复习：

1. 包装类：每一种基本类型都有对应的类形式---自动封箱拆箱---哈希码

2. 数学类：Math---提供的基本数学运算、BigDecimal---用于精确运算，在运算的时候要求以字符串形式传入

3. 日期类：Date---字符串和日期之间的转换---SimpleDateFormat

## 异常



异常是Java中一套用于反馈和处理问题的机制

### Throwable

异常的顶级父类。

Error表示错误。一旦出现无法处理。[OutOfMemoryError](mk:@MSITStore:C:\\Users\\Public\\Desktop\\开班软件\\JDK_API_1_6_zh_CN.CHM::/java/lang/../../java/lang/OutOfMemoryError.html" \o "java.lang 中的类), [StackOverflowError](mk:@MSITStore:C:\\Users\\Public\\Desktop\\开班软件\\JDK_API_1_6_zh_CN.CHM::/java/lang/../../java/lang/StackOverflowError.html" \o "java.lang 中的类)

Exception表示异常。出现之后可以处理。

编译时异常：在编译的时候就已经出现了。要求必须处理。

CloneNotSupportedException

ParseException

运行时异常：在编译的时候不出现到运行的时候才出现。可以处理，可以不处理 --- RuntimeException

ArithmeticException

ArrayIndexOutofBoundsException

NullPointerException

ClassCastException

StringIndexOutOfBoundsException

NumberFormatException

自定义异常去继承对应的异常类：如果继承的是Exception或者是其他子类，那么这个异常就是一个编译时异常；如果继承的是RuntimeException或者是其子类，那么这个异常就是一个运行时异常

### 异常的捕获方式

1. 如果多个异常的处理方式各不一样，那么多个catch分别捕获分别处理

2. 如果多个异常的处理方式都一样，那么捕获这些异常的父类进行统一的处理

3. 如果多个异常分了不同的组进行处理，那么同一组之间的异常可以用 | 隔开用一个catch来捕获处理 --- JDK1.7开始

运行时异常可以随意抛出随意捕获；编译时异常必须在抛出之后才能捕获。

总结：重载和重写的异同点

无论是方法的重载还是方法的重写，都是行为的多态。

重载：在同一个类中，存在方法名一致而参数列表不同的方法。方法的重载只和方法签名有关，和修饰符、返回值类型以及异常没有关系。本身是一种编译时多态。

重写：在父子类中，存在方法签名一致的非静态方法。在构成重写的时候，要求子类重写的方法的权限修饰符的范围要大于等于父类对应方法的权限修饰符的范围。如果方法的返回值类型是基本类型或者是void，那么子类在重写的时候要保持一致。如果方法的返回值类型是引用类型，那么子类在重写方法的时候返回值或者和父类对应方法的返回值类型一致，或者重写的方法的返回值类型是父类对应方法的返回值类型的子类。如果父类方法抛出了编译时异常，那么子类在重写的时候抛出的编译时异常不能超过父类异常的范围。重写本身是一种运行时多态。

#### finally

无论出现异常与否，都要执行一次。

项目在开发阶段，如果出现了异常，捕获异常，打印异常的栈轨迹，在catch中抛出一个运行时异常来停止

项目已经上线，如果出现了异常，记录错误日志，跳转错误页面。

## Collection<E>集合

能够存储多个数据的容器---大小不固定

E---泛型 --- 集合中的泛型是用于规定存储的元素类型的 --- 由于泛型的限定，集合中只能存储对象 --- 也就意味着泛型对应的必须是引用类型

int[] arr; arr的数据类型是数组，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; 5

Collection<int[]> c;

### List --- 列表

有序 --- 保证元素的存入顺序。可以对元素进行编号---下标，从0开始

#### ArrayList

底层存储元素用的是数组。默认初始容量是10，每次扩容是在上次容量的基础上增加一半，扩容底层是基于了右移来实现的。 10 -> 15 -> 22 -> 33 -> 49