复习：

1. 数学类：Math---最终类，私有构造方法，提供了基本类型的初等数学运算。BigDecimal---用于精确运算的类，要求参数以字符串形式传入，是因为底层是按照字符数组的下标位置来相减。BigInteger---用于存储任意大的整数，存储数据的时候是以数据的二进制的补码形式来存储，底层是以byte数组来实现

2. 日期类：Date---重点掌握字符串和日期的转换，SimpleDateFormat---如果字符串转化为日期-parse，如果是将日期转化为字符串用的是format

3. 异常：

异常的顶级父类是Throwable---Error（错误），Exception（异常）

Exception—可以处理---要么捕获要么抛出

编译时异常---要求必须处理

运行时异常---可以处理，可以不处理 ---RuntimeException

异常的捕获方式：单个处理、统一处理、分组处理

子类不能抛出比父类更多的异常---范围不能更大，编译时异常

finally的特点是无论出现异常与否，都会执行

## 集合--- Collection<E>

顶级的集合接口

存储多个同一类型的数据的容器---大小不固定

<E> --- 泛型---用于规定集合中的元素类型的---由于泛型的限制，集合中只能存储对象---要求泛型只能声明为引用类型

int[] arr; --- arr的数据类型是数组类型，元素类型是int

Collection<String> c; --- c的数据类型是Collection，元素类型是String

Collection<Integer> c ;

### List---列表

有序（保证元素的存入顺序）

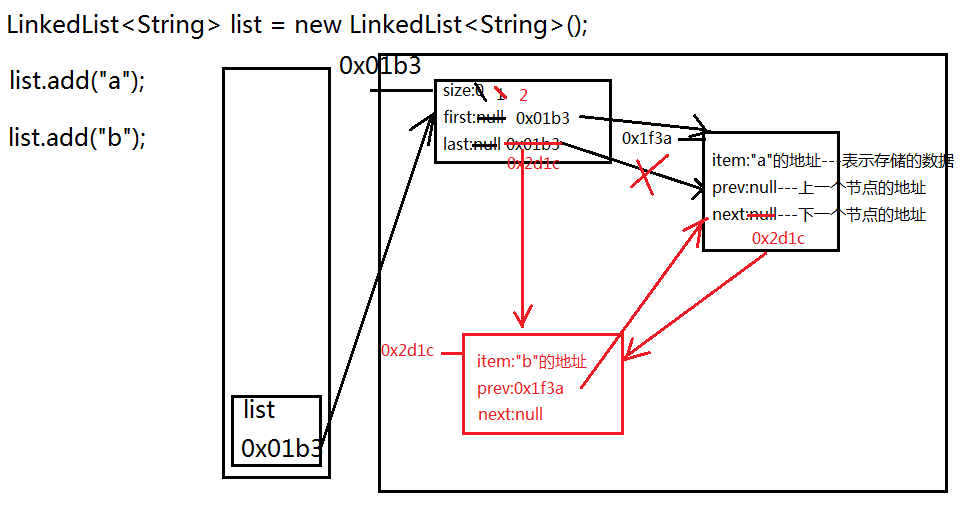
#### ArrayList---顺序表

底层是依靠数组来存储数据的。默认的初始容量是10，每次扩容是在原基础上加上一半--- 是基于右移来实现的。是一个线程不安全的集合。插入或者删除元素相对复杂，查询元素会相对较快

练习：手写一个顺序表 --- add / remove / set / get / indexOf / contains/size/isEmpty/toString

#### LinkedList---链表

底层是基于节点来实现的。节点在使用的时候再开辟空间，所以不需要初始容量。增删操作要相对容易，但是查询元素相对复杂。是一个线程不安全的集合。



思考：ArrayList和LinkedList的增删元素的效率

增删的位置靠前的时候LinkedList要快一些，如果增删的位置靠后ArrayList要快一些

#### Vector---向量

第一个集合。底层是使用数组存储数据的。默认初始容量是10，每次扩容都会使容量翻倍。是一个线程安全的

elements()---返回一个基于此向量的迭代器。

#### Stack---栈

继承了Vector。遵循后进先出的原则。栈顶元素---最后放入栈中的元素，栈底元素---最先放入栈中的元素。将元素放入栈中---入栈/压栈，将元素从栈中取出---出栈/弹栈

练习：用数组实现一个栈—empty/pop/peek/push/search