**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<>();　 // 初始化

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>();

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，第二个为要修改的值

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：

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));

}

}

}

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

import java.util.ArrayList;

public class RunoobTest {

public static void main(String[] args) {

ArrayList<Integer> myNumbers = new ArrayList<Integer>();

myNumbers.add(10);

myNumbers.add(15);

myNumbers.add(20);

myNumbers.add(25);

for (int i : myNumbers) {

System.out.println(i);

}

}

}

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