

一、选择题

1. 实现字符流的读操作类是 ( )。  
A. `FileInputStream`    B. `FileOutputStram`    C. `FileReader`    D. `Writer`
2. 下列关于 Java 语言特性的描述中, 错误的是( )。  
A. 支持多线程操作  
B. Java 程序与平台无关  
C. Java 语言删除了类似 C 语言中的指针和内存释放等语法, 有效避免了非法操作内存。  
D. 支持单继承和多继承
3. 下列关于运算符的描述中错误的是( )。  
A. `101100` 右移 2 位后结果是 `111011`  
B. `boolean b=20<45?true:false`; b 的返回值是 `true`  
C. 如果两个运算符具有相同的优先级, 那么左边的表达式要比右边的表达式先被处理  
D. `101100` 按位取反后的结果是 `110011`
4. 下列循环语句的循环次数是( )。  

```
int i = 5;
do{
System.out.println(i--);
i--;
}while(i!=0);
```

  
A. 0                      B. 1                      C. 5                      D. 无限
5. 下列循环语句中, 循环体被执行的次数为( )。  

```
for(int i = 0, j = 0; (j != 18) || (i < 4); i++);
```

  
A. 3                      B. 4                      C. 不确定                      D. 无限
6. 下列关于 Java 语言的数组描述中, 错误的是( )。  
A. 数组在赋初值和赋值时都不判越界  
B. 数组下标从 0 开始  
C. 数组作为对象允许用 `new` 关键字进行内存分配  
D. 数组的长度通常用 `length` 表示
7. 下列关于数组的定义形式, 正确的是( )。  
A. `int[]a; a = new int;`  
B. `char b[]; b = new char[80];`  
C. `int[]c = new char[10];`  
D. `int[]d[3]=new int[2][];`
8. 在类的修饰符中, 规定只能被同一包类所使用的修饰符是( )。  
A. `public`                      B. 默认                      C. `final`                      D. `abstract`
9. 在成员变量的修饰符中, 规定只允许该类自身访问的修饰符 ( )。  
A. `private`                      B. `public`                      C. 默认                      D. `protected`
10. 下列关于静态方法的描述中, 错误的是( )。  
A. 在类体内说明静态方法使用关键字 `static`。  
B. 静态方法只能处理静态变量或调用静态方法  
C. 静态方法不占用对象的内存空间, 非静态方法占用对象的内存空间  
D. 静态方法只能用类名调用

11. 下列关于抽象类的描述中, 错误的是( )。
- A.抽象类是用修饰符 **abstract** 说明的
  - B.抽象类是不可以定义对象的
  - C.抽象类是不可以有构造方法的
  - D.抽象类通常要有它的子类
12. 下列关于接口的描述中, 错误的是( )。
- A.接口实际上是由常量和抽象方法构成的特殊类
  - B.一个类只允许继承一个接口
  - C.定义接口使用的关键字是 **interface**
  - D.在继承接口的类中通常要给出接口中定义的抽象方法的具体实现
13. 下列关于类包的描述中, 错误的是( )。
- A.类包是一种特殊的类
  - B.类包是若干个类和接口的集合
  - C.使用 **import** 关键字导入
  - D.类包能够有效地避免同名类冲突的情况
14. 下列常用包中, 存放用户图形界面类库的包是( )。
- A. **java.awt**      B. **java.lang**      C. **java.util**      D. **java.io**
15. 下列关于接口的描述中, 错误的是( )。
- A.接口实际上是由常量和抽象方法构成的特殊类
  - B.一个类只允许继承一个接口
  - C.定义接口使用的关键字是 **interface**
  - D.在继承接口的类中通常要给出接口中定义的抽象方法的具体实现
16. 下列关于类包的描述中, 错误的是( )。
- A.类包是一种特殊的类
  - B.类包是若干个类和接口的集合
  - C.使用 **import** 关键字导入
  - D.类包能够有效地避免同名类冲突的情况
17. 下列常用包中, 存放用户图形界面类库的包是( )。
- A. **java.awt**      B. **java.lang**      C. **java.util**      D. **java.io**
18. 下列关于组件类的描述中, 错误的是( )。
- A. **Swing** 组件通常被称为“重量级组件”, 它完全是由 **java** 语言编写的。
  - B. 标签 (**Label**) 和按钮 (**Button**) 是组件类 (**Component**) 的子类
  - C. 面板 (**Panel**) 和窗口 (**Window**) 是容器类 (**Container**) 的子类
  - D. 文本框(**TextField**)和文本域 (**TextArea**)是文本组件类(**TextComponent**)的子类
19. 下列各种布局管理器中, **Window** 类和 **Frame** 类的默认布局是( )。
- A. **FlowLayout**    B. **CardLayout**    C. **BorderLayout**    D. **GridLayout**
20. 下列关于事件监听相关说法, 错误的是( )。

- A. Swing 事件模型中由 3 个分离的对象完成对事件的处理，分别为事件源、事件以及监听程序
- B. 按钮动作获取事件监听时需要实现 ActionListener 接口
- C. 所有的事件源都具有 addXXXListener() 和 removeXXXListener() 方法
- D. 按钮对象的事件处理不需要重写 actionPerformed() 方法

21. 下列关于 try-catch-finally 语句的描述中，错误的是( )。

- A. 异常产生后，如果不做任何处理，程序将会终止，java 结构化异常处理机制，使得程序不会因为一个异常而影响整个程序的执行。
- B. catch 语句块用来获取异常信息，当 try 语句中发生异常时，程序会自动跳转到 catch 代码块中执行，执行完毕后，不会再执行 try 代码块中发生异常语句后的代码。
- C. catch 语句块有一个参数，该参数是某种异常类的对象
- D. finally 语句块总是被执行的，该语句块起到提供统一出口的作用

22. 下列有关多线程的描述，错误的是( )。

- A. 线程是进程中的执行流程，一个进程中可以同时包括多个线程
- B. Java 中只提供了一种方式来实现线程，即：继承 java.lang.Thread 类
- C. 多个线程可以并发执行
- D. 完成线程真正功能的代码放在类的 run() 方法中，start() 方法用来启动线程

23. 在下列概念中，Java 语言只保留了 。

- A. 运算符重载
- B. 方法重载
- C. 指针
- D. 结构和联合

24. 下列关于 Java 语言特性的描述中，错误的是 。

- A. 支持多线程操作
- B. Java 程序与平台无关
- C. Java 程序可以直接访问 Internet 上的对象
- D. 支持单继承和多继承

25. 下列关于 Java Application 程序结构特点的描述中，错误的是 。

- A. Java 程序是由一个或多个类组成的
- B. 组成 Java 程序的若干个类可以放在一个文件中，也可以放在多个文件中
- C. Java 程序的文件名要与某个类名相同
- D. 组成 Java 程序的多个类中，有且仅有一个主类

26. Java 程序经过编译后生成的文件的后缀是

- A. .obj B. .exe C. .class D. .Java

27. 下列关于运行字节码文件的命令行参数的描述中，正确的是 。

- A. 第一个命令行参数(紧跟命令字的参数)被存放在 args[0] 中
- B. 第一个命令行参数被存放在 args[1] 中

C 命令行的命令字被放在 `args[0]` 中

D 数组 `args[]` 的大小与命令行参数的个数无关

28. 下列有关异常的描述中，错误的是：

A 异常是一种经过修正后程序仍可执行的错误

B 异常是一种程序在运行中出现的不可恢复执行的错误

C 不仅 `java` 语言有异常处理，`c++` 语言也有异常处理

D 出现异常不是简单结束程序，而是执行某种处理异常的代码，设法恢复程序的

29. 下列关于异常处理的描述中，错误的是：

A、程序运行时异常由 `java` 虚拟机自动进行处理

B、使用 `try-catch-finally` 语句捕获异常

c、使用 `throw` 语句抛出异常

D、捕获到的异常只能用当前方法中处理，不能用其他方法中处理

30. 下列关于 `try-catch-finally` 语句的描述中，错误的是：

A `try` 语句后面的程序段将给出异常处理的语句

B `catch()` 方法跟在 `try` 语句后面，它可以，是一个或多个

C `catch()` 方法有一个参数，该参数是某种异常类的对象

D `inally` 语句后面的程序段总是被执行的，该语句起到提供统一接口的作用

31. 下列关于抛出的异常的描述中，错误的是：

A 捕获到发生的异常可在当前的方法中处理，也可以抛到调用该方法的方法中处理

B 在说明要抛出异常的方法时应该加关键字

C (异常列表) 中可以有多个逗号用于分隔的异常

D 抛出异常的方法中要使用下述抛出异常的语句：`throw (异常名)`；其中，(异常名) 是异常类的类名

32. 下列关于用户创建自己的异常的描述中，错误的是：

A 创建自己的异常应该先创建一个异常类

B 为实现抛出异常，需在可能的抛出异常的方法中书写 `throw` 语句

C 捕获异常的方法是使用 `try-catch-fianlly` 语句格式

D 使用异常处理不会使整个系统更加安全和稳定

33. 下面的关键词和方法\_\_\_\_\_，不是线程同步所需要的。

A.`synchronized`    B.`wait()`    C.`notify()`    D.`sleep()`

34. 当对线程进行调度使其暂停时，暂停的时间单位是\_\_\_\_\_。

A.毫秒    B.秒    C.分钟    D.小时

35. 主线程属于\_\_\_\_\_。

A.守护线程    B.用户线程    C.`Thread` 线程    D.`Runnable` 线程

36. \_\_\_\_\_方法可以让某个线程等待其他线程的执行结束。

A.sleep()                  B.wait()                  C.notify()                  D.join()

37. 为了使线程处于暂停状态，可引用\_\_\_\_\_方法。

A.wait()                  B.sleep()                  C.join()                  D.notify()

38. \_\_\_\_\_方法是现在的 Java 版本建议使用的方法。

A. suspend()                  B.resume()                  C.stop()                  D.start()

39. Applet 类属于\_\_\_\_\_包的。

A Java.awt                  B.java.applet                  C.java.io                  D.java.lang

40. 下列关于 Applet 程序的描述中，错误的是\_\_\_\_\_。

- A.Applet 程序的主类必须是 Applet 类的子类
- B.Applet 程序的主类应该有一个 main () 方法
- C.Applet 不是完整的独立程序
- D.Applet 的字节码文件必须镶嵌在一个 HTML 文件中

41. 下列关于 Applet 程序的描述中，错误的是\_\_\_\_\_。

- A.将编辑好的 Java 源文件，使用 javac 命令生成字节码文件。
- B.将字节码文件嵌入 HTML 文件，并且存放在一个 Web 服务器中
- C.使用浏览器解释 HTML 文件中的标记
- D.使用编译器直接执行嵌套在 HTML 文件中的字节码文件

42. 在 Applet 类的主要方法中，用来实现初始化操作的是\_\_\_\_\_。

A.init()                  B.Stop()                  C.start()                  D.paint()

43. 下列关于 HTML 文件的描述中，错误的是\_\_\_\_\_。

- A .HTMLI 文件是使用一种超文本标记语言（HTML）书写的文件
- B.HTML 文件中使用的标记通常是用一对尖括号括起的，两个标记之间是正文内容
- C.HTML 文件中没有单独出现的标记
- D.HTML 中不区分大小写

44. 下列关于向 Applet 程序传递参数的描述中，错误的是\_\_\_\_\_。

- A.Applet 程序可以通过命令行获取外部参数
- B.Applet 程序可以通过 HTML 文件获取外部参数
- C.使用 Applet 标记中的 PARAM 标志来实现
- D.Applet 程序中使用 getParamrter()方法读取参数值

45. 下列关于字体的描述中，错误的是\_\_\_\_\_。

- A.Font 类提供了创建字体对象的方法
- B.字体风格(字型)使用了 3 个静态常量表示
- C.表示字型的字体风格只能单独使用，不可以组合
- D.字号表示字的大小，用点表示，一个点为 1/72 英寸

46. 下列用来获取当前颜色的方法是\_\_\_\_\_。
- A. getColor() B. setColor() C. getRed() D. Color()
47. 下列各种绘制矩形的方法中，绘制实心矩形的方法是\_\_\_\_\_。
- A. fillRect() B. drawRect() C. clearRect() D. drawRoundRect()
48. 下列演示图像的描述中，错误的是\_\_\_\_\_。
- A.使用图类 Image 定义图像对象  
B.使用方法 getImage()获取图像信息  
C.使用方法 drawImage()显示图像  
D.不可使用显示图的方法进行缩放
49. 关于下列容器的描述中，错误的是\_\_\_\_\_
- A.容器是由若干个组件和容器组成的  
B.容器是对图形界面中界面元素的一种管理  
C.容器是一种指定宽和高的矩形范围  
D.容器都是可以独立的窗口
50. 下列界面元素中，不是容器的是\_\_\_\_\_。
- A. List B. Frame C. Dialog D. Panel
51. 下列关于实现图形用户界面的描述中，错误的是\_\_\_\_\_。
- A.放在容器中的组件首先要定义，接着要初始化  
B.放在容器中的多个组件是要进行布局的、默认布局策略是 FlowLayout  
C.容器中所有组件都是事件组件，都可产生事件对象  
D.事件处理是由监听者定义的方法来实现的
52. 下列关于组件类的描述中，错误的是\_\_\_\_\_。
- A.组件类中包含了文本组件类(TextComponent)和装零组件类(MenuComponent)  
B.标签(Label)和按钮(Button)是组件类(Component)的子类  
C.面板(Panel)和窗口(Window)是容器类(Container)的子类  
D.文本框(TextField)和文本区(TextArea)是文本组件类(TextComponent)的子类
53. 在对下列语句的解释中错误的是\_\_\_\_\_。
- A. but 是某种事件对象,如按钮事件对象  
B. this 表示当前容器  
C. ActionListener 是动作事件的监听者  
D. 该语的功能是将 but 对象注册为 tthis 对象的监听者
54. 所有事件类的父类是\_\_\_\_\_。
- A.ActionEvent B.AwtEvent C.KeyEvent D.MouseEvent
55. 所有 GUI 标准组件类的父类是\_\_\_\_\_。
- A.Button B. List C. Component D. Container

56. 下列各种布局管理器中，Window 类、Dialog 类和 Frame 类的默认布局是。

A FlowLayout B. CardLayout C. BorderLayout D. GridLayout

57. 在下列各种容器中，最简单的无边框的又不能移动和缩放的只能包含在另一种容器中的容器是。

A. Window B. Dialog C. Frame D. Panel

58. 下列关于菜单和对话框的描述中，错误的是。

A.Frame 容器是可以容纳菜单组件的容器

B.菜单条中可包含若干个菜单，菜单中又可包含若干菜单项，菜单项中还可包含菜单子项

C.对话框与 Frame 一样都可作为程序的最外层容器

D.对话框内不含有菜单条，它是由 Frame 弹出

59. 下列对封装性的描述中，错误的是。

A 封装体包含了属性和行为

B 封装体中的属性和行为的访问权限是相同的

C 被封装的某些信息在封装体外是不可见的

D 封装使得抽象的数据类型提高了可重用性

60. 下列关于继承性的描述中，错误的是。

A 一个类可以同时生成多个子类

B 子类继承了父类的所有成员

C JAVA 语言支持单重继承和多重继承

D JAVA 语言通过接口可使子类使用多个父类的成员

61. 下列对多态性的描述中，错误的是。

A JAVA 语言允许运算符重载

B JAVA 语言允许方法重载

C JAVA 语言允许变量覆盖

D 多态性提高了程序的抽象性和简洁性

62. 在类的修饰符中，规定只能被同一包类所使用的修饰符是。

A public B 默认 C final D abstract

63. 在成员变量的修饰符中，规定只允许该类自身访问的修饰符是。

A private B public C 默认 D protected

64.在成员方法的访问控制修饰符中，规定访问权限包含该类自身、同包的其他类和其他包的该类的子类的修饰符是。

A public B private C 默认 D protected

65. 下列关于构造方法的描述中，错误的是。

A 不可重载

B 方法名同类名

C 无返回类型

D 系统自动调用

66. 下列关于静态方法的描述中，错误的是。

- A 在类体内说明静态方法使用关键字 `static`
- B 静态方法只能处理静态变量或调用静态方法
- C 静态方法不占用对象的内存空间，非静态方法占用对象的内存空间
- D 静态方法只能用类名调用

67. 下列对静态初始化器的描述中，错误的是 。

- A 静态初始化器是用来对类进行初始化的，而不是对某个对象初始化
- B 静态初始化器是由关键字 `static` 和一对花括号组成的语句组
- C 静态初始化器不同于构造方法，它不是方法
- D 静态初始化器是产生新对象时，由系统自动调用的

68. 下列关于抽象类的描述中，错误的是 。

- A 抽象类是用修饰符 `abstract` 说明的
- B 抽象类是不可以定义对象的
- C 抽象类是不可以有构造方法的
- D 抽象类通常要有它的子类

69. 下列关于类的继承性的描述中，错误的是 。

- A 继承是在已有类的基础上生成新类的一种方法
- B 子类继承父类的所有成员
- C JAVA 语言要求一个子类只有一个父类
- D 父类中成员的访问权限在子类中将被改变

70. 下列关于子类继承父类的成员的描述中，错误的是 。

- A 子类中继承父类中的所有成员都可以直接访问
- B 子类中定义有与父类同名变量是，子类继承父类的操作中，使用继承父类的变量；子类执行自己的操作中，使用自己定义的变量
- C 当子类中出现成员方法头与父类方法头相同的方法时，子类成员方法覆盖父类中的成员方法
- D 方法重载是编译时处理的，而方法覆盖是在运行时处理的

71. 下列关于接口的描述中，错误的是 。

- A.接口实际上是由常量和抽象方法构成的特殊类；
- B.一个类只允许继承一个接口；
- C.定义接口使用的关键字是 `interface`；
- D.在继承接口的类中通常要给出接口中定义的抽象方法的具体实现；

72. 下列关于包的描述中，错误的是 。

- A.包是一种特殊的类；
- B.包是若干个类的集合；
- C.包是使用 `package` 语句创建的；
- D.包有有名包和无名包两种；

73. 下列常用包中，存放用户图形界面类库的包是 。



A java.awt      B java.lang      C java.util      D java.io

74. 下列是系统提供的常用的类，是所有类的父类的类是

A Math      B Object      C System      D String

75. Java 语言所用的字符集中字符是      位。

A.8    B.16    C.32    D.64

76. 下列关于标识符的描述中，正确的是      。

- A.标识符中可以使用下划线和美元符；
- B.标识符中可以使用连接符和井号符；
- C.标识符中大小写字母是无区别的；
- D.标识符可选用关键字；

77. Java 语言和各种分隔符中，非法的是      。

A.空白符    B.分号    C.逗号    D.问号

78. 下列是 Java 语言中可用的注释，其中错误的是      。

A.//      B./\* ... \*/      C./\*\* ... \*\*/      D./\*\* ... \*/

79. Java 语言中字符型数据的长度是      位。

A.8      B.16      C.32      D.64

80. 下列描述的两种数据类型的长度不相等的是      。

- A.字符型和布尔型；
- B.字节型和布尔型；
- C.短整型和字符型 ；
- D.整型和单精度浮点型；

81. 下列是 Java 语言中的复合数据类型，其中错误的是      。

A.数组    B.接口    C.枚举    D.类

82. 下面是关于 Java 语言中数据类型的强制转换的描述，其中错误的是      。

- A.当数据类型从高转换为低时需强制转换；
- B.强制转换时使用强制转换去处符，形如 (<类型>)；
- C.浮点型数据强制转换为整型时，小数部分四舍五入；
- D.赋值操作中不都要采用强制类型转换；

83. 下列关于定义符号常量的描述中，错误的是      。

- A.定义符号常量使用关键字 const；
- B.定义符号常量时要赋初值 ；
- C.符号常量既可以定义为类成员，又可在方法中定义；
- D.定义符号常量时必须给出数据类型；

84. 下列关于变量的默认值的描述，其中错误的是 ☐ 。
- A. 定义变量而没有进行初始化时，该变量具有默认值；
  - B. 字符型变量的默认值为换行符；
  - C. 布尔型变量的默认值是 `false`；
  - D. 变量的默认值是可以被改变的；
85. 下列关于变量定义的描述中，正确的是 ☐ 。
- A. 定义变量时至少应指出变量名字和类型；
  - B. 定义变量时没有给出初值该变量可能是无意义值；
  - C. 定义变量时，同一个类型多个变量间可用分号分隔；
  - D. 定义变量时必须给变量初始化；
86. 下列关于变量作用域的描述，错误的是 ☐ 。
- A. 在某个作用域定义的变量，仅在该作用域内是可见的，而在该作用域外是不可见的；
  - B. 在类中定义的变量的作用域在该类中的方法内可以使用；
  - C. 在方法中定义的变量的作用域仅在该方法体内；
  - D. 在方法中作用域可嵌套，在嵌套的作用域中可以定义同名变量；
87. 下列关于增 1 (++) 减 1 (--) 运算符的描述中，正确的是 ☐ 。
- A. 增 1 减 1 运算符都是单目运算符
  - B. 增 1 减 1 运算符都只有前缀运算和后缀运算
  - C. 增 1 减 1 运算符不会改变变量的值
  - D. 增 1 减 1 运算符前缀运算和后缀运算后表达式值是相同的
88. 关于运算符优先级的描述中，错误的是 ☐ 。
- A. 在表达式中，优先级高的运算符先进行计算；
  - B. 赋值运算符优先级最高；
  - C. 单目运算符优先级高于双目和三目运算符；
  - D. 逻辑运算符优先级高于逻辑位运算符；
89. 下列关于表达式的描述中，正确的是 ☐ 。
- A. 任何表达式都有确定的值和类型；
  - B. 算术表达式的类型由第一个操作数的类型决定；
  - C. 逻辑表达式的操作数可以是任何形式的数据；
  - D. 赋值表达式的类型取决于右值表达式的类型；
90. 下列表达式中，非法的是 ☐ 。 `int a=5,b=6;double c=1.1,d=2.2`
- A. `a+c+++d`
  - B. `a--`
  - C. `c<<b`
  - D. `(a!=b)?c:d`
91. 下列关于条件语句的描述中，错误的是 ☐ 。
- A. `if` 语句可以有多个 `else` 子句和 `else if` 子句；
  - B. `if` 语句中可以没有 `else` 子句和 `else if` 子句；
  - C. `if` 语句中的〈条件〉可以使用逻辑表达式；
  - D. `if` 语句的 `if` 体、`else` 体内可以有循环语句；

92. 下列关于开关语句的描述中，正确的是\_\_\_\_\_。
- A. 开关语句中，**default** 子句可以省略
  - B. 开关语句中，**case** 子句的〈语句序列〉中一定含有 **break** 语句
  - C. 开关语句中，**case** 子句和 **default** 子句都可以有多个
  - D. 退出开关语句的唯一条件是执行 **break** 语句
93. 实现字符类的读操作类是\_\_\_\_\_。
- A. `FileInputStream`    B. `FileOutputStream`    C. `FileReader`    D. `Writer`
94. 进行文件操作的类是\_\_\_\_\_。
- A. `Reader`    B. `FileInputStream`    C. `FileWriter`    D. `Writer`
95. 可实现对基本数据类型的读操作类是\_\_\_\_\_。
- A. `FileReader`    B. `DataInputStream`    C. `DataOutput`    D. `FileWriter`
96. 随机文件访问是有\_\_\_\_\_类实现的。
- A. `File`    B. `BufferedInputStream`    C. `RandomAccessFile`    D. `FileWriter`
97. 可用于获得文件或目录名的路径名是\_\_\_\_\_。
- A. `File`    B. `RandomAccessFile`    C. `FileInputStream`    D. `BufferedReader`
98. 可提供管道连接的作用类是\_\_\_\_\_。
- A. `PipedWriter`    B. `OutputStream`    C. `FileReader`    D. `SequenceInputStream`
99. 用于关闭流的方法是\_\_\_\_\_。
- A. `close()`    B. `flush`    C. `read()`    D. `write()`
100. 若对 web 页面进行操作，一般会用到的类是\_\_\_\_\_。
- A. `URLs`    B. `URLConnection`    C. `Socket`    D. `DatagramSocket`
101. IP 地址或域名是由\_\_\_\_\_类来表示的。
- A. `URL`    B. `InetAddress`    C. `NetworkInterface`    D. `Socket`
102. 若访问网卡信息，需要用到的类是\_\_\_\_\_。
- A. `URL`    B. `Network`    C. `NetworkInterface`    D. `InetAddress`
103. 在套接字编程中，客户方需要用到 Java 类\_\_\_\_\_ 来创建 TCP 连接。
- A. `Socket`    B. `URL`    C. `ServerSocket`    D. `DatagramSocket`
104. 在套接字编程中，服务器方需要用到 Java 类\_\_\_\_\_ 来监听端口。
- A. `Socket`    B. `URL`    C. `ServerSocket`    D. `DatagramSocket`
105. 在 UDP 通信中，接收和发送数据报要用到的类是\_\_\_\_\_。

A.Socket B.DatagramSocket C.DatagramPacket D.MulticastSocket

106. 在 UDP 通信中, 数据报是由\_\_\_\_\_类来创建的。

A.Packet B. DatagramPacket C.DatagramSocket D.MulticastSocket

107. 在安全套接字编程中, 客户方一般用\_\_\_\_\_类来创建安全的套接字对象。

A.SSLSocket B.SSLSocketFactory C.SSLServerSocketFactory D.SSLServerSocket

108. \_\_\_\_\_可实现有序的对象操作。

A.HashMap B.HashSet C.TreeMap D.LinkedList

109. 在默认的情况下就实现了同步控制的类是\_\_\_\_\_。

A.Vector B.ArrayList C.HashMap D.Treelist

110. 下列不是迭代器 (Iterator) 接口所定义的方法是\_\_\_\_\_。

A.hasNext () B.next () C.remove () D.nextElement ()

111. 关于链表结构, 陈述错误的是\_\_\_\_\_。

A.在链表中查找对象是最有效的 B.链表课动态增长  
C.链表中每一个元素都有其前后元素的链接 D.链表中元素可以重复

112. Collections 类不可以对集合对象进行\_\_\_\_\_操作。

A.只读 B.同步 C.排序 D.删除

113. java 语言的集合框架类定义在\_\_\_\_\_语言包中。

A.java.util C.java.lang C.java.array D.java.collections

114. 下列关于循环语句的描述中, 错误的是\_\_\_\_\_。

A.任何一种循环体内都可以包含一种循环语句  
B.循环体内可以出现多个 break 语句  
C.循环体可以是空语句, 也可以是空  
D.循环语句中, 循环体至少被执行一次或不执行

115. 下列循环语句的循环次数是\_\_\_\_\_。

```
int i=5;
do{
    System.out.println(i--);
    i--;
}while(i!=0);
```

A.0 B.1 C.5 D.无限

116. 下列关于 Java 语言的数组描述中, 错误的是\_\_\_\_\_。

A.数组的长度通常用 length 表示;  
B.数组下表从 0 开始;

- C.数组元素是按顺序存放在内存的;
- D.数组在赋初值和赋值时都不判越界;

117. 下列关于数组的定义形式, 正确的是\_\_\_\_\_。

- A.int [ ]a; a=new int;
- B.char b[ ]; b=new cahr[80];
- C.int [ ]c=new cahr[10];
- D.int [ ]d[3]=new int[2][ ];

118. 下列关于字符串的描述中, 错误的是\_\_\_\_\_。

- A.Java 语言中, 字符 `cause` 分为字符串常量和字符串变量两种
- B.两种不同的字符串都是 `String` 类的对象
- C.Java 语言中不再使用字符数组存放字符串
- D.Java Application 程序的 `main ( )` 的参数 `args[ ]` 是一个 `String` 类的对象数组, 用它可以存放若干个命令行参数

## 二、 填空题

1. 对象具有 3 大要素, 它们分别是: \_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_。这三大变量分别用\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_表示。
2. 定义类时, 通常要使用的关键字有\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_。
3. 成员变量的定义格式如下:  
\_\_\_\_\_ <变量类型> <变量名> = \_\_\_\_\_;
4. 成员变量的非访问控制修饰符有\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_。
5. 在成员方法的定义中可以忽略的部分有\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_。
6. Java 语言中, 重载方法的选择是在\_\_\_\_\_时进行的, 系统根据\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_寻找匹配方法。
7. 没有子类的类称为\_\_\_\_\_, 不能被子类重载的方法称为\_\_\_\_\_, 不能被改变的值得量称为常, 又称为\_\_\_\_\_。
8. 创建一个对象时, 通常要包含 3 项内容: \_\_\_\_\_, 给对象分配空间, \_\_\_\_\_。
9. 系统规定 `int` 型变量默认值为\_\_\_\_\_, 浮点型变量的默认值为\_\_\_\_\_, 布尔型变量的默认值为\_\_\_\_\_。
10. Java 语言中, 定义子类时, 使用关键字\_\_\_\_\_来给出父类名。如果没有指出父类名, 则该类的默认父类为\_\_\_\_\_。
11. 系统规定: 表示当前类的构造方法用\_\_\_\_\_, 表示直接父类的构造方法用\_\_\_\_\_。
12. 接口中默认的成员变量的修饰符为\_\_\_\_\_, 默认的成员方法的修饰符是\_\_\_\_\_。
13. 实现接口中的抽象方法时, 必须使用\_\_\_\_\_方法头, 并且还要用\_\_\_\_\_修饰符。
14. 包名与子包名之间用\_\_\_\_\_分隔。加载包的关键字是\_\_\_\_\_。
15. 字符串类又包含两大类, 它们是\_\_\_\_\_和\_\_\_\_\_。

## 三、 判断题

1. Java 语言具有较好的安全性和可移植性及与平台无关等特性。
2. Java 程序编译后生成的是字节码文件, 其后缀是 `.exe`
3. 数据由高类型转换到低类型时, 采用强制转换, 数据精度要受到损失。
4. 常用的集合有 `List` 集合、`Set` 集合和 `Map` 集合。

5. 某类的对象可以作为另一个类的成员。
6. `InputStream` 类是字节输入流的抽象类，是所有字节输入流的父类。
7. 容器中只能包含有组件，而不能再含有容器。
8. 类中不可以再定义类。
9. 线程具有生命周期，包含出生状态、就绪状态、运行状态、等待状态、休眠状态、阻塞状态和死亡状态。
10. 异常处理是在编译时进行的。
11. Java 语言是在 C++语言之前问世的。
12. Java 语言具有较好的安全性和可移植性及与平台无关等特性
13. Java 语言中取消了联合概念，保留了结构概念。
14. Java 语言中数据类型占内存字节数与平台无关。
15. Java 语言的源程序不是编译型的，而是编译解释型的。
16. 操作系统中进程和线程两个概念是没有区别的。
17. Java 语言既是面向对象的又是面向网络的高级语言。
18. Java 程序分为两大类：一类是 `Java Application` 程序，另一类是 `Applet` 程序。前者 又称 `Java` 应用程序，后者又称 `Java` 小应用程序。
19. `Java Application` 程序是由多个文件组成的，其中可以有也可以没有主文件。
20. 组成 `Java Application` 的若干类中，有过仅有一个主类，只有主类中含有主方法 `Main()`
21. `Java Application` 程序中，必有一个主方法 `main()`，该方法有没有参数都可以。
22. Java 程序中是不区分大小写字母的。
23. 下述两个命令：`Javac` 和 `Java` 是不同的，编译时用前者，运行时用后者。
24. Java 程序编译后生成的是字节码文件，其后缀是 `.exe`。
25. 字节码文件机器是不能直接识别的，它要经过解释器，边解释边执行。
26. 在 Java 程序中，使用 `import` 语引入类是在解释器中实现的。
27. 在运行字节码文件时，使用 `Java` 命令，一定要给出字节码文件的扩展名(class)
28. 如果一个 Java 程序中有多个类，编译后只生成一个字节码文件，其名字同主类名。Java 程序中出现的输出方法 `println()`和 `print()`是完全一致的
29. 异常是一种特殊的运行错误的对象。
30. 异常处理可以使整个系统更加安全稳定。
31. 异常处理在编译时进行。
32. java 语言中的异常类都是 `java.lang.Throwable` 的子类。
33. `Throwable` 类有两个子类：`Error` 类和 `Exception` 类。前者由系统保留，后者供应用程序使用。
34. `Exception` 类只有一个子类 `RuntimeException`
35. 在异常处理中，出现异常和抛出异常是一回事
36. 运行异常是在运行时系统检测并处理的
37. 使用 `try-catch-finally` 语句只能捕获一个异常
38. 捕获异常时 `try` 语句后面通常跟有一个或多个 `catch ( )` 方法用来处理 `try` 块内生成的异常事件
39. 使用 `finally` 语句的程序代码为该程序提供一个统一的出口
40. 抛出异常的方法说明中要加关键字 `throws`，并在该方法中还应该添加 `throw` 语句
41. 创建异常类是要给出该异常类的父类
42. 如果异常类没有被捕获将会产生不正常的终止
43. 线程的启动是通过引用其 `start()`方法而实现的。

44. 当线程类所定义的 `run ()` 方法执行完毕，线程的运行就会终止。
45. 线程组的作用是将多个线程作为一个整体来进行控制。
46. 死锁的产生是因为多个线程间存在资源竞争。
47. 若所有的用户线程都终止了，Java 程序就会结束。
48. 线程优先级代表了线程的执行顺序。
49. 关键词 `synchronized` 只能对方法进行修饰。
50. Java 语言标识符中可以使用美元符。
51. Java 语言标识符中大小写字母是没区别的。
52. 分号 (;)、逗号 (,)、冒号 (:) 都可作为 Java 语言中的分隔符。
53. 文档注释符 `/**...*/` 是 Java 语言特有的注释符。
54. Java 语言的基本数据类型有 4 种：整型、浮点型、字符型和布尔型。
55. Java 语言的复合数据类型有 3 种：数组、类和包。
56. Java 语言中，字符型与字节型是一样的。
57. Java 语言是一种强类型语言，数据类型转换有两种：隐含转换和强制转换。
58. 数据由高类型转换到低类型时，采用强制转换，数据精度要受到损失。
59. 布尔型常量可以自动转换为短整型常量。
60. Java 语言使用的字符串中不隐含结束符。
61. Java 语言使用的是 Unicode 字符集，每个字符在内存中占 8 位。
62. Java 语言定义符号常量使用 `final` 关键字。
63. Java 语言中不同数据类型的长度是固定的，不随机器硬件不同而改变的。
64. Java 语言只有无符号的字符型。
65. 字符型变量中只存放一个字符。
66. 定义变量时必须进行初始化，否则变量具有无意义的值。
67. 若定义变量时不进行初始化，则该变量一定具有默认值。
68. Java 语言中，变量出现的位置只有一种，即为类体内定义的成员变量。
69. Java 语言规定在嵌套的程序块中不允许定义同名变量。
70. 求余运算符 `%` 可用于整数和浮点数。
71. 关系运算符组成的表达式是逻辑表达式。
72. 逻辑表达式 `&` 和 `&&` 是无区别的，都具有逻辑与的功能。
73. 运算符 `&` 是逻辑与运算符还是按位与运算符取决于操作数的类型。
74. 运算符 `>>` 和 `>>>` 都是右移位运算符，其区别在于对向右移出的位是舍弃还是保留。
75. 赋值运算符组成的赋值表达式中，通常将右边表达式的类型转换为左边变量的类型后再赋值。
76. 三目运算符的 3 个操作数中，第一个操作数的类型必须是逻辑型的。
77. 强制类型转换运算符的功能是将一个表达式的类型转换为所指定的类型。
78. 对象运算符 `instanceof` 是一个单目运算符，其表达式的值是类的对象。
79. 内存分配运算符 `new` 可以为创建的数组分配空间，但不能为创建的变量分配内存空间。
80. 运算符的优先级中，最低的是三目运算符。
81. 运算符的结合性中，除了三目运算符是自右向左的，其余都是自左向右。
82. 关系表达式和逻辑表达式的值都是布尔型的。
83. 赋值表达式的值不能是逻辑型的。
84. 条件表达式的类型总是 “?” 和 “:” 之间的操作数的类型。
85. 块语句可以嵌套，外层块语句定义的变量在内层块语句中是可见的。
86. 条件语句中的 `<条件>` 表达式可以是任何表达式。

87. 在条件语句的嵌套结构中，一个 if 子句最多有一个 else 子句与它配对，而且一定离它最近的。
88. 开关语句中，case 子句后面的<语句序列>可以是块语句。
89. 循环语句中，<条件>表达式必须是逻辑型表达式。
90. 在 Java 语言中，break 语句不能作用在<标号名>的左边。
91. Java 语言中，数组在静态和动态赋值时都判越界。
92. 说明或声明数组时不分配内存大小，创建数组时分配内存大小。
93. 基本数据类型的数组在创建时系统将指定默认值。
94. 数组在定义时可以进行初始化，使用初始值表。
95. 数组在定义时可以进行初始化，使用初始值表。
96. Java 语言中数组元素只有下标表示，没有指针表示。
97. 创建数组时，系统自动将数组元素个数存放在 length 变量中，可供用户对数组操作时使用。
98. 字符串可分为字符串变量和字符串常量两种，它们都是对象。
99. Java 语言中不使用字符数组存放字符串。
100. Applet 程序是通过浏览器中内置的 Java 解释器来解释执行的。
101. Applet 程序是以.java 为扩展名的 Java 语言源程序，该程序经过编译器后便成
102. 嵌入到 HTML 文件中的是 Applet 的源程序。
103. 运行 Applet 程序是使用 AppletViewer 命令运行嵌入了字节码文件的 HTML
104. Applet 程序中的主类必须是 Applet 类的子类。
105. Applet 类是 Java 语言类库中的一个重要的系统类，它被存放在 java.awt 包中。
106. init()方法是用来完成初始化操作的，在 Applet 程序运行期间只执行一次。
107. start()方法被系统自动调用来启动主线程运行。通常在 Applet 程序被重新启
108. paint()方法是在需要重画图形界面时被系统自动调用来显示输出结果的。
109. stop()方法是用来暂停执行操作的，它与 Start()方法不同，只被调用一次。
110. Init()、start()、stop()和 destroy()4 个方法构成 Applet 程序的生命周期。
111. HTML 语言中<HEAD>和是用来表示 HTML 文件开始和结束的标记。
112. HTML 中多数标记是成对出现的，也有不成对出现的标记。
113. HTML 中是区分大小写字母的。
114. 通过 HTML 文件中使用的 PARAM 标记可以向 Applet 程序传递参数的。
115. AWT 提供了许多标准的 GUI 组件和布局管理等类。
116. Java 语言采用 16 位颜色标准。Java 的调色板保证 128 色。
117. Graphics 类提供了 3 种绘制文本的方法，其方法名都是 drawstring()。
118. 绘制椭圆的方法是 drawOval()，使用该方法也可以绘制圆。
119. AWT 是抽象窗口工具包的英文缩写。
120. 容器中只能包含有组件，而不能再含有容器。
121. 容器分可以独立的窗口和不可以独立的窗口两种。
122. 单选按钮提供“多选一”的输入选择，下拉列表提供“多选多”的输入选择。
123. 所有容器的默认布局都是 FlowLayout。
124. 所有的组件都是事件源。
125. Java 的事件处理方法是采用委托事件处理模型。
126. 事件监听者除了得知事件的发生外，还应调用相应方法处理事件。
127. 图形用户界面是由用户自定义成分、控制组件和容器组成的。
128. 所有 GUI 标准组件类的父类是 Component 类。



- 129. 所有容器类的父类是 **Frame** 类。
- 130. 标签和按钮都是事件源。
- 131. 文本区和文本框都可以引发两种事件：文本改变事件和动作事件。
- 132. 每个复选框有两种状态：“打开”和“关闭”状态。
- 133. 复选框组(**CheckboxGroup**)是由若干个按钮组成的。在一组按钮中可选多个。
- 134. 列表(**List**)是由多个选项组成的，只可在列表框的若干个选项选取一个。
- 135. 下拉列表(**Choice**)是一种“多选多”的输入界面。
- 136. 画布(**Canvas**)是一种可容纳多个组件的容器。
- 137. 边界布局(**BorderLayout**)将容器分为东、西、南、北共 4 个区域。
- 138. 卡片布局(**CardLayout**)可使容器容纳多个组件，在同一时刻只显示若干个组件中的一个。
- 139. 容器可容纳组件和容器，同一个容器中的组件可被同时显示或同时隐藏。
- 140. 所有容器都是有边框的。
- 141. **Panel** 容器是可以嵌套使用的。
- 142. **Applet** 实际是一种特殊的 **Panel** 容器。**.JavaApplet** 程序只负责它拥有的 **Applet** 容器内的无边框区域。
- 143. 窗口(**Window**)容器是一种独立窗口，但它不能作为程序的最外层容器。
- 144. **Frame** 容器可以容纳菜单组件，它实现了 **Menucontainer** 接口。
- 145. **MenuComponent** 类是菜单组件的父类。
- 146. 菜单条(**MenuBar**)、菜单(**Menu**)、菜单项(**MenuItem**)是 3 种不同的菜单组件。
- 147. 对话框(**Dialog**)不是一种独立使用的容器
- 148. 文件对话框(**FileDialog**)是对话框(**Dialog**)的子类，它是一种打开文件和保留文件的窗口。
- 149. 类是一种类型，也是对象的模板。
- 150. **JAVA** 语言只支持单重继承，不支持多重继承。
- 151. 类中说明的方法可以定义在类体外。
- 152. **class** 不是定义类的唯一关键字。
- 153. 某类的对象可以作为另一个类的成员。
- 154. 某类的对象可以作为另一个类的成员。
- 155. 在类体内说明成员变量时不允许赋值。
- 156. 最终变量就是 **JAVA** 语言中的符号常量。
- 157. 静态变量的引用只能使用对象。
- 158. 静态方法只能处理静态变量。
- 159. 抽象方法是一种只有说明而无具体实现的方法。
- 160. 最终方法是不能被当前子类重新定义的方法。
- 161. **JAVA** 语言中，方法调用一律都是传址的引用调用。
- 162. 非静态方法中不能引用静态变量。
- 163. 静态初始化器是在构造方法被自动调用之前运行的。
- 164. 抽象方法只有方法头，而无方法体。
- 165. 抽象方法一定出现在抽象类中。
- 166. 最终类、最终方法和最终变量的修饰符都用 **final**。
- 167. 创建对象时系统将调用适当的构造方法给对象初始化。
- 168. 使用运算符 **new** 创建对象时，赋给对象的值实际上是一个地址值。
- 169. 使用构造方法只能给非静态成员变量赋初值。
- 170. 创建对象时，该对象占有的内存空间除了非静态的成员变量外，还有非静态的成员方法。

171. JAVA 语言中，对象成员的表示只使用运算符“.”。
172. JAVA 语言中，对象赋值实际上同一个对象具有两个不同的名字，因为它们都有同一个地址。
173. 对象可作方法参数，对象数组不能作方法参数。
174. JAVA 语言中，所创建的子类都应有一个父类。
175. JAVA 语言中，类的继承是可以传递的。
176. JAVA 语言中，构造方法是可以继承的。
177. 子类中构造方法应包含自身类的构造方法和直接父类的构造方法。
178. 调用 `super` 或 `this` 的构造方法的语句必须放在第一条语句。
179. 子类对象可以直接赋值给父类对象；而父类对象不可以赋值给子类对象。
180. 子类中所继承父类的成员都可以在子类中访问。
181. 成员方法的重载和覆盖是一回事。
182. 一个类可以实现多个接口。接口可以实现“多重继承”。
183. 实现接口的类不能是抽象类。
184. 当 `DataInputStream` 对象读到文件结束处，则返回-1。
185. 文件缓冲流的作用是提高文件的读/写效率。
186. 管道流可在一个线程中使用。
187. 通过 `File` 类可对文件属性进行修改。
188. 通过 `SequenceInputStream` 类可将两个或多个输入流有序的组合为一个输入流。
189. `RandomAccessFile` 对象是通过移动文件指针的方式来进行随机访问的。
190. 套接字是访问系统网络功能的标准接口。
191. 安全套接字是基于 SSL 协议的。
192. 基于 TCP 和基于 UDP 的网络编程不存在任何区别。
193. 安全套接字和普通套接字在流的处理上不存在区别。
194. `DatagramPacket` 对象表示要发送或接收的数据报。
195. IPv6 地址与 IPv4 地址是有区别的。
196. `Arrays` 类只要对数组进行操作。
197. 集合 `Set` 是通过键-值对的方式来存储对象的。
198. `Map` 接口是字 `Collections` 接口的继承而来。
199. “线程安全”是指当多个线程同时对一个集合对象进行修改和读取操作时，可以保证数据的一致性。
200. 在集合中元素类型必须是相同的。
201. 枚举接口中定义了具有删除功能的方法。

#### 四、简答题

1. 什么是对象？什么是类？
2. 对象的三大要素是什么？
3. 如何理解集合框架的含义？
4. Java Applet 在结构上有什么特点？
5. 什么是 HTML 文件？用来嵌套 Applet 的字节码文件的 HTML 文件有什么特点？

6. Applet 程序是如何实现的? (即工作原理)
7. Java 语言是什么时间问世的?它与 Internet 有什么关系?
8. Java 语言有哪些特点?
9. Java 语言取消了 C++语言中的哪些内容?取消了这些内容后,给 Java 语言带来了什么好处?
10. Java 语言的可移植性为什么比较高?
11. Java 语言程序分为哪两种?
12. Java Application 程序在结构上有哪些特点?
- 13.

Java 程序在书写上应注意哪些事项?

14. Java Application 编译后生成什么文件?该文件机器可以直接识别吗?
15. Java Application 程序是如何被运行的
16. Java Application 程序有哪两种输出界面
17. 什么是异常?异常和错误有什么区别?
18. 异常类的父类是什么类?它的子类有哪些?
19. 异常类有哪些常用的方法
20. 异常的处理方法有哪些?
21. 举例说明哪种异常是由系统通过默认的异常处理程序进行处理的?

22. 语句 `try` 和 `catch` 的主要功能是什么？
23. 语句 `finally` 的作用是什么？
24. 什么是抛出异常？如何实现抛出异常？语句 `throw` 的格式如何？
25. 用户 如何创建自己的异常？
26. 自定义的异常类格式如何？
27. 进程和线程的概念是什么？两者有什么区别与联系？
28. 线程有哪两种创建方式？
29. 如何实现线程的同步？
30. 什么是主线程？
31. 锁的含义是什么？如何给对象加锁？
32. 守护线程和用户线程的区别是什么？
33. 什么是线程池？
34. 如何理解死锁？
35. `Applet` 程序和 `Applet` 类有什么关系？
36. 如何实现下列绘制文本和图形的操作？
37. `Java Applet` 在结构上有何特点
38. `Applet` 类有哪些主要方法？下列各种方式的主要功能是什么？

- 39. Applet 的典型结构如何?
- 40. HTML 中常用的标记有哪些?
- 41. 如何使用 HTML 文件向 Applet 传递参数?
- 42. 写出下述英文速锁的中文译文

#### 1.GUI 2.AWT 3.HTML 4.URL

- 43. Java 语言有哪些多媒体特征?
- 44. 如何定义图像? 如何获取图像信息的方法? 如何显示图像?
- 45. 如何实现动画
- 46. 如何进行声音播放?
- 47. 什么是容器?什么是组件?容器有什么特点?
- 48. AWT 是什么含义? 有何特点?
- 49. 常用的图形用户界面标准化组件有哪些?
- 50. 简述实现图形用户界面的方法。
- 51. 在界面元素的类结构中, MenuBar 类的父类是什么?Container 类的父类是什么?Frame 类和 Dialog 类是哪个类的子类?
- 52. 什么是事件?如何进行事件处理?
- 53. Java2 中事件处理机制是什么?简述事件发生后的处理过程。
- 54. 如何注册事件监听者? 55. 简单解释 EventObject 类和 AwtEvent 类的意义。
- 56. 动作事件(ActionEvent)的监听者接口是什么?主要方法是什么?
- 57. 组件和容器有何不同?
- 58. 说明下述基本控制组件的创建方法和相关操作方法。(1)按钮 (2)标签 (3)文本框 (4)文本区 (5)复选框 (6)单选按钮组 (7)列表 (8)下拉列表 (9)滚动条 (10)画布
- 59. 什么是布局设计? 通常使用的布局管理器有哪些? 各种不同布局管理器有什么特点?
- 60. 常用的容器组件有哪些? 它们各自的特点是什么?
- 61. 解释下列装零组件的含义。(1) MenuBar (2) Menu (3) MenuItem (4) CheckboxMenuItem (5) PopupMenu
- 62. 简述菜单组件设计 步骤。
- 63. 说明下列各种容器的特点。(1) Panel (2) Windows (3) Frame (4) Dialog
- 64. 对话框容器的功能是什么?
- 65. 对话框(Dialog)类的构造方法有哪些?
- 66. 什么是对象? 什么是类? 二者有什么关系?
- 67. 对象的三大要素是什么?

68. 类的封装性、继承性和多态性各自的内容是什么？
69. 类的定义格式是什么？JAVA 语言和 C++ 语言中定义类的格式是否相同？
70. 定义类的修饰符有哪些？各自的特点是什么？
71. JAVA 语言中，成员变量的定义格式如何？它有那些修饰符？
72. JAVA 语言中，方法的定义格式如何？方法的修饰符有哪些？
73. 方法调用中应注意什么事项？
74. 如何定义重载方法？如何选择重载方法？
75. 构造方法的特点是什么？作用是什么？
76. 析构方法的特点是什么？作用是什么？
77. 什么是静态变量？什么是静态方法？各自有何特点？
78. 静态初始化器有何功能？如何定义？
79. 什么是抽象类？什么是抽象方法？各自有什么特点？
80. 什么是最终类？什么是最终变量？什么是最终方法？
81. 对象的定义格式如何？如何给对象初始化？如何给对象赋值？
82. 对象通常有哪些应用？
83. JAVA 语言的继承性有何特点？
84. 如何创建一个类的子类？
85. 什么是继承的传递性？
86. 子类的构造方法中应包含哪些内容？
87. 子类和父类对象之间转换的规则是什么？
88. 什么是成员变量的继承？什么是成员变量的覆盖？
89. 什么是成员方法的重载？什么是成员方法的覆盖？

90. 什么是接口？如何定义接口？接口与类有何区别？
91. 如何在类中实现接口？
92. 什么是包？如何创建包？如何引用包？
93. JAVA 语言提供的工具类中包含了哪些常用的包？
94. Object 类、Math 类和 System 类的功能各是什么？有哪些主要方法？
95. 字符串类 String 和 StringBuffer 有何不同？
96. 已知: `int a=3,b=5`; 求下列表达式的值：
- (1) `(a+b) %b`
  - (2) `b>>a`
  - (3) `-b>>>a`
  - (4) `A&b`
  - (5) `++a-b++`
- 已知: `double x=1.5,y=2.8`; 求下列表达式的值：
- (1) `x++>y`
  - (2) `x+4/5`
  - (3) `2*x==y`
  - (4) `y/6*6`
  - (5) `y+=x--`
- 已知: `Boolean b1=true, b2=false; int a=6`; 求下列表达式的值：
- (1) `b1&b2`
  - (2) `!b1&&b2||b2`
  - (3) `b2&b1|!b2`
  - (4) `b1!=b2`
  - (5) `!(a>6)&a6|b2=true)`
- 已知: `int a[]={3,5,7,9}`; 求下列表达式的值：
- (1) `a[0]+a[1]`
  - (2) `a[3]=6`
  - (3) `a[2]*a[3]>4*a[3]`
  - (4) `(a[0]+=a[1])+ ++a[2]`
  - (5) `for(int i=1;i<4;i++) a[0]+=a[i]`; 计算 `a[0]` 的值
100. 什么是流？
101. 输入流和输出流的不同体现在何处？
102. 字节流和字符流有何区别？
103. 节点流和处理流有何区别？
104. 文件的读/写流程是怎样的？
105. 文件缓冲流提供了什么样的功能？
106. 管道流的主要作用是什么？

107. 随机文件访问的含义是什么？
108. 什么是 URL？URL 类的基本操作步骤如何描述？
109. IP 地址和端口的作用是什么？什么是套接字（Socket）？
110. 客户/服务器模式有什么特点？Socket 类和 ServerSocket 类的区别是什么？
111. 安全套接字与普通套接字有什么区别？
112. UDP 通信的特点是什么？如何描述 UDP 数据报的收发流程？
113. 如何理解集合框架的含义？
114. 基本的集合接口有哪几个？
115. 映射、集合和列表的含义是什么？
116. HashMap 类和 TreeMap 类有什么区别？
117. HashSet 类和 TreeSet 类有何区别？
118. ArrayList 类和 LinkedList 类有何区别？
119. 历史集合类包含哪些类？
120. Java 的单词有哪些？Java 语言使用的字符集与 C++ 语言的相同吗？
121. 标识符有何用途？标识符的规则是什么？
122. 分隔符花括号通常用在什么地方？
123. 文档注释与多行注释符有何不同？
124. Java 语言的基本数据类型包含哪几种？
125. 字节型和字符型数据有何区别？
126. Java 语言中基本数据类型的长度是随计算机的不同而不同吗？
127. 长度为 32 位的基本数据类型有哪些？
128. Java 语言中，复合数据类型有哪些？
129. Java 语言中，数据类型转换有哪两种？当数据从低类型转换为高类型时需要用强制类型转换吗？
130. Java 语言中，常量的种类有哪些？
131. 整型常量有哪三种表示形式？浮点型变量有哪两种表示形式？
132. 布尔型常量可以转换成其他数据类型吗？例如整型。
133. 在 Java 语言的转义字符表示中，ASCII 码值对应的字符如何表示？Unicode 字符集中对应的字符如何表示？
134. 在 Java 语言中，表示字符串常量时应该注意哪些问题？
135. 在 Java 语言中，表示字符常量时应该注意什么问题？
136. 字符串变量的含义是什么？
137. Java 语言中，变量的值出来有初值、默认值和改变值外，还有无意义的值吗？
138. Java 语言中，一定变量至少要指出哪些要素？
139. 变量的作用域规则是怎样规定的？
140. Java 语言中的变量有哪两种？
141. Java 语言中是否允许在嵌套的程序中定义同名变量？
142. Java 语言的运算符中按其操作数来划分，可分为哪些种类？举例说明。
143. Java 语言的运算符按其功能通常分为哪些种类？
144. Java 语言的运算符中按其优先级可以划分几类？
145. Java 语言的运算符中按其结合性可以划分为哪些类？
146. Java 语言中，逻辑运算符有哪些？运算符&作为双目运算符，如何区分他是逻辑与还是按位与？
147. 移位运算符中，两种右移运算符>>和>>>有何不同？



148. 对象运算符中 instanceof 如何使用?
149. 内存分配运算符是什么? 功能如何?
150. 类成员使用什么运算符表示?
151. 有哪几种运算符组成的表达式的值是布尔值?
152. 有哪几种运算符组成的表达式的值是算数值?
153. 简洁逻辑运算符与非简洁逻辑运算符组成的表达式, 在求值上有何不同?
154. 什么是表达式语句? 什么是空语句?
155. 什么是块语句? 可以把块语句视为一条语句吗?
156. Java 语言中, 选择语句有哪两种形式?
157. If 语句中, <条件表达式>一定是逻辑型表达式吗?
158. Switch 语句中, <语句序列>里一定要有 break 语句吗?
159. While 循环语句与 do-while 循环语句有何不同?
160. For 循环语句中, 关键字 for 后面括号内的表达式是否可以使用多个逗号分隔表达式?
161. Break 语句和 continue 语句各有哪两种形式?
162. Import 语句和 package 语句的功能分别是什么?
163. 在 Java 语言中, 如何定义一个数组?
164. Java 中定义的数组都可以进行初始化吗?
165. 数组元素为基本数据类型的数组, 在创建时系统都给默认值吗? 布尔型的默认值是什么?
166. Java 中的数组实际上是一个隐含的“数组类”的对象, 而数组名实际是该独享的一个引用, 这种说法对吗?
167. 字符数组和字符串是两回事, 而字符串实际上是 String 类和 StringBuffer 类的对象, 这种说法对吗?
168. 字符串有两种, 分别是哪两种? 它们之间有何不同?

#### 一、 选择题

- 1.C 2.D 3.NO 4.NO 5.NO 6.A 7.C 8.B 9.A 10.D 11.C 12.B 13.C 14.A 15.B  
16.C 17.A 18.A 19.NO 20.NO 21.A 22.A 23.B 24.D 25.C 26.C 27.A 28.D  
29.D 30.A 31.D 32.D 33.D 34.A 35.B 36.D 37.B 38.D 39.B 40.B 41.D 42.A  
43.C 44.A 45.C 46.B 47.D 48.B 49.D 50.A 51.C 52.A 53.C 54.B 55.C 56.C  
57.D 58.C 59.B 60.C 61.A 62.B 63.A 64.D 65.A 66.D 67.D 68.C 69.D 70.A  
71.B 72.A 73.A 74.B 75.B 76.A 77.D 78.C 79.B 80.A 81.C 82.C 83.A 84.B  
85.A 86.D 87.A 88.D 89.A 90.C 91.A 92.A 93.C 94.B 95.B 96.C 97.A 98.A  
99.A 100.B 101.B 102.C 103.A 104.C 105.B 106.B 107.B 108.C 109.A 110.C  
111.A 112.D 113.A 114.B 115.D 116.D 117.B 118.B

#### 二、 填空题

- 1.对象名、对象类型、对象值; 同标识符、类名、默认值或初值 2.class; extends; implements  
3.修饰符; 初值 4.static; final; volatile 5.修饰符; 参数表; 异常类名列表 6.编译; 参数个数; 参数类型; 参数顺序 7.最终类; 最终方法; 最终变量 8.说明对象; 对象初始化 9.0;  
0.0; false 10.extends; object 11.this (); super () 12.final; final 13.完全相同; public 14.;  
import 15.String; StringBuffer

#### 三、 判断题

1.1	2.0	3.1	4.1	5.1	6.1	7.0	8.0	9.1	10.1	11.0	12.0	13.0	14.1	15.1	16.0
17.1	18.1	19.0	20.1	21.1	22.0	23.1	24.0	25.1	26.1	27.0	28.0	29.1	30.1		
31.0	32.1	33.1	34.0	35.0	36.0	37.0	38.1	39.1	40.1	41.1	42.1	43.1	44.1		
45.1	46.1	47.1	48.1	49.0	50.1	51.0	52.1	53.1	54.1	55.0	56.0	57.1	58.1		
59.0	60.1	61.0	62.1	63.1	64.1	65.1	66.0	67.1	68.0	69.1	70.0	71.0	72.0		
73.1	74.0	75.1	76.1	77.1	78.0	79.0	80.0	81.0	82.1	83.0	84.0	85.1	86.0		
87.1	88.1	89.1	90.0	91.1	92.1	93.1	94.1	95.1	96.1	97.1	98.1	99.1	100.1		
101.0	102.0	103.1	104.1	105.0	106.1	107.1	108.0	109.0	110.1	111.0	112.1				
113.0	114.1	115.1	116.0	117.1	118.1	119.1	120.0	121.1	122.0	123.0	124.0				
125.0	126.1	127.1	128.1	129.0	130.1	131.0	132.1	133.0	134.0	135.0	136.0				
137.0	138.1	139.1	140.0	141.0	142.1	143.0	144.1	145.1	146.0	147.0	148.0				
149.1	150.1	151.0	152.0	153.1	154.1	155.0	156.0	157.0	158.1	159.1	160.1				
161.0	162.0	163.1	164.1	165.1	166.1	167.1	168.1	169.1	170.1	171.1	172.1				
173.0	174.1	175.1	176.0	177.1	178.1	179.0	180.0	181.0	182.1	183.1	184.0				
185.1	186.0	187.1	188.1	189.1	190.1	191.1	192.0	193.1	194.1	195.1	196.1				
197.0	198.0	199.1	200.0	201.0											

#### 四、简答题

1. 2. 3. 4. 5. 6. 7. 8. 9. 10. 11. 12. 13. 14. 15. 16. 17. 18. 19. 20.  
21. 22. 23. 24. 25. 26. 27. 28. 29. 30. 31. 32. 33. 34. 35.  
36.

1.设置字体的方法

2.设置颜色的方法

3.绘制文本的方法

4.画直线的方法

5.画矩形的方法（二维和三维）

6.画圆弧和圆锥的方法

7.画多边形的方法

37.

38. 1.init()方法 2.start()方法 3.paint()方法 4.stop()方法 5.destroy()方法