第二周

## 第一章

一、对象的上转型对象（以及下转型对象）

1、上转型对象，父类引用指向子类对象，这样父类的引用可以在调用自己的被重写的方法时会调用引用向下转型的子类的方法

2、上转型对象，父类引用想要调用子类中有而父类中没有的方法则需将父类引用向下转型为子类类型

二、final关键字与static关键字

1、final关键字：

①被final修饰的成员变量是常量，在被声明时就必须初始化 且只能被初始化依次 之后再为其赋值会报错

②被final修饰的方法不能被重写，但是子类可以继承

③被final修饰的类 不能被继承 为最终类

④被final修饰的对象 其地址无法改变

2、static关键字：

①被static修饰的成员（成员方法或成员变量）是类类型的从属于类在类加载时就会被初始化，出生时间比构造器要早；另外一个类类型的变量可以通过类名.变量名访问（也可以通过对象.变量名访问一般不提倡）

②static静态代码块 在类加载时会被执行一次，执行顺序按照类的加载过程，程序从main方法进入加载main所在的类，遇到继承则加载父类代码加载类的时候遇到静态代码块就执行静态代码块。

提示：java中的非成员变量或者非成员方法，又叫类成员或静态成员，属于类所有。当我们定义一个类此时就会为类成员分配内存空间，并且类方法会在类变量之后分配空间。

三、方法的重载（overlord）

1、方法的重载发生在同一个类的中

2、方法的参数列表

四、方法的重写（override）

1、方法的重写发生在继承关系的父类与子类之间。

2、子类重写父类的方法的要求：

①方法的返回值类型、方法名、参数列表必须与父类被重写的方法完全一致

②方法的访问修饰符不能低于父类被重写时的访问权限。

五、单例设计模式

1、使用单例设计模式的目的：节省系统性能（内存与cpu占用率）

2、两种单例设计模式：

①懒汉式：需要才创建

public class User {

private static User user = null;

private User(){}

public static User getUser(){

if(user == null){

user = new User();

}

return user;

}

}

②饿汉式：定义的时候就创建好，需要直接拿

public class User {

private static User user = new User();

private User(){}

public static User getUser(){

return user;

}

}

总结： 这两种单态模式实现都是将自己类的对象变量 封装并static

并且将构造方法私有化

通过public的getUser方法 将对象的引用指向同一个对象

这里的同一个对象就有了两种实现方式

懒汉式就是将 将类类型的私有变量静态初始化为空 然后通过判断 如果为空则new一个不为空则返回初始化的那个（也就是说第一次 getUser方法就new一个user对象 第二次或者以后每次调用getUser方 法就是返第一次创建的象）

而饿汉式则是先将类类型的对象在静态初始化的时候就将其new出 来然后第一次 直到后来的每次都是返回的这个对象

# 第二章

一、抽象类与抽象方法（abstract）

1、抽象类不一定有抽象方法，但是抽象方法一定存在于抽象类中

2、抽象类可以有构造方法，但是不能实例化

3、一个子类如果继承自一个抽象父类，那么这个子类必须实现抽象父类的所有抽象方法，如果没有则这个子类一定是一个抽象子类

二、接口（interface）

1、接口中的属性就是常量用 public static final 修饰并且声明的时候一定要初始化

2、接口中所有的方法都是抽象方法除了 default修饰的方法 以及静态方法（static修饰的） 这两种方法由接口自己的实现类来访问

3、接口中的一般方法都是用public abstract修饰的

4、接口中没有构造方法当然也不能被实例化

5、一个接口可以继承自多个接口

6、一个类可以继承自另一个类同时实现多个接口 但是类的继承要写在接口的前面

7、接口主要用来定义规范

三、异常类

1、异常类的父类时throwable类 它有两个子类 一个Error 一个是Exception 其中Exception又有种，一种是编译期的异常 它出现则表示编译不能通过 另外一个是 RunTimeException 在运行中才能体现出来异常

2、常见的几种异常类：CheckedException：ClassNotFoundException,FileNotFoundException

RunTimeException：ArrayIndexOutOfBoundsException，ArithmeticException，IllegalArgumentException，NullPointException

3、处理异常的几种方法：

①最懒的方法直接在方法名后面throws 异常名 这种方法可以抛出多个异常 在方法后面用逗号隔开 谁调用这个方法就谁来处理异常 层层上抛知道虚拟机处理

②throw new 异常类名（""）字符串中输入当发生这个异常你要提示的内容，相当于new了一个异常类对象，并且初始化他，然后输出初始化的内容。此种方法可以灵活处理 放在判断结构中都行 只要发生异常就可以将异常对象用此种方法抛出并输出提示

③try-catch捕获并释放异常类

4、自定义异常类(定义的时候要继承自Exception)

①自定义一个异常类 可用上述的三种方法去实现对你定义的异常的处理

# 第三章

集合

一、Collection（接口）

Collection下的接口实现类储存的数据都是

1、List（接口）

三个常用的实现类

①Vector

②ArrayList

③LinkList

2、Set（接口）

两个常用的实现类

①HashSet

②TreeSet

二、Map（接口）储存键值对

1、TreeMap<K,V>（底层是红黑二叉树 加上实现Map接口）

我们的K为Integer类或者是String时Comparable接口中的方法会帮我们实现对K的排序 从而我们遍历的时候是看到的是以K的升序排列的

如果想把K设置成我们自己定义的类的类型 则需要在自定义的类中实现Comparable接口 并重写compareTo方法对我们传进来的自定义类对象进行属性的排序。

2、HashMap<K,V>（hash表 加上实现Map接口）

①允许K，V的值为空

②单线程安全

3、HashTable<K,V>

①K的值 或者V的值not Null

②多线程安全

# 第四章

日期类与日历

一、日期类的一些方法

1、Date date =new Date();产生一个日期对象 保存当前的时间日期 直接打印日期的时候输出的也是当前的时间与日期与星期

2、date.get 后面加year month day dayofweek dayofmonth ...可以返回当前日期的对应的属性但是年要-1900 月是0-11 对应 January-December 星期日为1 星期六为7

二、日历类

1、日历类的对象获取方式：

Calendar calendar= Calendar.getInstance();

2、日历类的一些方法

在实际的应用中日历类的一些方法由很多：set add get

三、格式化日历类

1、对象的获取方法：

SimpleDateFormat simpleDateFormat =new SimpleDateFormat("yyyy-MM-dd E");

2、方法：

①用格式化日历对象去调用 他的format方法 传入一个日期对象 返回的是一个new对象的时候格式化的字符串形式

②用格式化日历对象去调用 他的parseformat方法时 传入一个格式化的字符串 返回的是一个日期对象

四、Math类

直接撸方法：abs取绝对值 pow(a,b)a的b次幂 sqrt()开方 round（）四舍五入取整（加上0.5后取整数部分） min max sin asin 最大值最小值 三角函 反三角函数 random 取零到一的一个随机小数 ceil 向上取整 floor 向下取整

五、Random类（随机类数）

1、对象：Random random=new Random();

2、方法：random.nextInt(a);传入参数表示生成从0到a的一个左闭右开区间的随机整数

3、例子：想获得从50到80之间的随机整数：random.nextInt(30)+50

# 第五章

IO流

一、File 类的基本用法

1、创建File类的对象：File file = new File("a");a字符串表示传入你要操作文件的完整路径用反斜杠时注意转义 一般用正斜杆

2、一些方法：

file.getAbsolutePath();//获得绝对路径包括文件扩展名

file.createNewFile();//创建新的文件

file.exists();//文件或者文件夹是否存在

file.length();//文件的大小

file.getName();//获得文件的短命 为 文件名加上后缀名

file.getTotalSpace();//获得文件所在磁盘的总空间 返回的是字节数

file.getFreeSpace();//获得文件所在磁盘的可用空间 返回的是字节数

file.isDirectory();//文件对象是否是文件夹

file.isFile();//文件对象是否是文件

file.getParent();//获得文件的父目录 返回的是绝对路径减去 文件的段名的字符串

file.mkdirs();//创建文件对象传入的字符串多级目录

String [] s= file.list();//对文件对象如果不是文件夹 则返回一个null String类的数组 是文件夹则返回String类的数组将该文件夹下所有文件及文件夹存入

File [] f= file.listFiles();//对文件对象进行遍历 讲过该文件目录下的所有文件及文件夹 返回File类的一个数组存起来

二、FileInputStream与FileOutputStream 类(文件输入输出字节流)

1、对象构建它有两个构造方法：

①FileInputStream fileInputStream = new FileInputStream(file);//传入File对象

②FileOutputStream fileOutputStream = new FileOutputStream("C:/a");//传入路径

2、其中输入流读取文件到字节数组中 而输出流则是写入字节数组到文件中

3、写完之后记得要关闭流

三、FileReader与FileWriter类（文件输入输出字节流）

//新建一个字符流输入对象

FileReader fileReader = new FileReader("c:aaa.txt");

//将文件读取到字符数组中

char[] chars = new char[(int) file.length()];

fileReader.read(chars);

//新建一个字符流输出对象

FileWriter fileWriter = new FileWriter("C:bbb.txt");

//将字符数组写入字符流输出对象

fileWriter.write(chars);

//关闭流

fileReader.close();

fileWriter.close();