

## CS61B sp2018笔记 | Exceptions, Iterators, Iterables

2018年08月31日 08:13:36 隐秀\_ 阅读数: 42 更多

### 1. Throwing and catching

#### 1.1 Throwing Exceptions

在程序运行过程中，可能会遇到某些错误需要终止，这时程序就会抛出一个异常。

让我们看看一个常见的情况——`IndexOutOfBoundsException` 异常。下面的程序向`ArrayMap`中插入value为5，key为“hello”的一组数据，试图获取key为“yolp”的value。

```
1 public static void main (String[] args) {
2     ArrayMap<String, Integer> am = new ArrayMap<String, Integer>();
3     am.put("hello", 5);
4     System.out.println(am.get("yolp"));
5 }
```

运行这个程序时，发现这个key值根本不存在，然后造成混乱，结果打印了一些错误信息：

```
1 $ java ExceptionDemo
2 Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: -1
3 at ArrayMap.get(ArrayMap.java:38)
4 at ExceptionDemo.main(ExceptionDemo.java:6)
```

这是隐式的异常，是由Java本身抛出的错误。这个错误告诉我们 `ArrayIndexOutOfBoundsException` 错误造成了程序混乱，但是它并没有明确那个数据项造成了这个错误，对用户来讲并没有给出很多有价值的提示。

不过，我们可以使用 `throw` 关键字，抛出自己的异常信息。比如说，我们可以重新实现`get`方法：

```
1 public V get(K key) {
2     int location = findKey(key);
3     if(location < 0) {
4         throw new IllegalArgumentException("Key " + key + " does not exist in map.");
5     }
6     return values[findKey(key)];
7 }
```

这时候再运行程序，我们就可以清楚的看到出错的位置：

```
1 $java ExceptionDemo
2 Exception in thread "main" java.lang.IllegalArgumentException: Key yolp does not exist in map.
3 at ArrayMap.get(ArrayMap.java:40)
4 at ExceptionDemo.main(ExceptionDemo.java:6)
```

#### 1.2 Catching Exceptions

我们已经知道，当程序运行出错时会抛出异常，那么当抛出异常时我们还能做些什么呢？考虑一下下面这些异常情况：

- 你试图使用3831240亿个字节的内存
- 你试图把一个对象转换成dog，但这个对象的动态类型不是dog
- 你试图调用一个指向null的引用类型的方法
- 你试图访问数组-1位置的元素

我们进行如下操作：

```
1 Object o = "mulchor";
2 Planet x = (Planet) o;
```

```

1 Exception in thread "main" java.lang.ClassCastException:
2 java.lang.String cannot be cast to Planet

```

接着，我们用显示的异常处理提供错误信息：

```

1 public static void main(String[] args) {
2     System.out.println("ayyy lmao");
3     throw new RuntimeException("For no reason.");
4 }

```

这种情况下运行程序：

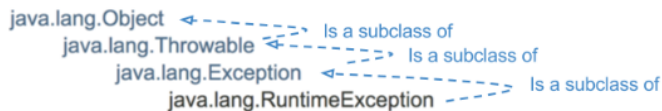
```

1 $ java Alien
2 ayyy lmao
3 Exception in thread "main" java.lang.RuntimeException: For no reason.
4 at Alien.main(Alien.java:4)

```

于是我们发现，即使程序中没有错误，我们也可以实例化 `RuntimeException` 类来抛出异常

## Class RuntimeException



Constructors	
Modifier	Constructor and Description
	<code>RuntimeException()</code> Constructs a new runtime exception with null as its detail message.
	<code>RuntimeException(String message)</code> Constructs a new runtime exception with the specified detail message.

[https://blog.csdn.net/qq\\_40950957](https://blog.csdn.net/qq_40950957)

更为常规的做法是，使用 `try` 和 `break` 方法捕捉可能发生异常的代码，考虑下面的例子：

```

1 Dog d = new Dog("Lucy", "Retriever", 80);
2 d.becomeAngry();
3
4 try {
5     d.receivePat();
6 } catch (Exception e) {
7     System.out.println("Tried to pat: " + e);
8 }
9 System.out.println(d);

```

程序的结果可能会是这样：

```

1 $ java ExceptionDemo
2 Tried to pat: java.lang.RuntimeException: grrr... snarl snarl
3 Lucy is a displeased Retriever weighing 80.0 standard lb units.

```

## 1.3 The Philosophy Of Exceptions

异常的设计不仅仅是为了捕获程序中的错误，提供给用户更准确的信息。更重要的是，它能把处理错误的程序彼此分开。考虑一下下面的情况：

```

1 func readFile: {
2     open the file;
3     determine its size;
4     allocate that much memory;
5     read the file into memory;

```

```
6     close the file;
7 }
```

这是一些对文件操作的程序，每个部分都可能导致错误，没有 `try-catch` 的异常处理手段，我们可能写出这样的代码：

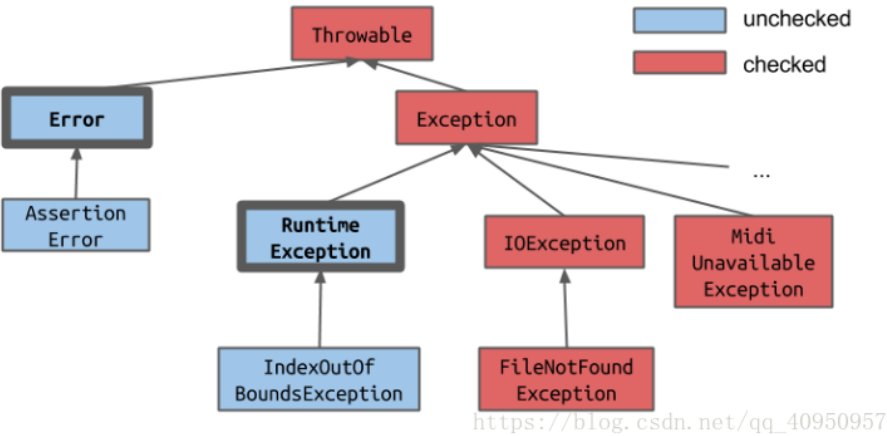
```
1 func readFile: {
2     open the file;
3     if (theFileIsOpen) {
4         determine its size;
5         if (gotTheFileLength) {
6             allocate that much memory;
7         } else {
8             return error("fileLengthError");
9         }
10        if (gotEnoughMemory) {
11            read the file into memory;
12            if (readFailed) {
13                return error("readError");
14            }
15        ...
16    } else {
17        return error("memoryError");
18    }
19 } else {
20     return error("fileOpenError")
21 }
22 }
```

然而，用异常处理的手段代码就变得简洁多了：

```
1 func readFile: {
2     try {
3         open the file;
4         determine its size;
5         allocate that much memory;
6         read the file into memory;
7         close the file;
8     } catch (fileOpenFailed) {
9         doSomething;
10    } catch (sizeDeterminationFailed) {
11        doSomething;
12    } catch (memoryAllocationFailed) {
13        doSomething;
14    } catch (readFailed) {
15        doSomething;
16    } catch (fileCloseFailed) {
17        doSomething;
18    }
19 }
```

## 2. Checked vs Unchecked Exceptions

上面谈论到的异常都是运行时异常，我们可以称之为 **“unchecked” exceptions**；可有些错误简直连编译器都看不过去了，编译时期就会报错，我们称之为 **“checked” exceptions**。



3. Iterators and Iterables

3.1 Iteration

我们知道Java允许我们这样来遍历一个链表：

```
1 List<Integer> friends =
2   new ArrayList<Integer>();
3 friends.add(5);
4 friends.add(23);
5 friends.add(42);
6 for (int x : friends) {
7     System.out.println(x);
8 }
```

我们也可以使用 `iterator` 来遍历，先在 `List.java` 中定义：

```
1 public Iterator<E> iterator();
```

然后我们就可以使用它来遍历：

```
1 List<Integer> friends = new ArrayList<Integer>();
2 ...
3 Iterator<Integer> seer = friends.iterator();
4
5 while (seer.hasNext()) {
6     System.out.println(seer.next());
7 }
```

股市第三次机会来了！不看就亏大了！

唐煌投资 · 爍爍

想对作者说点什么？

我来说一句

伯克利计算机低年级核心课程之CS61A-SICP

8775

<http://bbs.eol.cn/forum.php?mod=viewthread&extra=page%3D1&tid=2391800> 给想... 来自: [fengjiexyb](#)的专栏

CS61B Homework5作业回顾（附代码）

408

HW5一开始做不出来的同学可以继续先看Lecture，Encapsulated List那一节老师会回... 来自: [everest115](#)的博客

CS61B sp2018笔记 | Generics and Autoboxing

80

1. Automatic Conversions 1.1 Autoboxing and Unboxing Java中的泛型用到了&lt;code>it... 来自: [隐秀\\_](#)



完全理解 Python 迭代对象、迭代器、生成器

本文源自RQ作者的一篇博文，原文是Iterables vs. Iterators vs. Generators，俺写的这... 来自：python\_tty的专栏

(原创笔记) 加州伯克利大学CS61b数据结构 (Java描述) 一：对象

OOP(object-oriented programming): object: a repository of data; class: type of object... 来自：Emacsor的博客

CS61B Homework6作业回顾 (附代码)

本次作业重点是hascode和compress function，评价一个compress function好坏的标... 来自：everest115的博客

Python标准库：内置函数zip(\*iterables)

本函数是实现从多个序列生一个元组列表迭代子返回，即是从每个序列里获取一项，... 来自：大坡3D软件开发

CS61BHomework3作业回顾 (附代码)

Part 1.smoosh()函数，目的是将数组中出现的所有数字产生重复的部分删除，例如11... 来自：everest115的博客

股市第三次机会来了！不看就亏大了！

唐煌投资 · 嬾嫻

关于使用vector时出现“Vector iterators incompatible!”问题注意细节

在使用SLT中vector时，但用for loop时，稍不注意，就会出现"vector iterators incomp... 来自：feixuedudiao的...

Python 编程思想 —— 可迭代对象 (iterables)

1. 可迭代对象包括 基本数据结构：list、tuple set、dict list comprehension>> [i for i i... 来自：Zhang's Wikipedia



caimouse

关注 1858篇文章



ctthuangcheng

关注 655篇文章



SunnyYoona

关注 823篇文章

Iterables vs. Iterators vs. Generators

来源：http://nvie.com/posts/iterators-vs-generators/ 译文：http://www.jianshu.com... 来自：bubblelone的专栏

python 中zip (\*iterables) 函数的使用

1.zip()函数的定义 从参数中的多个迭代器取元素组合一个新的迭代器 返回：一个zip... 来自：youyongyong1的...

C++ 迭代器 (Iterators)

迭代器用于存取(access)一个序列(sequence)中的元素，其用法类似于指针。例如，我... 来自：cuiyifang的专栏

北京股王8年不亏铁律“1272”曝光，震惊众人！

陕西信息科技 · 嬾嫻

final review for berkeley cs61b

关于伯克利公开课cs61b的期末复习文档

07-21

PHP 5.5 yield and promises

Coroutines generalize subroutines to allow multiple entry points for suspending and resuming execution at certain locat...

05-29

伯克利大学数据结构课程

http://datastructur.es/sp17/ CS61B-Data Structures UC berkeley 来自：yaoxiaofeng\_000...

C++实现控制台版2048 (内附彩蛋)

前言 之前做过一个JavaScript版本的2048游戏，最近在学习C++，昨天晚上突然... 来自：隐秀\_

深入浅出 Python Iterators 迭代器

本文由浅入深解释了 Python Iterators。Iterator 即实现了 \_\_iter\_\_() 和 \_\_next\_\_() 方法... 来自：sx1lgc的博客



## 什么是核心期刊 核心期刊有哪些

百度广告

### CS61B Homework9作业回顾（附代码）

👁 57

本次作业主要学会运用Disjoin Sets, DisjointSets的class作业已经写好给出。class中... 来自: [everest115的博客](#)

### CS61B Homework7作业回顾(附代码)

👁 36

发现编程真的是很强调逻辑。首先Track down the whole tree, split 3-key node into 2 ... 来自: [everest115的博客](#)

### ES6学习——迭代器（**Iterators**）：迭代器接口高级应用

👁 900

迭代器自身可迭代 迭代器返回{done:true} 换成非对象类型, 例如undefined, null等... 来自: [Challenge Next ...](#)

### 阿里巴巴2017年秋招：最少砖块数被切中编程题

👁 166

```
# -*-coding:utf-8 -*- #code=utf-8 #code=utf-8 import sys if __name__ == "__main__": r...
```

 来自: [yangxy的博客](#)

### cs61a课程总结--lecture7 递归（和一种数据结构）

👁 520

数据结构 tuple 是一种至类型 来自: [vczhfan的专栏](#)

### 手机赚钱新项目，短短几个小时，赚了2000！

桥至投资 · 熯熯

### 下载 Data Structures Lecture Notes (UCB **CS61b**)

10-20

Data Structures Lecture Notes (UCB CS61b) Data Structures Lecture Notes (UCB CS61b)

### ES6学习——迭代器（**Iterators**）：迭代器接口应用示例

👁 422

这篇文章我们来看看迭代器的一些基本应用例子。前面给的一些例子都是用for-of循... 来自: [Challenge Next ...](#)

### c++教程（二十三：**Exceptions**）

👁 3011

Exceptions（异常） Exceptions 异常提供了一种方法来对程序中的异常情况（如运行... 来自: [我爱智能](#)

### C#2.0介绍之**Iterators**(迭代器)

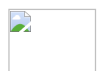
👁 609

转自: <http://wdxinren.cnblogs.com/archive/2005/07/12/191652.html>C#的foreach... 来自: [ido的专栏](#)

### CS61A 系列课程**笔记**（一）

👁 1338

嗯 今天刚看第二周的课程, 大量的 reading 材料我是真的要崩溃了, 全英。。。我... 来自: [柠檬黄先生的博客](#)



## 招聘海归硕士

百度广告

### ES6学习——迭代器（**Iterators**）：内置可迭代对象汇总

👁 934

这篇文章我们看一下哪些内置对象是可迭代的： 1）数组Arrays console.log([][Symbol... 来自: [Challenge Next ...](#)

### 《STL源码剖析》**笔记-迭代器iterators**

👁 30

上一篇: 《STL源码剖析》笔记-空间配置器 来自: [WizardtoH](#)

### Python学习**笔记**系列（一）：异常

👁 1万

try: except : # if name1 is raised except : # if name2 is raised except (name3, name4... 来自: [xudacheng06的...](#)

### 异常（**Exceptions**）

👁 1090

关于Java异常处理的一些总结 来自: [无常](#)

### 【Finereport】Multiple **exceptions** & 预览找不到模板文件

👁 124

问题: 在公司堡垒机上使用Finereport 8.0总是会遇到预览找不到模板文件, 明明已经... 来自: [喜东东的博客](#)

### 股市第三次机会来了！不看就亏大了！

唐煌投资 · 熯熯

android 经典错误

6463

07-27 12:33:55.503: DEBUG:dalvikvm(100): threadid=31 wakeup: interrupted07-27... 来自: [tqtihihc的专栏](#)

一个jetty启动报错问题的排查过程（Multiple exceptions）

1332

Multiple exceptions jetty启动服务时错误信息如下:WARN [main] o.e.j.w.WebAppCont...

【Java】异常（Exceptions）

563

概念 exception是“exceptional event”的缩写，是指执行程序中发生的事件，破坏... 来自: [Just do it!](#)

有关Iterators and Reverse Iterators的程序

```
#include <vector>
#include <string>
using namespace std;
void print (int elem){cout << elem << " ";}
int main(...
```

Java高级系列——何时使用、如何使用异常（Exceptions）

694

一、介绍 在Java中，异常是一个非常重要的工具，在程序流中异常可以发出一些反常... 来自: [Ron.Zheng](#)

老中医说：饭后用一物，体重不过百！北京人必看！

尹承熙 · 熾燄

python3.5 django 文档相关动作错误及解决办法

783

1、动作：StringIO 参考：http://stackoverflow.com/questions/11914472/stringio-in-... 来自: [gavinking0110的...](#)

Python中itertools模块用法详解

591

本文实例讲述了Python中itertools模块用法，分享给大家供大家参考。具体分析如下... 来自: [bbbeoy的专栏](#)

OpenVolumeMesh编译的一些错误解决

1501

OpenVolumeMesh编译的一些错误解决 来自: [XjtrX的专栏](#)

springboot-mybatis 实现数据库操作

46

之前的项目是建立在使用已有的JPA接口操作数据库，我们这次不使用JPA的接口，自... 来自: [qq\\_42685050的...](#)

boost库实现Base64编解码

3390

boost库对Base64编解码进行很了很好的封装，以下代码用boost库实现Base64编解... 来自: [nk\\_wang的专栏](#)

 别再用MongoDB了

百度广告

下载

A Java Reference: Assorted Java Reference Material

10-08

UC Berkeley CS61B Data Structure推荐阅读 Fall 2018

下载

CS61B java intro

03-31

berkeley cs61b java introduction, 对初学者很有帮助

下载

Exceptions in MIPS.

05-04

深入MIPS架构，展示MIPS异常处理的细节。对CPU异常的学习有帮助。

MS08067/MS10061漏洞靶机环境搭建总结

8340

MS08067/MS10061漏洞靶机环境搭建总结 来自: [SQL\\_injection的...](#)

ES6学习——迭代器（Iterators）：接口规范概述

3111

这篇文章我们仔细看一下ES6规范中是如何定义Iterator的接口，在25.1章节中：@@i... 来自: [Challenge Next ...](#)

 别再用MongoDB了

百度广告

Python中的异常和模块

51

一、异常 1.捕获异常 在程序开发中,如果对某些代码的执行不确定(程序的语法完全正... 来自: [Code for need .C...](#)

Iterators (迭代器)

关于迭代器， 总共有五种：

560

来自： a130737的专栏



隐秀\_

关注

原创

26

粉丝

22

喜欢

24

评论

13

等级：

博客 3

访问：3万+

积分：503

排名：11万+

勋章：



1024



about

cs专业一名挣扎的后端  
热爱交友

GitHub：[github.com/seriouszyx](https://github.com/seriouszyx)

cnblog：[cnblogs.com/henuzyx](https://cnblogs.com/henuzyx)

独立博客：[seriouszyx.github.io](https://seriouszyx.github.io)

热门文章

最新超详细VMware虚拟机下载与安装

阅读量：19735

最新超详细虚拟机VMware安装Kali Linux

阅读量：5596

上机考试作弊，也不是不可以呢

阅读量：2642

hexo搭建个人独立博客 | Next主题深度美化方案

阅读量：660

JSONArray.fromObject不执行且不报错问题的解决

阅读量：431

最新文章

Tomcat 引入jar包导致的

NoClassDefFoundError 报错的问题解决

Spring IoC容器浅析及简单实现

CS61B sp2018笔记 | Efficient Programming

Linux 基本操作总结（后端必备）

Java生成解析二维码

博主专栏

Spring 2018

CS61B sp2018笔记





阅读量：514    6 篇

个人分类

C++4 篇

JavaScript5 篇

Linux4 篇

游戏2 篇

Java13 篇

展开

归档

2018年10月2 篇

2018年9月1 篇

2018年8月4 篇

2018年7月8 篇

2018年6月4 篇

展开

最新评论

hexo搭建个人独立博客 | Ne...  
qq\_40950957: [reply]qq\_39901385[/reply] 我用的是这个主题 Next.Gemini, 我...

hexo搭建个人独立博客 | Ne...  
qq\_39901385: 我知道什么问题了是没清缓存 楼主就是那个主页怎么把一篇文字折叠起来呀 用一个按钮来控制打开这个怎么...

hexo搭建个人独立博客 | Ne...  
qq\_39901385: 本地能设置next的背景和头像旋转 使用了hexo g -d放在了git上 绑定了域名 但是通过g...

hexo搭建个人独立博客 | Ne...  
qq\_39901385: 楼主方便加个q吗我有些hexo搭配next的主题优化的问题想问你

最新超详细VMware虚拟机下载与...  
qq\_43318544: 不错顶起来, 现在有15了VMware Workstation Pro15虚拟机破解版: https...



客户关系管理系统



联系我们



扫码联系客服



官方公众号

 QQ客服

 kefu@csdn.net

 客服论坛


 400-660-0108

工作时间 8:00-22:00

https://blog.csdn.net/qq\_40950957/article/details/81324531

9/10

[关于我们](#)[招聘](#)[广告服务](#)[网站地图](#)

 百度提供站内搜索 京ICP证09002463号

©2018 CSDN版权所有

---

[网络110报警服务](#)[经营性网站备案信息](#)

[北京互联网违法和不良信息举报中心](#)

[中国互联网举报中心](#)