1. 类是一种自定义的数据类型，可以使用类来定义变量，这种变量称为引用数据类型变量。
2. 构造方法是一个类创建对象的根本途径，如果没有创建构造方法，系统则自动创建一个无参的构造方法。
3. this关键字调用的是类里的元素。
4. 布尔类型不能与其他任何数据类型做转换。
5. Static修饰的方法和变量，既可以通过类来调用，也可以通过对象来调用，静态成员不能直接访问给静态成员。
6. Static修饰的方法中不能使用this引用。所以static修饰的方法不能使用没有static修饰的普通成员。
7. 栈是Java的指令区，堆数据区，对象被创建成功后就会保存在堆中，而对象的引用会保存在栈中。
8. Java方法中的基本类型参数传递方式是值传递。所谓值传递，就是将实际参数值的副本传入方法内，而实际参数本身不会受到影响。
9. 值传递是将栈内存中变量的值复制后传递给方法，而引用类型的数据做方法参数时（如参数为方法时），虽然属于是引用传递，传递的是地址，相当于指针。
10. 在定义方法时，如果最后一个形参的类型后增加“…”，则表明该形参可以接受多个参数值，多个参数值被当成数组传入，个数可变的形参只能处于形参列表的最后，一个方法中最后只能有一个长度可变的形参。
11. 确定一个方法需要三要素：调用者，方法名，形参列表。方法重载的要求是两同，一不同：同一个类中方法名相同，参数列表不同。
12. 变量包括成员变量和局部变量，成员变量包括实例变量（没有static）和类变量（static修饰）。成员变量无需显示初始化。
13. Java中共有4个访问控制符，控制级别从小到大为：private，default，protect，public。其中当不使用任何控制符修饰时，系统默认采用default。Default访问的范围是通一个类和通一个包下。
14. Java的子类不能直接调用父类的构造方法，Java的继承是单继承，每个子类只能有一个直接的父类，一次只能继承一个父类。
15. 方法的重写（方法覆盖）遵循“三同一小一大”的规则，即方法名相同，形参列表相同，返回值类型相同，“一小”：子类方法声明抛出的异常类应比父类方法声明抛出的异常类更小或相等。“一大”：子类方法的访问权限应比父类方法更大或相等。
16. 覆盖方法或者别覆盖方法要么都是类方法，要么都是实例方法。
17. 如果子类方法中调用父类被覆盖的实例方法，可以用super所谓调用者进行调用,super是父类对象的默认引用。
18. 在一个构造方法中调用另一个重载的构造方法，一般使用this调用实现；在子类构造方法中调用父类构造方法，一般使用super调用实现。
19. 子类的构造过程中必须调用其父类的构造方法，如果调用super（）必须写在子类构造方法中的第一行。如果子类的构造方法没有显式的调用父类的构造方法，则系统默认调用父类的无参构造方法。
20. 如果子类构造方法中既没有显式调用父类构造方法，而父类又没有无参的构造方法，则编译出错。
21. 创建Java对象时，系统总是先调用该类里定义的初始化块。初始化块只能在创建对象时自动执行而且在执行构造方法之前执行。
22. 静态初始化块是和类相关的，系统将在类初始化阶段执行它，而不是在创建对象的时候才执行，静态初始化块总是比普通初始化块先执行。
23. 如果一个类始终只能创建一个实例，则这个类被称为单例类。创建单例类的构造方法需要用private修饰。
24. 使用final修饰的成员变量必须显示的初始化，普通方法不能为final修饰的成员变量赋值。
25. Final修饰的方法仅仅是不能被重写，并不是不能被重载。Final修饰的类没有子类，如java.lang.Math类就没有子类。
26. 当一个类没有足够的信息以描绘一个具体的对象时，这样的类就是抽象类。是对一系列看上去不同但本质上相同的具体概念的抽象。抽象类是从多个具体类中抽象出来的具有共同特点的父类。
27. 抽象类不能被实例化，只能被子类继承。子类继承抽象类时，如果子类不是抽象类，则必须实现从抽象类继承来的所有抽象方法。
28. 抽象方法必须定义在抽象类中，有抽象方法的类只能定义成抽象类，类里可以没有抽象方法。
29. Abstract不能用于修饰属性和修饰构造方法。Abstract不能和final，static，private同时使用。
30. 接口不包含普通的方法，只有抽象方法，和抽象类的区别是接口是规范，不提供任何实现，接口的继承可以是多继承。接口内没有构造方法和初始化块。
31. 接口的成员变量都是共有的，静态的，最终的常量。
32. 内部类可以直接访问外部类的私有数据，因为内部类被当成其外部类成员，但外部类不能访问内部类的实现细节。
33. 非静态内部类不能声明static成员，只有静态的内部类才能声明static成员。
34. 静态内部类没有父类，不能引用外部类实例成员或其他内部类成员。当可以访问外部类的任一静态字段或静态方法。
35. Java数组既可以存储基本数据类型，也可以存储引用数据类型的数据。
36. 一旦属猪初始化完成，数组在内存中所占的存储空间将被固定下来，因此数组的长度不可改变。
37. 声明数组变量时不允许指定数组的长度，如：int[3] a; 这语句是违法的。
38. Java基本数据类型：boolean，byte，char，short，int，float，long，double。String不属于Java的基本数据类型。
39. equals和==的区别：  
    1. equals返回的是布尔类型的值，用来比较两个对象是否等价，只能用来比较引用类型，判断的是内容。不能用来比较基本类型。

2. ==属于运算符，既可以比较基本类型，也可以比较引用类型，判断的是内容和内存地址。

3. equals是方法在Object中初始的功能判断的是内存地址，在某些特定的类中重写了该方法。equals（）方法在比较一些特定的引用类型（如java.lang.String，java.io.File，java.util.java以及封装类等）数据时，允许改变先前严格的等价性标准，只要两个对象的内容呢等价即返回true。

例子：

String s1 = new String(“hello”);

String s2 = new String(“hello”);

System.out.println(s1==s2); //输出为false

System.out.println(s1.equeals(s2)); //输出为true

S1=s2;

System.out.println(s1==s2); //输出为true

System.out.println(s1.equeals(s2)); //输出为true

1. String类的对象已经创建，其内容不可改变。每次进行改变的时候等同于创建了一个String对象。
2. Java.lang.StringBuffer类表示的是内容可以修改的Unicode编码字符序列，其对象创建后，所保存的字符串内容和长度均可以修改。
3. Stringbuffer对象都拥有一个可变容量的字符串缓冲区，可以随着内容自动增减。初始值为16个字符。
4. Stringbuilder也是一个可变的字符序列，和stringbuffer相似，但有些区别。
5. StringBuffer是线程安全的可变字符序列，可将字符缓冲区安全的用于多个线程，即线程同步。StringBuilder是与StringBuffer兼容的API，但不保证同步，只能用于单线程中，速度比StringBuffer快。执行速度为，stringbuilder 》 Stringbuffer》string。
6. Length()方法计算的是字符串的总长度，而capacity（）计算的是对象的总分配。
7. System类中所有成员都是静态的。
8. toString（）方法能够处理任何引用类型的数据，但不能处理基本类型数据。
9. 封装类均被定义为final，因此不能被继承。所封装的基本类型为只读，所以创建后不能改变内容。
10. 当程序发生异常时，产生一个异常事件，生成一个异常对象，并把它提交给运行时系统，再有运行时系统寻找相应的代码来进行处理，这个过程称为抛出异常。
11. 异常抛出后，运行时系统从生成对象的代码开始，沿方法的调用栈逐层回溯查找，直到包含相应的处理方法，并把异常对象交给该方法为止，这个过程捕获异常。
12. 在使用多个catch语句时，要注意catch子句的排列顺序，子类必须要在父类的前面。
13. 简单类（如int或char）以及非Throwable类（如String或Object）不能用在throw语句中。
14. Throw语句可以引发明确的异常。
15. 获取Throwable对象：使用catch子句中的参数或者是用new操作符创建。
16. 程序执行到throw语句后立即终止，后面的语句不在执行，然后在包含它的所有try块中从里到外寻找含有与其类型匹配的catch子句。
17. 无论有没有异常被引发，finally块中的语句都会执行。
18. 断言是java开发中的调试方式，通常用类检测关键值，方式：assert expression；如果expression返回为true，则继续执行，否则抛出异常。
19. 在泛型中，除了异常类，枚举类和匿名类外，任何类都可以具有类型参数。
20. 同一个泛型类应用不同的类型参数复合而成的类型间并不存在继承关系。
21. 对于一个static方法，无法访问泛型类的类型参数，否则就必须使其成为泛型方法。
22. Java集合类有三种：Set，List，Map。Set和List继承Collection接口。
23. Set集合不记录元素加入的顺序，不包含重复的元素。List集合可以区分元素的顺序，通过索引寻找元素，可以包含重复的元素。Map中不包含重复的键，并且每个“键”最多只能映射一个值。
24. Map不支持Iterator()迭代器。
25. HashMap不支持线程安全的，而Hashtable支持线程安全。在单线程下HashMap的速度较快。HashMap可以接受null值，而Hashtable不可以。在Java5以后，ConcurrentHashMap替代了Hashtable。
26. Java支持二进制字节流和字符流，InputStream是字节输入流的超类，OutputStream是字节输出流的超类，Reader是字符输入流的超类，Writer是字符输出流的超类。
27. File类中delete方法删除的是一个空的文件。
28. FileFilter和FilenameFilter用于实现对文件名字符串的过滤。两者都有accept方法，FileFilter是accept（File pathname）, FilenameFilter是accept（File dir,String name）。两者的区别在于accept（）方法的参数不同。
29. RandomAccessFile类提供“随机访问文件”方式，有两个特点：1.对一个文件可以同时进行既读又写的操作。2.可以在文件中指定的任意位置读取数据或写入操作。
30. InputStreamReader 和InputStreamWriter是将字节流转换为字符流。
31. 序列化是指把并行数据转换成串行数据的处理过程，而对象序列化是指把对象的状态数据以字节流的形式进行处理，一般用于实现完全的对象。
32. 带有static和transient关键字的属性或方法将不会被序列化。
33. 在GBK编码中文占用两个字节，英文占用1个，在UTf-8中文占用3个字节，英文占用1个字节。