以下是填空、简答题可能考的知识，请大家务必在考试前多看几遍，混个眼熟

填空题：

1、Java 源文件的扩展名是 .java 。

2、编译 Java Application 源程序文件将产生相应的字节码文件，这些字节码文件的扩展名为 .class 。

3、HelloWorld.java 编译成功后会生成一个 HelloWorld.class 文件。

4、开发与运行 Java 程序需要经过的三个主要步骤为 编译源程序 、 编译生成字节码 、 解释运行字节码 。

5、面向对象程序设计的基本特征是 抽象 、 封装 、 继承 、 多态 。

6、在 Java 语言中，体现多态性有两个方面： overloading(重载) 和 overriding(重写、覆盖) 。

7、package 语句要放在 文件开头 ，且必须放在 import 语句之前。

8、Java 源程序的文件名必须与 public 类的类名 相同。

9、Java 源文件中可以有 多 个类。Java 源文件中有 至多 1 个类可以是 public 类。

10、 类 是 Java 中基本的结构单位。

11、Java 应用程序由若干个 类 所构成，它们可以分布在 1 个或多 个源文件中，其中必须有 1 个源文件含有 主类 。

12、Java 应用程序总是从主类的 main 方法开始执行。

13、Java 源文件如果含有主类，主类可以是 public 类、 默认(缺省) 类。

14、String 类在 java.lang 包中。

15、Java 的字符类型采用的是 Unicode 编码。

16、Java 的各类数据所占用的长度与具体的软硬件平台环境 无 关。

17、在 Java 的基本数据类型中，char 型采用 Unicode 编码方案，每个编码占用 2 字节内存空间。无论是中文字符还是英文字符，都是占用 2 字节内存空间。

18、在 Java 的方法中，定义一个常量必须用关键字 final 。

19、基本数据类型包括 布尔 型、 整数 型、 浮点 型、 字符 型。

20、浮点型数据根据存储长度和精度的不同，进一步分为 double 和 float 两种具体类型， double 的精度高于另一者。

21、Java 中的默认浮点型是 double 。

22、字符串分为两大类，一类是字符串常量，使用 String 类的对象表示；另一类是字符串变量，使用 StringBuffer 类的对象表示。

23、 StringBuffer 对象的字符序列可以修改， String 对象的字符序列不可以修改。

24、整型数组中的各元素的值必须是 整 型。

25、数组的下标从 0 开始。

26、Java 语言的标识符 是 区分大小写的。

27、在 Java 中，\是转义字符，\n 表示 回车 ，\t 表示 tab 键(制表键) ，\\表示 \字符本身 。

28、Java 程序中的单行注释符是 // ，多行注释符是 /**/ 。

29、即使条件不满足， do-while 循环体内的语句也至少执行一次。

30、for 循环、while 循环、do-while 循环 可以 互相嵌套。

31、for 循环 可以 产生死循环，while 循环 可以 产生死循环，do-while 循环 可以 产生死循环。

32、for 循环的循环体 可以 为空，while 循环的循环体 可以 为空，do-while 循环的循环体 可以 为空。

33、方法中可以包含 任意 个 return 语句。return 语句可以用来返回 任何 数据类型。

34、同一方法 不可以 同时用 static 和 abstract 修饰。

35、方法的修饰(如果有)例如 public、abstract，必须放在 方法的类型 的前面。

36、 abstract 方法是一种仅有方法头，没有方法体的方法；这种方法只能存在于 abstract 类和接口中。

37、在面向对象方法中，类的实例称为 对象 。

38、类声明 class A 等价于 class A extends java.lang.Object 。

39、在 Java 中若定义抽象类则需要加关键字 abstract 来修饰。

40、abstract 和 final 不可以 同时修饰同一个类。

41、abstract 类中可以有 常量以及变 量。

42、abstract 类中 可以 有 abstract 方法， 可以 有非 abstract 方法。

43、interface 中只可以有 abstract 方法，不可以有 非 abstract 方法。

44、类中的实例方法 不可以 用类名直接调用，类中的静态方法 可以 用类名直接调用。

45、同一个类的对象使用 不 同的内存段。

46、静态成员使用 相 同的内存空间。

47、成员变量 有 默认值，局部变量 没有 默认值。

48、成员变量的名字 可以 和局部变量的名字相同。

49、类成员的权限修饰符的访问权限大小关系是 public > protected > private 。

50、方法的参数的名字 不可以 和方法中声明的局部变量的名字相同。

51、定义私有的成员函数或成员变量时， 不必 在类的开头部分集中定义。

52、定义私有的成员函数或成员变量时，利用关键字 private 定义。

53、被私有访问控制符 private 修饰的成员变量，只能被 该类自身 所访问和修改。

54、被私有访问控制符 private 修饰的成员变量， 不能 被与它在同一个包中的其他类、在其他包中的该类的子类所访问和修改。

55、String 类是 final 类。

56、final 类可以有 0 个子类，非 final 类可以有 多 个子类。

57、除了 java.lang.Object 类，任何一个类有且仅有一个父类。

58、除了 java.lang.Object 类，任何一个类有 且仅有一 个父类。

59、在 Java 中，所有类的根类/父类/超类是 java.lang.Object 。

60、子类和父类 可以不 在一个包中。

61、在 Java 中，关键字 final 使类不能派生出子类。

62、子类 可以 声明和父类的成员变量同名的成员变量。

63、子类声明的成员的变量的名字和从父类继承来的成员变量的名字相同，子类就会 隐藏掉 所继承的成员变量。

64、一个类中可以有 多 个构造方法。

65、构造方法 不可以 用 final 或 static 修饰。

66、构造方法是类的一种特殊方法，它的主要作用是 完成对类的对象的初始化工作 ，它的方法名必须与 类名 相同。

67、一般在创建新对象时，系统会自动调用 构造方法 ，自动调用是通过关键字 new 实现的。

68、创建类的对象时，使用运算符 new 给对象分配内存空间。

69、Java 语言对构造方法的返回类型的规定是： 没有返回类型 。

70、Java 语言规定构造方法 可以 重载。

71、即使一个类中未显式定义构造方法，也会有一个默认的构造方法，默认的构造方法的参数情况是： 无参 ，函数体情况是： 函数体为空 。

72、子类 不继承 父类的构造方法。

73、子类构造方法中 可以 有多条 super 调用父类的某个构造方法的语句。

74、如果在子类的构造方法中，没有显示地写出 super 关键字来调用父类的某个构造方法，那么编译器就默认地有 super(); 调用父类的无参数的构造方法。

75、如果在子类的构造方法中，显示地写出了 super 关键字来调用父类的某个构造方法，那么编译器就 不再提供 默认的 super 语句。

76、super 关键字形成的语句必须是子类构造方法中的第 1 条语句。

77、子类中想使用被子类隐藏的实例成员变量或实例方法就需要使用关键字 super 。

78、重载方法是指，一个类中定义两个方法，名字 相同 ，参数 不同 。

79、this 不可以 出现在 static 方法中， 可以 出现在实例方法和构造方法中。

80、子类 可以 定义和父类的方法同名的方法。

81、子类在方法重写时， 不可以 把父类的类（static）方法重写为实例方法。

82、子类在方法重写时，方法的访问权限 不可以降低，但可以提高 。

83、接口 可以 用 public 修饰， 不可以 用 private 或 proteced 修饰。

84、接口中只可以有 常 量，不可以有 变 量。

85、接口中只可以有 abstract 方法，不可以有 非 abstract 方法。

86、接口中的常量 必须 指定初值。

87、接口中的常量 可以 用接口名直接访问。

88、除了 final 属性，接口中定义的常量还具有 public、static 属性

89、接口中的方法的访问权限一定都是 public 。

90、接口中的方法 不可以 用 private 或 protected 或 final 修饰。

91、接口中 void f(); 方法声明等价于 public abstract void f(); 。

92、接口的常量中 可以 存放实现该接口的类的实例的引用。

93、abstract 类 可以 实现接口。final 类 可以 实现接口。

94、类使用关键字 implements 实现接口。

95、定义一个类继承父类的关键字是 extends ，定义一个接口继承接口的关键字是 extends 。

96、一个类可以继承 1 个类。一个类可以实现 多 个接口。一个接口可以继承 多 个接口。

97、一个类 可以 同时继承一个类和实现一个接口

98、一个类 不可以 重复实现同一个接口。

99、类和它所实现的接口 不一定 在同一个包里。

100、一个类声明实现一个接口，但没有重写接口中的所有方法，那么这个类必须是 abstract 类。

101、抽象类可以 重写 接口中的方法，也可以 继承 接口中的方法。

102、如果一个非 abstract 类实现某个接口，该类 必须 重写接口中的全部 abstract 方法。

103、如果一个非 abstract 类实现某个接口，该类必须重写接口中的 全部 abstract 方法。

104、子接口将继承父接口中的 全部 方法和 全部 常量。

105、Integer.parseInt("");双引号内 不是整数 时会触发 NumberFormatException 异常。

106、FileNotFoundException 类是 IOException 类的子类。

107、Throwable 类有两个重要的子类—— Exception (异常) 和 Error (错误) 。

108、所有异常的父类都是 Throwable 类。

109、捕捉异常通过 try-catch-finally 语句实现。

110、try-catch 语句可以由 多 个 catch 组成。

111、在编写异常处理的 Java 程序中，每个 catch 语句块都应该与 try 语句块对应。

112、在异常处理中，将可能产生异常的语句放在 try 块中，用 catch 语句去处理异常。

113、如果想在方法头抛出异常，那么需要用关键字 throws ；如果想在方法体内抛出异常，那么需要用关键字 throw 。

114、FileReader 输入流按 字符(char) 读取文件的内容，FileWriter 输出流按 字符(char) 写出数据。

115、FileOutputStream 输出流按 字节(byte) 写出数据。

116、程序如果需要读取程序“外部”的数据，可以创建指向外部的 输入 流。

117、程序如果需要将程序中数据，写入到程序“外部”，可以创建指向外部的 输出 流。

118、如果程序要读取一个文件，可以创建指向文件的 FileInputStream 流、 FileReader 流。

119、如果程序要写入一个文件，可以创建指向文件的 FileOutputStream 流、 FileWriter 流。

120、 import java.sql.*; 命令可以使我们在程序中创建数据库相关的对象。

121、 import java.io.*; 命令可以使我们在程序中创建输入输出流相关的对象。

122、 import java.net.*; 命令可以使我们在程序中创建网络相关的对象。

123、线程状态可以分为五大状态： 新建 、 就绪 、 运行 、 阻塞 、 死亡 。

124、多线程系统中，多个线程之间有 同步 和 互斥 两种关系。

表格：

基本数据类型

类型	关键字	占用大小		取值范围	默认值
		(单位：bit) (单位：位)	(单位：byte) (单位：字节)		
布尔型	boolean	未定义		false 或 true	false
字节型	byte	8	1	−128~127	0
短整型	short	16	2	−32768~32767	0
整型	int	32	4	-2^{31}~$2^{31}-1$	0
长整型	long	64	8	-2^{63}~$2^{63}-1$	0L
单精度浮点型	float	32	4	大约$\pm 3.4\times 10^{38}$	0.0f
双精度浮点型	double	64	8	大约$\pm 1.7\times 10^{308}$	0.0d
字符型	char	16	2	16 位 Unicode	'\0'

注：java 中没有明确规定布尔类型的数据所占字节大小。在不同的环境下，布尔类型所占字节可能不同，但是无论布尔类型占据多少字节，都只有 1bit 的数据是有意义的。

例 1：在 Java 的基本数据类型中，char 型采用 Unicode 编码方案，每个编码占用 2 字节内存空间。

例 2：在 Java 中，整型常量 123 占用的存储字节数是 4 。

简答题：

1、请简述 Java 语言的特点。

答：简单易学：Java 风格类似于 C++，但它摒弃了 C++中复杂、不安全的特性；

面向对象：Java 的设计是完全面向对象的，它具有面向对象的封装、继承和多态三大特点；

安全性：Java 提供了字节校验器、文件访问限制机制、类装载器和运行时内存布局四级安全保证机制；

跨平台（体系结构中立）：Java 程序能够在网络上任何地方执行；语言版本完全同一；具有字节代码与平台无关性；

多线程：Java 环境本身就是多线程的，并且 Java 提供了对多线程的语言级支持；

动态性：Java 所需要的类是运行时动态装载的，也可从网络载入。在分布环境中动态地维护应用程序和类库的一致性，类库的更新不需要重新编译程序，不影响用户程序的执行；

健壮性：Java 提供强类型机制、异常处理、垃圾自动收集等，并且 Java 摒弃了指针。

除上述七点以外，Java 还是一种分布的、解释的、可移植的、高性能的程序设计语言。

2、请简述 Java 中异常处理的机制。

答：首先 Java 的异常是面向对象的，一个 Java 的 Exception 是一个描述异常情况的对象。当出现异常情况时，一个 Exception 对象就产生了，并放到异常的成员函数里。

Java 的异常处理是通过 5 个关键词来实现的：try，catch，throw，throws 和 finally。异常的处理结构由 try，catch，finally 三个块组成。其中 try 块存放将可能发生异常的 Java 语言，并管理相关的异常指针；catch 块紧跟在 try 块后面，用来激发被捕获的异常；finally 块包含清除程序没有释放的资源、句柄等。不管 try 块中的代码如何退出，都将执行 finally 块。

Java 语言可以不在方法中直接捕获，而用 throw 语句将异常抛给上层的调用者。throw 语句就是来明确地抛出一个异常。首先你必须得到一个 throwable 的实例句柄，通过参数传到 catch 中，或者采用 new 操作符来创建一个。

3、Java 中有哪几种访问控制符？并说明各自的作用范围。

答：private：仅在本类内的代码可以访问。

默认（无修饰符）：在同一包内的代码可以访问。

protected：在同一包内及其子类中的代码可以访问。

public：所有包中的代码都可以访问。

4、请说明 final 和 static 各自的作用。

答：final：final 修饰的变量的值不可以被改变，final 修饰的方法不可以被重写，final 修饰的类不可以被继承。

static：static 修饰的成员变量称为静态变量，被类的所有的对象共享，可以通过类本身直接调

用。static 修饰的方法称为静态方法，可以通过类本身直接调用。静态方法中不能访问类的非静态成员变量和非静态成员方法，静态方法中不可以有 this。

5、什么是继承？

答：通过必要的说明能够实现某个类无需重新定义就拥有另一个类的某些属性和方法，这种关系就称为继承，并且允许多层的继承关系。先定义的类称为父类，后定义的类称为子类。

6、在 Java 中，实现代码复用主要有两种方式——继承和组合。请描述下这两种方式的区别。

答：继承：复用父类的属性和方法，在编译期间就确定了类的层次结构。在实际编程中，不好的设计经常会导致父类中带有子类的行为，导致父类子类互相依赖，丧失了重用的好处，破坏了封装。

组合：在运行期间通过对象之间的引用，动态确定彼此之间的关系，对象之间只能通过接口相互作用，对象的封装性就得到了良好的维护。在运行期间，任何对象都能够被替换为其他相同类型的对象。

总而言之，在 Java 中优先使用组合，而不是继承。

7、请简述重载和重写的区别。

答：方法的重写（Overriding）和重载（Overloading）是 Java 多态性的不同表现。重写是父类与子类之间多态性的一种表现，重载是一个类中多态性的一种表现。

如果在子类中定义某方法与其父类有相同的名称和参数，我们说该方法被重写。子类的对象使用这个方法时，将调用子类中的定义，对它而言，父类中的定义如同被“屏蔽”了。

如果在一个类中定义了多个同名的方法，它们或有不同的参数个数或有不同的参数类型，则称为方法的重载。重载的方法是可以改变返回值的类型。

8、构造方法是特殊的方法，请简述它的特殊之处。

答：①构造方法名必须与类名相同；

②构造方法没有返回值，但不用 void 声明；

③构造方法不能被 static、final、abstract、synchronized 和 native 等修饰符修饰；

④构造方法不能像一般方法那样用“对象.构造方法”显式地直接调用，应该用 new 关键字调用构造方法，给新对象初始化。

9、实例方法和类方法的区别是什么？

答：被 static 修饰的方法称为类方法（或静态方法），而没有被 static 修饰的方法称为实例方法。

二者的调用方式不同。实例方法属于实例，必须通过实例调用；类方法属于类，一般通过类名调用，也可以通过实例调用。

二者访问的成员不同。实例方法可以直接访问该类的实例变量和实例方法，也可以访问类变量和类方法；类方法只能访问该类的类变量和类方法，不能直接访问实例变量和实例方法。类方法要访问实例变量或调用实例方法，必须首先获得该实例，然后通过该实例访问相关的实例变量或调用实例方法。

10、说明抽象类和接口的异同点。

答：相同点：抽象类和接口都可以有抽象方法，都不可以被实例化。

不同点：①创建关键字不同：抽象类用关键字 abstract 创建，接口用 interface 关键字创建；

②成员变量不同：抽象类可以包含普通的变量，接口内的变量只能是 final 的；

③方法不同：抽象类可以包含普通的方法，接口内只能有抽象方法且都是 public 的；

④继承/实现不同：接口可以被继承，也可以被实现，而抽象类只能被继承。接口被接口继承时用关键字 extends，且可以多继承；接口被类实现时用关键字 implements，且可以多实现；抽象类被类继承时用关键字 extends，且只能单继承。

11、在 Java 中怎样创建一个线程？

答：①定义类来实现 Runnable 接口

```
class TestThread implements Runnable{
    public void run() {...}
}
```

②继承 Thread 类来实现

```
class TestThread extends Thread {
    TestThread(String name){
        super(name);
        start();
    }
    public void run() {...}
}
```

12、Error 和 Exception 有什么区别？

答：Error 是程序无法处理的错误，比如 OutOiMemoryError、ThreadDeath 等。这些异常发生时，Java 虚拟机一般会选择线程终止。

Exception 是程序本身可以处理的异常，这种异常分两大类——运行时异常和排运行时异常。程序中应当尽可能去处理这些异常。

13、finally 的作用是什么？什么时候会被执行？

答：finally 代码块主要用来释放资源，比如 I/O 缓冲区、数据库连接。

无论是否抛出异常，finally 代码块总是会被执行。就算是没有 catch 语句同时又抛出异常的情况下，finally 代码块仍然会被执行。

14、finally、finalize 有什么区别？

答：finally 是异常处理语句结构的部分，表示总是执行。

finalize 是 java.lang.Object 类的一个方法，在垃圾收集器执行的时候会调用被回收对象的此方法，可以覆盖此方法提供垃圾收集时的其他资源回收，例如关闭文件等。JVM 不保证此方法总被调用。

15、字节流与字符流有什么区别？

答：①字节流操作的基本单元为字节，字符流操作的基本单元为 Unicode 码元；

②字节流默认不使用缓冲区，字符流使用缓冲区；

③字节流可用于任何类型的对象，包括二进制对象，而字符流只能处理字符或者字符串。