返回
- 数组
 - java数组与c不同
 - java定义数组, 数组空间在数组名前
 -

完整模式

```
double[] a;           ← 声明数组
a = new double[N];    ← 创建数组
for (int i = 0; i < N; i++)
    a[i] = 0.0;       ← 初始化数组
```

简化写法

```
double[] a = new double[N];
```

声明初始化

```
int[] a = { 1, 1, 2, 3, 5, 8 };
```

声明、创建并初始化一个数组

- Java允许两个数组变量名指向同一位置

- 1.1.5.4 起别名

请注意，数组名表示的是整个数组——如果我们将一个数组变量赋予另一个变量，那么两个变量将会指向同一个数组。例如以下这段代码：

```
int[] a = new int[N];
...
a[i] = 1234;
...
int[] b = a;
...
b[i] = 5678; // a[i] 的值也会变成 5678
```

这种情况叫做起别名，有时可能会导致难以察觉的问题。如果你想将数组复制一份，那么应该声明、创建并初始化一个新的数组，然后将原数组中的元素值挨个复制到新数组，如表 1.1.4 的第三个例子所示。

- java初始化数组，如不赋值，则默认初始化为0，布尔数组为false
-

```
double[][] a = new double[M][N];
```

我们将这样的数组称为 $M \times N$ 的数组。我们约定，第一维是行数，第二维是列数。和一维数组一样，Java 会将数值类型的数组元素初始化为 0，将布尔型的数组元素初始化为 `false`。默认的初始化对二维数组更有用，因为可以节约更多的代码。下面这段代码和刚才只用一行就完成创建和初始化的语句是等价的：

```
double[][] a;
a = new double[M][N];
for (int i = 0; i < M; i++)
    for (int j = 0; j < N; j++)
        a[i][j] = 0.0;
```

在将二维数组初始化为 0 时这段代码是多余的，但是如果想要初始化为其他值，我们就需要嵌套的 `for` 循环了。

• 静态方法

- 静态方法是指处理好的函数程序
- 调用方式同 `c`
- 一个方法可以存在多个返回语句，但至多有一个返回值，可以返回 `void`（无返回值）
- 小记外部库,数学函数 `math`, `integer`、`double` 转化字符串为整形、浮点型数据
 - 系统标准库 `java.lang.*`：这其中包括 `Math` 库，实现了常用的数学函数；`Integer` 和 `Double` 库，能够将字符串转化为 `int` 和 `double` 值；`String` 和 `StringBuilder` 库，我们稍后会在本节和第 5 章中详细讨论；以及其他一些我们没有用到的库。
 - 导入的系统库，例如 `java.util.Arrays`：每个标准的 Java 版本中都含有上千个这种类型的库，不过本书中我们用到的并不多。要在程序的开头使用 `import` 语句导入才能使用这些库（我们也是这样做的）。

头文件备注 `public class + housename`，可调用库；调用函数 `static void equation()`

表 1.1.7 Java 的 `Arrays` 库节选（`java.util.Arrays`）

<code>public class Arrays</code>	
<code>static void sort(int[] a)</code>	将数组按升序排序

注：其他原始类型和 `Object` 对象也有对应版本的方法。

public class Math	
static double abs(double a)	a 的绝对值
static double max(double a, double b)	a 和 b 中的较大者
static double min(double a, double b)	a 和 b 中的较小者
注 1: abs()、max() 和 min() 也定义了 int、long 和 float 的版本。	
static double sin(double theta)	正弦函数
static double cos(double theta)	余弦函数
static double tan(double theta)	正切函数
注 2: 角用弧度表示, 可以使用 toDegrees() 和 toRadians() 转换角度和弧度。	
注 3: 它们的反函数分别为 asin()、acos() 和 atan()。	
static double exp(double a)	指数函数 (e^a)
static double log(double a)	自然对数函数 ($\log_e a$, 即 $\ln a$)
static double pow(double a, double b)	求 a 的 b 次方 (a^b)
static double random()	[0, 1) 之间的随机数
static double sqrt(double a)	a 的平方根
static double E	常数 e (常数)
static double PI	常数 π (常数)

- 字符串
- 标准输入输出
- 数据抽象
 - 数据抽象是指, 可以面向对象, 创建新的数据类型, 进而进行编程



模块化编程——API (应用程序编程接口)

一些常用的库

表 1.1.8 我们的随机数静态方法库的 API

public class StdRandom		
static	void initialize(long seed)	初始化
static	double random()	0 到 1 之间的实数
static	int uniform(int N)	0 到 N-1 之间的整数
static	int uniform(int lo, int hi)	lo 到 hi-1 之间的整数
static	double uniform(double lo, double hi)	lo 到 hi 之间的实数
static	boolean bernoulli(double p)	返回真的概率为 p
static	double gaussian()	正态分布, 期望值为 0, 标准差为 1
static	double gaussian(double m, double s)	正态分布, 期望值为 m, 标准差为 s
static	int discrete(double[] a)	返回 i 的概率为 a[i]
static	void shuffle(double[] a)	将数组 a 随机排序

注: 库中也包含为其他原始类型和 Object 对象重载的 shuffle() 函数。

表 1.1.9 我们的数据分析静态方法库的 API

public class StdStats		
static	double max(double[] a)	最大值
static	double min(double[] a)	最小值
static	double mean(double[] a)	平均值
static	double var(double[] a)	采样方差
static	double stddev(double[] a)	采样标准差
static	double median(double[] a)	中位数

表 1.1.10 StdRandom 库中的静态方法的实现

期望的结果	实 现
随机返回 [a,b) 之间的一个 double 值	<pre>public static double uniform(double a, double b) { return a + StdRandom.random() * (b-a); }</pre>
随机返回 [0..N) 之间的一个 int 值	<pre>public static int uniform(int N) { return (int) (StdRandom.random() * N); }</pre>
随机返回 [lo,hi) 之间的一个 int 值	<pre>public static int uniform(int lo, int hi) { return lo + StdRandom.uniform(hi - lo); }</pre>
根据离散概率随机返回的 int 值（出现 i 的概率为 a[i]）	<pre>public static int discrete(double[] a) { // a[] 中各元素之和必须等于 1 double r = StdRandom.random(); double sum = 0.0; for (int i = 0; i < a.length; i++) { sum = sum + a[i]; if (sum >= r) return i; } return -1; }</pre>
随机将 double 数组中的元素排序（请见练习 1.1.36）	<pre>public static void shuffle(double[] a) { int N = a.length; for (int i = 0; i < N; i++) { // 将 a[i] 和 a[i..N-1] 中任意一个元素交换 int r = i + StdRandom.uniform(N-i); double temp = a[i]; a[i] = a[r]; a[r] = temp; } }</pre>

数据转换

- parseInt(string s) 将string转化为int数据
- toString(int a/double x)将int/double转化为string数据
- parseDouble(string s)将string转化为double数据

而类 Integer 和 Double 库中也提供了为类 String 类型相互转化的静态方法（见表 1.1.12）。

表 1.1.12 String 值和数字之间相互转换的 API

public class Integer		
static	int parseInt(String s)	将字符串 s 转换为整数
static	String toString(int i)	将整数 i 转换为字符串
public class Double		
static	double parseDouble(String s)	将字符串 s 转换为浮点数
static	String toString(double x)	将浮点数 x 转换为字符串