2016-09-27日工作日报

1. **已完成的工作内容**

(1)正则表达式的作用：验证字符串是否与指定规则匹配

本质：字符串规则，规则描述;(2)常用正则表达式

相关代码:

public class Test5 {

public static void main(String[] args) {

// 定义一个正则表达式字符串

String expression = "-?\d{3}";

// 定义一个需要验证的字符串

String str = "-a222222";

// 套路用法

Pattern p = Pattern.compile(expression);

Matcher m = p.matcher(str);

// 请问 str符合 expression的规则吗？

// 表示str中的字符串必须完全匹配expression

boolean bln1 = m.matches();

// 表示str中从开始位置字符到找到能匹配expression的字符串或者子字符串就可以了

boolean bln2 = m.lookingAt();

// 表示匹配expression成功一次后，是否还有下一个组字符串与expression匹配

boolean bln3 = m.find();

// matches()方法返回true的话，那么就表示str负责expression的规则

// 反之则不符合

System.out.println("bln1="+bln1);

System.out.println("bln2="+bln2);

System.out.println("bln3="+bln3);

}

}-

(2)group()方法的使用

import java.util.regex.Matcher;

import java.util.regex.Pattern;

/\*\*

- Created by chenyan on 2016/9/27.

\*/

public class Test6 {

public static void main(String[] args) {

String expression = "A(B(C))D";

Pattern p = Pattern.compile(expression);

Matcher m = p.matcher("ABCD");

System.out.println("matches:" + m.matches());

System.out.println("groupCount:" + m.groupCount());

System.out.println("group():" + m.group());

System.out.println("group(1):" + m.group(1));

System.out.println("group(2):" + m.group(2));

}

}

(3)list

(4)ArrayList 可变数组

import java.util.ArrayList;

import java.util.Iterator;

import java.util.List;

/\*\*

- Created by chenyan on 2016/9/27.

\*/

public class ListDemo {

public static void main(String[] args) {

// 创建

List<String> list = new ArrayList<String>();

// 添加

list.add("毛泽东");

list.add("周恩来");

list.add("刘少奇");

list.add(2,"朱德");

// list.add("林彪");

// 修改

// list.set(0,"林彪");

// 删除

// list.remove(0);

list.remove("毛泽东");

// 查询遍历

for(int index = 0;index< list.size();index ++) {

String name = list.get(index);

System.out.println(name);

}

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

for (String name : list) {

System.out.println(name);

}

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

Iterator<String> it = list.iterator();

while(it.hasNext()) {

String name = it.next();

System.out.println(name);

}

}

}

(4) set

- Set接口也是Collection接口的子接口，但是与Collection或List接口不同的是，Set接口中不能加入重复的元素。

- Set接口的定义： public interface Set<E> extends Collection<E>

- Set接口的主要方法与Collection是一致的

- Set接口的实例无法像List接口那样进行双向输出

- Set接口的常用子类

- HashSet (散列的存放) :

HashSet是Set接口的一个实现类，

主要的特点是：

里面不能存放重复元素，

采用散列的存储方式，所以是没有顺序(插入顺序)的。

import java.util.HashSet;

import java.util.Iterator;

import java.util.LinkedList;

/\*\*

- Created by chenyan on 2016/9/27.

\*/

public class HashSetDemo {

public static void main(String[] args) {

// 创建

HashSet<String> list = new HashSet<String>();

// 添加

list.add("毛泽东");

list.add("周恩来");

list.add("刘少奇");

// list.add("林彪");

// 修改

// list.set(0,"林彪");

// 删除

// list.remove(0);

// list.remove("毛泽东");

// 查询遍历

for (String name : list) {

System.out.println(name);

}

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

Iterator<String> it = list.iterator();

while(it.hasNext()) {

String name = it.next();

System.out.println(name);

}

}