Java语言基础

【Day14】

Map集合

Map集合的概述

- Map接口定义的集合又称查找表，用于存储所谓“Key-Value”映射对。Key可以看成是Value的索引，作为Key的对象在集合中不可以重复。
- 根据内部数据结构的不同，Map接口有多种实现类，其中常用的有内部为哈希表实现的HashMap和内部为二叉树实现的TreeMap。

Map集合的常用方法

- Map集合中常用的方法如下：

V put(K key, V value);	将Key-Value对存入Map，若集合中已经包含该Key，则替换该Key所对应的Value，返回值为该Key原来所对应的Value，若没有则返回null

Map集合的常用方法

- Map集合中常用的方法如下：

V get(Object key);	返回与参数Key所对应的Value对象，如果不存在则返回null
boolean containsKey(Object key);	判断集合中是否包含指定的Key
boolean containsValue (Object value);	判断集合中是否包含指定的Value

Map集合的遍历方式

- 当处理Map集合中每个元素时需要迭代Map集合，迭代方式一：迭代Key

```
map =  ....//初始化map
Set<Character> keySet = map.keySet();
for (Iterator<Character>  i = keySet.iterator();  i.hasNext();  ) {
    Character key = i.next();
    Integer value = map.get(key);
    System.out.println(key+":"+value);
}
```

Map集合的遍历方式

- Map集合的迭代方式二：迭代Entry.

```
map = ....//初始化map
Set<Entry<Character, Integer>>  entries = map.entrySet();
for (Entry<Character, Integer> e : entries) {
    Character key = e.getKey();
    Integer value = e.getValue();
    System.out.println(key+":"+value);
 }
```

Map集合的性能调优

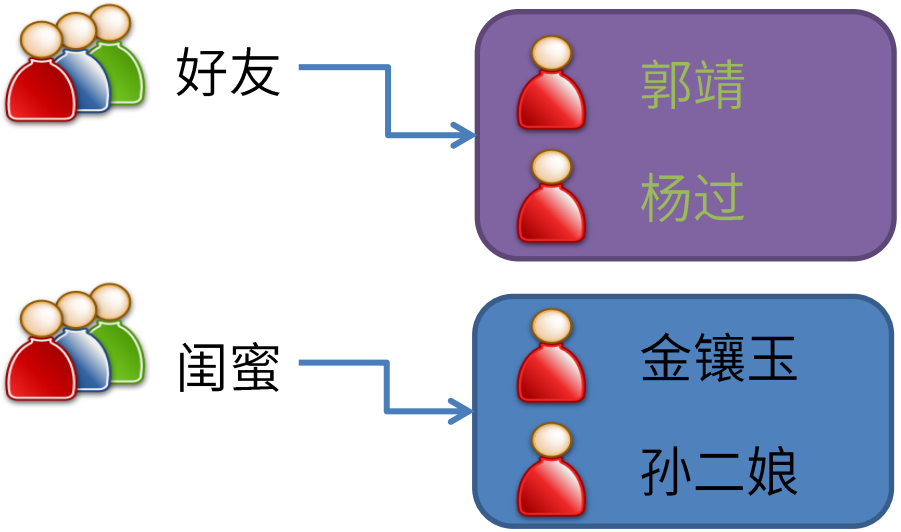
- Capacity：容量, hash表里bucket(桶)的数量, 也就是散列数组大小.
- Initial capacity：初始容量, 创建hash表的时 初始bucket的数量, 默认构建容量是16. 也可以使用特定容量.
- Size： 大小, 当前散列表中存储数据的数量.
- Load factor：加载因子, 默认值0.75(就是75%), 当向散列表增加数据时如果size/capacity 的值大于Load factor则发生扩容并且重新散列(rehash).
- 性能优化: 加载因子较小时候散列查找性能会提高, 同时也浪费了散列桶空间容量. 0.75是性能和空间相对平衡结果. 在创建散列表时候指定合理容量, 减少rehash提高性能.

Map集合的实际应用

- Map集合是面向查询优化的数据结构, 在大数据量情况下有着优良的查询性能.
- 经常用于根据key检索value的业务场景

Map集合的实际应用

- 联系人列表的编程实现
  Map<String, List<Contact>> contacts;



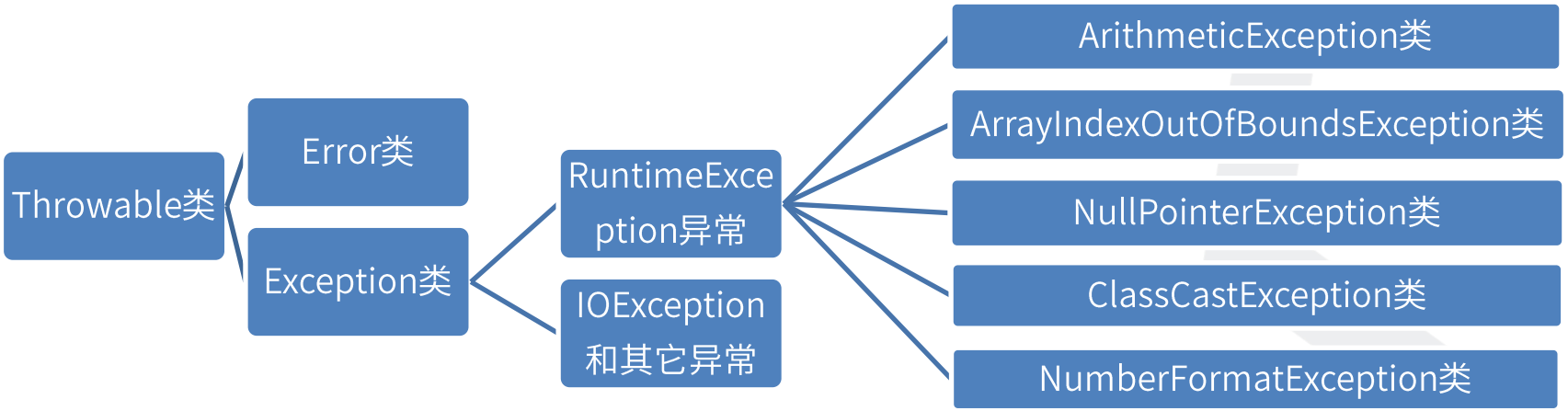
异常处理机制

异常机制的概述

- Java异常结构中定义有Throwable类，Exception和Error是其派生的两个子类
- 其中Exception表示由于网络故障、文件损坏、设备错误、用户输入非法等情况导致的异常；
- Error表示Java运行时环境出现的错误，例如：JVM内存资源耗尽等。

异常机制的框架

- Java异常相关类的框架如下：



异常的捕获

- 异常的捕获语法如下：

```
try{
    编写可能发生异常的代码;
  }
  catch(异常类型 引用变量名){
    编写针对该类异常的处理代码;
  }
  ...
  finally{
    编写无论是否发生异常都要执行的代码;
  }
```

异常捕获的注意事项

- 当需要编写多个catch分支时，切记小类型应该放在大类型的前面；
- 懒人的写法：catch(Exception e){}
- finally通常用于进行善后处理，如：关闭已经打开的文件等。

异常的抛出

- 程序中会定义许多方法，这些方法中可能会因某些错误而引发异常，但你不希望直接在这个方法中处理这些异常，而希望交给调用它的方法统一处理，这时候可以使用“throws”关键词来声明这个方法将会抛出异常。
- 例如:

```
public static void stringToDate(String str) throws  ParseException{
}
```

重写方法的抛出规则

- 如果使用继承时，在父类的某个方法上抛出某些异常，而在子类别中重新定义该方法时注意：

  1 不抛出异常

  2 抛出父类方法异常中的子类异常

  3 抛出和父类一样的异常

- 重写的方法不可以：

  1 抛出同级不一样的异常

  2 抛出更大的异常

自定义异常

- 很多的软件在开发时都采用异常做错误处理的方式，因此用户可以根据需要自定义异常。
- 自定义异常的方式为:

  1 继承Exception或者异常的子类。

  2 提供两个构造，无参构造和String做参数的构造。

File类

File类的简介

- java.io.File类用于表示文件和目录，通过File类在程序中操作硬盘上的文件和目录。
- File类只用于表示文件和目录的属性信息（名称、大小等），不能对文件的内容进行访问。

File的常用方法

- File类中常用的方法如下：

File(String pathname)	根据参数指定的路径名来构造对象
boolean exists()	测试此抽象路径名表示的文件或目录是否存在
boolean delete()	用于删除文件，当删除目录时要求是空目录
boolean createNewFile()	用于创建新的空文件
boolean mkdir()	用于创建目录
boolean mkdirs()	用于创建多级目录

总结与答疑

技术资料

IT兄弟连

ITXDL.CN

变态严管　让学习成为一种习惯