

# 常用API(String、ArrayList)



黑马程序员  
[www.itheima.com](http://www.itheima.com)

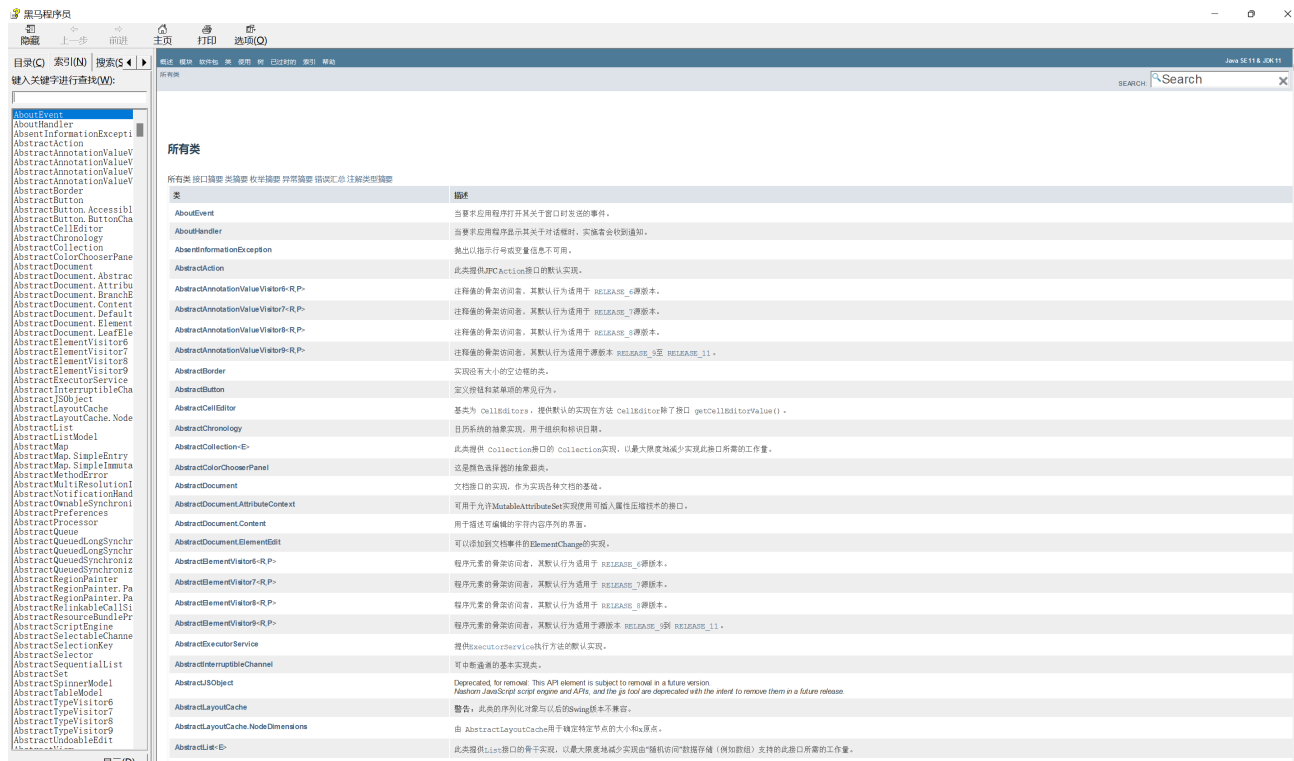
传智教育旗下  
高端IT教育品牌



多一句没有，少一句不行，用最短时间，教会最实用的技术！

## API ( Application Programming Interface,应用程序编程接口 )

- Java写好的程序(功能)，咱们可以直接调用。
- Oracle 也为Java提供的这些功能代码提供了相应的 **API文档(使用说明书)**。



## 高级软件人才培养专家

## String简单介绍

- String类定义的变量可以用于存储字符串，同时String类提供了很多操作字符串的功能，我们可以直接使用。



扫码登录更安全

密码登录 短信登录

itheima

.....

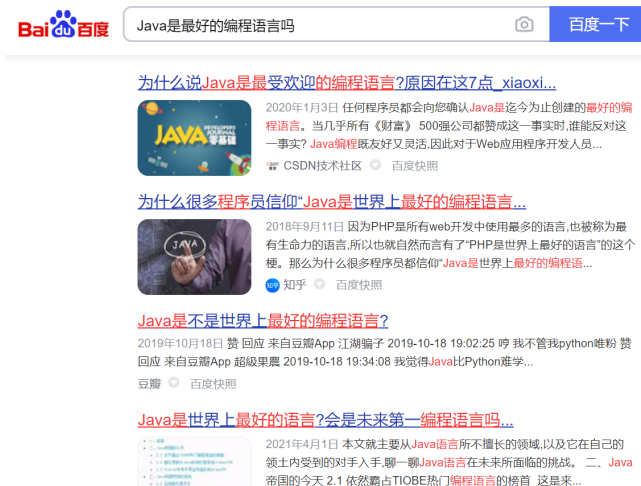
登录

微博登录 支付宝登录

忘记密码 忘记用户名 免费注册

需要跟正确的用户名和密码进行比较。

```
[12:24上午]: 还玩***
[12:24上午] Da1nA呀: 为什么都是
[12:25上午] 喜欢你不爱流浪: 还玩***
[12:25上午] Da1nA呀: 我***
[12:25上午] 喜欢你不爱流浪: 玩***
??
[12:25上午] Da1nA呀: 我***了
[12:25上午] 喜欢你不爱流浪: 为什么啊
```



## 关于String类同学们需要学会什么

String定义变量存储字符串

需要知道如何创建字符串对象，并使用String定义变量指向该字符串对象。

String的内存原理

字符串对象在内存中的原理是什么样。能够解决一些字符串的常见面试题

String类提供了哪些API

能够说出并使用String类提供的操作字符串的功能：遍历、替换、截取、相等，包含...

String解决实际案例

能够利用String的常用API去解决实际场景的业务需求，真正做到学以致用



# 目录

Contents

## ➤ String

- ◆ String类概述
- ◆ String类的常用方法-遍历、替换、截取、分割操作
- ◆ String类案例实战
- ◆ String类创建对象的2种方式
- ◆ String类常见面试



# 目录

Contents

## ➤ String

### ◆ String类概述

### ◆ String类常用方法-字符串内容比较

### ◆ String类的常用方法-遍历、替换、截取、分割操作

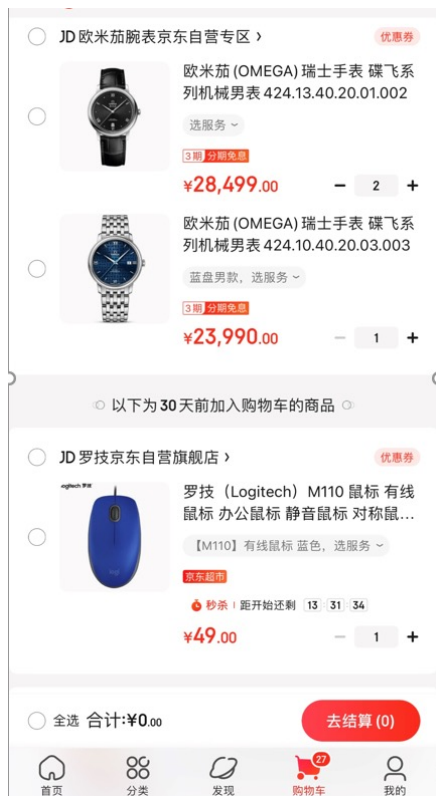
### ◆ String类案例实战

### ◆ String类创建对象的2种方式

### ◆ String类常见面试

## ArrayList 简单介绍

- ArrayList代表的是集合类，集合是一种容器，与数组类似，不同的是**集合的大小是不固定的**。
- 通过创建ArrayList的对象表示得到一个集合容器，同时ArrayList提供了比数组更好用，更丰富的API (功能)给程序员使



### 购物车使用集合对象来存储商品对象更合适

- 随时可能添加新商品对象进来（个数不确定）
- 也随时可能删除商品对象

## 关于ArrayList类同学们需要学会什么

ArrayList集合如何创建对象

ArrayList常用API

ArrayList存储自定义对象

ArrayList解决实际问题

要知道如何利用ArrayList创建对象代表集合容器来存放数据。

能够说出并使用ArrayList类提供的丰富的元素操作的功能：添加、获取、删除、修改等功能

能够使用ArrayList存储自定义的对象，并清楚ArrayList集合存储对象的底层原理

能够使用ArrayList存储对象，并完成数据搜索，删除等常见业务需求





# 目录

Contents

## ➤ ArrayList

- ◆ 集合概述
- ◆ ArrayList集合快速入门
- ◆ ArrayList对泛型的支持
- ◆ ArrayList常用方法、遍历
- ◆ ArrayList集合案例：遍历并删除元素
- ◆ ArrayList集合案例：存储自定义类型的对象
- ◆ ArrayList集合案例：元素搜索



# 目录

Contents

## ➤ String

### ◆ String类概述

### ◆ String类常用方法-字符串内容比较

### ◆ String类的常用方法-遍历、替换、截取、分割操作

### ◆ String类案例实战

### ◆ String类创建对象的2种方式

### ◆ String类常见面试

## ➤ ArrayList

## String 概述

- java.lang.String 类代表字符串，String类定义的变量可以用于指向字符串对象，然后操作该字符串。
- Java 程序中的所有字符串文字（例如“abc”）都为此类的对象。

```
String name = "小黑";  
String schoolName = "黑马程序员";
```

## String类的特点详解

- String其实常被称为不可变字符串类型，它的对象在创建后不能被更改。

```
public static void main(String[] args) {  
    String name = "传智";  
    name += "教育";  
    name += "中心";  
    System.out.println(name);  
}
```

C:\soft\Java\jdk-15.0.1\bin\java.exe  
传智教育中心



靓仔疑问

从上述代码可以看出字符串变量name指向的字符串对象，那为何还说字符串不可变呢？

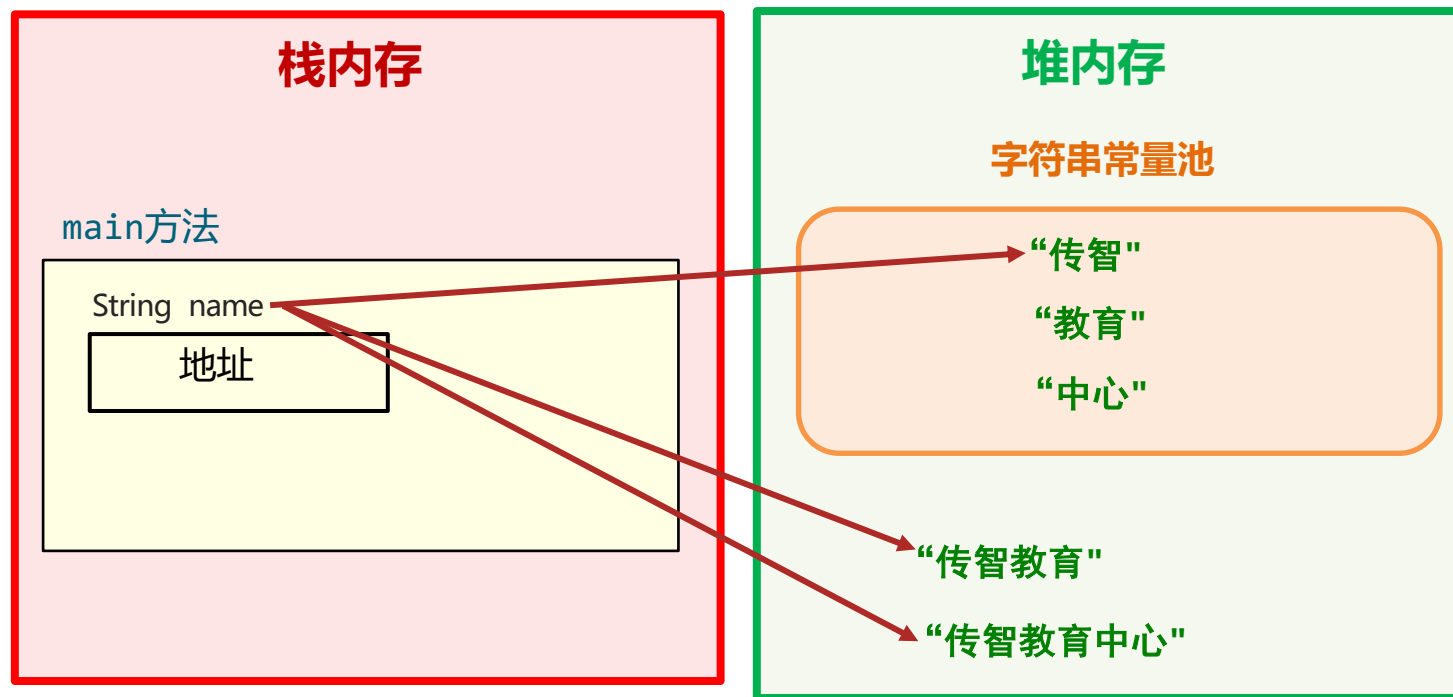
## 字符串对象存在哪里？

- 以 "" 方式给出的字符串对象，在字符串常量池中存储。

```
public static void main(String[] args) {  
    String name = "传智";  
    name += "教育";  
    name += "中心";  
    System.out.println(name);  
}
```

控制台

传智教育中心





## 总结

1. String是什么，可以做什么？

- **字符串类型，可以定义字符串变量指向字符串对象。**

2. String是不可变字符串的原因？

- **String变量每次的修改其实都是产生并指向了新的字符串对象。**
- **原来的字符串对象都是没有改变的，所以称不可变字符串。**



# 目录

## Contents

### ➤ String

- ◆ String类概述

- ◆ String类的常用方法-字符串内容比较

- ◆ String类的常用方法-遍历、替换、截取、分割操作

- ◆ String类案例实战

- ◆ String类创建对象的2种方式

- ◆ String类常见面试

### ➤ ArrayList

## 字符串的内容比较



扫码登录更安全

密码登录 短信登录

itheima

.....

登录

微博登录 支付宝登录

忘记密码 忘记用户名 免费注册

```
public class StringDemoAPI3 {  
    public static void main(String[] args) {  
        String sysLoginName = "itheima";  
  
        Scanner sc = new Scanner(System.in);  
        System.out.println("请您输入您的登录名称");  
        String loginName = sc.next();  
  
        System.out.println(sysLoginName == loginName);  
    }  
}
```

**结论：字符串的内容比较不适合用“==”比较。**



## 字符串的内容比较：

- 推荐使用String类提供的“equals”比较：只关心内容一样即可

方法名	说明
public boolean equals (Object anObject)	将此字符串与指定对象进行比较。只关心字符内容是否一致！
public boolean equalsIgnoreCase (String anotherString)	将此字符串与指定对象进行比较，忽略大小写比较字符串。只关心字符内容是否一致！



## 总结

- 1、如果是字符串比较应该使用使用什么方式进行比较，为什么？
  - 使用String提供的equals方法。
  - 只关心内容一样就返回true。
- 2、开发中什么时候使用==比较数据
  - 基本数据类型比较时使用。



# 目录

## Contents

### ➤ String

- ◆ String类概述
- ◆ String类的常用方法-字符串内容比较
- ◆ String类的常用方法-遍历、替换、截取、分割操作
- ◆ String类案例实战
- ◆ String类创建对象的2种方式
- ◆ String类常见面试

### ➤ ArrayList

## String常用API

方法名	说明
<code>public int length ()</code>	返回此字符串的长度
<code>public char charAt (int index)</code>	获取某个索引位置处的字符
<code>public char[] toCharArray () :</code>	将当前字符串转换成字符数组返回
<code>public String substring(int beginIndex, int endIndex)</code>	根据开始和结束索引进行截取，得到新的字符串（包前不包后）
<code>public String substring(int beginIndex)</code>	从传入的索引处截取，截取到末尾，得到新的字符串
<code>public String replace(CharSequence target, CharSequence replacement)</code>	使用新值，将字符串中的旧值替换，得到新的字符串
<code>public String[] split(String regex)</code>	根据传入的规则切割字符串，得到字符串数组返回



# 目录

Contents

## ➤ String

- ◆ String类概述
- ◆ String类的常用方法-字符串内容比较
- ◆ String类的常用方法-遍历、替换、截取、分割操作
- ◆ String类案例实战
- ◆ String类创建对象的2种方式
- ◆ String类常见面试

## ➤ ArrayList

## 案例 String类开发验证码功能



系统登录

admin

\*\*\*\*\*

验证码, 单击图片刷新

请输入验证码

重置 登录

The image shows a web form for system login. It has a title '系统登录' (System Login) with a dropdown arrow. Below the title are three input fields: a text field containing 'admin', a password field with masked characters '\*\*\*\*\*', and a captcha field. The captcha field has a label '验证码, 单击图片刷新' (Captcha, click image to refresh) and a red text prompt '请输入验证码' (Please enter the captcha). To the right of the captcha input is a small image showing the captcha 'ayf2n'. At the bottom are two buttons: '重置' (Reset) and '登录' (Login).

需求：

- 随机产生一个5位的验证码，每位可能是数字、大写字母、小写字母。

分析：

- ① 定义一个String类型的变量存储a-zA-Z0-9之间的全部字符。
- ② 循环5次，随机一个范围内的索引，获取对应字符连接起来即可。

## 案例 模拟用户登录功能



A mockup of a system login form. It features a title '系统登录' (System Login) at the top left and a close button (X icon) at the top right. Below the title are two input fields: the first contains the text 'admin', and the second contains six asterisks '\*\*\*\*\*'. At the bottom of the form are two buttons: '重置' (Reset) on the left and '登录' (Login) on the right.

### 需求：

- 模拟用户登录功能，最多只给三次机会。

### 分析：

- ① 系统后台定义好正确的登录名称，密码。
- ② 使用循环控制三次，让用户输入正确的登录名和密码，判断是否登录成功，登录成功则不再进行登录；登录失败给出提示，并让用户继续登录。

## 案例 手机号码屏蔽

### 需求

- 键盘录入一个手机号，将中间四位号码屏蔽，最终效果为：

158\*\*\*\*7839

### 分析

- ① 键盘录入一个字符串。
- ② 调用字符串对象的截取API，截取字符串前三位、后四位。
- ③ 将前三位 连接 “\*\*\*\*” 然后继续连接后四位，输出最终结果即可。





# 目录

## Contents

### ➤ String

- ◆ String类概述
- ◆ String类的常用方法-字符串内容比较
- ◆ String类的常用方法-遍历、替换、截取、分割操作
- ◆ String类案例实战
- ◆ String类创建对象的2种方式
- ◆ String类笔试题

### ➤ ArrayList

## 创建字符串对象的2种方式

- 方式一：直接使用 "" 定义。（推荐方式）

```
String name = "传智教育";
```

- 方式二：通过String类的构造器创建对象。

构造器	说明
public String()	创建一个空白字符串对象，不含有任何内容
public String(String original)	根据传入的字符串内容，来创建字符串对象
public String(char[] chs)	根据字符数组的内容，来创建字符串对象
public String(byte[] chs)	根据字节数组的内容，来创建字符串对象

## 有什么区别吗？（面试常考）

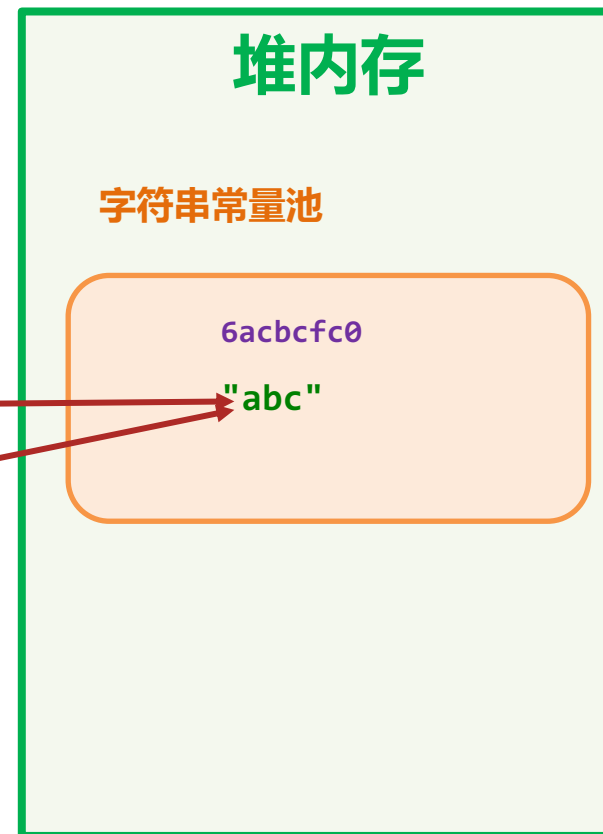
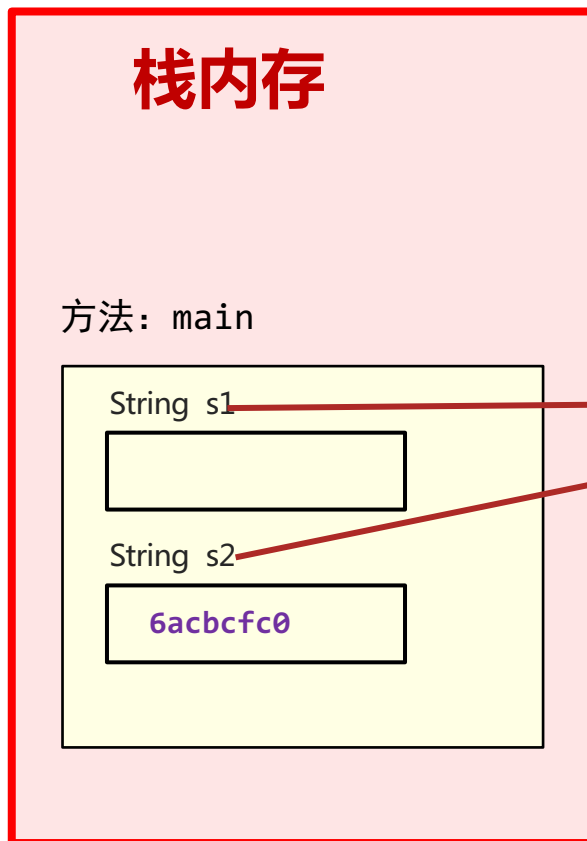
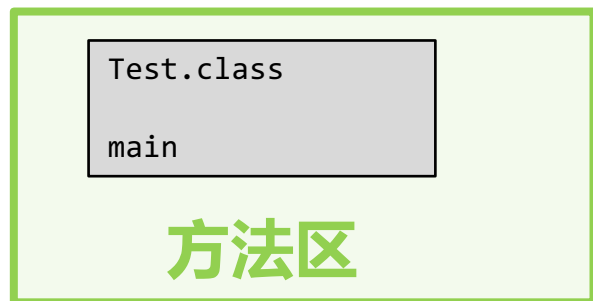
- 以 "" 方式给出的字符串对象，在字符串常量池中存储，而且相同内容只会在其中存储一份。
- 通过构造器new对象，每new一次都会产生一个新对象，放在堆内存中。

```
String s1 = "abc";  
String s2 = "abc";  
System.out.println(s1 == s2); // true  
  
char[] chs = {'a', 'b', 'c'};  
String s3 = new String(chs);  
String s4 = new String(chs);  
  
System.out.println(s3 == s4); // false
```

## 通过 "" 定义字符串内存原理

```
public class Test {  
    public static void main(String[] args) {  
        String s1 = "abc";  
        String s2 = "abc";  
        System.out.println(s1 == s2);  
    }  
}
```

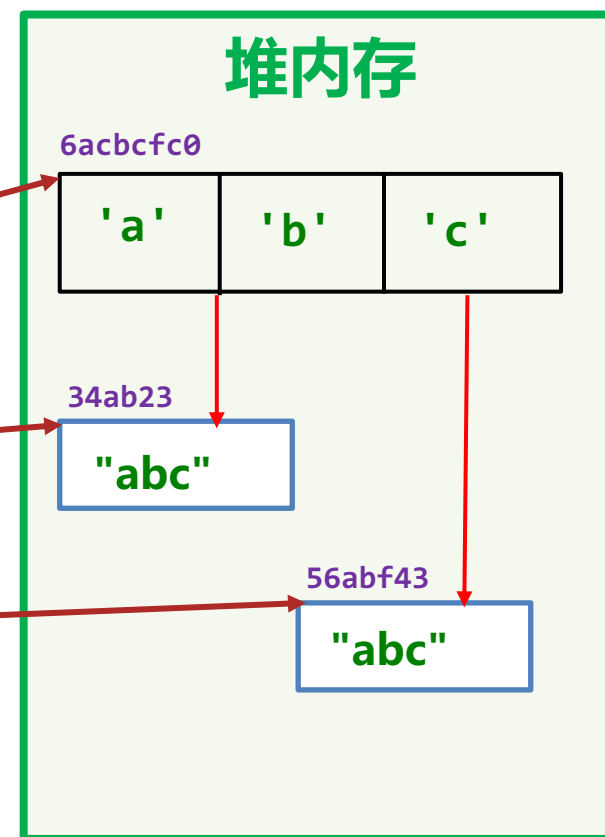
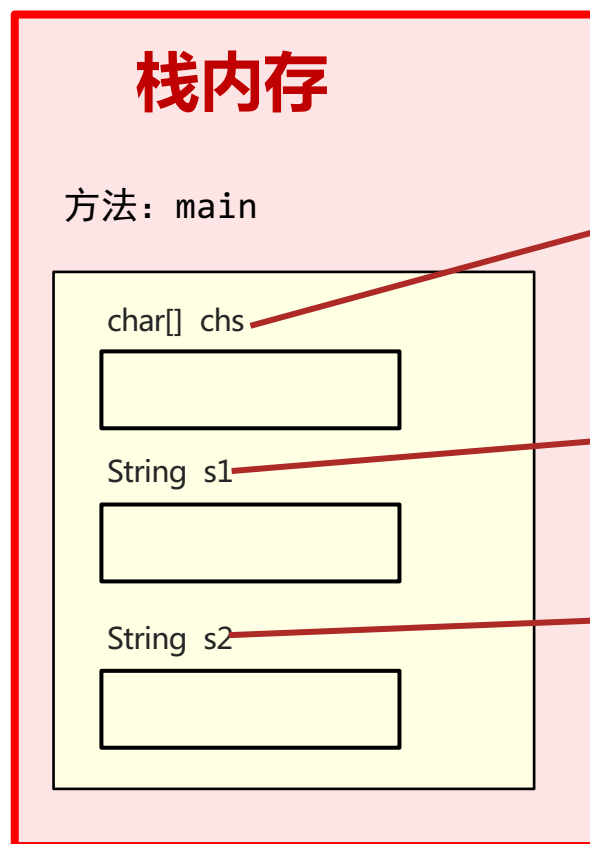
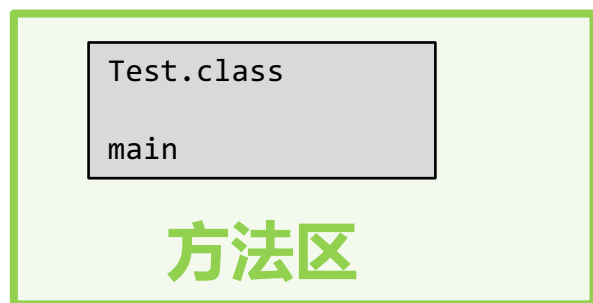
控制台 true



## 通过new构造器得到字符串对象

```
public class Test {  
    public static void main(String[] args) {  
        char[] chs = {'a', 'b', 'c'};  
        String s1 = new String(chs);  
        String s2 = new String(chs);  
        System.out.println(s1 == s2);  
    }  
}
```

控制台 false





# 总结

## 1、字符串对象的特点有哪些？

- 双引号创建的字符串对象，在字符串常量池中存储同一个。
- 通过new 构造器创建的字符串对象，在堆内存中分开存储。



# 目录

## Contents

### ➤ String

- ◆ String类概述
- ◆ String类的常用方法-字符串内容比较
- ◆ String类的常用方法-遍历、替换、截取、分割操作
- ◆ String类案例实战
- ◆ String类创建对象的2种方式
- ◆ String类笔试题

### ➤ ArrayList

## String常见面试题

- 问题：下列代码的运行结果是？

```
public class Test2 {  
    public static void main(String[] args) {  
        String s2 = new String("abc");  
  
        String s1 = "abc";  
        System.out.println(s1 == s2);  
    }  
}
```

false

这句代码实际上创建了两个对象

堆内存

字符串常量池

"abc"

"abc"



## String常见面试题

```
public class Test3 {  
    public static void main(String[] args) {  
        String s1 = "abc";  
        String s2 = "ab";  
        String s3 = s2 + "c";  
        System.out.println(s1 == s3);  
    }  
}
```

false

```
public class Test4 {  
    public static void main(String[] args) {  
        String s1 = "abc";  
        String s2 = "a" + "b" + "c";  
        System.out.println(s1 == s2);  
    }  
}
```

true

Java存在编译优化机制，程序在编译时：“a” + “b” + “c” 会直接转成 “abc”



# 目录

## Contents

### ➤ String

- ◆ String类概述
- ◆ String类创建对象的2种方式
- ◆ String类常见面试题
- ◆ String类常用方法-字符串内容比较
- ◆ String类常用方法-遍历、替换、截取、分割操作
- ◆ String类案例实战

### ➤ ArrayList

## 字符串的内容比较



扫码登录更安全

密码登录 短信登录

itheima

.....

登录

微博登录 支付宝登录

忘记密码 忘记用户名 免费注册

```
public class StringDemoAPI3 {  
    public static void main(String[] args) {  
        String sysLoginName = "itheima";  
  
        Scanner sc = new Scanner(System.in);  
        System.out.println("请您输入您的登录名称");  
        String loginName = sc.next();  
  
        System.out.println(sysLoginName == loginName);  
    }  
}
```

**结论：字符串的内容比较不适合用“==”比较。**

## 字符串的内容比较：

- 推荐使用String类提供的“equals”比较：只关心内容一样即可

方法名	说明
<code>public boolean equals (Object anObject)</code>	将此字符串与指定对象进行比较。只关心字符内容是否一致！
<code>public boolean equalsIgnoreCase (String anotherString)</code>	将此字符串与指定对象进行比较，忽略大小写比较字符串。只关心字符内容是否一致！



## 总结

- 1、如果是字符串比较应该使用使用什么方式进行比较，为什么？
  - 使用String提供的equals方法。
  - 只关心内容一样就返回true。
- 2、开发中什么时候使用==比较数据
  - 基本数据类型比较时使用。



# 目录

## Contents

### ➤ String

- ◆ String类概述
- ◆ String类创建对象的2种方式
- ◆ String类常见面试题
- ◆ String类常用方法-字符串内容比较
- ◆ String类常用方法-遍历、替换、截取、分割操作
- ◆ String类案例实战

### ➤ ArrayList

## String常用API

方法名	说明
<code>public int length ()</code>	返回此字符串的长度
<code>public char charAt (int index)</code>	获取某个索引位置处的字符
<code>public char[] toCharArray () :</code>	将当前字符串转换成字符数组返回
<code>public String substring(int beginIndex, int endIndex)</code>	根据开始和结束索引进行截取，得到新的字符串（包前不包后）
<code>public String substring(int beginIndex)</code>	从传入的索引处截取，截取到末尾，得到新的字符串
<code>public String replace(CharSequence target, CharSequence replacement)</code>	使用新值，将字符串中的旧值替换，得到新的字符串
<code>public String[] split(String regex)</code>	根据传入的规则切割字符串，得到字符串数组返回



# 目录

## Contents

### ➤ String

- ◆ String类概述、不可变原理
- ◆ String创建对象的2种方式
- ◆ String常见面试题
- ◆ String常用方法-字符串内容比较
- ◆ String常用方法-截取、分割、遍历、替换、等
- ◆ String案例操作

### ➤ ArrayList





# 目录

## Contents

➤ String

➤ ArrayList

◆ 集合概述

◆ ArrayList集合快速入门

◆ ArrayList对于泛型的支持

◆ ArrayList常用方法、遍历

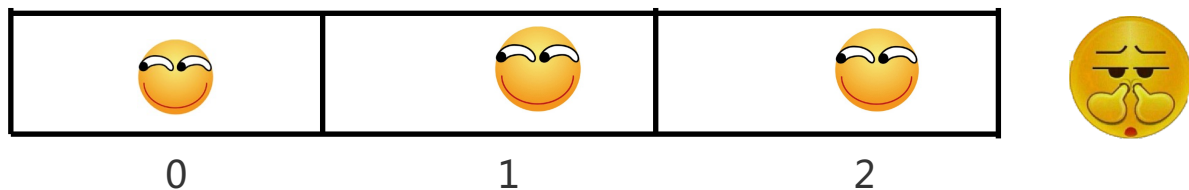
◆ ArrayList集合案例：遍历并删除元素

◆ ArrayList集合案例：存储自定义类型

◆ ArrayList集合案例：元素搜索

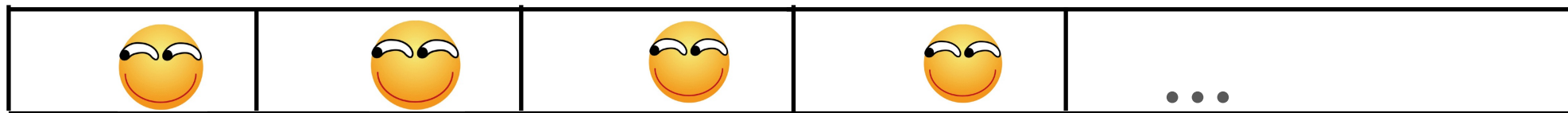
集合是与数组类似，也是一种容器，用于装数据的。

## 数组的特点



- 数组定义完成并启动后，**类型确定、长度固定**。
- **问题**：在个数不能确定，且要进行增删数据操作的时候，数组是不太合适的。

## 集合的特点



- 集合的大小不固定，启动后可以动态变化，类型也可以选择不固定。
- 集合非常适合做元素个数不确定，且要进行增删操作的业务场景。
- 集合还提供了许多丰富、好用的功能，而数组的功能很单一。

## 集合的应用场景

← → ↺ search.51job.com

☐ 全选 已选条件: java(全文) ☆ 收藏 批量申请

1-1.5万/月 广州 | 3-4年经验 | 大专 | 招5人

申请职位

☐ Java开发工程师 (微服务方向) 02-22发布 上海金指软件技术有限公司

1.2-1.8万/月 上海-浦东新区 | 3-4年经验 | 大专 | 招6人 民营企业 | 50-150人

五险一金 通讯补贴 绩效奖金 员工旅游 年终奖金 专业培训 餐饮补贴 ... 互联网/电子商务

☐ Java开发工程师 02-22发布 用友汽车信息科技(上海)股份...

1.2-1.8万/月 北京-朝阳区 | 2年经验 | 本科 | 招3人 民营企业 | 1000-5000人

五险一金 交通补贴 餐饮补贴 出国机会 年终奖金 定期体检 弹性工作 计算机软件

☐ JAVA开发工程师 02-22发布 杭州天迈网络科技有限公司

0.8-1.2万/月 杭州-江干区 | 2年经验 | 本科 | 招5人 民营企业 | 50-150人

五险一金 弹性工作 补充公积金 餐饮补贴 年终奖金 员工旅游 互联网/电子商务

☐ JAVA 数据库开发工程师 02-22发布 成都虚谷伟业科技有限公司

6:34 5G

购物车 编辑 14

全部 (35) 降价 (8) 常买 (0) 分类 ~

优惠券

Java 基础案例教程 黑马程序员 9787115439376

选服务 ~

¥31.32 1 +

Java 自学宝典 黑马程序员 9787302475415

选服务 ~

¥74.24 1 +

Java Web 程序开发入门 传智播客 高教产品研发部著 9787302387... 仅剩5件

选服务 ~

¥25.81 1 +

Java EE 企业级应用开发教程 黑马程序员 9787115461025

选服务 ~

¥28.88 1 +

JD 京东自营 优惠券

全选 合计:¥0.00 去结算 (0)

首页 分类 发现 购物车 35 我的



# 总结

## 1、数组和集合的元素存储的个数问题?

- 数组定义后类型确定，长度固定
- 集合类型可以不固定，大小是可变的。

## 2、数组和集合适合的场景

- 数组适合做数据个数和类型确定的场景
- 集合适合做数据个数不确定，且要做增删元素的场景



# 目录

Contents

➤ **String**

➤ **ArrayList**

◆ 集合概述

◆ **ArrayList集合快速入门**

◆ ArrayList对于泛型的支持

◆ ArrayList常用方法、遍历

◆ ArrayList集合案例：遍历并删除元素

◆ ArrayList集合案例：存储自定义类型的对象

◆ ArrayList集合案例：元素搜索

## ArrayList集合

- ArrayList是集合中的一种，它支持索引。（暂时先学习这个，后期课程会学习整个集合体系）

### ArrayList集合的对象的创建

构造器	说明
<code>public ArrayList()</code>	创建一个空的集合对象

### ArrayList集合的添加元素的方法

方法名	说明
<code>public boolean add(E e)</code>	将指定的元素追加到此集合的末尾
<code>public void add(int index,E element)</code>	在此集合中的指定位置插入指定的元素



# 总结

1、ArrayList类如何创建集合对象的，如何添加元素？

- **public ArrayList()**
- **public boolean add(E e)**
- **public void add(int index,E element)**





# 目录

Contents

➤ **String**

➤ **ArrayList**

◆ 集合概述

◆ ArrayList集合快速入门

◆ **ArrayList对于泛型的支持**

◆ ArrayList常用API、遍历

◆ ArrayList集合案例：遍历并删除元素

◆ ArrayList集合案例：存储自定义类型的对象

◆ ArrayList集合案例：元素搜索

## 泛型概述

- `ArrayList<E>`：其实就是一个泛型类，可以在编译阶段约束集合对象只能操作某种数据类型。

### 举例：

- `ArrayList<String>`：此集合只能操作字符串类型的元素。
- `ArrayList<Integer>`：此集合只能操作整数类型的元素。

**注意：泛型只能支持引用数据类型，不支持基本数据类型。**



# 总结

1、怎么去统一ArrayList集合操作的元素类型，泛型需要注意什么？

- 使用泛型：<数据类型>
- 创建集合对象都应该使用泛型。
- `ArrayList<String> list1 = new ArrayList();`
- 泛型只能支持引用数据类型，不支持基本数据类型。



# 目录

## Contents

➤ **String**

➤ **ArrayList**

◆ 集合概述

◆ ArrayList集合快速入门

◆ ArrayList对于泛型的支持

◆ **ArrayList常用方法、遍历**

◆ ArrayList集合案例：遍历并删除元素

◆ ArrayList集合案例：存储自定义类型

◆ ArrayList集合案例：元素搜索

## ArrayList集合常用方法

方法名称	说明
public E get(int index)	返回指定索引处的元素
public int size()	返回集合中的元素的个数
public E remove(int index)	删除指定索引处的元素，返回被删除的元素
public boolean remove(Object o)	删除指定的元素，返回删除是否成功
public E set(int index,E element)	修改指定索引处的元素，返回被修改的元素



# 目录

Contents

➤ **String**

➤ **ArrayList**

◆ 集合概述

◆ ArrayList集合快速入门

◆ ArrayList对于泛型的支持

◆ ArrayList常用方法、遍历

◆ **ArrayList集合案例：遍历并删除元素**

◆ ArrayList集合案例：存储自定义类型

◆ ArrayList集合案例：元素搜索

## 案例

## 遍历并删除元素值

**需求：**

- 某个班级的考试在系统上进行，成绩大致为：98，77，66，89，79，50，100
- 现在需要先把成绩低于80分以下的数据去掉。

**分析：**

- ① 定义ArrayList集合存储多名学员的成绩。
- ② 遍历集合每个元素，如果元素值低于80分，去掉它。



# 总结

- 1、从集合中遍历元素，并筛选出元素删除它，应该怎么解决？
  - 从集合后面遍历然后删除，可以避免漏掉元素。





# 目录

Contents

➤ **String**

➤ **ArrayList**

◆ 集合概述

◆ ArrayList集合快速入门

◆ ArrayList对于泛型的支持

◆ ArrayList常用方法、遍历

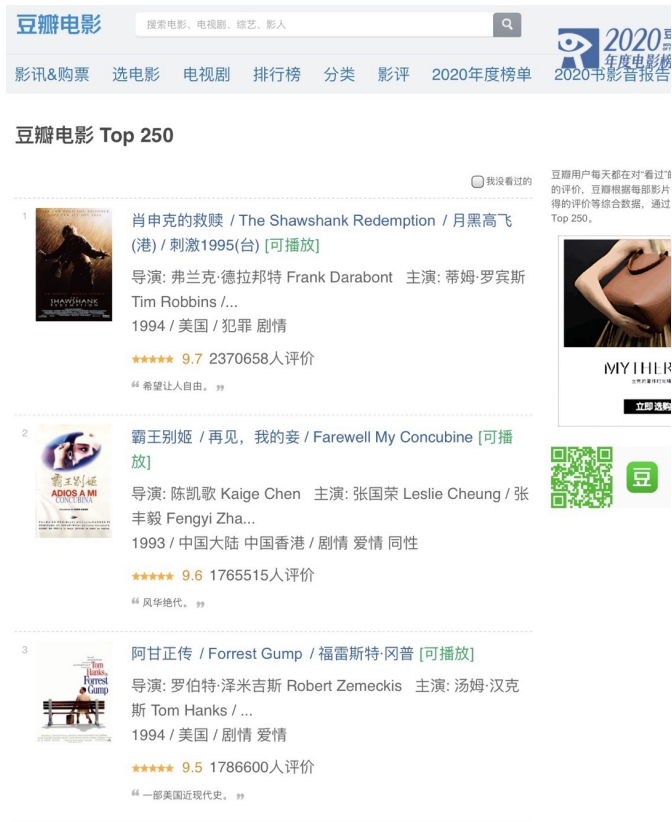
◆ ArrayList集合案例：遍历并删除元素

◆ **ArrayList集合案例：存储自定义类型的对象**

◆ ArrayList集合案例：元素搜索

## 案例

## 影片信息在程序中的表示



## 需求

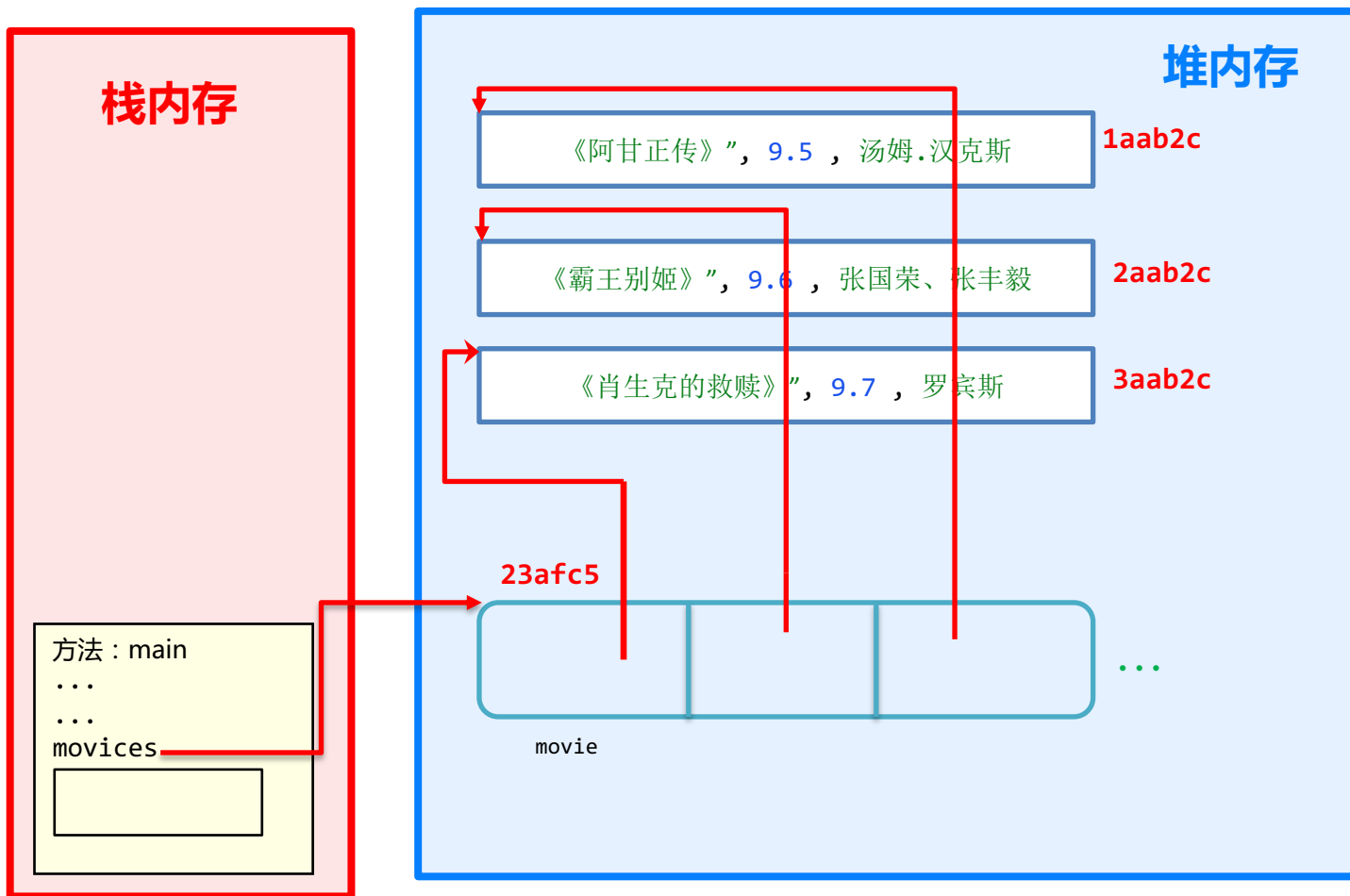
- 某影院系统需要在后台存储上述三部电影，然后依次展示出来。

## 分析

- ①：定义一个电影类，定义一个集合存储电影对象。
- ②：创建3个电影对象，封装相关数据，把3个对象存入到集合中去。
- ③：遍历集合中的3个对象，输出相关信息。

```
public class Movie {  
    private String name;  
    private double score;  
    private String acotr;  
  
    public Movie(String name, double score, String acotr) {  
        this.name = name;  
        this.score = score;  
        this.acotr = acotr;  
    }  
    // ... getter + setter  
}
```

```
public class SystemDemo {  
    public static void main(String[] args) {  
        ArrayList<Movie> movies = new ArrayList<>();  
        movies.add(new Movie("《肖生克的救赎》", 9.7, "罗宾斯"));  
        movies.add(new Movie("《霸王别姬》", 9.6, "张国荣、张丰毅"));  
        movies.add(new Movie("《阿甘正传》", 9.5, "汤姆.汉克斯"));  
        System.out.println(movies);  
  
        for (int i = 0; i < movies.size(); i++) {  
            Movie movie = movies.get(i);  
            System.out.println("片名: " + movie.getName());  
            System.out.println("评分: " + movie.getScore());  
            System.out.println("主演: " + movie.getAcotr());  
        }  
    }  
}
```



**结论：集合中存储的元素并不是对象本身，而是对象的地址。**



# 目录

## Contents

➤ **String**

➤ **ArrayList**

◆ 集合概述

◆ ArrayList集合快速入门

◆ ArrayList对于泛型的支持

◆ ArrayList常用方法、遍历

◆ ArrayList集合案例：遍历并删除元素

◆ ArrayList集合案例：存储自定义类型的对象

◆ ArrayList集合案例：元素搜索

## 案例 学生信息系统的数据搜索

学号	姓名	年龄	班级
20180302	叶孤城	23	护理一班
20180303	东方不败	23	推拿二班
20180304	西门吹雪	26	中药学四班
20180305	梅超风	26	神经科2班

### 需求

- 后台程序需要存储如上学生信息并展示，然后要提供按照学号搜索学生信息的功能。

### 分析

- ① 定义Student类，定义ArrayList集合存储如上学生对象信息，并遍历展示出来。
- ② 提供一个方法，可以接收ArrayList集合，和要搜索的学号，返回搜索到的学生对象信息，并展示。
- ③ 使用死循环，让用户可以不停的搜索。



传智教育旗下高端IT教育品牌