首页 HTML CSS **JAVASCRIPT JQUERY** VUE2 VUE3 **BOOTSTRAP** PYTHON3 PYTHON2 JAVA

Java 教程

Java 教程

Java 简介

Java 开发环境配 置

Java 基础语法

Java 对象和类

Java 基本数据类 型

Java 变量类型

Java 修饰符

Java 运算符

Java 循环结构

Java 条件语句

Java switch case

Java Number & Math 类

Java Character 类

Java String 类

Java StringBuffer

Java 数组

Java 日期时间

Java 正则表达式

Java 方法

Java Stream,

File, IO

Java Scanner 类

Java 异常处理

Java 面向对象

Java 继承

Java

Override/Overload

← Java 集合框架

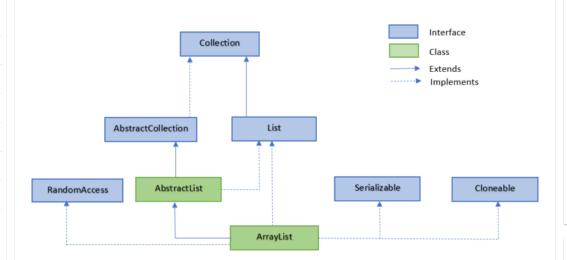
Java LinkedList >

Java ArrayList



ArrayList 类是一个可以动态修改的数组,与普通数组的区别就是它是没有固定大小的限 制,我们可以添加或删除元素。

ArrayList 继承了 AbstractList , 并实现了 List 接口。



ArrayList 类位于 java.util 包中,使用前需要引入它,语法格式如下:

```
import java.util.ArrayList; // 引入 ArrayList 类
ArrayList<E> objectName =new ArrayList<>();
```

E: 泛型数据类型,用于设置 objectName 的数据类型,只能为引用数据类型。 objectName: 对象名。

ArrayList 是一个数组队列,提供了相关的添加、删除、修改、遍历等功能。

添加元素

ArrayList 类提供了很多有用的方法,添加元素到 ArrayList 可以使用 add() 方法:

实例

```
import java.util.ArrayList;
public class RunoobTest {
    public static void main(String[] args) {
        ArrayList<String> sites = new ArrayList<String>();
```

Ⅲ 分类导 航

HTML / CSS

JavaScript

服务端

数据库

移动端

XML 教程

ASP.NET

Web Service

开发工具

网站建设

Advertisement





反馈/建议

Java 抽象类 Java 封装 Java 接口 Java 枚举 Java 包(package) Java 高级教程 Java 数据结构 Java 集合框架 Java ArrayList Java LinkedList Java HashSet Java HashMap Java Iterator Java Object Java 泛型 Java 序列化 Java 网络编程

Java 多态

Java 发送邮件 Java 多线程编程

Java Applet 基础

Java 文档注释

Java 实例

Java 8 新特性

Java MySQL 连 接

Java 9 新特性

Java 测验

```
sites.add("Google");
sites.add("Runoob");
sites.add("Taobao");
sites.add("Weibo");
System.out.println(sites);
}
}
```

以上实例,执行输出结果为:

```
[Google, Runoob, Taobao, Weibo]
```

访问元素

访问 ArrayList 中的元素可以使用 get() 方法:

```
实例
```

```
import java.util.ArrayList;

public class RunoobTest {
    public static void main(String[] args) {
        ArrayList<String> sites = new ArrayList<String>();
        sites.add("Google");
        sites.add("Runoob");
        sites.add("Taobao");
        sites.add("Weibo");
        System.out.println(sites.get(1)); // 访问第二个元素
    }
}
```

注意:数组的索引值从0开始。

以上实例,执行输出结果为:

Runoob

修改元素

如果要修改 ArrayList 中的元素可以使用 set() 方法:

实例

```
import java.util.ArrayList;

public class RunoobTest {
    public static void main(String[] args) {
        ArrayList<String> sites = new ArrayList<String>();
        sites.add("Google");
        sites.add("Runoob");
        sites.add("Taobao");
        sites.add("Weibo");
        sites.set(2, "Wiki"); // 第一个参数为索引位置,第二个为要修改的

        System.out.println(sites);
    }
}
```





以上实例, 执行输出结果为:

```
[Google, Runoob, Wiki, Weibo]
```

删除元素

如果要删除 ArrayList 中的元素可以使用 remove() 方法:

```
实例
```

```
import java.util.ArrayList;

public class RunoobTest {
    public static void main(String[] args) {
        ArrayList<String> sites = new ArrayList<String>();
        sites.add("Google");
        sites.add("Runoob");
        sites.add("Taobao");
        sites.add("Weibo");
        sites.remove(3); // 删除第四个元素
        System.out.println(sites);
    }
}
```

以上实例,执行输出结果为:

```
[Google, Runoob, Taobao]
```

计算大小

如果要计算 ArrayList 中的元素数量可以使用 size() 方法:

实例

```
import java.util.ArrayList;

public class RunoobTest {
   public static void main(String[] args) {
        ArrayList<String> sites = new ArrayList<String>();
        sites.add("Google");
        sites.add("Runoob");
        sites.add("Taobao");
        sites.add("Weibo");
        System.out.println(sites.size());
    }
}
```

以上实例,执行输出结果为:

4

迭代数组列表

我们可以使用 for 来迭代数组列表中的元素:





믦

实例

```
import java.util.ArrayList;

public class RunoobTest {
    public static void main(String[] args) {
        ArrayList<String> sites = new ArrayList<String>();
        sites.add("Google");
        sites.add("Runoob");
        sites.add("Taobao");
        sites.add("Weibo");
        for (int i = 0; i < sites.size(); i++) {
            System.out.println(sites.get(i));
        }
    }
}</pre>
```

以上实例,执行输出结果为:

```
Google
Runoob
Taobao
Weibo
```

也可以使用 for-each 来迭代元素:

实例

```
import java.util.ArrayList;

public class RunoobTest {
    public static void main(String[] args) {
        ArrayList<String> sites = new ArrayList<String>();
        sites.add("Google");
        sites.add("Runoob");
        sites.add("Taobao");
        sites.add("Weibo");
        for (String i : sites) {
             System.out.println(i);
        }
    }
}
```

以上实例,执行输出结果为:

```
Google
Runoob
Taobao
Weibo
```

其他的引用类型

ArrayList 中的元素实际上是对象,在以上实例中,数组列表元素都是字符串 String 类型。





如果我们要存储其他类型,而 <E> 只能为引用数据类型,这时我们就需要使用到基本类型的包装类。

基本类型对应的包装类表如下:

基本类型	引用类型
boolean	Boolean
byte	Byte
short	Short
int	Integer
long	Long
float	Float
double	Double
char	Character

此外,BigInteger、BigDecimal 用于高精度的运算,BigInteger 支持任意精度的整数,也是引用类型,但它们没有相对应的基本类型。

```
ArrayList<Integer> li=new Arraylist<>(); // 存放整数元素
ArrayList<Character> li=new Arraylist<>(); // 存放字符元素
```

以下实例使用 ArrayList 存储数字(使用 Integer 类型):

实例

以上实例, 执行输出结果为:

```
10
15
20
25
```





ArrayList 排序

Collections 类也是一个非常有用的类,位于 java.util 包中,提供的 sort() 方法可以对字符或数字列表进行排序。

以下实例对字母进行排序:

```
实例
import java.util.ArrayList;
import java.util.Collections; // 引入 Collections 类
public class RunoobTest {
    public static void main(String[] args) {
        ArrayList<String> sites = new ArrayList<String>();
        sites.add("Taobao");
        sites.add("Wiki");
        sites.add("Runoob");
        sites.add("Weibo");
        sites.add("Google");
        Collections.sort(sites); // 字母排序
        for (String i : sites) {
            System.out.println(i);
        }
    }
}
```

以上实例, 执行输出结果为:

```
Google
Runoob
Taobao
Weibo
Wiki
```

以下实例对数字进行排序:

实例

```
import java.util.ArrayList;
import java.util.Collections; // 引入 Collections 类

public class RunoobTest {
    public static void main(String[] args) {
        ArrayList<Integer> myNumbers = new ArrayList<Integer>();
        myNumbers.add(33);
        myNumbers.add(15);
        myNumbers.add(20);
        myNumbers.add(34);
        myNumbers.add(8);
        myNumbers.add(12);

        Collections.sort(myNumbers); // 数字排序

        for (int i : myNumbers) {
            System.out.println(i);
```







```
}
}
```

以上实例,执行输出结果为:

```
8
12
15
20
33
34
```

Java ArrayList 方法

Java ArrayList 常用方法列表如下:

方法	描述
<u>add()</u>	将元素插入到指定位置的 arraylist 中
addAll()	添加集合中的所有元素到 arraylist 中
<u>clear()</u>	删除 arraylist 中的所有元素
clone()	复制一份 arraylist
contains()	判断元素是否在 arraylist
g <u>et()</u>	通过索引值获取 arraylist 中的元素
indexOf()	返回 arraylist 中元素的索引值
removeAll()	删除存在于指定集合中的 arraylist 里的所有元素
remove()	删除 arraylist 里的单个元素
size()	返回 arraylist 里元素数量
isEmpty()	判断 arraylist 是否为空
subList()	截取部分 arraylist 的元素
<u>set()</u>	替换 arraylist 中指定索引的元素
sort()	对 arraylist 元素进行排序
toArray()	将 arraylist 转换为数组
toString()	将 arraylist 转换为字符串





ensureCapacity()	设置指定容量大小的 arraylist
lastIndexOf()	返回指定元素在 arraylist 中最后一次出现的位置
<u>retainAll()</u>	保留 arraylist 中在指定集合中也存在的那些元素
containsAll()	查看 arraylist 是否包含指定集合中的所有元素
trimToSize()	将 arraylist 中的容量调整为数组中的元素个数
removeRange()	删除 arraylist 中指定索引之间存在的元素
replaceAll()	将给定的操作内容替换掉数组中每一个元素
removelf()	删除所有满足特定条件的 arraylist 元素
forEach()	遍历 arraylist 中每一个元素并执行特定操作

更多 API 方法可以查看: https://www.runoob.com/manual/jdk11api/java.base/java/util/Arra yList.html



<u></u> Java 集合框架

← Java 集合框架

Java LinkedList →

② 点我分享笔记

在线实例	字符集&工	最新更新
· HTML 实例	具	· PHP
· CSS 实例	· HTML 字符 集设置	array_key_l · PHP
· JavaScript 实例	· HTML	array_key_f
	ASCII 字符集	· 9.2 Verilog
· Ajax 实例	· HTML ISO-	可
· jQuery 实例	8859-1	· 9.1 Verilog
· XML 实例	· HTML 实体	逻
	符号	· 8.5 Verilog
· Java 实例	· HTML 拾色	
	器	· 8.4 Verilog ACC
	· JSON 格式 化工具	· 8.3 Verilog TF

站点信息

- ·意见反馈
- · 免责声明
- · 关于我们
- · 文章归档

关注微信



Copyright © 2013-2021 菜乌 runoob.com All Rights Reserv 号:闽ICP备15012807号







