javaSE学习笔记

1、继承父类时，若有新增同名属性或方法，调用父类的属性或方法必须用super关键字，**被重写方法不能比父类方法抛更大的异常（比如父类抛FileNotFoundException,子类方法不能抛IOException，父类无异常，子类重写方法也不能抛出异常）**

2、**Java虚拟机拥有堆（存储 new出来的对象及数组值）、本地方法栈（存放本地方法状态，有的虚拟机将其与虚拟机栈放在一起）、方法区（存放常量，包名、类名、方法信息等相关信息，有个运行时常量池，用于存储编译时生成的常量）、虚拟机栈（占用的是操作系统内存，线程私有存储对象引用，局部变量，栈溢出和内存不足是针对此而言的），程序计数器（线程私有，每个线程都有，本地方法计数器为空）。**

3、String类---不通过new创建的字符串存放在常量池，再次创建，不在创建新字符串，而是指向初次创建的字符串，new出来的字符串存放在对堆空间，而值仍然是指向常量池的值，静态域存放静态变量。

引用（对象名 ，其实是地址）、局部变量放在栈空间，对象实体的地址、属性地址存放在堆空间，属性的值存放在常量池。double类型不能直接比较，比如 x=0.1,y=0.2,z=x+y,这时，z！=0.3；

4、equals()方法---只比较属性值，不比较方法，object类的equals方法比较的是地址值

5、toString()方法---打印对象的引用时默认调用该对象的toString()方法，没有重写Object类toString()方法，默认调用Object类的toString()方法，返回该对象所在的类，以及实体的对应堆空间的首地址

6、八种基本数据类型---bollean、 byte、char、short、int、float、

double、long，对应的有八种包装类比如Integer，Short，Character等

7、String.ValueOf()、Integer.parseInt(String)、Bollean.parseBoolean(String)............

8、父类的两个不同子类不能直接强转，但可把一个子类转为父类、再转为另一个子类

9、Vector.addElement(object),参数必须是对象

10、自动装箱和拆箱---装箱是把基本类型转为包装类，拆箱相反

11、三目运算符？：会自动提升返回值类型（精度）

12、单例模式---构造函数声明私有，两种方式创建--懒汉模式、饿汉模式（线程安全）

13、属性赋值操作顺序---默认值->显式或非静态代码块（看代码位置）->构造器->赋值方法

14、每次创建对象，非静态代码块加载一次

15、final属性不能默认初始化，可以通过显示、代码块、构造器初始化，如有多个构造器，每个构造器都要初始化

16，abstract抽象类有构造器

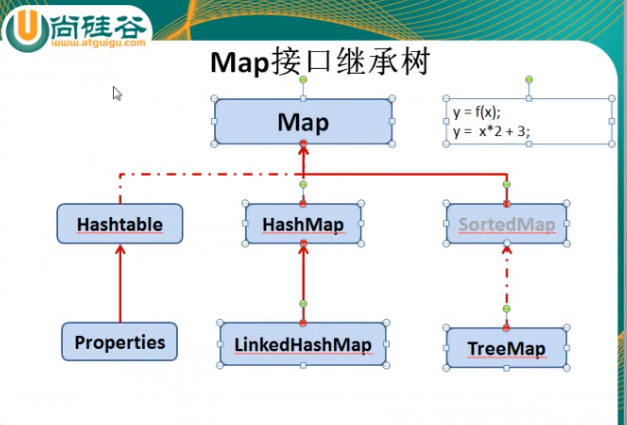
17、接口的属性都是 public static final 修饰，无构造器，接口的两个应用，**代理模式（代理类创建并调用被代理类的方法，相当于帮别人干事情）和工厂方法（在一个类的方法中创建另一个类的实例并返回）**

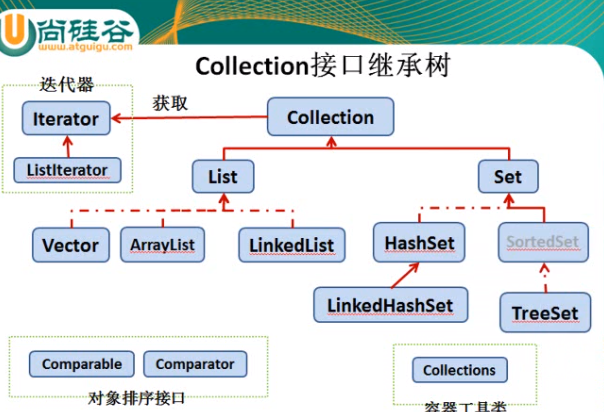
18、内存溢出--申请的超出可用的，内存泄漏--申请的无法释放

19、for(String i: str)遍历不影响原数组的值

20、掌握Collection接口的15个方法

21、List 实现类ArrayList、LInkedList、Vector，Set实现类**HashSet（可添加null值）**、LinkedHashSet、TreeSet（只能添加同一类型，添加自定义类型必须实现Comparable接口，之后可以按照指定自然排序遍历），**Set的无序性是更具HashCode实现的，不可重复性是根据HashCode和equals方法实现的，因为equals比较费时，所以先用HashCode，算出HashCode值就把根据HashCode把元素放在哪，HashCode相同时再使用equals比较，**Map键值不可重复，重复添加时，会覆盖原来的（与set区分,set不会覆盖），实现类HashMap、LinkedHashMap、TreeMap、HashTable（子类Propeties---重点，可用于处理.properties文件，键值对存储）





HashMap------所有键值key是一个Set（不重复）集合，所有值value是一个collection集合，每一个键值对是一个Entry<key,value>,所有键值对是一个Set集合

22、排序接口Comparable、Comparator

23、自定义泛型类--class<T> 自定义泛型方法 public <T> void fun(T t)

24、泛型为通配符的集合或类可以读，无需实例化，但不能写，写null除外；

25、枚举类---自定义---私有化构造器，类内部构造枚举类对象，对象必须用 public static final 修饰，

使用enum关键字定义枚举类--私有化构造器，类内部构造枚举类对象，对象无需定义和修饰符，多个对象用逗号分隔

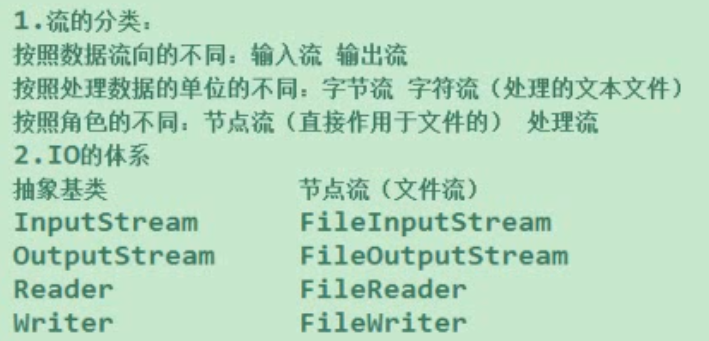
枚举类内部定义自己的方法可以返回不同结果

26、注解---元注@Retention（RetentionPolicy.SOURCE、CLASS、RUNTIME)生命周期，@Target（声明注解什么，是类还是方法还是别的），@Document（生成文档时可以被提取成文档，@Retention必须为RUNTIME），@Inherited（声明注解是否可以被继承）

27、File--File对象对应一个文件或目录，File类只能对文件本身进行操作（创建、删除、重命名等）**文件路分隔符为 / 或 \ 都行但 \ 要转义**，涉及到文件内容，需用文件流，new FIle()是获取已存在文件或目录的对象，不存在的话，可以调用createNewFile()创建，makedir（）创建文件目录，上层目录存在才能创建，上层目录不存在，要用makdirs（）

File的renameTo方法f1.renameTo（f2），f1重命名为f2，要求 f1 一定存在，f2一定不存在，并且f1和f2同为文件或同为目录，f1和f2不一定是同一个路径但不可跨盘符，File的length方法（当对象是文件）返回字符个数

28‘、IO流---直接操作文件的叫节点流，其他的叫处理流

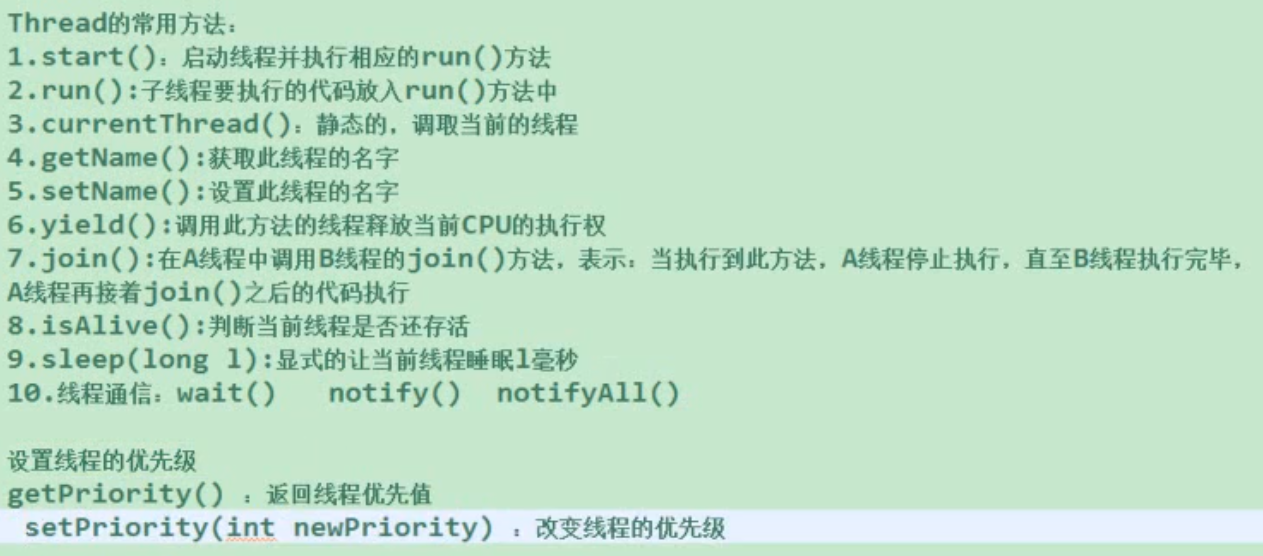


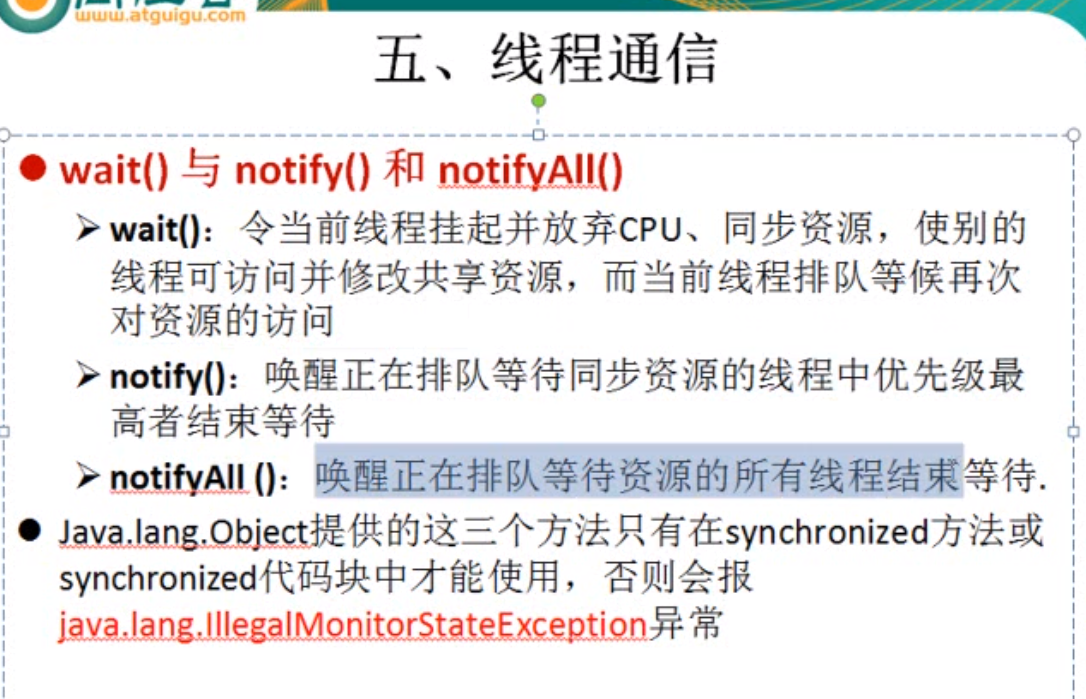
29、输出时文件可以不存在，会自动创建，若存在，会覆盖

30、FileInputStream处理字节，FileReader处理字符，FileReader只能处理文本文件（纯字符，txt文件，word文档不属于纯文本文件），FileInputStream二进制文件（非文本文件，几乎所有文件，也能处理文本文件），BufferReader可以读一行（reanLine（）），Buffer类输出要flush（），数据流、对象流序列化实现serializable或Externable接口，RandomAccessFile类即可输入又可·输出

31、多线程----设置优先级不能保证线程先执行完，只是抢到执行权的概率增大（默认优先级是5，最大是10，最小是1），设置优先级在线程开启前。启动多线程必须调用Thread的start方法，直接调用run方法是不行的，实现Runnable接口的类，所有线程共享一个run方法，继承Thread的类，多有线程拥有执行自己的run方法，**java线程有自己的工作内存，对值的操作都是在工作内存进行，只是从主内存拷贝值到工作内存，不同线程不共享工作内存**

32、线程同步------同步代码块和同步方法（关键字synchronized(lock)lock必须是唯一的对象针对同步代码块，同步方法不需要对象只要关键字，默认对象是当前对象），对于继承Thread，同步的锁对象是当前对象时，对象不是唯一的**(注意),volatile关键字，java虚拟机栈为每个线程分配一块空间，称为线程栈，**线程栈保存了线程运行时候变量值信息。当线程访问某一个对象时候值的时候，首先通过对象的引用找到对应在堆内存的变量的值，然后把堆内存变量的具体值load到线程本地内存中，建立一个变量副本，之后线程就不再和对象在堆内存变量值有任何关系，而是直接修改副本变量的值，在修改完之后的某一个时刻（线程退出之前），自动把线程变量副本的值回写到对象在堆中变量。这样在堆中的对象的值就产生变化，虚拟机只保证线程读取的值是最新的，load之后，有别的线程写入值到内存，此线程无法读取





33、StringBuffer---线程安全，StringBuilder---线程不安全，效率高于StringBuilder

34、BigDecimal大数（可以自定义任意精度，默认三位）、BigInteger

35、反射---通过反射可以直接访问私有类型，需调用setAccessible（）方法，搭配getDeclaredMethod()（表示本身所有声明方法）使用，getMethod()（表示本身和父类的public方法）

36、反射获取Class对象----有三种方法，通过Class类的方法（forName（））获取，通过自定义类本身属性（class）获取，通过自定义类对象的方法（getClass（））获取，需要用newInstance（）（实际是调用运行时类空参构造器）方法获取对象，也可以通过指定构造器获取对象，默认是Object，需要强转。还可以通过类加载器加载。反射方法调用----方法引用.invoke（类对象,方法参数）

37、反射实现动态代理---可应用于调试、远程调用。

38、动态代理与aop----创建被代理类，创建aop,动态创建代理类。

39---线程池——