

A hand in a suit points towards the word "Java". The background is a blurred image of a person in a suit. Various hexagonal icons are scattered around, including a location pin, a house, a magnifying glass, a gear, a clock, a globe, an envelope, a smartphone, a refresh arrow, and a location pin.

Java

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

2 难点重点

3 知识内容

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

```
h3 {font-size: 20px !important;}
th {font-size: 11px; text-align: left;}
tr {margin: 3px !important; padding: 0px !important; padding-top: 5px !important; border-top: 1px solid #ccc !important;}

#container {margin: auto; width: 850px; padding-top: 90px;}

#info_bar_line1 {font-weight: bold; font-size: 20px; margin: 0; padding: 0; text-align: left;}
#info_bar_line2 {font-size: 14px; margin: 0; text-align: left;}
.info_bar {width: 100%; background-color: #4285C4; position: fixed; padding: 10px 20px; z-index: 10;}
.info_bar p {color: #ffffff !important;}

.hide {display: none;}

#field_information {cursor: pointer; float: left; margin: 1px 0 0 5px;}
#field_information_container {float: left;}
.label {font-size: 32K !important;}
.btn_copy_text {width: 110px;}
.btn_get_first {width: 110px;}

.title {width: 701px !important;}
.description {width: 701px !important; height: 73px !important;}

.tag_editor {line-height: 25px !important; height: 225px; padding: 5px 0px !important; border: 1px solid #ccc !important; border-radius: 4px;}
.tag_editor_delete {height: 25px !important;}
.tag_editor_delete i {line-height: 25px !important;}
.tag_editor_spacer {width: 10px !important;}

#btn_settings { webkit-user-select: none; -ms-user-select: none; ms-user-select: none; user-select: none; user-select: none; transition: all 0.5s ease-out 0s;}
#btn_settings:hover { cursor: pointer; transform: rotate(100deg); transition: all 0.5s ease-out 0s;}

#select_theme_container {width: 280px;}
#google_api_key {width: 400px;}
#get_first_n_value {width: 50px;}
.simple_text {text-decoration: none !important;}
.panel_settings {padding: 10px !important;}
.panel_settings_container {margin-bottom: 5px !important;}

#google_translate_api_info {font-size: 10px; margin-left: 35px;}
.checkbox_comment {font-size: 10px;}
.btn_default .badge {margin-left: 5px; border-radius: 5px !important;}
mark {padding: 0 !important;}

#add_and_translate {font-size: 10px;}

.tooltipster-box {background: #fff !important;}
.tooltipster-arrow-background {border-top-color: #fff !important;}
.tooltipster-box {-webkit-box-shadow: 0 1px 4px rgba(0,0,0,.2); box-shadow: 0 1px 4px rgba(0,0,0,.2)}
```



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


```
port class MainActivity extends AppCompatActivity {
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
    public void onClick(View view) {
        Intent i = new Intent("net.1
```

在前面的章节中介绍过在程序中可以通过数组来保存多个对象，但在某些情况下无法确定到底需要保存多少个对象，此时数组将不再适用，因为数组的长度不可变。

JDK中提供了一系列特殊的类，这些类可以存储任意类型的对象，并且长度可变，统称为集合。

集合按照其存储结构可以分为两大类，即单列集合Collection和双列集合Map。

```
h3 {font-size: 20px !important;}
th {font-size: 11px; text-align: left;}
tr {margin: 3px !important; padding: 0px !important; padding-top: 5px !important; border-top: 1px solid #ccc !important;}

#container {margin: auto; width: 850px; padding-top: 90px;}

#info_bar_line1 {font-weight: bold; font-size: 20px; margin: 0; padding: 0; text-align: left;}
#info_bar_line2 {font-size: 14px; margin: 0; text-align: left;}
.info_bar {width: 100%; background-color: #4285c4; position: fixed; padding: 10px 20px; z-index: 10;}
.info_bar p {color: #ffffff !important;}

.hide {display: none;}

#field_information {cursor: pointer; float: left; margin: 1px 0 0 5px;}
#field_information_container {float: left;}
.label {font-size: 32k !important;}
.btn_copy_text {width: 110px;}
.btn_get_first {width: 110px;}

.title {width: 701px !important;}
.description {width: 701px !important; height: 73px !important;}

.tag_editor {line-height: 25px !important; height: 225px; padding: 5px 0px !important; border: 1px solid #ccc !important; border-radius: 4px;}
.tag_editor_delete {height: 25px !important;}
.tag_editor_delete i {line-height: 25px !important;}
.tag_editor_spacer {width: 10px !important;}

#btn_settings { webkit-user-select: none; -ms-user-select: none; ms-user-select: none; user-select: none; user-select: none; transition: all 0.3s ease-out 0s;}
#btn_settings:hover { cursor: pointer; transform: rotate(100deg); transition: all 0.3s ease-out 0s;}

#select_theme_container {width: 280px;}
#google_api_key {width: 400px;}
#get_first_n_value {width: 50px;}
.simple_text {text-decoration: none !important;}
.pamel_settings {padding: 10px !important;}
.pamel_settings_container {margin-bottom: 5px !important;}

#google_translate_api_info {font-size: 10px; margin-left: 35px;}
.checkbox_comment {font-size: 10px;}
.btn_default .badge {margin-left: 5px; border-radius: 5px !important;}
mark {padding: 0 !important;}

#add_and_translate {font-size: 10px;}

.tooltipster-box {background: #fff !important;}
.tooltipster-arrow-background {border-top-color: #fff !important;}
.tooltipster-box {-webkit-box-shadow: 0 1px 4px rgba(0,0,0,.2); box-shadow: 0 1px 4px rgba(0,0,0,.2)}
```



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



List接口



Set接口



Map接口



JDK5.0新特性——泛型



Collections工具类



Arrays工具类



反射基础



```
h3 {font-size: 20px !important;}
th {font-size: 11px; text-align: left;}
tr {margin: 3px !important; padding: 0px !important; padding-top: 5px !important; border-top: 1px solid #ccc !important;}

#container {margin: auto; width: 850px; padding-top: 90px;}

#info_bar_line1 {font-weight: bold; font-size: 20px; margin: 0; padding: 0; text-align: left;}
#info_bar_line2 {font-size: 14px; margin: 0; text-align: left;}
.info_bar {width: 100%; background-color: #4285C4; position: fixed; padding: 10px 20px; z-index: 10;}
.info_bar p {color: #ffffff !important;}

.hide {display: none;}

#field_information {cursor: pointer; float: left; margin: 1px 0 0 5px;}
#field_information_container {float: left;}
.label {font-size: 32K !important;}
.btn_copy_text {width: 110px;}
.btn_get_first {width: 110px;}

.title {width: 701px !important;}
.description {width: 701px !important; height: 73px !important;}

.tag_editor {line-height: 25px !important; height: 225px; padding: 5px 0px !important; border: 1px solid #ccc !important; border-radius: 4px;}
.tag_editor_delete {height: 25px !important;}
.tag_editor_delete i {line-height: 25px !important;}
.tag_editor_spacer {width: 10px !important;}

#btn_settings { webkit-user-select: none; -ms-user-select: none; ms-user-select: none; user-select: none; user-select: none; transition: all 0.5s ease-out 0s;}
#btn_settings:hover { cursor: pointer; transform: rotate(100deg); transition: all 0.5s ease-out 0s;}

#select_theme_container {width: 280px;}
#google_api_key {width: 400px;}
#get_first_n_value {width: 50px;}
.simple_text {text-decoration: none !important;}
.panel_settings {padding: 10px !important;}
.panel_settings_container {margin-bottom: 5px !important;}

#google_translate_api_info {font-size: 10px; margin-left: 35px;}
.checkbox_comment {font-size: 10px;}
.btn_default .badge {margin-left: 5px; border-radius: 5px !important;}
mark {padding: 0 !important;}

#add_and_translate {font-size: 10px;}

.tooltipster-box {background: #fff !important;}
.tooltipster-arrow-background {border-top-color: #fff !important;}
.tooltipster-box {-webkit-box-shadow: 0 1px 4px rgba(0,0,0,.2); box-shadow: 0 1px 4px rgba(0,0,0,.2)}
```



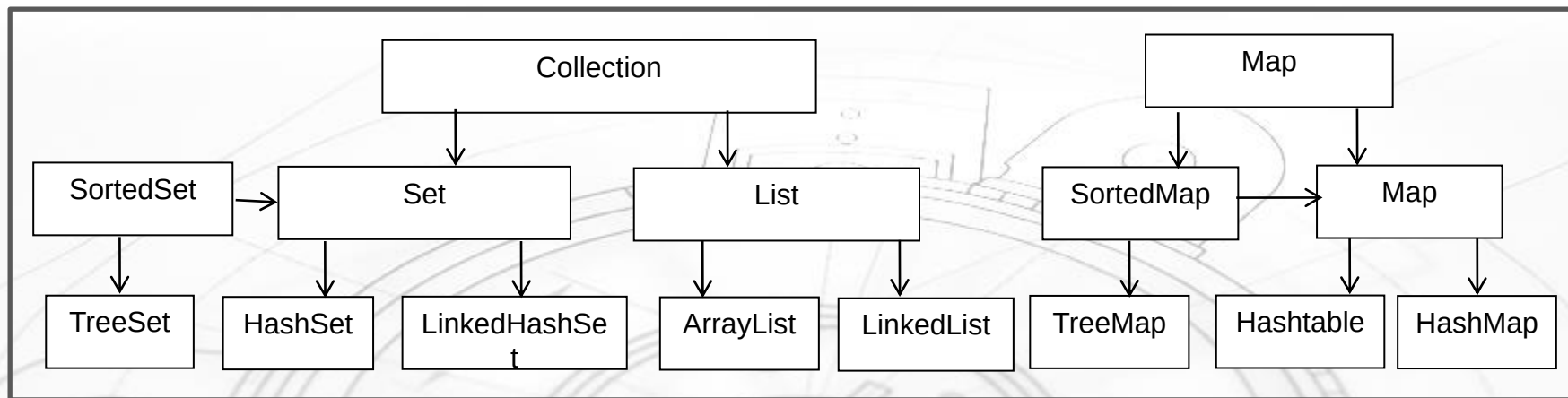
3

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. 集合概述



集合体系架构图

Collection是所有单列集合的父接口，因此在Collection中定义了单列集合(List和Set)通用的一些方法，这些方法可用于操作所有的单列集合。

2. List接口

2.1 List接口简介

List接口继承自Collection接口，是单列集合的一个重要分支，习惯性地会将实现了List接口的对象称为List集合。

在List集合中允许出现重复的元素，所有的元素是以一种线性方式进行存储的，在程序中可以通过索引来访问集合中的指定元素。

另外，List集合还有一个特点就是元素有序，即元素的存入顺序和取出顺序一致。

List不但继承了Collection接口中的全部方法，而且还增加了一些根据元素索引来操作集合的特有方法。

2. List接口

2.2 ArrayList集合

ArrayList是List接口的一个实现类，它是程序中最常见的一种集合。

在ArrayList内部封装了一个长度可变的数组对象，当存入的元素超过数组长度时，ArrayList会在内存中分配一个更大的数组来存储这些元素，因此可以将ArrayList集合看作一个长度可变的数组。

ArrayList集合中大部分方法都是从父类Collection和List继承过来的，其中add()方法和get()方法用于实现元素的存取。

2. List接口

2.2 ArrayList集合

在集合ArrayList的定义语句中，我们使用了泛型的符号<>，该符号的作用是用来指明集合中的元素类型，关于泛型的更详细的讲解，将在后面的章节中进行，这里只需要记住使用方式以及作用即可。

ArrayList集合不仅可以存储系统已有数据类型，也可以存储自定义数据类型，下面请看例11-2，在该例中，有自定义类Point，将若干Point对象放入集合ArrayList中。

2. List接口

2.3 LinkedList集合

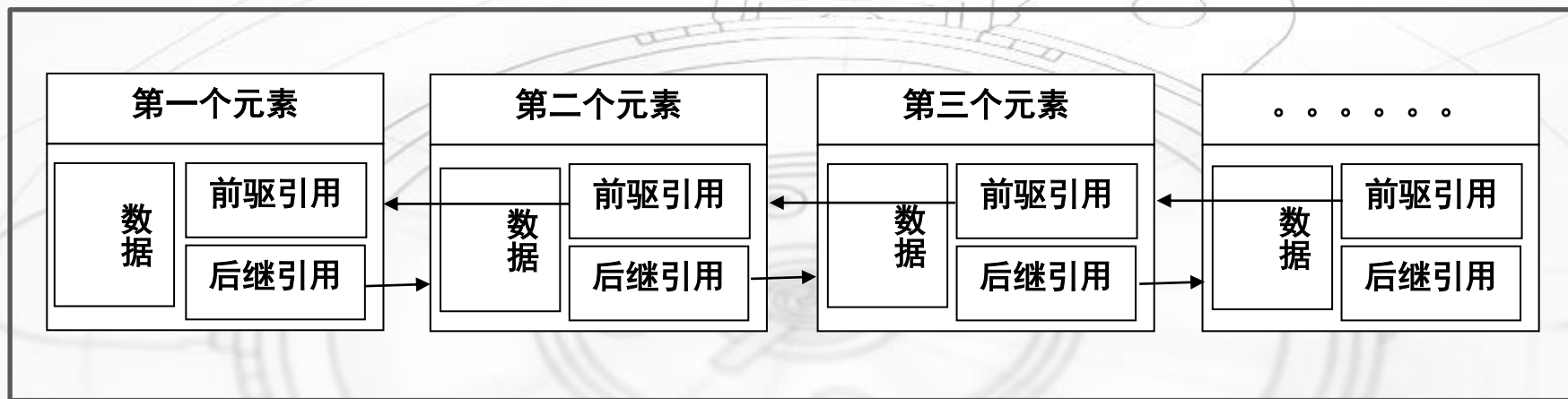
List接口的另一个实现类LinkedList，克服了ArrayList集合在查询元素时速度很快，但在增删元素时效率较低的局限性。

该集合内部维护了一个双向循环链表，链表中的每一个元素都使用引用的方式来记住它的前一个元素和后一个元素，从而可以将所有的元素彼此连接起来。

当插入一个新元素时，只需要修改元素之间的这种引用关系即可，删除一个节点也是如此。

2. List接口

2.3 LinkedList集合



2. List接口

2.4 Iterator集合

Iterator接口也是Java集合框架中的一员，但它与Collection、Map接口有所不同，Collection接口与Map接口主要用于存储元素，而Iterator主要用于迭代访问（即遍历）Collection中的元素，因此Iterator对象也被称为迭代器。

通过一个案例来学习如何使用Iterator迭代集合中的元素。

2. List接口

2.5 JDK5.0新特性——foreach循环

虽然Iterator可以用来遍历集合中的元素，但写法上比较繁琐，为了简化书写，从JDK5.0开始，提供了foreach循环。foreach循环是一种更加简洁的for循环，也称增强for循环。foreach循环用于遍历数组或集合中的元素，其具体语法格式如下：

foreach的语句格式：

```
for(元素类型t 元素变量x : 遍历对象obj){
```

```
    引用了x的java语句;
```

```
}
```

2. List接口

2.5 JDK5.0新特性——foreach循环

与for循环相比，foreach循环不需要获得容器的长度，也不需要
根据索引访问容器中的元素，但它会自动遍历容器中的每个元素。

但是foreach循环也有其局限性的一面。当使用foreach循环遍历
集合或数组的时候，只能访问集合中的元素，却不能对其进行修改。
接下来以一个String类型的数组为例来进行演示。

2. List接口

在JDK1.2以前还没有Iterator接口的时候，遍历集合需要使用Enumeration接口，它的用法和Iterator类似。

JDK中提供了一个Vevtor集合，该集合是List接口的一个实现类，用法与ArrayList完全相同，区别在于Vector集合是线程安全的，而ArrayList集合是线程不安全的。在Vector类中提供了一个elements()方法用于返回Enumeration对象，通过Enumeration对象就可以遍历该集合中的元素。

通过一个案例来演示如何使用Enumeration对象遍历Vector集合：

3. Set接口

3.1 Set接口简介

Set接口和List接口一样，同样继承自Collection接口，它与Collection接口中的方法基本一致，并没有对Collection接口进行功能上的扩充，只是比Collection接口更加严格了。与List接口不同的是，Set接口中元素无序，并且都会以某种规则保证存入的元素不出现重复。

3. Set接口

3.1 Set接口简介

Set接口主要有两个实现类，分别是HashSet和TreeSet。其中，HashSet是根据对象的哈希值来确定元素在集合中的存储的位置，因此具有良好的存取和查找性能。TreeSet则是以二叉树的方式来存储元素，它可以实现对集合中的元素进行排序。

3. Set接口

3.2 HashSet集合

HashSet是Set接口的一个实现类，它所存储的元素是不可重复的，并且元素都是无序的。当向HashSet集合中添加一个对象时，首先，会调用该对象的hashCode()方法来确定元素的存储位置，然后再调用对象的equals()方法来确保该位置没有重复元素。

3. Set接口

3.2 HashSet集合

HashSet集合之所以能确保不出现重复的元素，是因为它在存入元素时做了很多工作。当调用HashSet集合的add()方法存入元素时，首先调用当前存入对象的hashCode()方法获得对象的哈希值，然后根据对象的哈希值计算出一个存储位置。如果该位置上没有元素，则直接将元素存入，如果该位置上有元素存在，则会调用equals()方法让当前存入的元素依次和该位置上的元素进行比较，如果返回的结果为false就将该元素存入集合，返回的结果为true则说明有重复元素，就将该元素舍弃。

3. Set接口

3.2 HashSet集合

根据前面的分析不难看出，当向集合中存入元素时，为了保证HashSet正常工作，要求在存入对象时，需要重写Object类中的hashCode()和equals()方法。例程7-9中将字符串存入HashSet时，String类已经重写了hashCode ()和equals()方法。但是如果将Student对象存入HashSet，结果又如何呢？

3. Set接口

3.2 HashSet集合

接下来对Student类进行设计，假设name相同的学生就是同一个学生，代码如下所示。

【例11-91】 Student类进行设计。

【例11-92】 HashSet集合的特点，不可以添加重复元素。

4. Map接口

4.1 Map接口简介

在现实生活中，每个人都有唯一的身份证号，通过身份证号可以查询到这个人的信息，这两者是一一对应的关系。

在应用程序中，如果想存储这种具有对应关系的数据，则需要使用JDK中提供的Map接口。

Map接口是一种双列集合，它的每个元素都包含一个键对象Key和值对象Value，键和值对象之间存在一种对应关系，称为映射。

从Map集合中访问元素时，只要指定了Key，就能找到对应的Value。

4. Map接口

4.1 Map接口简介

首先了解一下Map接口中定义的一些通用方法。

Map接口提供了大量的实现类，最常用的有HashMap，接下来针对这个类进行详细地讲解。

HashMap集合是Map接口的一个实现类，它用于存储键值映射关系，但必须保证不出现重复的键。

4. Map接口

4.2 HashMap集合

在程序开发中，经常需要取出Map中所有的键和值，那么如何遍历Map中所有的键值对呢？

有两种方式可以实现，第一种方式就是先遍历Map集合中所有的键，再根据键获取相应的值。

Map集合的另外一种遍历方式是先获取集合中的所有的映射关系，然后从映射关系中取出键和值。

5. JDK5.0新特性——泛型

当把一个对象存入集合后，集合会“忘记”这个对象的类型，将该对象从集合中取出时，这个对象的编译类型就变成了Object类型。

换句话说，我们在程序中无法确定一个集合中的元素到底是什么类型的。那么在取出元素时，如果进行强制类型转换就很容易出错。

5. JDK5.0新特性——泛型

5.1 为什么使用泛型

为了解决这个问题，在Java中引入了“参数化类型(parameterized type)”这个概念，即泛型。

它可以限定方法操作的数据类型，在定义集合类时，可以使用“<参数化类型>”的方式指定该类中方法操作的数据类型。

上面这种写法就限定了ArrayList集合只能存储String类型元素，将改写后的程序再次编译，程序在编译时期就会出现错误提示。

5. JDK5.0新特性——泛型

5.1 为什么使用泛型

程序编译报错的原因是修改后的代码限定了集合元素的数据类型，`ArrayList<String>`这样的集合只能存储String类型的元素，程序在编译时，编译器检查出Integer类型的元素与List集合的规定的类型不匹配，编译不通过，这样就可以在编译时期解决错误，避免程序在运行时期发生错误。

5. JDK5.0新特性——泛型

5.2 自定义泛型

泛型的作用是什么，在程序中是否能自定义泛型呢？



6. Collections工具类

JDK提供了一个工具类专门用来操作集合，这个类就是 Collections，它位于java.util包中。Collections类中提供了大量的方法用于对集合中元素进行排序、查找和修改等操作，接下来对这些常用的方法进行介绍。

6. Collections工具类

6.1 查找、替换操作

Collections类还提供了一些常用方法用于查找、替换集合中的元素。



6. Collections工具类

6.2 对象排序

Collections类中的排序方法，经常会在以后的开发中使用，上面的例子中排序的集合元素为单纯的数字，比较好理解，如果集合中存储的是若干对象，要求按照对象的某一个字段信息对集合中的元素进行排序，程序该怎么写呢？

7. Arrays工具类

java.util包中还提供了一个专门用于操作数组的工具类——**Arrays**。Arrays工具类提供了大量的静态方法。



7. Arrays工具类

7.1 使用Arrays的sort()方法排序

在前面学习数组时，要想对数组进行排序就需要自定义一个排序方法，其实也可以使用Arrays工具类中的静态方法sort()来实现这个功能，接下来通过一个案例来学习sort()方法的使用。

7. Arrays工具类

7.2 使用Arrays的binarySearch

(Object[] a, Object key) 方法查找元素

程序开发中，经常会在数组中查找某些特定的元素，如果数组中元素较多时查找某个元素就会非常繁琐，为此，Arrays类中提供还了一个方法binarySearch(Object[] a, Object key)用于查找元素，接下来通过一个案例来学习该方法的使用。

7. Arrays工具类

7.3 使用Arrays的fill

(Object[] a, Object val)方法填充元素

程序开发中，经常需要用一個值替换数组中的所有元素，这时可以使用Array的fill(Object[] a, Object val)方法，该方法可以将指定的值赋给数组中的每一个元素，接下来通过一个案例来演示如何填充元素。

7. Arrays工具类

7.4 使用Arrays的toString

(int[] arr)方法把数组转换为字符串

在程序开发中，经常需要把数组以字符串的形式输出，这时就可以使用Arrays工具类的另一个方法toString(int[] arr)。

需要注意的是，该方法并不是对Object类toString()方法的重写，只是用于返回指定数组的字符串形式，接下来通过一个案例来演示如何将数组转换为字符串。

8. 反射基础

Java的反射机制是Java特性之一，Java程序的运行需要Java类要被Java虚拟机加载。Java类如果不被Java虚拟机加载，是不能正常运行的。但是编译时期并不确定是哪个类被加载了，而是在程序运行的时候才加载。使用在编译期并不知道类。这样的特点就是反射。

在Java中，每个类都有一个相应的Class对象。也就是说，当我们编写一个类，编译完成后，在生成的.class文件中，就会产生一个Class对象，用于表示这个类的类型信息。

8. 反射基础

获取Class实例的三种方式：

- (1) 利用对象调用getClass()方法获取该对象的Class实例；
- (2) 使用Class类的静态方法forName()，用类的名字获取一个Class实例；
- (3) 运用.class的方式来获取Class实例，对于基本数据类型的封装类，还可以采用.TYPE来获取相对应的基本数据类型的Class实例。

```
h3 {font-size: 20px !important;}
th {font-size: 11px; text-align: left;}
tr {margin: 3px !important; padding: 0px !important; padding-top: 5px !important; border-top: 1px solid #ccc !important;}

#container {margin: auto; width: 850px; padding-top: 90px;}

#info_bar_line1 {font-weight: bold; font-size: 20px; margin: 0; padding: 0; text-align: left;}
#info_bar_line2 {font-size: 14px; margin: 0; text-align: left;}
.info_bar {width: 100%; background-color: #4285c4; position: fixed; padding: 10px 20px; z-index: 10;}
.info_bar p {color: #ffffff !important;}

.hide {display: none;}

#field_information {cursor: pointer; float: left; margin: 1px 0 0 5px;}
#field_information_container {float: left;}
.label {font-size: 32k !important;}
.btn_copy_text {width: 110px;}
.btn_get_first {width: 110px;}

.title {width: 701px !important;}
.description {width: 701px !important; height: 73px !important;}

.tag_editor {line-height: 25px !important; height: 225px; padding: 5px 0px !important; border: 1px solid #ccc !important; border-radius: 4px;}
.tag_editor_delete {height: 25px !important;}
.tag_editor_delete i {line-height: 25px !important;}
.tag_editor_spacer {width: 10px !important;}

#btn_settings { webkit-user-select: none; -ms-user-select: none; ms-user-select: none; user-select: none; user-select: none; transition: all 0.3s ease-out 0s;}
#btn_settings:hover { cursor: pointer; transform: rotate(160deg); transition: all 0.3s ease-out 0s;}

#select_theme_container {width: 280px;}
#google_api_key {width: 400px;}
#get_first_n_value {width: 50px;}
.simple_text {text-decoration: none !important;}
.pamel_settings {padding: 10px !important;}
.pamel_settings_container {margin-bottom: 5px !important;}

#google_translate_api_info {font-size: 10px; margin-left: 35px;}
.checkbox_comment {font-size: 10px;}
.btn_default .badge {margin-left: 5px; border-radius: 5px !important;}
mark {padding: 0 !important;}

#add_and_translate {font-size: 10px;}

.tooltipster-box {background: #fff !important;}
.tooltipster-arrow-background {border-top-color: #fff !important;}
.tooltipster-box {-webkit-box-shadow: 0 1px 4px rgba(0,0,0,.2); box-shadow: 0 1px 4px rgba(0,0,0,.2)}
```



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

本章详细介绍了几种Java常用集合类，从Collection、Map根接口开始讲起，重点介绍了List集合、Set集合、Map集合之间的区别，以及常用实现类的使用方法和需要注意的问题，另外还介绍了泛型的使用以及Collections工具类和Arrays工具类，最后对Java的反射做了简单的讲解。

通过本章的学习，必须熟练掌握各种集合类的使用场景，以及需要注意的细节。掌握泛型的使用以及工具类的使用。



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