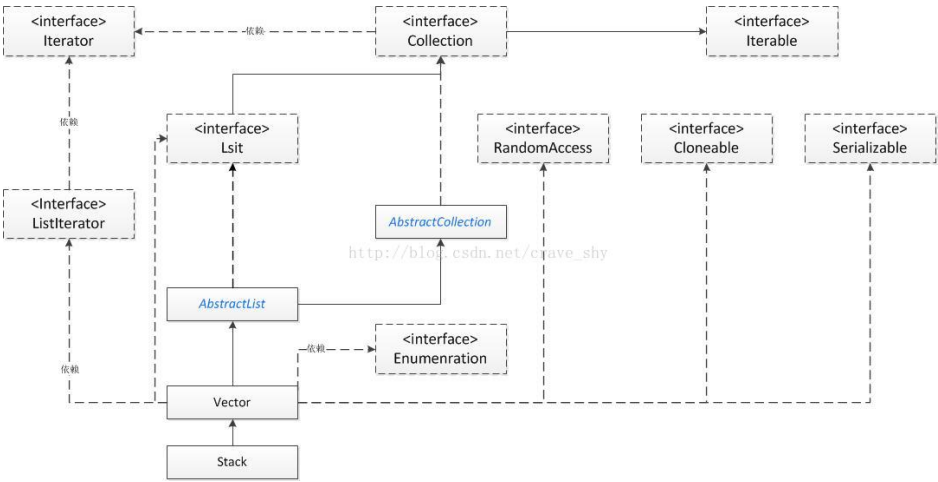


java\_集合体系之Stack详解、源码及示例——06

原创 2013年12月23日 15:25:19 2995 0 7

java\_集合体系之Stack详解、源码及示例——06

一：Stack结构图



- 简单说明：
- 1、上图中虚线且无依赖字样、说明是直接实现的接口
  - 2、虚线但是有依赖字样、说明此类依赖与接口、但不是直接实现接口
  - 3、实线是继承关系、类继承类、接口继承接口

二：Stack类简介：

Stack（栈）、见名知意、是以栈的形式来存储数据、其特点是先进后出（FILO）、同时Stack继承与Vector、所以如果把Stack内部的实现也是通过动态数组来实现的、而不是其他的数据结构。在List体系中有个LinkedList是以双向链表的数据结构来存储数据的、所以使用LinkedList完全可以达到Stack的效果、但是Stack不能作为双向链表来使用、并且LinkedList不是线程安全的、而Stack是线程安全的、对于Stack、连他父类Vector都不被推荐使用了、Stack的命运也逃不掉被雪藏、他和其父类的存在意义只是为Collection兼容性做点贡献。

三：Stack API

1、构造方法

Stack()

2、一般方法



Oscar Chen (http://blog....)

+ 关注

(http://blog.csdn.net/chenghuaying)

原创 182 粉丝 14 喜欢 0

- > CentOS 集群机器之间ssh免密 (/crave\_shy/article/details/72964997)
- > JVM-内存管理-运行时数据区域 (/crave\_shy/article/details/56675052)
- > JVM-Blog目录 (/crave\_shy/article/details/56675032)
- > JVM-为什么要学JVM (/crave\_shy/article/details/56673439)

更多文章 (http://blog.csdn.net/chenghuaying)

在线课程



(http://edu.csdn.net/huiyiCourse/series\_detail?utm\_source=blog7)

【直播】机器学习&数据挖掘7周实训--韦玮

(http://edu.csdn.net/huiyiCourse/series\_detail/54?utm\_source=blog7)



(http://edu.csdn.net/combo/detail/471?utm\_source=blog7)

【套餐】系统集成项目管理工程师顺利通关--徐朋

(http://edu.csdn.net/combo/detail/471?utm\_source=blog7)

```
        boolean    empty()
synchronized E    peek()
synchronized E    pop()
        E          push(E object)
synchronized int   search(Object o)
```

## 四：Stack源码分析

```
package com.chy.collection.core;

public class Stack<E> extends Vector<E> {
    /** 默认无参构造方法*/
    public Stack() {
    }

    /** 将一个元素压入Stack中*/
    public E push(E item) {
        addElement(item);
        return item;
    }

    /** 将栈顶元素弹出*/
    public synchronized E pop() {
        E    obj;
        int   len = size();
        obj = peek();
        removeElementAt(len - 1);

        return obj;
    }

    /** 查看栈顶元素*/
    public synchronized E peek() {
        int   len = size();

        if (len == 0)
            throw new EmptyStackException();
        return elementAt(len - 1);
    }

    /** 查看是否为空栈*/
    public boolean empty() {
        return size() == 0;
    }

    /** 查找指定对象o在栈中的索引*/
    public synchronized int search(Object o) {
        int i = lastIndexOf(o);

        if (i >= 0) {
            return size() - i;
        }
        return -1;
    }

    /** 默认版本号 */
    private static final long serialVersionUID = 1224463164541339165L;
}
```

总结：Stack继承与Vector、它具有Vector所具有的所有功能、同时从Stack源码中可以看出、作为Stack结构、它提供了用于操作栈结构的方法、压入、弹出、查看、搜索、这几个方法内部的实现都是通过调用父类Vector的方法来实现的。

## 五：Stack示例

1、遍历方式：Stack遍历方式与Vector完全相同、可以参考前面一章：Vector的相关内容、这里不再赘述。

2、示例：

```

package com.chy.collection.example;

import java.util.Stack;

public class StackTest {

    /**
     * 这里仅仅测试Stack源码中提供的方法、而不再测试Stack从Vector中继承的方法、所以示例也很简单
     */
    private static void testStack(){
        //初始化一个含有字符“abcde”的Stack
        Stack<String> stack = new Stack<String>();
        stack.push("a");
        stack.push("b");
        stack.push("c");
        stack.push("d");
        stack.push("e");

        //查看Stack中栈顶元素
        System.out.println("the top element of stack : " + stack.peek());
        System.out.println(stack);
        //查看并弹出Stack栈顶元素
        System.out.println("the top element of stack : " + stack.pop());
        System.out.println(stack);
        //查看Stack是否包含“e”
        System.out.println("the index of 'e' in stack : " +stack.search("e"));
        //查看Stack是否为空
        System.out.println("is the stack empty?  " + stack.empty());
    }

    public static void main(String[] args) {
        testStack();
    }
}

```

## 总结：

对Stack、同其父类一样、早已不推荐使用、这里也只是针对Stack源码中提供的方法进行了分析、如果真要使用栈这种结构来实现数据存储、推荐使用Deque () 接口及其实现提供了LIFO堆栈操作的更完整和更一致的 set，应该优先使用此 set，而非此类。权当了解！


更多内容：[java\\_集合体系之总体目录——00](http://blog.csdn.net/crave_shy/article/details/1741671)  
([http://blog.csdn.net/crave\\_shy/article/details/1741671](http://blog.csdn.net/crave_shy/article/details/1741671))

版权声明：本文为博主原创文章，未经博主允许不得转载。



标签：Stack (<http://so.csdn.net/so/search/s.do?q=Stack&t=blog>) /  
 java集合 (<http://so.csdn.net/so/search/s.do?q=java集合&t=blog>) /  
 Collection框架图 (<http://so.csdn.net/so/search/s.do?q=Collection框架图&t=blog>) /  
 Vector (<http://so.csdn.net/so/search/s.do?q=Vector&t=blog>) /  
 Enumeration (<http://so.csdn.net/so/search/s.do?q=Enumeration&t=blog>) /

0条评论

 qq\_36596145 ([http://my.csdn.net/qq\\_36596145](http://my.csdn.net/qq_36596145))  
 ([http://my.csdn.net/qq\\_36596145](http://my.csdn.net/qq_36596145))



发表评论

暂无评论

---

## 相关文章推荐

### Java中的堆栈API——Stack (/gaohuanjie/article/details/45075261)

Java中的堆栈API——Stack



wangshuxuncom 2015-04-16 15:18 1677

---

### Java集合之Stack 源码分析 (/as02446418/article/details/47123581)

1.简介栈是数据结构中一种很重要的数据结构类型，因为栈的后进先出功能是实际的开发中有很多的应用场景。  
Java API中提供了栈 ( Stack)的实现，简单使用如下所示 import java.util...



as02446418 2015-07-29 10:43 732

---

### 恶补java(十一)-----Stack类的使用 (/android\_jiangjun/article/details/39183891)

package com.gc.Stack; /\*\* \* java中stack的使用方法.堆栈是一种"后进先出"(LIFO)的数据结构,只能在一端进行插入(称为"压栈")或删除(称为"出栈")数据的操...



gc\_gongchao 2014-09-10 17:31 3112

---

### java数据结构与算法之栈（Stack）设计与实现 (/javazejian/article/details/53362993)

【版权申明】转载请注明出处（请尊重原创，博主保留追究权）  
<http://blog.csdn.net/javazejian/article/details/53362993> 出自【zejia...



javazejian 2016-11-28 12:27 4998

---

### java数据结构与算法之双链表设计与实现 (/javazejian/article/details/53047590)

转载请注明出处（万分感谢！）：<http://blog.csdn.net/javazejian/article/details/53047590> 出自【zejian的博客】 关联文章j...



javazejian 2016-11-06 22:37 4489

---

### java数据结构与算法之顺序表与链表深入分析 (/javazejian/article/details/52953190)

开篇直奔主题，无论是顺序表还是链表，它们都是线性表的一种，用比较官方的话来讲，线性表是其组成元素间具有线性关系的一种线性结构，而我们恰好可以采用顺序存储和链式存储结构来表示线性表。接下来将从以下几点...



javazejian 2016-11-05 16:24 8028

---

### Java集合的Stack、Queue、Map的遍历 (/ocean1010/article/details/7628042)

一、Map的遍历 import java.util.HashMap; import java.util.Iterator; import java.util.Map; ...



ocean1010 2012-06-03 15:06 4859

---

### Java集合的Stack、Queue、Map的遍历 (/lv18092081172/article/details/71513121)

转载自：<http://lavasoft.blog.51cto.com/62575/181781/> 在集合操作中，常常离不开对集合的遍历，对集合遍历一般来说一个f...



lv18092081172 2017-05-09 23:45 146

## Java集合的Stack、Queue、Map的遍历 (/w46434137/article/details/9005422)

一、Map的遍历 import java.util.HashMap; import java.util.Iterator; import java.util.Map; /\*\* \* Map的...



w46434137 2013-06-01 21:53 622

## Java遍历List的几种方法及其性能比较 (/yasi\_xi/article/details/24797807)

参考：http://blog.csdn.net/player26/article/details/3955906 http://my.oschina.net/trydofor/blog/99...



yasi\_xi 2014-04-30 16:06 1826

## Java\_io体系之PipedWriter、PipedReader简介、走进源码及示例——14 (http://810364804.iteye.com/blog/1992802)

Java\_io体系之PipedWriter、PipedReader简介、走进源码及示例——14 ——管道字符输出流、必须建立在管道输入流之上、所以先介绍管道字符输出流。可以先看示例或者总结、总结写的有点Q、不喜可无视、有误的地方指出则不胜感激。一：PipedWriter 1、类功能简介：管道字符输出流、用于将当前线程的指定字符写入到与此线程对应的管道字符输入流



810364804 2013-12-08 18:50 57

## java\_集合体系之:LinkedList详解、源码及示例——04 (/crave\_shy/article/details/17440835)

摘要：本文通过对LinkedList内部存储数据的结构、LinkedList的结构图、示例、源码、多方面深入分析LinkedList的特性和使用方法。



chenghuaying 2013-12-20 15:11 6349



目录



喜欢

## 常见异常解析 (http://fuyou0104.iteye.com/blog/1174579)

ConcurrentHashMap与CopyOnWriteArrayList比较。博客分类：Java ConcurrentHashMap  
ConcurrentHashMap引入了Segment，每个Segment又是一个hashtable，相当于是两级Hash表，然后锁是在Segment一级进行的，提高了并发性。缺点是对整个集合进行操作的方法如size()或isEmpty()的实现很困难，基本无法得到精准的数据。Segment的read不加锁，只有在读到null的情况(一般不会有null的，只有在其他线程操作Map的时候，所以就用锁来等他操作完)下调用



fuyou0104 2011-09-18 16:29 4162

## java\_集合体系之HashMap详解、源码及示例——09 (/crave\_shy/article/details/17552679)

摘要：本文通过HashMap的结构图分析HashMap所具有的特性、通过源码深入了解HashMap实现原理、使用方法、通过实例加深对HashMap的应用的理解。篇幅较长、慎入！



chenghuaying 2013-12-25 14:54 2541

## java\_集合体系之ArrayList详解、源码及示例——03 (http://810364804.iteye.com/blog/1992789)

java\_集合体系之ArrayList详解、源码及示例——03 一：ArrayList结构图 <img

src="http://img.blog.csdn.net/20131220102938781?

watermark/2/text/aHR0cDovL2Js2cuY3Nkbi5uZXQvY3JhdmVfc2h5/font/5a6L5L2T/fontsize/400/fill/10JBQkFCMA==/dissolve/



810364804 2013-12-20 10:56 163

## java\_集合体系之总体目录——00 (/crave\_shy/article/details/17416791)

摘要：java集合系列目录、不断更新中、、、水平有限、总有不足、误解的地方、请多包涵、也希望多提意见、多多讨论 ^\_^



chenghuaying 2013-12-19 15:41 👁 3210

---

## Android界面特殊全汇总 (http://yuanlanjun.iteye.com/blog/1616453)

(一) Activity 页面切换的效果 Android 2.0 之后有了 overridePendingTransition ( ) , 其中里面两个参数, 一个是前一个 activity 的退出, 另一个 activity 的进入, Java 代码 1. @Override public void onCreate(Bundle savedInstanceState) { 2. super.onCreate(savedInstanceState); 3. 4



yuanlanjun 2012-04-04 11:12 👁 1483

---

## java\_集合体系之Vector详解、源码及示例——05 (/crave\_shy/article/details/17504279)

摘要：本文通过对Vector的结构图中涉及到的类、接口来说明Vector的特性、通过源码来深入了解Vector各种功能的实现原理、通过示例加深对Vector的理解。



chenghuaying 2013-12-23 14:40 👁 2129

---

## Java\_io体系之OutputStreamWriter、InputStreamReader简介、走进源码及示例——17 (http://810364804.iteye.com/blog/1992797)

Java\_io体系之OutputStreamWriter、InputStreamReader简介、走进源码及示例——17 一：  
OutputStreamWriter 1、类功能简介：输入字符转换流、是输入字节流转向输入字符流的桥梁、用于将输入字节流转换成输入



810364804 2013-12-10 09:51 👁 60

---

## 一个程序员的Java和C++学习之路(整理) (/xugangwen/article/details/44811783)

转载：http://blog.csdn.net/ajian005/article/details/8003655 Java学习之路 一直有这么个想法，列一下我个人认为在学习和使用J...



xugangwen 2015-04-01 19:30 👁 16374

---