JAVA程序设计

中国科技出版传媒股份有限公司
China Science Publishing & Media Ltd.（CSPM）
科 学 出 版 社

目 录 CONTENTS

1 学习指南

中国科技出版传媒股份有限公司
China Science Publishing & Media Ltd.（CSPM）
科学出版社

学习指南

本章主要通过详尽的实例，配以合理的练习，首先介绍了数组的基本概念，以一维数组和二维数组为例加深读者理解数组的基本操作，然后介绍了一种重要的数据类型即字符串，通过两种具体的字符串类型String和StringBuffer加深对字符串基本结构和基本操作的理解，最后介绍了java类库中常用的几种类型。

中国科技出版传媒股份有限公司
China Science Publishing & Media Ltd.（CSPM）
科 学 出 版 社

2 难点重点

中国科技出版传媒股份有限公司
China Science Publishing & Media Ltd.（CSPM）
科学出版社

难点重点

- 数组的概念
- 二维数组和多维数组
- 字符和字符串的基础知识
- String类
- 使用StringBuffer类
- 正则表达式
- Java类库
- 基本数据类
- 实用工具类

3 知识内容

中国科技出版传媒股份有限公司
China Science Publishing & Media Ltd.（CSPM）
科学出版社

1. 数组的概念

数组是若干个具有相同数据类型的数据元素的集合。引用这些元素时可用同一个名字，不同的下标来指明。表示数组元素的下标个数称为该数组的维数。只有一个下标的数组称为一维数组，有两个下标的数组称为二维数组，依此类推。

中国科技出版传媒股份有限公司
China Science Publishing & Media Ltd.（CSPM）
科 学 出 版 社

1. 数组的概念

1.1 一维数组的声明

一维数组的定义定义的格式为：

〈数据类型〉 〈数组名〉[ ];

或

〈数据类型〉[ ] 〈数组名〉;

1. 数组的概念

1.1 一维数组的声明

说明:

（1）〈数据类型〉可以为Java语言中任意的数据类型，〈数组名〉为合法的变量名，[ ]指明该变量是一个数组类型。每个数组的下限都是从0开始的,且不可改变。

（2）数组在被创建后，其元素被系统自动初始化了。字符元素被初始化为'\u0000'，而对于对象数组都被初始化为null。 如果不初始化的话，在内存是找不到它的位置的。

（3）数组中的第一元素记做第0个，i[0]是数组i 的第一个元素。

1. 数组的概念

1.1 一维数组的声明

例如：

float a[]; // 定义了一个数组a，其中的元素可以存放float型数据。

在Java中，数组是基于类的，所以在数组使用之前必须对它进行实例化。在Java中，在使用数组时可使用new运算符创建数组，当数组元素的数量发生变化时，可使用new再向它分配数组元素的数量。

1. 数组的概念

1.1 一维数组的声明

如下语句对已声明的rain数组创建一个有10个double类型数据组成的数组：

```
double []rain；  //声明一个double数组rain，其大小可以是任意的。
rain=new double[10];  //rain是10个元素的数组，数组下标从0到9。
rain=new double[20];  //rain是20个元素的数组。
```

1. 数组的概念

1.2 一维数组的初始化

在数组定义的同时可以给数组元素赋值，称为数组的初始化。格式如下：

数据类型　数组名 [ ] = {值 1，值 2，…值 N} ；

或

数据类型 [ ]　数组名 = {值 1，值 2，…值 N} ；

Java编译程序会自动根据值的个数算出整个数组的长度，并分配相应的空间。如果为基本数据类型，元素会初始化为0；如果为boolean类型，元素会初始化为false；如果为引用类型，则元素会初始化为空（null）。

1. 数组的概念

1.2 一维数组的初始化

例如：int[] array1 = { 2, 3, 5, 7, 11, 13, 17, 19 };

声明数组和创建数组可以一起完成：

```
float  boy[]=new float[4] { 1.2f,2.3f,15.2f,7.6f } ;
```

也可以

```
float  boy[ ]= { 1.2f,2.3f,15.2f,7.6f }
```

相当于

```
float  boy[]=new float[4];
boy[0]=1.2f; boy[1]=2.3f; boy[2]=15.2f; boy[3]=7.6f;
```

1. 数组的概念

1.3 元素的引用

定义一个数组并为它分配存储空间后，才可以使用数组中的元素。访问数组元素的表示形式如下：

数组名[下标]；

下标即数组索引，可以是整数或整形表达式。数组元素访问的结果是变量，即由下标选定的数组元素。Java数组从0开始建立索引，即数组索引从0开始。

1. 数组的概念

1.4 把数组传递给方法

如果类型为基本数据类型，Java 采用值传递的方式把实际参数传递给方法的形式参数，传递的是实际参数参的值，方法运行之后实参变量的值不改变。

中国科技出版传媒股份有限公司
China Science Publishing & Media Ltd.（CSPM）
科 学 出 版 社

1. 数组的概念

1.4 把数组传递给方法

[例5-1]数组采用引用（传地址）调用change方法对数组内容进行改变。

程序源码：

ChangeTest.java

1. 数组的概念

1.5 编程应用

[例5-2] 班里有30位学生，使用动态录入并赋值的方式计算平均分：

程序源码：

chap05_02.java

中国科技出版传媒股份有限公司
China Science Publishing & Media Ltd.（CSPM）
科 学 出 版 社

1. 数组的概念

1.5 编程应用

[例5-3] 对输入的数组元素进行排序：

程序源码：

chap05_03.java

中国科技出版传媒股份有限公司
China Science Publishing & Media Ltd.（CSPM）
科 学 出 版 社

1. 数组的概念

1.5 编程应用

[例5-4]使用起泡排序算法对一个有十个元素的数组进行排序：

程序源码：

BubbleSort.java

中国科技出版传媒股份有限公司
China Science Publishing & Media Ltd.（CSPM）
科 学 出 版 社

2. 二维数组和多维数组

一维数组由排列在一行中所有元素组成，它只有一个索引。

从概念上讲，二维数组就像一个具有行和列的表格一样。

中国科技出版传媒股份有限公司
China Science Publishing & Media Ltd.（CSPM）
科学出版社

2. 二维数组和多维数组

2.1 二维数组的定义和初始化

（1）二维数组的定义

二维数组定义如下：

〈数据类型〉〈数组名〉[ ][ ];

或

〈数据类型〉[ ][ ]〈数组名〉;

与一维数组一样，这时对数组元素也没有分配内存空间，同样要使用new关键字来分配内存，然后才可以访问元素。

2. 二维数组和多维数组

2.1 二维数组的定义和初始化

（1）二维数组的定义

对多维数组来说，分配内存空间时可以直接为每一维分配空间。

例如：

```
int a[ ][ ]=new int[2][3] ;
```

也可以从最高维开始，分别为每一维分配空间，例如：

```
int a[ ][ ]=new  int[2][ ];
a[0]=new  int[3];
a[1]=new  int[3];
```

2. 二维数组和多维数组

2.1 二维数组的定义和初始化

（2）二维数组的初始化

可以在声明数组时将其初始化，例如：

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

也可以初始化数组，但不指定级别：

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

2. 二维数组和多维数组

2.1 二维数组的定义和初始化

（2）二维数组的初始化

如果要声明一个数组变量但不将其初始化，则必须使用new运算符将数组分配给此变量。例如：

```
int  [ ][ ]arr；
arr=new int[][]{{1,2},{3,4},{5,6},{7,8}}; //正确
arr = {{1,2},{3,4},{5,6},{7,8}};    //错误
```

也可以给数组元素赋值，例如：

```
arr[2][1]=10；
```

2. 二维数组和多维数组

2.2 二维元素的引用

二维数组中每个元素，其引用方式为：数组名[下标1][下标2]；

例如以下代码按索引顺序读取二维数组r各元素的值并计算它们的和，将它们的和存于变量sum中：

```
double  sum =0;
for (int i=0;i<rain.length;i++){
                    for (int j=0;j<rain[i].length;j++){
                    sum+=rain[i][j];
                    }
}
```

2. 二维数组和多维数组

2.3编程应用

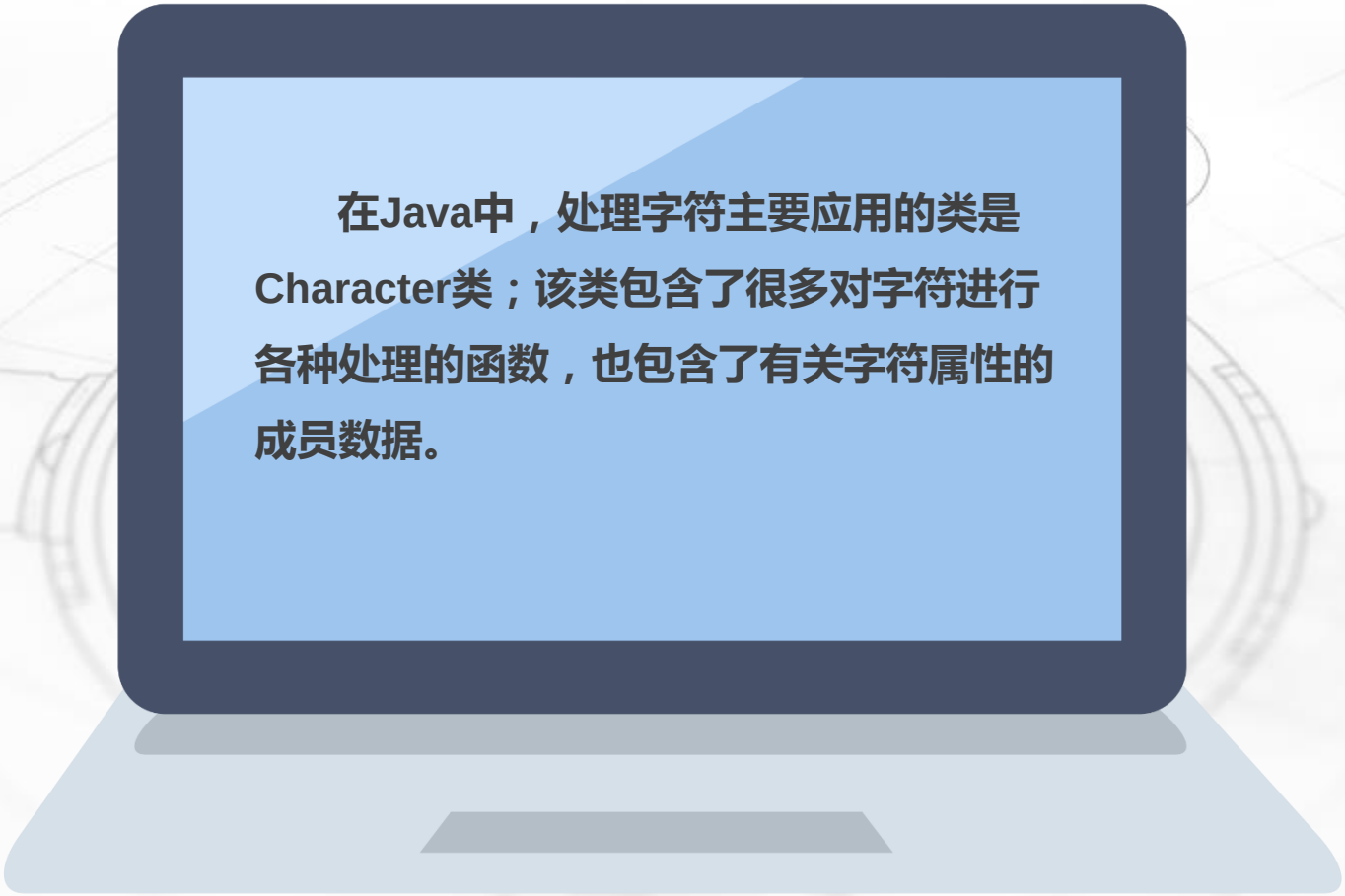
[例5-5] 以下实例演示了一维、二维数组的声明、创建、初始化。

程序源码：

Test.java

first ‹ 1 2 3 4 › last cancel ok

3. 字符和字符串的基础知识



3. 字符和字符串的基础知识

字符串是字符的序列，它是组织字符的基本数据结构，从某种程度上来说有些类似于字符的数组。在Java中，字符串被当作对象来处理。程序中需要用到的字符串可以分为两大类，一类是创建之后不会再做修改和变动的字符串常量String类；另一类是创建之后允许再做更改和变化的字符串变量StringBuffer类。

4. 创建对象

4.1 创建String对象

Java中将字符串作为String类型对象来处理。String构造函数 如下：

（1）String()， 默认构造函数，无参数；

```
String s1 = new String();
```

（2）String(char chars[])，传入字符数组 ；

（3）String(char chars[], int startIndex, int numChars) ，传入一个字符数组，从指定下标位置开始获取指定个数的字符，用这些字符来初始化字符串变量。

（4）字符串对象的内容初始化

（5）String(byte asciiChars[])

4. 创建对象

4.1 创建String对象

[例5-6]使用构造函数初始化字符串。

程序源码：

chap05_06.java

中国科技出版传媒股份有限公司
China Science Publishing & Media Ltd.（CSPM）
科 学 出 版 社

4. 创建对象

4.2 创建StringBuffer对象

StringBuffer定义了下面三个构造函数：

StringBuffer()　// 默认构造函数，预留了16个字符的空间，该空间不需再分配。

StringBuffer(int size)　// 设置指定缓冲区大小。

StringBuffer(String str) // 设置StringBuffer对象初始化的内容并预留16个字符空间，且不需再分配空间。

5. String类

5.1 求字符串长度

使用String类中的length（ ）方法可以获取一个字符串的长度，

[例5-7]使用length（ ）方法求字符串的长度。

程序源码：

chap05_07.java

5. String类

5.2 连接字符串

使用"+"运算符可以将两个字符串连接起来产生一个新的String对象。如果原来的字符串有空格，新的字符串会保留原来的空格。在Java类中还有一个用于连接字符串的方法：public String concat(String str)；这个方法是将参数str指定的字符串连接到当前字符串后面。返回的字符串的值，会保留其中的空格及相应的符号。

5. String类

5.2 连接字符串

[例5-8]"+"与concat()的运用

程序源码：

chap05_08.java

中国科技出版传媒股份有限公司
China Science Publishing & Media Ltd.（CSPM）
科学出版社

5. String类

5.3 比较字符串

使用字符串类中的equals(String str)可以判断两个字符串是否相等，这个方法返回一个boolean型的值，如果为true则说明两个字符串相等，如果为false说明两个字符串不相等。返回的boolean型的值经常会用做条件判断。

5. String类

5.3 比较字符串

[例5-9] equals()方法的运用

程序源码：

CompareString.java

中国科技出版传媒股份有限公司
China Science Publishing & Media Ltd.（CSPM）
科 学 出 版 社

5. String类

5.3 比较字符串

在Java中还有一种比较字符串大小的方法，equalsIgnoreCase(String another),这是忽略字符大小写的一种比较方法。例如：

[例5-10]equalsIgnoreCase ()的运用

程序源码：

chap05_10.java

first < 1 2 3 4 > last cancel ok

5. String类

5.4 搜索(截取)字符串

在Java的字符串类中提供两个方法搜索(截取)字符串：

String substring(int beginIndex) ：截取从beginIndex位置开始到结束的子字符串。

[例5-11] substring()的运用

程序源码：

chap05_11.java

String substring(int beginIndex, int endIndex) ：截取从beginIndex位置开始到endIndex位置的子字符串。

中国科技出版传媒股份有限公司
China Science Publishing & Media Ltd.（CSPM）
科学出版社

5. String类

5.5 搜索(截取)字符

从一个字符串对象中求取指定的字符，就要用到求取字符的方法：charAt(int index)。这个方法能也只能返回一个单一的字符，这与上面的求取子串是不一样的。其中，参数index是一个整数，它是指字符串序列中字符的位置。注意：这个整数是从0开始的。

5. String类

5.5 搜索(截取)字符

[例5-12] charAt()方法的运用

程序源码：

chap05_12.java

5. String类

5.6 修改字符串

修改字符串的目的是为了得到新的字符串，类String提供了如下几种方法用来修改字符串。

replace() 用另一个字符取代指定字符串中指定字符,具体形式：

public String replace(char oldchar,char newChar)

在字符串中用newChar字符替换oldChar字符 ， 例如：

String s="Hello" ;

replace('l','w'); // 执行后 s ="Hewwo";

6. 使用StringBuffer类

6.1 把字符串添加到缓冲区

一旦创建了StringBuffer类的对象，就可以使用StringBuffer类的大量方法和属性。最常用的是 append()方法，它把字符串添加到缓冲区，将任一其他类型数据的字符串形式连接到调用StringBuffer对象的后面，对所有内置的类型和Object，它都有重载形式。下面是它的几种重载形式：

- StringBuffer append(String str)
- StringBuffer append(int num)
- StringBuffer append(Object obj)；

6. 使用StringBuffer类

6.1 把字符串添加到缓冲区

[例5-13] append()方法的运用

程序源码：

chap05_13.java

6. 使用StringBuffer类

6.2 把字符串插入到缓冲区

insert()方法将一个字符串插入另一个字符串中。下面是它的几种形式：

- StringBuffer insert(int index,String str)；
- StringBuffer insert(int index,char ch)；
- StringBuffer insert(int index,Object obj)。

6. 使用StringBuffer类

6.3 从缓冲区中获取字符

要得到StringBuffer对象中指定位置上的字符，可以使用charAt()方法，它的一般形式为：char charAt(int where)；为字符串中的单个字符赋值或进行替换可以使用setCharAt()方法，它的一般形式如下：void setCharAt(int where,char ch) 。对于这两种方法，where值必须是非负的，同时不能超过或等于StringBuffer对象的长度。还有getChars(int suorceStart,int sourceEnd,char target[] , int targetStart) 方法，该方法将字符串拷贝到字符数组中。

6. 使用StringBuffer类

6.4 修改缓冲区中字符串

StringBuffer类提供了如下几种方法用以修改缓冲区中的字符串：

append( )： 将任一其他类型数据的字符串形式连接到调用StringBuffer对象的后面。

insert( )： 在字符串指定位置插入值。

setCharAt( )：为字符串中的单个字符赋值或进行替换。

reverse()：倒置StringBuffer的内容。

6. 使用StringBuffer类

6.4 修改缓冲区中字符串

StringBuffer类提供了如下几种方法用以修改缓冲区中的字符串：

StringBuffer delete(int startIndex,int endIndex)：删除指定位置的字符串。

StringBuffer deleteCharAt(int loc)： 删除指定位置的字符。

replace() ：它完成在StringBuffer内部用一个字符串代替另一个指定起始位置和结束位置的字符串的功能。

6. 使用StringBuffer类

6.5 String和StringBuffer的区别

Java 平台提供了两个类：String和StringBuffer，它们可以储存和操作字符串， String类提供了数值不可改变的字符串，而StringBuffer类提供的字符串可以进行修改。当字符数据要改变的时候可以使用StringBuffer。

7. 正则表达式

正则表达式是一种可以用于模式匹配和替换的规范，一个正则表达式就是由普通的字符（例如字符a到z）以及特殊字符（元字符）组成的文字模式，它 用以描述在查找文字主体时待匹配的一个或多个字符串。正则表达式作为一个模板，将某个字符模式与所搜索的字符串进行匹配。

7. 正则表达式

java.util.regex 包主要包括以下三个类：

- Pattern 类：Pattern 对象是一个正则表达式的编译表示。
- Matcher 类：Matcher 对象是对输入字符串进行解释和匹配操作的引擎。
- PatternSyntaxException：PatternSyntaxException 是一个非强制异常类，它表示一个正则表达式模式中的语法错误。

8. Java类库

类库就是Java API(Application Programming Interface，应用程序接口)，是系统提供的已实现的标准类的集合。在程序设计中，合理和充分利用类库提供的类和接口，不仅可以完成字符串处理、绘图、网络应用、数学计算等多方面的工作，而且有了Java类库，只需自己编写类继承系统提供的有关标准就可以解决相关的问题。这样可以大大提高编程效率，使程序简练、易懂。

8. Java类库

8.1 类库的使用

Java类库中的类和接口大多封装在特定的包里，每个包具有自己的功能。类库的使用方式有两种：

（1）在每个类名前添加完整的包名：java.until.Date today=new java.until.Date( );

（2）使用import语句。这个语句导入一个特定的类或整个包。例如，可以使用下面的语句导入java.until包中的所有类：import java.until.*；然后使用：Date today=new Date ( );无需在前面加上包前缀。或者在源代码前面加上这样的语句：import java.until.Date，这样在程序中也可以直接使用Date类。

中国科技出版传媒股份有限公司
China Science Publishing & Media Ltd.（CSPM）
科 学 出 版 社

8. Java类库

8.2 常用类库介绍

java.lang是Java语言最广泛使用的包。它所包括的类是其他包的基础，由系统自动引入，程序中不必用import语句就可以使用其中的任何一个类。java.lang中所包含的类和接口对所有实际的Java程序都是必要的。它提供利用 Java 编程语言进行程序设计的基础类。

中国科技出版传媒股份有限公司
China Science Publishing & Media Ltd.（CSPM）
科 学 出 版 社

8. Java类库

8.2 常用类库介绍

java.util是Java语言中另一个使用广泛的包，它包括集合类、时间处理模式、日期时间工具等各种常用工具。Java的集合类是java.util包中的重要内容，它允许以各种方式将元素分组，并定义了各种使这些元素更容易操作的方法。

JAVA的核心库java.io提供了全面的IO接口，包括：文件读写，标准设备输出等等。Java中IO是以流为基础进行输入输出的，所有数据被串行化写入输出流，或者从输入流读入。

中国科技出版传媒股份有限公司
China Science Publishing & Media Ltd.（CSPM）
科学出版社

9. 基本数据类

为了能将基本类型视为对象来处理,并能连接相关的方法，java为每个基本类型都提供了包装类。这些包装类提供的方法具有某些基本功能，比如：类型转换、值测试、相等性检查等。这些类在java.lang包中，分别是：Boolean，Byte，Short，Character，Integer，Long，Float等。

每个Java的原始数据类型都有一个相应的包装类。包装类只是用来封装一个不可变值的类。Integer类封装int值，Float类封装float值。包装类名称与相应的原始数据类型名称不完全匹配。

9. 基本数据类

9.1 Interger类

（1）属性

① static int MAX_VALUE：返回最大的整型数；

② static int MIN_VALUE：返回最小的整型数；

③ static Class TYPE ：返回当前类型。

例如：

```
System.out.println("Integer.MAX_VALUE: " +
Integer.MAX_VALUE );
```

结果为：Integer.MAX_VALUE: 2147483647

9. 基本数据类

9.1 Interger类

（2）构造方法

① Integer(int value) ：通过一个int的类型构造对象；

② Integer(String s) ：通过一个String的类型构造对象；

例如：

Integer i = new Integer("123");

生成了一个值为123的Integer对象。

9. 基本数据类

9.1 Interger类

Integer的常用方法如下：

byteValue()：取得用byte类型表示的整数；

int compareTo(Integer anotherInteger) ：比较两个整数。

相等时返回0；小于时返回负数；大于时返回正数。

double doubleValue() ：取得该整数的双精度表示。

int intValue() ： 返回该整型数所表示的整数。

9. 基本数据类

9.1 Interger类

Integer的常用方法如下：

long longValue() ：返回该整型数所表示的长整数。

static int parseInt(String s) ：将字符串转换成整数。s必须是时进制数组成，否则抛出NumberFormatException异常。

static int parseInt(String s, int radix) ：以radix为基数radix返回s的十进制数。所谓的基数，就是"几进制"。

9. 基本数据类

9.1 Interger类

Integer的常用方法如下：

short shortValue() ：返回该整型数所表示的短整数。

static String toBinaryString(int i) ：将整数转为二进制数的字符串。

static String toHexString(int i) ：将整数转为十六进制数的字符串。

static String toOctalString(int i) ：将整数转为八进制数的字符串。

9. 基本数据类

9.1 Interger类

Integer的常用方法如下：

String toString() ：将该整数类型转换为字符串。

static String toString(int i) ：将该整数类型转换为字符串。不同的是，此为类方法。

static String toString(int i, int radix) ：将整数i以基数radix的形式转换成字符串。

static Integer valueOf(String s) ：将字符串转换成整数类型。

9. 基本数据类

9.2 包装类实例

[例5-16]整型包装类实例

程序源码：

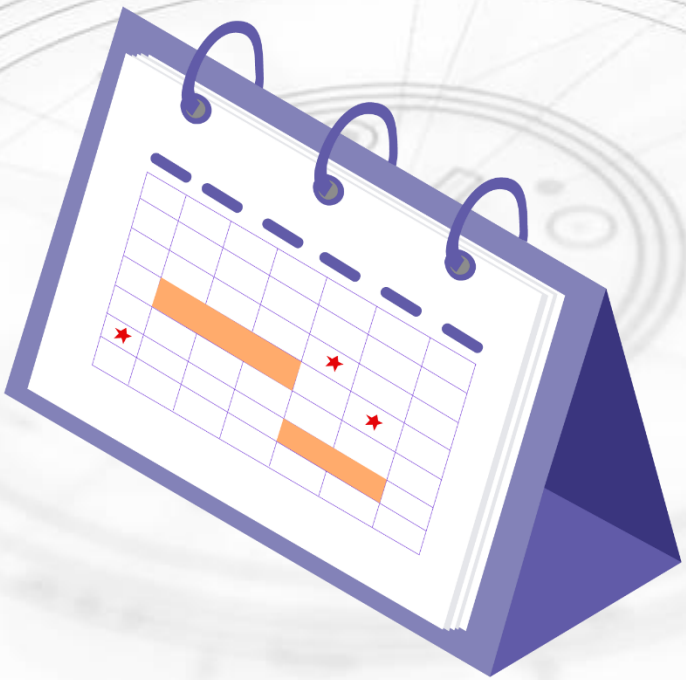
Convert.java

中国科技出版传媒股份有限公司
China Science Publishing & Media Ltd.（CSPM）
科 学 出 版 社

10. 实用工具类

10.1 日期类

Java最常用的表示日期的工具类有Data类和Calendar类。



中国科技出版传媒股份有限公司
China Science Publishing & Media Ltd.（CSPM）
科 学 出 版 社

10. 实用工具类

10.1 日期类

（1）Data类

Data类对象表示当前日期与时间，并提供操作日期和时间各组成部分的方法。必须将Data对象装换为字符串，才能将其输出。

10. 实用工具类

10.1 日期类

（1）Data类

① Data类常用的构造方法

public Date()：创建的日期类对象的日期时间被设置成创建时刻相对应的日期时间。例如：

Date today=new Date()；//today被设置成创建时刻相对应的日期时间。

public Date (long date)：long 型的参数date可以通过调用Date类中的static方法parse(String s)来获得。

10. 实用工具类

10.1 日期类

（1）Data类

② Data类常用方法

String toString（）:返回日期的格式化字符串，包括星期几。

void getTime():设置日期对象，以表示自1970年1月1日起指定的毫秒数。

[例5-17] Data类的用法。

程序源码：

chap05_17.java

10. 实用工具类

10.1 日期类

（2）Calendar类

根据给定的Data类， Calendar类可以以整型(即用一组整型如YEAR，MONTH，DAY，HOUR)的形式检索信息。类Calendar是一个抽象类，使用Calender 类的static方法getInstance（ ）可以初始化一个日历对象，

例如：

Calender calendar1= Calender.getInstance();

10. 实用工具类

10.1 日期类

Calender对象定义后，可以调用set方法来设置日期和时间，set方法主要调用3种方式如下：

- public final void set(int year，int month，int date)
- public final void set(int year，int month，int date，int hour，int minute)
- public final void set(int year，int month，int date，int hour；int minute，int second)

10. 实用工具类

10.1 日期类

要获取有关年份、月份、小时、星期等信息，calendar对象调用方法 public int get(int field)，其中参数field的有效值由Calendar的静态常量指定，例如：

calendar.get(Calendar.MONTH);

运行后将返回一个整数，如果该整数是0，表示当前日历是在一月；该整数是1，表示当前日历是在二月等。

Calender对象还可以调用getTimeInMillis()获取，可以返回从1970 年 1 月 1 日至现在所经过的毫秒数。

中国科技出版传媒股份有限公司
China Science Publishing & Media Ltd.（CSPM）
科 学 出 版 社

10. 实用工具类

10.1 日期类

[例5-18]使用Calendar类显示时间信息。

程序源码： chap05_18.java

10. 实用工具类

10.2 Random类与随机数

java实用工具类库中的类java.util.Random提供了产生各种类型随机数的方法，它可以产生int、long、float、double以及Goussian等类型的随机数。

类Random中的方法十分简单，它只有两个构造方法和六个普通方法。

中国科技出版传媒股份有限公司
China Science Publishing & Media Ltd.（CSPM）
科学出版社

10. 实用工具类

10.2 Random类与随机数

（1）构造方法：

public Random()

public Random(long seed)

Java产生随机数需要有一个基值seed，在第一种方法中基值缺省，则将系统时间作为seed。

10. 实用工具类

10.2 Random类与随机数

（2）普通方法：

public synonronized void setSeed(long seed)：该方法是设定基值seed。

public int nextInt()：该方法是产生一个整型随机数。

public long nextLong()：该方法是产生一个long型随机数。

public float nextFloat()：该方法是产生一个Float型随机数。

public double nextDouble()：该方法是产生一个Double型随机数。

public synchronized double nextGoussian()：该方法是产生一个double型的下一个高斯分布的随机数。生成的高斯值的中间值为0.0，而标准差为1.0。

10. 实用工具类

10.2 Random类与随机数

[例5-20]模拟自然界抛硬币10次，输出正面和反面各多少次的例子。

程序源码：

chap05_20.java

中国科技出版传媒股份有限公司
China Science Publishing & Media Ltd.（CSPM）
科 学 出 版 社

10. 实用工具类

10.3 Math类

java.lang.Math类主要用于对数字的一些基本操作，常用的方法包括：

10. 实用工具类

10.3 Math类

（1）三角函数：

public static double sin (double radians)

public static double cos(double radians)

public static double tan(double radians)

public static double toRadians (double degree)

public static double toDegrees (double radians)

public static double asin (double a)

public static double acos (double b)

public static double atan (double a)

10. 实用工具类

10.3 Math类

```
Math.toDegrees ( Math.PI/2 );//返回值为90.0
Math.toRadians ( 30 ) ;    //返回值为π/6
Math.sin ( 0 );     //返回值为0.0
Math.sin ( Math.toRadians(270) );     //返回值为-1.0
Math.sin ( Math.PI/6 );     //返回值为0.5
Math.sin ( Math.PI/2 );     //返回值为1.0
Math.cos ( 0 ) ;     //返回值为 1.0
Math.cos ( Math.PI/6 ) ;     //返回值为0.866
Math.cos ( Math.PI/2 );     //返回值为0
Math.asin ( 0.5 );     //返回值为π/6
```

10. 实用工具类

10.3 Math类

（2）指数函数：

public static double exp (double x) : 表示e^x。

public static double log (double x) : 表示ln x。

public static double log10 (double x) ：表示 log10 (x)。

public static double pow (double a, double b) : 表示a的b次方。

public static double sqrt (double x) ：表示x的开平方。

10. 实用工具类

10.3 Math类

```
Math.exp ( 1 );     //返回值为2.71828
Math.log ( Math.E );     //返回值为1.0
Math.log10 ( 10 );     //返回值为 1.0
Math.pow ( 2, 3 );     //返回值为8.0
Math.pow ( 3, 2 );     //返回值为 9.0
Math.pow (3.5, 2.5 );     //返回值为 22.91765
Math.sqrt ( 4 );     //返回值为2.0
Math.sqrt ( 10.5 );     //返回值为3.24
```

中国科技出版传媒股份有限公司
China Science Publishing & Media Ltd.（CSPM）
科 学 出 版 社

10. 实用工具类

10.3 Math类

（3）取整方法：

public static double ceil (double x)：向上取整。

public static double floor (double x)：向下取整。

public static int round (float x) ：四舍五入取整。

Public static long round (double x) ：四舍五入取整。

10. 实用工具类

10.3 Math类

（4）常量

java.lang.Math类中还包含E和PI两个静态常量，定义值如下：

public static final Double E = 2.718281828459045235 4

public static final Double PI = 3.14159265358979323846

4 本章小结

中国科技出版传媒股份有限公司
China Science Publishing & Media Ltd.（CSPM）
科学出版社

本章小结

本章主要介绍了两部分内容：第一部分是数组的定义和结构，第二部分是常见字符串类型、正则表达式，以及.Java类库。通过本章学习，要求读者掌握数组的使用和类库的常用功能。

中国科技出版传媒股份有限公司
China Science Publishing & Media Ltd.（CSPM）
科 学 出 版 社


THANK YOU
中国科技出版传媒股份有限公司
China Science Publishing & Media Ltd.（CSPM）
科 学 出 版 社