

JAVA 面向对象 程序设计习题集



主编 姚骏屏

目 录

引 言	7
第一章 JAVA 基础	8
一、填空题	8
二、判断题	11
三、选择题	12
四、程序阅读题	31
第二章 控制结构	37
一、选择题	37
二、填空题	54
三、判断题	57
四、程序阅读题	58
五、编程题	63
第三章 字符串	68
一、选择题	68
二、填空题	73
三、判断题	74
四、编程题	75
第四章 数组	77
一、选择题	77
二、判断题	80
三、填空题	82
四、编程题	83
第五章 类和对象	85
一、选择题	85
二、填空题	92
三、程序填空	93
四、问答题	106
五、判断题	106
六、编程题	107
第六章 面向对象程序设计	110
一、选择题	110
二、填空题	134
三、判断题	138
四、编程题	142
第七章 数据库的连接	149
一、选择题	149
二、填空题	151
三、判断题	152

四、编程题	152
第八章 多线程	153
一、填空题	153
二、选择题	154
三、多项选择	169
四、判断题	170
五、程序分析题	171
六、程序设计题	172
第九章 异常处理	175
一、选择题	175
二、填空题	180
三、判断题	183
四、程序阅读题	185
五、程序题	190
第十章 文件操作	191
一、选择题	191
二、填空题	198
三、判断题	202
四、读程序题	206
五、问答题	212
六、编程题	213
第十一章 GUI 界面设计	215
一、选择题	215
二、填空题	226
三、判断题	229
四、程序填空	231
五、简答题	247
六、编程题	248
参考答案	265
第一章 java 基础	265
一、填空题	265
二、判断题	267
三、选择题	267
四、程序阅读题	267
第二章 控制结构	268
一、选择题	269
二、填空题	269
三、判断题	270
四、程序阅读题	270
五、编程题	271

第三章 字符串290

 一、选择题290

 二、填空题291

 三、判断题291

 四、编程题291

第四章 数组301

 一、选择题301

 二、判断题301

 三、填空题301

 四、编程题302

第五章 类和对象305

 一、选择题305

 二、填空题305

 三、程序填空307

 五、判断题310

 六、编程题310

第六章 面向对象程序设计330

 一、选择题330

 二、填空题330

 三、判断题332

 四、编程题332

第七章 数据库的连接392

 一、选择题392

 二、填空题392

 三、判断题392

 四、编程题392

第八章 多线程395

 一、填空题395

 二、选择题396

 三、多项选择396

 四、判断题396

 五、程序分析题396

 六、程序设计题397

第九章 异常处理432

 一、选择题432

 二、填空题432

 三、判断题433

 四、程序阅读题433

 五、程序题434

第十章 文件操作438

一、选择题	438
二、填空题	438
三、判断题	440
四、读程序题	440
五、问答题	440
六、编程题	443
第十一章 GUI 界面设计	460
一、选择题	460
二、填空题	460
三、判断题	462
四、程序填空	462
五、简答题	463
六、编程题	465

引 言

随着教学改革的深入推进，项目制教学模式已经受到我校广大师生的好评，也收到了很好的效果。为了配合我校自编教材的使用，扩大同学们的知识深度和广度，巩固 Java 学习知识点，本书积累了我校教学经验丰富的 JAVA 老师，历年来的教学经验总结，收集了大量 Java 相关试题，供大家参考。

感谢参加本书编写的有金鹏、李再友、谭登超、王丽萍、牟兰、税世兵、贺华川、张曾、杨怡菡、樊豪、陈肖、李虹锋、任晶、黄荣等。

由于编者水平有限，时间仓促，本书错误在所难免，敬请广大读者批评指正。

意见、建议邮箱 xscqfm@163.com

编 者

2010 年 7 月

第一章 JAVA 基础

一、填空题

- 1、_____是唯一标志一个变量 (variable)、方法 (method) 和类 (class) 等的名字。
- 2、标识符是区分_____的，可以为任意长度 (unlimited-length)，但一般来说，长度适中即可。
- 3、标识符以_____、_____或_____开始，并包含这些字符和数字的字符序列。
- 4、标识符不能使用 java 语言定义的_____。
- 5、_____数据类型是不能简化的、内置的数据类型，用以表示真是的数字、字符等，通常用来构建更复杂的数据类型。
- 6、java 的开发工具包括 _____， _____， _____三种不同的工具包。
- 7、JVM 是 (_____--java virtual machine)，JDK 是 (_____--Java Developer's Kit)。
- 8、java 中最简单的一种数据类型是 _____，其类型说明符为 _____。
- 9、布尔类型变量的值只有两种值，其值为 _____和_____。
- 10、整数类型包括 _____、_____、_____、_____，他们都是带符号的（即有正负之分），表示数学中的整个数。
- 11、byte 类型数据占 _____位空间，short 类型数据占 _____位空间，int 类型数据占 _____位空间，long 类型数据占 _____位空间。
- 12、除了常见的十进制表示外，_____还可以用 _____进制和 _____进制来表示整数。
- 13、浮点类型包括 _____和_____两种数据类型，他们是带符号的（有正负之分），表示的是数学中的小数。
- 14、float 类型数据占据 _____位空间，double 类型数据占据 _____位空间。
- 15、对于浮点型数据，其十进制形式由 _____和_____组成，且必须含有小数点。
- 16、在 java 中_____也可以使用科学计数法来表示。
- 17、常用的算术运算符有 +, -, *, / 在 java 中用来表示 _____、_____、和_____。
- 18、_____运算符能够用于 String 的连接操作是 java 提供的一种特别支持。
- 19、/ 运算对于两个整数的除法，返回值是他们的商的整数部分 (_____ int 型)。可以通过取模运算符 _____来获取他们的余数。
- 20、java 中递增符号为 _____，java 中递减符号为 _____，赋值运算符是_____。

- 21、实际上，递增递减使用的是一种所谓的后缀形式，即递增，递减运算符在变量的后面，其实，递增递减也可以试用_____缀形式。
- 22、新建一个对象的运算符是_____，调用对象的属性和方法的运算符是_____。
- 23、关系运算符运算后的产生一个_____类型的结果。
- 24、关系运算中如果关系成立（为真），将返回_____，否则返回_____。
- 25、关系运算符包括_____（>）、_____（<）、_____（==）、_____（>=）、_____（<=）和_____（!=）。
- 26、boolean 类型变量只能使用_____和_____两个关系运算符进行比较。
- 27、与关系运算一样，_____运算也会产生 boolean 结果。
- 28、equals 方法比较的是对象的_____内容。
- 29、逻辑运算符包括_____（逻辑与），_____（逻辑或），和_____（逻辑非）。
- 30、逻辑运算符中_____和_____是二目运算符，而_____是一目运算符。
- 31、一个基本的 Java 程序包括_____，_____，_____，_____和_____五部分。
- 32、短路逻辑运算包括_____和_____。
- 33、在文件的路径使用中 \path1\path2\path3，java 对他的描述为_____。
- 34、java 语言中使用“{”开始，应用“_____”结束。
- 35、java 允许的三种注释方法为、_____，_____，_____。
- 36、Java 语言中，小数默认为_____类型。
- 37、在 java 中数据类型分为_____和_____两大类。
- 38、逻辑表达式：（34<=25）&&（68!=90）||（48*2<100）的值为_____。
- 39、Math 类的 random()方法可以产生一个_____之间的_____型随机数。
- 40、在 Java 语言中，逻辑常量只有_____和_____两个值。
- 41、所谓 N 目运算符，是指参加运算的变量的个数为_____。
- 42、当参加逻辑运算的两个变量都为_____时，逻辑与运算才返回 true，否则返回 false。
- 43、有一个参与逻辑运算的变量为真时，_____逻辑或运算便会返回_____。
- 44、当且仅当逻辑运算的变量都为假时，_____逻辑或运算才会返回_____。
- 45、逻辑非就是简单地取反、true 变为_____，false 变为_____。
- 46、&&、||和!只能作用于_____类型值上。
- 47、位运算允许直接操作一个整数的每个_____进制位。

- 48、位运算符包括按位与 (_____)、按位或 (_____)、按位异或 (_____)等。
- 49、位运算符会对参与运算的变量所对应的二进制位进行 _____ 运算，并产生一个结果。
- 50、在位运算中可以把 0 视为 _____，1 视为 _____。
- 51、当且仅当参与位运算的两个位都为 1 时，位与运算才返回 _____，否则返回 _____。
- 52、只要有一个参与位运算的位为 1 时，位或运算就返回 _____。
- 53、当且仅当参与位运算的两个位都为 0 时，位或运算才返回 _____。
- 54、位运算中，非就是简单的取反、1 变为 _____，0 变为 _____。
- 55、在 java 中好哎呦左移位运算符 (_____)和右移位运算符 (_____)。
- 56、若 a 为 int 型变量且赋值 6。执行语句 a--后 a 的值是 _____。执行语句 a++后 a 的值是 _____。
- 57、表达式 (10/3) 的值是 _____。表达式 (-10/3) 的值是 _____。
- 58、在一个 Java 源文件中定义了 3 个类和 15 个方法，编译该 Java 源文件时会产生 _____ 个字节码文件，其扩展名是 _____。
- 59、字符变量以 char 类型表示，它在内存中占 _____ 位 bit。
- 60、在 Java 语言中，如数字后没有任何字母，计算机默认为 _____ 类型。
- 61、在 java 中，赋值运算是通过数学符号中的 _____ 来完成的。
- 62、在 java 赋值运算中等号的意思是获取等号 _____ 边的值并将他复制到 _____ 边。
- 63、赋值运算中等号左边的值称为左值，右边的称为右值。右值可以是任何数、变量或是表达式，而左值则必须是一个 _____。
- 64、可以将一个常数赋值给 _____ (如 x=1)，但是不能为常数赋值。
- 65、每个 Java 应用程序可以包括许多方法，但必须有且只能有一个方法。
- 66、Java 编程的三个步骤、编辑源程序、_____和解释运行字节码。
- 67、根据结构组成和运行环境的不同，Java 程序可以分为两类、____和_____。
- 68、_____状态和行为是对象的主要属性。对象的状态在计算机内部是用_____来表示，而对象的行为在计算机内部是用_____来表示。
- 69、关于 main 方法的方法头正确的书写格式是_____。
- 70、Java 语言的各种数据类型之间提供两种转换、_____和强

制转换。

71、开发与运行 Java 程序需要经过的三个主要步骤为、____、____和____。

72、若 $x = 100$, $y = 120$, 则 $x > y$ 和 $x \leq y$ 的逻辑值分别为和_____。

73、设 $x = 60$, 则表达式 $(x++) / 3$ 的值是_____。

74、Java 语言的整型常量有十进制、_____十六进制和八进制三种形式, 从这角度来说, 4589 是_____整数, 0x3890 是_____整数, 02548 是_____整数。

75、Java 源文件中最多只能有一个_____类, 其他类的个数不限。

76、创建包的语句应该放在程序的位置_____。

77、定义包的关键字_____, Java 源程序经编译生成的字节码文件的扩展名为_____。

二、判断题

1、Java 语言不区分大小写。()

2、Java 程序源文件的扩展名为 .class。()

3、Java 程序中都含有 main 方法, 因为它是所有 Java 程序执行的入口。()

4、Java 程序可以运行在任何操作系统上, 只要有对应操作系统的虚拟机。()

5、Java 程序源文件名应该与主类名保持一致。()

6、“//”即可以表示单行注释, 也可以表示多行注释。()

7、Java 中的整型 int 占 2 个字节, 取值范围为 -32768 ~ 32767。()

8、在 Java 语言中, 执行语句 "boolean t=1 && 0;" 的结果是给 boolean 类型变量 t 赋初值为 false。()

9、声明变量时必须定义一个类型。()

10、注释的作用是使程序在执行时在屏幕上显示注释符号之后的内容。()

11、Java 认为变量 Sum 与 sum 是相同的。()

12、求模运算符 (%) 只可用于整型操作数。()

13、算术运算符 * , / , % , + 和 - 有相同的优先级。()

14、Java 语言中的标识符可以以数字、_____字母或下划线开头。()

15、Java 中小数常量的默认类型为 float 类型, 所以表示单精度浮点数时, 可以不在后面加 F 或 f。()

16、Java 语言是 IBM 公司设计的一种面向对象程序设计语言。()

17、一个 Unicode 字符占 4 个字节。()

18、假设 x、y 都是 float 型, 分别等于 5 和 7, 而 x1、y1 则是 double 型, 分别等于 5.0 和 7.0, 那么 x/y 与 x1/y1 的值不相等()

19、用 Javac 编译 Java 源文件后得到代码叫字节码。 ()

20、applet 可以运行在浏览器中。 ()

21、Java 可以用来进行多媒体及网络编程。 ()

三、选择题

1、语法错误由 () 发现的。

A、JRE B、编译程序 C、命令提示符 D、应用程序

2、Java 的源代码文件具有 () 的扩展名。

A、class B、java C、javac D、source

3、JAVA 语言是在 1991 年由 Jame Gosling 在 () 创立的。

A、Apple B、IBM C、Microsoft D、Sun Microsystems

4、java 运行环境只可以识别出 () 文件。

A、java B、jre C、exe D、class

5、拥有扩展名为 () 的文件可以存储程序员所编写的 Java 源代码。

A、java B、class C、exe D、jre

6、Java 是 () 语言的一个实例，其单条程序语句便可以完成任务。

A、机器 B、汇编 C、高级 D、以上都不对

7、Java 编程提供了可以由开发人员直接使用的 ()，因而每个应用程序不必从头创建，只需导入相应需要的就可以了。

A、现有库类 B、TCP C、汇编代码 D、二级存储器

8、() 可将一个 java 文件转换成一个 class 文件

A、调试程序 B、编译程序 C、转换器程序 D、JRE

9、在查找到应用程序的一个语法错误以后，编译程序将给用户通告此错误 ()

A、出现的行号 B、可用于改正错误的正确代码
C、一个简短的描述 D、选择 A 和 B

10、语法错误可因多种原因产生，例如，当 ()

A、应用程序突然中止 B、缺少括号 C、单词拼写错误 D、选择 B 和 C

11、为了编译应用程序，输入命令 () 并跟上相应文件的名称。

A、build B、java C、javac D、create

12、下面对标识符的描述，正确的是 ()

A、可以以任意字符作为起始，但不能包含空格
B、必须以数字作为起始，但不能包含空格
C、不能以数字作为起始，也不能包含空格
D、不能以数字作为起始，但可以包含空格

13、方法 () 可将一个数值转换成文本。

A、Integer.valueOf B、Integer.parseInt C、String.parseInt D、String.valueOf

14、下面的数据类型强制转换中，需要把一个字符串转换为数字型的方法中，哪个是正确的（ ）。

- A、Double.parseDouble() B、String.parseDouble()
C、Double.parseDouble() D、String.parseString()

15、乘法运算符一侧的表达式被引述为它的（ ）

- A、运算符的值 B、结果 C、操作数 D、参数

16、在数据类型的强制转化中，其中方法 Integer.parseInt（ ）的功能描述正确的是（ ）。

- A、将一个整数转换成文本 B、将文本转换为一个整数

- C、同 String.valueOf 一样完成相同的任务 D、以上答案都不对

17、（ ）字符是一个乘法运算符

- A、星号（*） B、正斜杠（/）字符 C、分号（;） D、以上答案都不对

18、一个（ ）运算拥有两个操作数

- A、注释 B、文本 C、二目 D、以上答案均不正确

19、下面对注释的功能描述正确的是（ ）

- A、有助于增强应用程序的可读性 B、位于两个正斜杠之后
C、会编译程序忽略 D、以上答案均正确

20、Java 语句在遇到（ ）时将结束。

- A、, B、; C、。 D、{ }

21、文档注释的 @version 标识表示（ ）。

- A、类与方法的更新过程 B、版权信息
C、作者名称 D、接口与类的版本

22、文档注释的 @param 标识表示（ ）。

- A、参数说明 B、还回参数的说明
C、异常处理的说明 D、抛出异常类的说明

23、（ ）是规范的类名。

- A、CountClassl B、car C、moon D、demo

24、（ ）是规范的类包名字。

- A、java.Lang B、java..lang C、Java.lang D、1java.lang

25、（ ）是规范的常量名称。

- A、Min_Value B、Min_value C、MIN_V ALUE
D、min_Value

26、Java 语言中的关键字是（ ）。

- A、标识符 B、由 Java 保留使用的 C、区分大小写 D、B 和 C
- 27、类的声明是以 () 作为结束
A、花括号 B、分号 (;) C、end 关键字 D、class 关键字
- 28、方法 () 可以将文本转换成数值
A、Integer.parseInt B、String.valueOf C、Integer.parseInt D、String.value
- 29、变量名必须是一个 ()。
A、关键字 B、有效的标识符 C、以上两个答案都对 D、以上答案都不对
- 30、一些已在 Java 中定义的类型，如 int，被认为是 () 类型。
A、创建 B、现存 C、已定义的 D、基本
- 31、在 Java 中，() 可由 “ ” 来表示。
A、空字符 B、空字符串 C、空值 D、以上答案都不对
- 32、当一个变量被放置在内存中的某一个单元中时，此值会 () 该单元中先前所存放的值。
A、复制 B、替换 C、将其值添加到 D、移动
- 33、在 Java 中，在编写算术表达式的时候，必须 () 才能输入进计算机中。
A、使用括号 B、在多行上 C、以值的形式 D、以上答案都不对
- 34、赋值运算符 (=) 右侧的表达式总是在赋值发生之 () 进行
A、前 B、后 C、同一时刻 D、以上都不对
- 35、断点不能设置在 () 中
A、注释 B、可执行的代码行 C、赋值语句 D、算术语句
- 36、当应用程序在某个断点处暂时执行时，下一条即将执行的语句将位于该断点所在执行语句 () 的语句
A、之前 B、之后 C、处 D、以上都不对
- 37、为便于阅读表达式而添加在表达式中的括号被认识是 () 括号
A、必要的 B、冗余的 C、嵌入 D、嵌套的
- 38、运算符 () 用于执行除法运算
A、/ B、\ C、% D、*
- 39、每一个变量都拥有一个 ()

A 、 名字 B、 类型 C 、 以上两个都对 D 、 以上都不对

40、在 Java 中，使用 () 可强制运算符的执行次序。

A、 括号 B、 变量 C、 调试程序 D、 内存

41、如果一个表达式中包含多个表示乘法、除法、和求余的运算符，那么他们将按照 () 的顺序进行执行

A、从右到左 B、从左到右 C、以上两个答案都对 D、以上答案都不对

42、调试程序中的 () 命令能够对应应用程序源代码中的一个可执行的代码行设置断点。

A 、 stop B、 run C 、 print D 、 clear

43、注释内容在代码编写中经常出现，下面对注释功能的说明正确的是 ()。

A、 解释说明代码 B、 注销部分代码 C、 强调代码的重要性 D、 A 和 B

44、用于完成某个特定任务并且可能会返回一个值的代码部分被认为是 ()

A、 变量 B、 方法 C、 操作数 D、 标识符

45、用于存储整数值的变量应使用关键字 () 进行申明

A、 integer B、 int C 、 intValue D、 Int

46、() 是帮助程序员开发算法的一个人工的、非正式的语言

A、 伪代码 B、 Java 语言 C、 注释 D、 执行

47、下面对伪代码的描述中，正确的是 ()

A、 通常只能描述声明 B、 在计算机中执行
C、 通常只描述可执行的代码行 D 、 通常描述声明和可执行的代码行

48、符号 () 并不属于一个 Java 运算符

A 、 * B、 != C、 <> D、 %

49、下面哪些是 java 语言中的关键字？ ()

A 、 sizeof B 、 abstract C、 NULL D、 Native

50、下面语句哪个是正确的？ ()

A 、 char a='abc'; B、 long l=0xfff; C、 float f=0.23; D、 double b=0.7E-3;

51、以下程序测试 String 类的各种构造方法，试选出其运行效果。
()

```
class STR{  
    public static void main(String args[]){  
        String s1=new String();  
        String s2=new String("String 2");  
        char chars[]={'a',' ','s','t','r','i','n','g'};
```

```

String s3=new String(chars);
String s4=new String(chars,2,6);
byte bytes[]={0,1,2,3,4,5,6,7,8,9};
StringBuffer sb=new StringBuffer(s3);
String s5=new String(sb);
System.out.println("The String No.1 is "+s1);
System.out.println("The String No.2 is "+s2);
System.out.println("The String No.3 is "+s3);
System.out.println("The String No.4 is "+s4);
System.out.println("The String No.5 is "+s5);
}
}

```

- A、 The String No.1 is
The String No.2 is String 2
The String No.3 is a string
The String No.4 is string
The String No.5 is a string
- B 、 The String No.1 is
The String No.2 is String 2
The String No.3 is a string
The String No.4 is tring
The String No.5 is a string
- C 、 The String No.1 is
The String No.2 is String 2
The String No.3 is a string
The String No.4 is strin
The String No.5 is a string
- D 、 以上都不对

52、下面语句段的输出结果是什么？ ()

```

int i = 9;
switch (i) {
default:
System.out.println("default");
case 0:
System.out.println("zero");
break;
case 1:
System.out.println("one");
case 2:
System.out.println("two"); }

```


B 、 default, zero

D、 no output displayed

()

D、 Guang zhou

$$(\quad)$$

C、 (-128) ~ 127

D、 $(-256) \sim 255$

()

B、变量 d2没有初始化 ,d1初始化为

4.0

D、变量 d1没有初始化 ,d2初始化为

4.0

)

```
int i = 1;
```

```
switch (i) {
```

case 0:

```
System.out.println("zero");
```

```
break;
```

case 1:

```
System.out.println("one");
```

case 2:

```
System.out.println("two");
```

default:

```
System.out.println("default");
```

}

D、

default

57、Java 的字符类型采用的是 Unicode 编码方案，每个 Unicode 码占用（ ）个比特位。

D、64

)

C、 My Variable

D、small-pig

)

C、 $X=3+X^2$

D、 $X = X^*3 + X^*2$

) 0

D、

implements

61、 下列代码中

```
if(x>2 ){System.out.println(      “ first      ” );}  
else if (x>-2 ){System.out.println(      “ second ” );}  
else { System.out.println(      “ third ” );}
```

要求打印字符串为 “ second”时，x 的取值范围是 ()

A、x<=2 并且 x>-2 B、x>2 C、x>-2 D、x<=-2

62、 以下标识符中哪项是不合法的 ()

A. Agig0lInt\$223 B、_sdf C、\$45sd D、45df

63、下面的数据类型是 float 型的是 ()

A、55.6 B、15 C、95L D、8.6F

64、 定义一个长整型变量 a 的语句是 ()

A、int a B、a long C、float a D、long a

65、下式的运算后 i 的结果是： ()

```
int i=3;    i*=4;
```

A、 6 B、 8 C、 12 D、 3

66、下面赋值语句正确的是： ()

A、int b=9.3 B、boolean a=false C、float f=23.1 D、
char b= 'b ';

67、下面说法错误 的是： ()

A、java 是一种网络编程语言 B、java 是由 sun 公司发布的
C、java 允许使用指针访问内存 D、java 的前身是 Oak

68、下述字符组合不能作为 Java 程序标识符的是 ()

A、Aa12 B、X C、y D、5x

69、以下程序代码的输出结果是 ()

```
int x=53;
```

```
System.out.println(1.0+x/2);
```

A、27.0 B、27.5 C、1.026 D、1.026.5

70、编译 java 程序时出现 error: cannot read: aaa.java , 则下列原因最正确的是 ()

A、原因是执行的当前目录下没有找到 aaa.java 文件。
B、原因是没有安装 JDK 开发环境。
C、原因是 java 源文件名后缀一定是以 .txt 结尾。
D、原因是 JDK 安装后没有正确设置环境变量 PATH 和 Classpath。

71、指出不正确的表达式 ()

A、double a=1.0; B、Double a=new Double(1.0); C、byte a = 120;
D、Byte a = 200;

72、关于 Float , 下列说法不正确的是 ()

A、Float 是一个类 B、Float 在 java.lang 包中
C、float a=1.0 是正确的赋值方法 D、Float a= new Float(1.0) 是
正确的赋值方法

- 73、下面哪个单词是 Java 语言的关键字 ()
A、Float B、this C、string D、unsigned
- 74、以下代码段执行后的输出结果为 ()
int x=3 ; int y=10 ;
System.out.println(y%x);
A、0 B、1 C、2 D、3
- 75、下列选项中不是 JAVA 语言特点的是 ()
A、面向对象 B、平台无关 C、安全性高 D、单线程
- 76、下列选项中，哪一个是打印语句的正确表达式： ()
A、System.out.println(); B、System.out.Println();
C、System.Out.print(); D、System.Out.println();
- 77、下列哪一个是 JAVA 语言的关键字 ()
A、hello B、void C、num_01 D、abc
- 78、关于数据类型转换下面描述错误的是 ()
A、当两个数据的类型不统一时，必须先进行数据类型的转换，再运算或赋值
B、byte 类型数据可以转换为 short,int,long 类型数据
C、long 类型数据有可能转换为 byte,short,int 类型数据
D、在语句 int i=(int) 78.67; 中，变量 i 的值为 79
- 79、下列定义 String 变量 s 的语句中，错误的是 ()
A、String s=new "Yes! "; B、String s=new String("yes! ");
C、String s; D、String s= "Yes! " ;
- 80、下列关于注释语句的描述错误的是 ()
A、多行注释语句的内容，从 /* 开始，到 */ 结束
B、多行注释语句也可以是单行语句
C、文档注释语句的内容，从 /** 开始，到 */ 结束
D、单行注释语句的内容，从 // 开始，以分号为结尾句
- 81、下列哪一个是合法的标识符 ()
A、12class B、+viod C、-5 D、_black
- 82、下列哪一个不是 JAVA 语言中保留字、 ()

A、 if B、 sizeof C、 orivate D、 null

83、下列描述中正确的一项是 ()

A、标识符首字符的后面可以跟数字； B、标识符不区分大小写；

C、符合数据类型变量包括布尔型、字符型、浮点型； D、数组属于基本数据类型；

84、下列哪一项不属于变量的组成部分、 ()

A、变量名 B、变量属性 C、变量初值 D、变量大小

85、下列关于变量组成部分描述正确的一项是 ()

A、变量名不是标识符 B、变量名是标识符
C、浮点型属于复合类型 D、变量属性分为基本类型和复合类型

86、下列选项中，哪一项不属于 JAVA 语言的简单数据类型

()
A、整数型 B、数组 C、字符型 D、浮点型

87、下列关于基本数据类型的说法中，不正确的一项是 ()

A、boolean 是 JAVA 特殊的内置值，或者为真或者为假
B、float 是带符号的 32位浮点数
C、double 是带符号的 64位浮点数
D、char 应该上是 8位的 Unicode 字符

88、下列关于溢出的说法中，正确的一项是 ()

A、一个整型的数据达到整型能表示的最大数值后，再加 1，则机器产生上溢，结果为整型的最大值
B、一个整型的数据达到整型能表示的最小数值后，再减 1，则机器产生下溢，结果为整型的最小值

C、实型变量在运算中不会产生溢出的问题
D、实型变量在运算中和整型一样会产生溢出问题

89、下列关于 char 类型的数据说法中，正确的一项是 ()

A、?r?表示换行符
B、?n?表示回车符
C、char 类型在内存中占 16位
D、凡是在 Unicode 的字符都可以用单引号括起来作为 char 类型常量

90、下列关于基本数据类型的取值范围描述中， 正确的是 ()

A、byte 类型范围是 -128~128
B、boolean 类型范围是真或者假
C、char 类型范围是 0~65536
D、short 类型范围是 -32767~32767

- 91、下列哪个不是正确标识符 ()
 A、\$million B、\$_million C、1 \$_million D、\$ 1_million
- 92、下列关于 JAVA 语言简单数据类型的说法中，哪个是正确的 ()
 A、一 0 开头的整数代表 8 进制常量
 B、以 0x 或者 0X 开头的整数代表 8 进制整型常量
 C、boolean 类型的数据作为类成员变量的时候，系统默认值初始为 true
 D、double 类型的数据占计算机存储的 32 位
- 93、下列关于 JAVA 语言中要使用的一个变量，不正确的是 ()
 A、在 JAVA 程序中要使用一个变量，必须先对其进行声明
 B、变量声明语句可以在程序的任何地方，只要在变量使用前就可以
 C、变量不可以在其作用域之外使用
 D、局部变量在使用之前可以不初始化，系统使用默认的初始值
- 94、下列 JAVA 语句中，不正确的一项是 ()
 A、int \$ e,a,b =10; B、char c,d = ?a?; C、float e = 0.0d;
 D、double e = 0.0f;
- 95、在编写 JAVA 程序时候，如果不为类的成员变量定义初始值， JAVA 会给它默认值，下面说法不正确的是、 ()
 A、 byte 是 0 B、 int 是 0 C、 long 是 0.0l D、 float 是 0.0f
- 96、下列关于自动类型转型的说法中，哪个正确 ()
 A、char 类型数据可以自动转换为任何简单的数据类型的数据
 B、char 类型数据只能自动转换为 int 类型数据
 C、char 类型数据不能自动转换 boolean 类型数据
 D、char 类型不能做自动类型转换
- 97、下列语法中不正确的一个是 ()
 A、float a = 1.1f; B、byte d =128;
 C、double c =1.1/0.0; D、char b =(char)1.1f;
- 98、下列哪一项属于 JAVA 语言的复合数据类型 ()
 A、无符号整数类型； B、整型；
 C、联合类型； D、接口；
- 99、下列对整型常量定义的解释中，正确的是 ()
 A、034 代表八进制的数 1C； B、034 代表八进制的数 34；
 C、034 代表十六进制的数 28； D、34L 代表 34 的 64 位长整数；
- 100、下列关于整型类型的说法中，正确的是 ()
 A、short 类型的数据存储顺序先低后高 B、

Integer.MAX_VALUE 表示整型最大值

C、Long.MIN_VALUE 表示整型最大值 D、long 类型表示数据范围和 int 类型一样

101、Java application 中的主类需包含 main 方法，main 方法的返回类型是什么？（ ）

A、int B、float C、double D、void

102、下面用来编译 Java 应用程序的命令是（ ）

A、javac B、jdk C、appletviewer D、java

103、下面用来运行 Java 应用程序的命令是（ ）

A、javac B、jdk C、appletviewer D、java

104、（ ）字符是一个乘法运算符。

A、星号 (*) B、正斜杠 (/) C、分号 (;) D、乘号 (x)

105、下列语句中，不正确的是（ ）

A、float e = 1.1f B、char f = 1.1f

C、double g = 1.1f D、byte h = 1

106、下列哪一个不属于 JAVA 语言的数据类型（ ）

A、指针类型 B、类 C、数组 D、浮点类型

107、下列哪个数代表八进制整数？（ ）

A、0XA6 B、0144 C、1840 D、-1E3

108、下列哪个代表单精度浮点数？（ ）

A、0652 B、3.4457D C、0.298f D、0L

109、下列哪个代表十六进制整数？（ ）

A、0123 B、1900 C、fa00 D、0xa2

110、下列哪个是反斜杠字符的正确表示？（ ）

A、\\ B、*\ C、\ D、\'

111、下列说法中，哪个是正确的？（ ）

A、JAVA 是不区分大小写的，源文件与程序类名不允许相同

B、JAVA 语言以方法为程序的基本单位

C、Applet 是 JAVA 的一类特殊应用程序，它嵌入 HTML 中，随主页发布到互联网上

D、以 // 符开始的为多行注释语句

112、JAVA 虚拟机的执行过程有多个特点，下列哪个特点不属于 JVM 执行特点？（ ）

A、多线程性 B、动态连接 C、异常处理 D、异步处理

113、在 JAVA 语言有多种实现的机制，下列哪种技术属于垃圾回收机制？（ ）

A、字节编译 B、内存跟踪 C、语法检查 D、堆栈 / 下溢检查

114、设有定义 :float a=2,b=4,h=3;, 以下表达式中与代数式 $\frac{1}{2}(a+b)h$ 计

算结果不相符的是 ()

A、 $(a+b)*h/2$

B、 $(1/2)*(a+b)*h$

C、 $(a+b)*h*1/2$

D、 $h/2*(a+b)$

115、 下列正确的标识符是 ()

A、 12ab

B、 float

C、 aw~1e

D、 b6ty

116、 下列选项中 ,不能用作标识符的是 ()

A、 _1234_

B、 _1_2

C、 int_2_

D、 2_int_

117、 若 x,a,b,c 四个变量都是 int 型变量 ,则执行下面的表达式后 ,x 的值是 () $x=(a=1,b=2,c=a+b+3);$

A、 1

B、 2

C、 3

D、 6

118、 下列描述中不正确的是 ()

A、 字符型数组中可以存放字符串

B、 可以对字符型数组进行整体输入、输出

C、 可以对整型数组进行整体输入、输出

D、 不能在赋值语句中通过赋值运算符 "=" 对字符型数组进行整体赋值

119、 以下 4 个选项 ,不能看作一条语句的是 () .

A、 {;}

B、 $a=0,b=0,c=0;$

C、 $if(a>0);$

D、 $if(b==0)m=1;n=2;$

120、 以下叙述中错误的是 () .

A、 对于 double 类型数组 ,不可以直接用数组名对数组进行整体输入或输出

B、 数组名代表的是数组所占存储区的首地址 ,其值不可改变

C、 当程序执行中 ,数组元素的下标超出所定义的下标范围时 ,系统将给出 "下标越界 " 的出错信息

D、 可以通过赋初值的方式确定数组元素的个数

121、 可以正确表达 $x \geq 0$ 或 $x \leq 1$ 的关系的表达式是 () .

A. $(x \geq 1) || (x \leq 0)$

B、 $x \geq 1 | x \leq 0$

C、 $x \geq 1 \text{ OR } x \leq 0$

D、 $x \geq 1 \&\& x \leq 0$

122、 double 类型可以用做存储 ()

A、 字母和数字

B、 带有小数点的数

C、 字符串

D、 以上答案都不对

123、 常量应使用 () 关键字进行声明

A、 fixed

B、 constant

C、 final

D、 const

124、 () 运算符能够在其左操作数大于右操作数时返回 false

A、 ==

B、 <

C、 <=

D、 以上答案都正确

125、 一个类型为 Boolean 的变量可以被赋予的值为 ()

A、 ture false

B、 off on

C、 one zero

D、 yes

no

126、 一种在初始声明变后其值不能再改变的变量称为 ()

A、 double

B、 常量

C、 标准

D、 Boolean

127、()运算符是将左操作数与右操作数的和再赋予左操作数的一种运算符

A、+ B、+= C、+= D、+=

128、单目运算符()

A、只需要一个操作数 B、需要两个操作数
C、必须使用 && D、可以没有操作数

129、()运算符可以用确保两个条件都为真

A、^ B、&& C、and D、||

130、下面对运算符 && 描述正确的是()

A、执行短路运算
B、是一个比较运算
B、如果两个操作数都为真，则计算的结果为假
D、以上答案都不对

131、当()则条件 expression1&&expression2 的计算结果为 true

A、expression1 为 true 并且 expression2 为 false
B、expression1 为 false 并且 expression2 为 true
C、expression1 和 expression2 为 true
D、expression1 和 expression2 为 false

132、当()则条件 expression1||expression2 的计算结果为 false

A、expression1 为 true 并且 expression2 为 false
B、expression1 为 false 并且 expression2 为 true
C、expression1 和 expression2 为 true
D、expression1 和 expression2 为 false

133、当()则条件 expression1^expression2 的计算结果为 true

A、expression1 为 true 并且 expression2 为 false
B、expression1 为 false 并且 expression2 为 true
C、expression1 和 expression2 为 true
D、A 和 B

134、假设 a=3，当赋值操作 b=a—完成以后，变量 a 和 b 的值变为()

A、3, 3 B、2, 3 C、3,2 D、2,2

135、假设 c=5,当赋值操作 d=c*++c 完成后，变量 d 的值变为()

A、25 B、30 C、36 D、以上答案都不对

136、一个 int 型变量能够被隐式的转换为()

A、long B、double C、float D、以上答案都对

137、当一个 int 型值和 double 型值相加，会出现()

A、隐式转换 B、造型 C、赋值 D、以上答案都不对

138、一个 double 型值可被隐式转换为()型值

A、int B、float C、long D、以上答案都不对

139、一个 Boolean 型值可被隐式转换为()型值

A、int B、double C、long D、以上答案都不对

140、通过使用关键字 () 来定义常量

A、readOnly B、final C、constants D、constant

141、() 对象的 println 方法用于在命令提示符窗口中显示文本

A、System B、System.out C、System.output D、以单行答案都不对

142、System 类在哪个包中 ? ()

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

143、给出下面代码：

```
public class test{
    static int a[] = new a[10];
    public static void main(String args[]) {
        System.out.println(arr[10]);
    }
}
```

那个选项是正确的 ? ()

A、编译时将产生错误； B、编译时正确，运行时将产生错误；
C、输出零； D、输出空。

144、switch 语句不能用于下列哪种数据类型 ()

A、double B、byte C、short D、char

145、下列哪个数代表十六进制整数 ()

A、0123 B、1900 C、fa00 D、0xa2

146、定义整型变量： int n=456;，表达式的值为 5 的是 ()

A、n/10%10 B、(n-n/100*100)/100 C、n%10 D、n/10

147、符号 () 并不属于一个 java 运算符。

A、* B、!= C、<> D、%

148、已知 ch 是字符型变量，下面正确的赋值语句是 () 。

A、ch= ' a+b ' B、ch= ' \0 ' C、ch= ' 7 ' + ' 9 ' D、ch=5+9

149、Java 语言中数值数据的类型能自动转换，按照从左到右的转换次序为 ()

A、byte int short long float double
B、byte short int long float double
C、byte short int float long double

D、 short byte int long float double

150、下面供选字符序列中，不属于 Java 语言关键字的是 ()

A、 goto B、 null C、 false D、 native

151、以下程序代码的输出结果是 ()

```
int x=73 ; System.out.println(x>>1) ;
```

A、 36 B、 36.5 C、 37 D、 true

152、 下列变量定义错误的是 ()

A、 int a B、 double b=4.5 C、 boolean b=true D、 float f=9.8

153、下列哪个是 Java 语言的编译工具 ()

A、 javac B、 java C、 jalap D、 javadoc

154、java 平台是由那部分组成的 ()

A、 java 虚拟机和 java 编译器 B、 java 应用程序编程接口和 java 虚拟机

C、 java 编译器和 java 解释器 D、 java 应用程序编程接口和解释器

155、定义字符串常量用 () 括起来的字符序列。 ()

A、 ?? B、 " " C、 {} D、 []

156、在 java 中，表示换行的转义字符是 ()

A、 \r B、 \n C、 \f D、 b

157、一下的选项中能正确表示 Java 语言中的一个整形常量的是 ()

A、 35.d B、 -20 C、 1,234 D、 "123 "

158、下列哪个不是 Java 的关键字 ()

A、 TRUE B、 boolean C、 int D、 public

159、JDK 中提供的文本文档生成器是 ()

A . jav.exe B、 javap.exe C、 javadoc.exe D、 javaprof.exe

160、下列叙述中正确的是 ()

A、 生命变量时必须指定一个类型 B、 java 认定变量 number 与 Number 相同

C、 Java 中唯一的注释方法是 "// " D、 源文件中 public 类可以有 0 或多个

161、下列表达式不合法的是 ()

A、 String x= "Sky "; int y=5;x+=y;

B、 String x= "Sky ";int y=5;if(x==y){}

C、 String x= "Sky ";int y=5;x=x+y;

D、 String x=null;int y=(x!=null)&&(x.length()>0)?x.length:0

162、下面描述错误的是 ()

A、 java 代码中严格区分大小写 B、 java 中的所有字符都是英文状态下的

B、java 是以类为程序的基本单位 D、java 是面向过程的语言

163、能够支持 javadoc 命名的注释语句是 ()

A 、 /*-----**/

B、 /*-----*/

C、 //

D、 /**-----*/

164、下列变量定义中，正确的是 ()

A、 long 1=123L

B、 long 1=3.14156L

C、 int i= ?K?

D、 double d=1.55989E2f

165、在下列说法中，选出最正确的一项是 ()

A、 java 语言是以类为程序的基本单位的

B、 java 语言是不区分大小写的

C、多行注释语句必须以 // 开始

D、在 java 语言中、类的源文件名和该类可以不相同

166、下列注释语句的描述中，正确的一项是 ()

A、以 // 开始的是多行注释语句

B、以 /* 开始， */ 结束的单行注释语句

C、以 /** 开始， */ 结束的是可以生成帮助文档的注释语句

D、以 /** 开始， */ 结束的是单行注释语句

167、下列说法中错误的一个是 ()

A、 \$1_abc 是合法的 java 标识符

B、局部变量必须先明后使用

C、 void 是合法的 java 保留字
标识符命名的

D、在 java 中，类不是用标

168、下列标识符中，正确的是 ()

A、 1_Back

B、 \$_Money

C、 .\$-money

D、

2-Forward

169、下面哪一个是符合 java 语言语法的语句 ()

A、 int a=0;

B 、 int a=0

C、 double i=1.3%12;

D、

int a=0;

Boolean

b=a!=

int

b=2;

int b=2;

int

c=12;

int 12=7;

A=12+b+e!=4;

a=a+12=c+b;

170、Java 语言能做到 “编写一次，随处可用”，是指 Java 语言具有 ()

A、解释特性

B、可移植性

C、很高的通用性

D、

很高的安全性

171、.ASCII 码与 Unicode 码的区别是 ()

A、 ASCII 码是 16 位的字符集，而 Unicode 码也是 16 位的字符集

- B、ASCII 码是 16 位的字符集，而 Unicode 码是 8 位的字符集
C、ASCII 码和 Unicode 码都是 8 位，只是 Java 中使用 Unicode 码。
D、ASCII 码和 Unicode 码都是 16 位，只是 Java 中使用 Unicode 码。

172、JDK 的组成描述中，以下哪几部分组成是不正确的（ ）

- A、Java 编程语言 B、工具及工具的 API
C、Java EE 扩展 API D、Java 平台虚拟机

173、以下哪个包不是 Java 标准库中常用的包？（ ）

- A、javax.servlet.http B、java.lang
C、java.util D、java.sql

174、使用 JDK 工具生成的 Java 文档的文件格式是（ ）

- A、XML 格式 B、HTML 格式
C、二进制格式 D、自定义格式

175、下列有关 main() 方法的签名正确的是（ ）

- A、public static void main(String[] args){}
B、public static void main(){}
C、public static main(String args[]){}
D、public void static main(String[] args){}

176、以下关于 Applet 的描述，错误的是（ ）

- A、一个 Applet 就是一段 Java 程序
B、Applet 是一种特殊的 Java 程序，它需要运行在 Web 服务器上
C、Applet 是一种特殊的 Java 程序，它需要运行在 Web 浏览器上
D、Applet 是一种 Java SE 平台的应用程序

177、下列不是 int 类型数据的选项是（ ）

- A、20 B、077 C、0xABBC D、u03A6

178、下列不是 Java 语言中所规定的注释样式的是（ ）

- A、// 单行注释 B、-- 单行注释
C、/* D、/**
* 单行或多行注释 * 文档注释
*/ */

179、下列不是 Java 语言的关键字的是（ ）

- A、goto B、instanceof C、sizeof D、volatile

180、Java 语言中表达式 10/3 的结果是（ ）

- A、3.3 B、3.33 C、3 D、3.0

181、Java 语言中表达式 -12>>3 的结果是（ ）

- A、-4 B、-9 C、-1 D、-2

182、下列赋值语句正确的是（ ）

- A、long val=6; B、int age = 23L;
C、short x=1,y=2,z;
 z=x+y; D、int a= ,A?

183、假设有 2 个整数 x 和 y，表达式 x>=y ? x : y 的含义是（ ）

- A、 求 2 个数的较大数 B、 求 2 个数的较小数
C、 把 y 的值赋给 x D、 把 x 的值赋给 y

184、Java 语言中常量的定义是 ()

- A、 public static B、 public static final
C、 final D、 public static abstract

185、作为 Java 程序应用程序入口的 main 方法，其声明格式是：
()

- A、 public static int main (String args[])
B 、 public static void main (String args[])
C、 public void main (String args[])
D 、 public int main (String args[])

186、设有一个 Java 小程序，源程序名为 Appfirst.java, 其 HTML 文件为 Appfirst.html，则 运行该小程序的命令为。 ()

- A、 java Appfirst B 、 javac Appfirst.java
C 、 appletviewer appfirst.java D 、 appletviewer appfirst.html

187、下面命令正确的是。 ()

- A、Java appfirst. java B、Java appfirst C Java appfirst. class D、Javac appfirst

188、对主类的说明正确的是。 ()

```
public class Helloworld {  
public static void main(String args[ ] ) {  
System.out.println(" 欢迎访问 Java 世界!");  
}
```

- A、源程序无法通过编译，因为主方法的声明不正确
B、源程序可以通过编译，但无法运行，因为该文件没有 public 类
C、源程序可以通过编译，但在运行时会出现“程序中没有主方法的异常”
D、程序能正常运行

189、如果已经编写了 Java 小程序 Appfirst.java，在编写它所对应的 HTML 文件时能正确将其嵌入到 HTML 网页文件的语句是。
()

- A、 <applet code= " Appfirst.java width=300 height=60></applet>
B 、 <applet code= " Appfirst width=300 height=60></applet>
C 、 <applet code= " Appfirst.class width=300 height=60></applet>
D 、 以上都不对。

190、下面这些标识符那个是错误的： ()

- A 、 Javaworld B 、 sum C 、 2Java Program D 、 \$abc

191、下列那组基本数据类型所占的字节数相同。 ()

- A、布尔型和字符型 B、整型和单精度型 C、字节型和长整型
D、整型和双精度型

192、下列那一组运算符的优先级顺序是那由高到低排序的。 ()

A、|、&、!; B、&、^、||; C、!、%、++; D、<、<<、++

193、下面哪个赋值语句不会产生编译错误 ()

A、char a=?abc?; B、byte b=152; C、float c=2.0; D、double d=2.0;

194、下面哪个单词是 Java 语言的关键字。 ()

A、False B、FOR C、For D、for

195、有如下的程序： ()

```
public class Welcome3 {  
    public static void main( String args[] ) {  
        System.out.println( "How\nare\nyou ! \n" );  
    }  
}
```

则它的输出结果是：

A、How are you! B、How are you!

C、How are you! D、How are you!

196、执行下面程序后，哪个结论是正确的 ()

```
int a , b , c;
```

```
a=1;
```

```
b=3;
```

```
c=(a+b>3 ?++a : b++)
```

A、a 的值为 2，b 的值为 3，c 的值为 1

B、a 的值为 2，b 的值为 4，c 的值为 2

C、a 的值为 2，b 的值为 4，c 的值为 1

D、a 的值为 2，b 的值为 3，c 的值为 2

197、设各个变量的定义如下，哪些选项的值为 true ()

```
int a=3 , b=3;
```

```
boolean flag=true;
```

A、++a==b B、++a==b++ C、(++a==b) || flag D、(++a==b) & flag

198、使整值变量 x 加 1，下面写出的形式不对的是： ()

A、x++ B、++x C、x=x+1 D、x=+1

199、下面表达式的值的类型为 ()

```
( int ) 6.5/7.5*3
```

A、short B、int C、double D、float

200、设 a，b，x，y，z 均为 int 型变量，并已赋值，下列表达式的结果属于非逻辑值的是 ()。

A、x>y && b<a B、-z>x-y C、y==++x D、y+x*x++

201、下面语句输出的结果为 ()

```
system.out.println(5^2)
```

A 、 6 B、 7 C、 10 D、 25

202、设 x 为 int 型变量，则执行以下语句段后， x 的值为 ()
x=15;

x+=x-=x-x;

A 、 10 B、 20 C、 30 D 、 40

203、对下面的语句执行完正确的说法是 ()

int c=?c?/3;

System.out.println(c) ;

A 、 输出结果为 21 B 、 输出结果为 22 C、 输出结果为 32
D、 输出结果为 33

204、以下选项中变量 a 以定义类型，合法的赋值语句为 ()

A、 a= int(y) B、 a==1 C、 a=a+1=3 D、 ++a

205、阅读以下程序，选择正确的答案 ()

int x=7,y=8,z=9;

6 char ch='A';

x++;z--;

System.out.println(x==y);

System.out.println(y==z--);

System.out.println(x==z);

System.out.println(ch++== 'A');

System.out.println(ch++== 'B');

A、 程序输出结果均为布尔值 B、 程序中输出 3 个 true
C、 程序运行后， ch 的值为 ' C ' D、 程序运行后， z 的值为 8

206、执行下列程序段后， b , x , y 的值正确的是 ()

int x=3,y=4;

boolean ch;

ch=x<y||++x==--y;

A 、 true , 3 , 4 B、 true , 4 , 3 C、 false , 3 , 4 D、
false , 4 , 3

207、下列叙述不正确的是 ()

A 、 JAVA 语言是一种面向过程的高级程序设计语言

B、 JAVA 语言源程序经过编译后生成字节码文件

C、 JAVA 语言源程序文件扩展名为 .java

D、 JAVA 语言程序有两种类型 Application 程序和 Applet 程序

四、程序阅读题

1、 int x=10 ;

Int y=x++ ; // 执行后， y=_____， x=_____

Y=++x ; // 执行后， y=_____， x=_____.

2、 写出下列程序输出结果 :

Int i = 1,j = 2,k = 3;

```

System.out.println("(i<j)==" + (i<j));
System.out.println("(k<j)==" + (k<j));
System.out.println("(i<j)&&(k<j)==" + (i<j)&&(k<j));
System.out.println("(i<j)|| (k<j)==" + (i<j)|| (k<j));
System.out.println("! (i<j)==" + ! (i<j));

```

解：

```

(i<j)== _____
(k<j)== _____
(i<j)&&(k<j)== _____
(i<j)|| (k<j)== _____
! (i<j)== _____

```

- 3、 下面这个简单例子示范了算术运算符， 也说明了浮点型除法和整型除法之间的差别，写出以下程序的运行结果：

```

public class BasicMath {
    public static void main(String args[]) {
        // arithmetic using integers
        System.out.println("Integer Arithmetic");
        int a = 1 + 1;
        int b = a * 3;
        int c = b / 4;
        int d = c - a;
        int e = -d;
        System.out.println("a = " + a);
        System.out.println("b = " + b);
        System.out.println("c = " + c);
        System.out.println("d = " + d);
        System.out.println("e = " + e);
        // arithmetic using doubles
        System.out.println("\nFloating Point Arithmetic");
        double da = 1 + 1;
        double db = da * 3;
        double dc = db / 4;
        double dd = dc - a;
        double de = -dd;
        System.out.println("da = " + da);
        System.out.println("db = " + db);
        System.out.println("dc = " + dc);
        System.out.println("dd = " + dd);
        System.out.println("de = " + de);
    }
}

```



```
}  
}
```

该程序的输出结果为：

- 4、 下面的示例程序说明了模运算符 % 的用法 , 写出以下程序的运行结果：

```
public class Modulus {  
    public static void main(String args[]) {  
        int x = 42;  
        double y = 42.25;  
        System.out.println("x mod 10 = " + x % 10);  
        System.out.println("y mod 10 = " + y % 10);  
    }  
}
```

该程序的输出结果为：

- 5、 下面的例子显示了几个赋值运算符的作用 , 写出以下程序的运行结果：

```
// Demonstrate several assignment operators. public class OpEquals {  
    public static void main(String args[]) {  
        int a = 1;  
        int b = 2;  
        int c = 3;  
        a += 5;  
        b *= 4;  
        c += a * b;  
        c %= 6;  
        System.out.println("a = " + a);  
        System.out.println("b = " + b);  
        System.out.println("c = " + c);  
    }  
}
```

该程序的输出结果为：

- 6、 下面的程序说明了递增运算符的使用 , 写出以下程序的运行结果：

```

public class IncDec {
    public static void main(String args[]) {
        int a = 1;
        int b = 2;
        int c;
        int d;
        c = ++b;
        d = a++;
        c++;
        System.out.println("a = " + a);
        System.out.println("b = " + b);
        System.out.println("c = " + c);
        System.out.println("d = " + d);
    }
}

```

该程序的输出结果为：

7、 下面的例子说明了位逻辑运算符，写出以下程序的运行结果：

```

public class BitLogic {
    public static void main(String args[]) {
        String binary[] = { "0000", "0001", "0010", "0011", "0100", "0101",
            "0110", "0111", "1000", "1001", "1010", "1011", "1100", "1101",
            "1110", "1111" };
        int a = 3; // 0 + 2 + 1 or 0011 in binary
        int b = 6; // 4 + 2 + 0 or 0110 in binary
        int c = a | b;
        int d = a & b;
        int e = a ^ b;
        int f = (~a & b) | (a & ~b);
        int g = ~a & 0x0f;
        System.out.println(" a = " + binary[a]);
        System.out.println(" b = " + binary[b]);
        System.out.println(" a|b = " + binary[c]);
        System.out.println(" a&b = " + binary[d]);
        System.out.println(" a^b = " + binary[e]);
        System.out.println(" ~a&b|a&~b = " + binary[f]);
        System.out.println(" ~a = " + binary[g]);
    }
}

```

```
}
```

该程序的输出结果为：

- 8、 下面的示例程序说明了左移运算符的用法 ,写出以下程序的运行结果：

```
public class ByteShift {  
    public static void main(String args[]) {  
        byte a = 64, b;  
        int i;  
        i = a << 2;  
        b = (byte) (a << 2);  
        System.out.println("Original value of a: " + a);  
        System.out.println("i and b: " + i + " " + b);  
    }  
}
```

该程序的输出结果为：

- 9、 下面的示例程序说明了右移运算符的用法 ,写出以下程序的运行结果：

```
public class HexByte {  
    static public void main(String args[]) {  
        char hex[] = { '0', '1', '2', '3', '4', '5', '6', '7', '8', '9', 'a', 'b', 'c', 'd', 'e', 'f' };  
        byte b = (byte) 0xf1;  
        System.out.println("b = 0x" + hex[(b >> 4) & 0x0f] + hex[b & 0x0f]);  
    }  
}
```

该程序的输出结果为：

- 10、 下面的示例程序说明了位运算符的用法 ,写出以下程序的运行结果：

```
public class OpBitEquals {  
    public static void main(String args[]) {  
        int a = 1;  
        int b = 2;
```

```

int c = 3;
a |= 4;
b >>= 1;
c <<= 1;
a ^= c;
System.out.println("a = " + a);
System.out.println("b = " + b);
System.out.println("c = " + c);
}
}

```

该程序的输出结果为：

- 11、 下面的示例程序说明了布尔型运算符的用法 ,写出以下程序的运行结果：

```

public class BoolLogic {
public static void main(String args[]) {
boolean a = true;
boolean b = false;
boolean c = a | b;
boolean d = a & b;
boolean e = a ^ b;
boolean f = (!a & b) | (a & !b);
boolean g = !a;
System.out.println(" a = " + a);
System.out.println(" b = " + b);
System.out.println(" a|b = " + c);
System.out.println(" a&b = " + d);
System.out.println(" a^b = " + e);
System.out.println("!a&b|a&!b = " + f);
System.out.println(" !a = " + g);
}
}

```

该程序的输出结果为：

- 12、 下面的示例程序说明了？运算符的用法 ,写出以下程序的运行结果：

```

public class Ternary {
    public static void main(String args[]) {
        int i, k;
        i = 10;
        k = i < 0 ? -i : i; // get absolute value of i
        System.out.print("Absolute value of ");
        System.out.println(i + " is " + k);
        i = -10;
        k = i < 0 ? -i : i; // get absolute value of i
        System.out.print("Absolute value of ");
        System.out.println(i + " is " + k);
    }
}

```

该程序的输出结果为：

13、写出下面表达式的运算结果，设

a=2,b=-4,c=true。

(1) -- a % b++

(2) (a >= 1 && a <= 10 ? a : b)

(3) c ^ (a > b)

(4) (-- a) << a

(5) (double)(a+b)/5+a/b

答案为：

第二章 控制结构

一、选择题

1、所有的 java 应用程序都是按照 () 类型的程序控制来进行的编写的。()

A、一种
种

B、两种

C、三种

D、四

2、应用程序中的语句按照它们所缩写的次序逐条的执行过程被称为 ()

A、控制转移
均不对

B、顺序执行

C、工作流程

D、以上

3、下面几个选项中是循环语句的是 ()

A、if 语句

B、if-else 语句

C、for 语句

D、try-catch

语句

4、ifelse 是一个 () 的选择语句

A、 单向 B、 双向 C 、 三向 D 、 嵌套

5、将一个 if ... else 语句放置在另一个 if ... else 语句的内部，属于() 的一个例子

A、 嵌套 if ... else 语句 B、 堆积 if ... else 语句

C、 创建顺序的 if ... else 语句 D、 以上答案都不对

6、能从循环语句的循环体中跳出的语句是 ()

A、 if 语句 B、 break 语句 C、 for 语句 D、 continue 语句

7、下面对 continue 的用法叙述正确的是： ()

A、跳过循环体下面的语句，继续下次循环 B、跳出循环体

C、下次循环前不需要判断终止条件 D、以上说法

都不对

8、下面哪条语句可对某次考试中凡成绩在 90 或者 90 分以上的学生显示出 A ? ()

A、 if(studentGrade !=90)
System.out.println("student received an A")

B、 if(studentGrade >90)
System.out.println("student received an A")

C、 if(studentGrade <=90)
System.out.println("student received an A")

D、 if(studentGrade >=90)
System.out.println("student received an A")

9、执行语句 for(int i=1;i++<4;); 后变量 i 的值是 .()

A、 3 B、 4 C、 5 D、 不定

10、以下的描述中，正确的是 ()

A、 continue 语句的作用是结束整个循环的执行

B、只能在循环体内和 switch 语句体内使用 break 语句

C、在循环体内使用 break 语句或 continue 语句的作用相同

D、从多层循环嵌套中退出时，只能使用 goto 语句

11、下面程序段的运行结果是 ()

```
int n=0;
```

```
while(n++<=2); System.out.println (n);
```

A、 2 B、 3 C、 4 D、 有语法

错

12、下面程序的运行结果是 ()

```
{
```

```
int num=0;
```

```
while(num<=2){
```

```

        num++;
        System.out.println(num);
    }
}

```

- A、 1 B、 1 C、 1 D、 1
 2 2 2
 3 3
 4

13、能从循环语句中跳出来的语句是 ()
 A、 for 语句 B、 break 语句 C、 while 语句 D、 continue 语句

14、对 java 嵌套语句描述错误的是 ()
 A、 for 语句可以嵌套 if 语句 B、 if 语句可以嵌套循环语句
 C、 while 语句可以嵌套 for 语句 D、 循环语句中可以用 goto 语句

15、java 程序基本结构不包括 ()
 A、 顺序结构 B、 分支结构 C、 循环结构 D、 逻辑结构

16、运行以下程序段，其正确的运行结果是 ()

```

x=-1;
do{
    x=x*x;
}

```

while(!x);
 A、 是死循环 B、 循环执行二次 C、 循环执行一次 D、 有语法错误

17、若有如下语句

```

int x=3;
do{ System.out.println(x);
    x-=2;}
while(--x>0);

```

则上面程序段运行的结果为 ()

A、 输出的是 1 B、 输出的是 1 和-2 C、 输出的是 3 D、 是死循环

18、设有程序段

```

int k=10;
while(k==0)k=k-1;

```

这下面描述中正确的是 ()

- A、 while 循环执行 10 次 B、 循环是无限循环
 C、 循环体语句一次也不执行 D、 循环体语句执行一次

19、下面语句执行结果为 ()

```
public static void main(String[] args){  
    int i=4;  
    if(i<0){  
        i=1;}  
    else{i=2;}  
    System.out.println(i);
```

A、 0

B、 1

C、 2

D、 4

20、下面执行语句输出结果为 ()

```
public class nam {  
    public static void main(String[] args) {  
        int sum = 0;  
        for(int i=0;i<=10;i++){  
            sum=sum+i;  
        }  
        System.out.println(sum);  
    }  
}
```

A、 50

B、 55

C、 0

D、 54

21、下列语句序列执行后 ,m 的值是 ().

```
int a=10, b=3, m=5;  
if( a==b ) m+=a; else m=++a*m;  
System.out.println(m);
```

A、 15

B、 50

C、 55

D、 5

22、下列语句序列执行后 ,x 的值是 ()

```
int a=2, b=4, x=5;  
if( a<--b ) x*=a;  
System.out.println(x);
```

A、 5

B、 20

C、 15

D、 10

23、位于 switch 关键字之后且处于圆括号内的表达式被称为一个 ()

A 、 警戒条件
表达式

B 、 控制表达式

C、 选择语句

D、 case

24、() 将用于确定循环是否继续迭代

A、 控制变量的初始值

B 、 右花括号

C 、 左花括号

D 、 控制变量

25、在一个 for 循环中 , 对控制变量的自增或自减运算时发生在 ()

A、 循环体执行结束以后

B、 循环体执行以前

C 、 当循环 -继续条件为 FALSE 时

D、 当执行循环体时

26、switch 是一条 () 选择语句

A 、 单向 B 、 双向 C 、 多向 D 、 以上答案都不对

27、() 将导致程序控制进入位于 switch 后的条件语句

A、 分号 B 、 stop 关键字 C、 break 语句 D、 case 语句结束

28、() 表示一条 switch 语句结束

A 、 end 关键字 B、 } 字符 C、 break 关键字 D、 default

29、下列语句序列执行后 ,k 的值是 ()

```
int i=10, j=18, k=30;
```

```
switch( j - i )
```

```
{ case 8 : k++;
```

```
case 9 : k+=2;
```

```
case 10: k+=3;
```

```
default : k/=j;
```

```
}
```

A、 31 B、 32 C、 2 D、 33

30、若 a 和 b 均是整型变量并已正确赋值 ,正确的 switch 语句是 ()

A、 switch(a+b); { } B、 switch(a+b*3.0) { }

C、 switch a { } D、 switch (a%b) { }

31、下列语句序列执行后 , i 的值是 ()

```
int i=16;
```

```
do { i/=2; } while( i > 3 );
```

```
System.out.println(i)
```

A、 16 B、 8 C、 4 D、 2

32、下列语句序列执行后 , k 的值是 ()

```
int j=8, k=15;
```

```
for( int i=2; i!=j; i++ ) { j-=2; k++; }
```

```
System.out.println(k)
```

A、 18 B、 15 C、 16 D、 17

33、下列语句序列执行后 ,r 的值是 ()

```
char ch='8'; int r=10;
```

```
switch( ch+1 )
```

```
{
```

```
case '7': r=r+3;
```

```
case '8': r=r+5;
```

```
case '9': r=r+6; break;
```

```
default: r=r+8;
```

}

A、 13 B、 15 C、 16 D、 18

34、 设 int 型变量 a,b,float 型变量 x,y,char 型变量 ch 均已正确定义并赋值 ,正确的 switch 语句是 ().

A、 switch (x + y) { } B、 switch (ch + 1) { }

C、 switch ch { } D、 switch (a + b); { }

35、 下列语句序列执行后 ,k 的值是 ().

int x=6, y=10, k=5;

switch(x%y)

{

case 0: k=x*y;

case 6: k=x/y;

case 12: k=x-y;

default: k=x*y-x;

} System.out.println(k);

A、 60 B、 5 C、 0 D、 54

36、 下列语句序列执行后 ,r 的值是 ().

char ch='A'; int r=6;

switch(ch+5)

{

case 'A': r=r+3;

case 'B': r=r+5;

case 'C': r-=6; break;

default : r/=2;

}

A、 11 B、 3 C、 2 D、 9

37、 下列语句序列执行后 ,j 的值是 ().

int j=1;

for(int i=5; i>0; i-=2) j*=i;

A、 15 B、 1 C、 60 D、 0

38、 包含多条语句的 if 语句体应放置在一对 () 中

A 、 () B、 [] C 、 <> D、 {}

39、 以下 for 循环的执行次数是 ().

for(int x=0;(x==0)&(x0; i-=2) j*=2;

A、 15 B、 1 C、 60 D、 32

40、 以下 for 循环的执行次数是 ().

for(int x=0,y=0;(y!=0)&&(x0; i--) j+=i;

A、 3 B、 4 C、 5 D、 6

41、 以下由 for 语句构成的循环执行的次数是 ().

for (int i = 0; true ; i++) ;

A、有语法错 ,不能执行

B、无限次

C、执行 1 次

D、一次也不执行

42、下面语句执行后 ,i 的值是 ().

```
for( int i=0, j=1; j < 5; j+=3 ) i=i+j;
```

A、 4

B、 5

C、 6

D、 7

43、下列语句序列执行后 ,k 的值是 ().

```
int j=8, k=15;
```

```
for( int i=2; i!=j; i++ ) { j-=2; k++; }
```

A、 18

B、 15

C、 16

D、 17

44、下列语句序列执行后 ,k 的值是 ().

```
int m=3, n=6, k=0;
```

```
while( (m++) < ( -- n) ) ++k;
```

A、 0

B、 1

C、 2

D、 3

45、下列语句序列执行后 ,i 的值是 ().

```
int s=1, i=1;
```

```
while( i<=4 ) {s*=i; i++;}
```

A、 6

B、 4

C、 24

D、 5

46、 下列语句序列执行后 ,k 的值是 ().

```
int m=3, n=6, k=0;
```

```
while( m > 4 ) --j;
```

A、 5

B、 6

C、 7

D、 8

47、下列语句序列执行后 ,j 的值是 ().

```
int j=8, i=6;
```

```
while( i > 4 ) i-=2;
```

```
- j;
```

A、 5

B、 6

C、 7

D、 8

48、下列语句序列执行后 ,j 的值是 ().

```
int j=9, i=6;
```

```
while( i-- > 3 ) --j;
```

A、 5

B、 6

C、 7

D、 8

49、下列语句序列执行后 ,j 的值是 ().

```
int j=3, i=2;
```

```
while( --i!=i/j ) j=j+2;
```

A、 2

B、 4

C、 6

D、 5

50、 若有循环 :

```
int x=5, y=20;
```

```
do { y-=x; x+=2; }while(x<Y);
```

则循环体将被执行 ().

A、 2 次

B、 1 次

C、 0 次

D、 3 次

51、 以下由 do-while 语句构成的循环执行的次数是 ().

int m = 8;

do { ++m; } while (m 1);

A、 1 B、 5 C、 2 D、 0

52、下列语句序列执行后 ,i 的值是 ().

int i=10;

do { i/=2; } while(i--> 1);

A、 1 B、 5 C、 2 D、 -1

53、下列语句序列执行后 ,i 的值是 ().

int i=10;

do { i-=2; } while(i>6);

A、 10 B、 8 C、 6 D、 4

54、下列语句序列执行后 ,i 的值是 ().

int i=16;

do { i/=2; } while(i > 3);

A、 16 B、 8 C、 4 D、 2

55、以下由 do-while 语句构成的循环执行的次数是 ().

int k = 0;

do { ++k; }while (k < 1);

A、 一次也不执行 B、 执行 1 次 C、 无限次 D、 有语法错,不能执行

56、 若有循环 :

int x=5,y=20;

do{ y-=x; x++; } while(++x<--y);

则循环体将被执行 ().

A、 0 次 B、 1 次 C、 2 次 D、 3 次

57、MAX_LENGTH 是 int 型 public 成员变量 , 变量值保持为常量 100 , 用简短语句定义这个变量。 ()

A、 public int MAX_LENGTH=100;

B、 final int MAX_LENGTH=100;

C、 final public int MAX_LENGTH=100;

D、 public final int MAX_LENGTH=100;

58、在一个 do...while 语句中,关键字 while 出现在 ()

A 、 do 关键字之后, do...while 循环体之前

B 、 do 关键字之后,并作为循环体中的最后一条语句

C 、 do 关键字之前

D 、 do 关键字之后, do...while 循环体之后

59、计数器控制循环中的 () 要素将用于确定循环变量在每次的循环迭代中如何进行修改

A、 名称 B 、 初始值 C、 自增或自减运算 D、 最终值

- 60、for 语句内，位于第一个分号前的值表示（ ）
 A、 计数器变量的初始值 B、 计数器变量的最终值
 C、 自增运算 D、 语句迭代的次数
- 61、for 循环语句的第 1 行常量被非正式的称为（ ）
 A、 for 语句体 B、 for 首部 C、 for 计数器 D、 for 表达式
- 62、若 switch 语句中控制表达式不与任何一个 case 标签相等且不存在 default case，则（ ）
 A、 出现一个错误 B、 出现一个无限循环
 C、 不执行任何语句 D、 首个 case 语句将得带执行
- 63、若存在一个 case 标签与控制表达式相匹配，则应用（ ）
 把该 case 标签所执行的代码相分离
 A、 一个冒号 B、 一个下划线 C、 break 关键字 D、 一个分号
- 64、基本类型（ ）不能用与 switch 语句相兼容
 A、 double B、 int C、 short D、 char
- 18、switch 语句中表达式（ expression ）的值不允许使用的类型是（ ）
 A、 byte B、 int C、 boolean D、 char
- 65、对于下列代码、

```

public class Parent {
    public int addValue( int a, int b) {
        int s;
        s = a+b;
        return s;
    }
}
class Child extends Parent {

```

 下述哪些方法可以加入类 Child？（ ）
 A、 int addValue(int a, int b){// do something...}
 B、 public void addValue (int a, int b){// do something...}
 C、 public int addValue(int a){// do something...}
 D、 public int addValue(int a, int b)throws MyException {//do something...}
- 66、下面哪些选项是正确的 main 方法说明？（ ）
 A、 public main(String args[])
 B、 public static void main(String args[])
 C、 private static void main(String args[])

D、 void main()

67、已知代码、

```
Public class ex_3_1_2_if{
Public static void main(String []asd){
int a=5;
if(a>=5){
System.out.println( “ a>=5 是真值 ” );}
else{System.out.println( “ a>=5 是假值 ” );}
}
}
}
```

请问最终将打印什么语句 ()

A 、 False B、 a>=5 是真值 C 、 True D 、 a>=5 是假值

68、已知代码、

```
public class ex_3_1_2_switch{
public static void main(String []asd){
int a=2
switch(a){
case 1:
System.out.println( “ a=1 ” );
case 2:
System.out.println( “ a=2 ” );
break;
case 3:
System.out.println( “ a=3 ” );
break;
default:
System.out.println( “ a!=1&&a!=2&&a!=3 ” );
break;
}
}
}
```

请问结果将输出什么 ()

A 、 1 B、 a=1 C、 a=1a=2 D、 a=2

69、下列语句序列执行后， i 的值是 ()。

```
int i=16;
do { i/=2; } while( i > 3 );
```

A、 16 B、 8 C、 4 D、 2

70、下列语句序列执行后， k 的值是 ()。

```
int j=8, k=15;
for( int i=2; i!=j; i++ ) { j-=2; k++; }
```

A、 18 B、 15 C、 16 D、 17

71、 阅读下面的代码，回答问题， ()

```
for( m = 0 ; m > -2 ; m -- ){ , . }
```

For 循环执行多少次、

A、 0 B、 1 C、 2 D、 3

72、 阅读下面的代码，回答问题，

```
for( m = 0; m < 5; m++ )
```

```
{
```

```
    System.out.print( m + "," );
```

```
    if( m == 3 )
```

```
        break;
```

```
}
```

执行结果是、 ()

A、 0,1,2,3,4,5 B、 0,1,2,3,4 C、 0,1,2,3 D、 0,1,2

73、 编译运行下面的代码，会有什么结果产生、 ()

```
int i = 1;
```

```
switch (i) {
```

```
case 0:
```

```
System.out.println("zero");
```

```
break;
```

```
case 1:
```

```
System.out.println("one");
```

```
case 2:
```

```
System.out.println("two");
```

```
default:
```

```
System.out.println("default");
```

```
}
```

A、 One B、 one, default C、 one, two, default D、

Default

74、 编译运行下面的代码，会有什么结果产生、 ()

```
int i = 9;
```

```
switch (i) {
```

```
default:
```

```
System.out.println("default");
```

```
case 0:
```

```
System.out.println("zero");
```

```
break;
```

```
case 1:
```

```
System.out.println("one");
case 2:
System.out.println("two");
}
```

- | | |
|-------------------------------------|------------------|
| A、 Default | B、 default, zero |
| C、 error default clause not defined | D、 no output |

displayed

75、下面不会在编译时出错的是 ()

- A、 int i=0;
- ```
if(i)
{
 System.out.println("Hello");
}
```
- B、 boolean b = true;
- ```
boolean b2 = true;
if(b==b2)
    System.out.println("So true");
```
- C、 int i=1;
- ```
int j = 2;
if(i ==1 & j==2)
 System.out.println("OK");
```
- D、 int i=1;
- ```
int j = 2;
if(i ==1 &| j==2)
    System.out.println("OK")
```

76、当 x 的值为 2 时，下面代码的运行结果是、 ()

```
switch (x) {
case 1:
System.out.println(1);
case 2:
case 3:
System.out.println(3);
case 4:
System.out.println(4);
}
```

- | | | | |
|------------|------|-------|--------|
| A、 什么都不会输出 | B、 3 | C、 34 | D、 134 |
|------------|------|-------|--------|

77、以下哪个不是循环语句的内容 ()

- | | | | |
|----------|---------|---------|----------|
| A、 初始化部分 | B、 循环条件 | C、 迭代部分 | D、 布尔表达式 |
|----------|---------|---------|----------|

78、以下 4 个 for 语句，合法的是 ()

A 、 int i;
for(i=5,int
for(i=0,j=10;i<10,j>0;i++,j--){}

B 、 int i,j
j=10;i<10;j--){}

C、 int i,k;
for(i=0,k=0;(i<10&&k>0);i++,k--){}
for(i=0;j=10;i<10;i++;j--){}

D 、 int i,j;

79、下面循环体内的运算 x+=i 执行了 () 次

int x=0;
for(int i=0;i<6;i++){
x+=i;}

A、 4 B、 5 C、 6 D、 7

80、下面循环体内的运算 x+=j 执行了 () 次。

int x=0;
for(int i=0;j<6;i++){
for(int j=0;j<6;j++){
x+=j;}
}

A 、 34 B、 36 C、 35 D 、 40

81、当一个被执行的语句并不是按照应用程序编写的次序来执行的，
则是被认为是发生了 ()

A、 转换 B、 流动 C、 逻辑错误 D、 控制转移

82、 if 语句被称为是一个 () 语句，因为它能够选择忽略一个
操作

A、 单向选择 B 、 多向选择 C、 双向选择 D、 循
环

83、程序控制的三种类型是顺序、选择和 ()

A、 简化 B、 判定 C、 分支 D、 循环

84、 while 语句的循环体将 ()

A 、 至少执行一次 B、 从不执行
C、 在其条件为 true 时执行 D、 在其条件为 false 时执
行

85、计数器控制的循环有时也被称为 () 因此新欢次数在循环
开始执行前是已知的

A、 有限循环 B、 已知循环 C 、 顺序循环 D、 计数
器循环

86、 () 语句将一直执行直至其循环 -继续条件变为 FALSE 时为止

A、 while B、 if C、 until D 、 if ...else

87、当 while 语句中的条件语句不会变为 FALSE 时，为导致出现
() 循环

A、 不确定 B、 未定义 C、 嵌套 D、 无限

- 88、do...while 语句对循环 -继续条件的测试时发生在 ()
 A、 循环体执行完成后 B、 循环体执行完成以前
 C 、 A 和 B D、 以上答案都不对
- 89、当 do...while 语句中循环 -继续条件 () 时，会出现一个无限循环
 A、 不会变为 TRUE B 、 不会变为 FALSE
 C 、 为 FALSE D、 被反复测试
- 90、当一个 do...while 中循环 -继续条件不会变为 FALSE 时，会出现 ()
 A、 无限循环 B、 计数器控制的循环 C、 控制语句 D、 潜逃的控制语句
- 91、() 语句将至少执行一次他的语句体， 且能够继续执行下去直至其循环 -继续条件变为 FALSE 时为止
 A、 while B 、 if C 、 do...while D 、 if...else
- 92、某循环其执行的迭代次数多一次或少一次的现象被称为 ()
 A 、 无限循环 B、 计数器控制循环 C、 偏一错误 D、 嵌套的控制 语句
- 93、如果循环 -继续条件在首次计算中就为 FALSE ，则接下来的 do...while 循环语句 ()
 A、 执行器循环体知道条件为 TRUE B 、 执行其循环体知道条件为 FALSE
 C 、 不会执行 D、 只对循环体执行一次

94、阅读下列代码

```
Public class Test2006{
  Public static void main(String[] args){
    String s= "Test";
    Switch(s){
    case"Java":System.out.println( "Java");break;
    case"Test":System.out.println( "Test");break;
    case".Net ":System.out.println( ".Net ");break;
    }}}
  其运行结果是 ( )
```

- A . Java B、 Language C、 Test D、 编译出错
- 95、关于以下程序段，正确的说法是 ()
- ```
1 . String s1= " abc " + " def " ;
2 . String s2=new String (s1);
3 . if (s1==s2)
4 . System.out.println(" == succeeded ");
5 . if (s1.equals(s2))
```

6.                      System.out.println(                      “ .equals() succeeded                      ” );

A、行 4 与行 6 都将执行

B、行 4 执行，行 6 不执行

C、行 6 执行，行 4 不执行

C、行 4、行 6 都不执行

96、若有如下程序段，其中 s、a、b、c 均已定义为整型变量，且 a、c 均已赋值（c 大于 0）。

s=a;

for(b=1;b<=c;b++) s=s+1;

则与上述程序段功能等价的赋值语句是（                      ）。

A、s=a+b;

B、s=a+c;

C、s=s+c;

D、s=b+c;

97、认真阅读下面的程序：

```
class TestApp{
 public static void main(String[] args){
 for(int i=0;i<5;i++)
 System.out.print(i+1);
 System.out.println(i);
 }
}
```

上述程序运行后的结果是（                      ）

A、123456

B、123455

C、123450

D、编译

错误

98、认真阅读下面的程序：

```
class TestApp{
 public static void main(String[] args){
 for(int i=0;i<10;i++){
 if(i==3)
 break;
 System.out.print(i);
 }
 }
}
```

程序运行后的输出是（                      ）

A、0123

B、012

C、0123456789

D、

012456789

99、所有的程序均可以用几种类型控制结构编写。（                      ）

A、顺序结构、选择结构、循环结构

B、顺序结构、循环结构

C、顺序结构、选择结构

D、选择结构、循环结构

100、当条件为真和条件为假时，（                      ）控制结构可以执行不同的动作。

A、switch

B、while

C、for

D、if/else

101、下面程序片段输出的是什么？（                      ）

```
int a=3;
int b=1;
if(a=b)
System.out.println(" a= " +a);
```

A、 a=1    B、 a=3    C 、 编译错误 , 没有输出    D、 正常运行 , 但没有输出

102、下面程序的输出结果是 : (            )

```
public class Test{
void printValue(int m){
do { System.out.println(" The value is " +m);
}
while(-- m>10);
}
public static void main(String arg[]){
int i=10;
Test t= new Test();
t.printValue(i);
}
}
```

A 、 8            B 9            C 、 10            D 、 11

103、下列语句执行后 , x 的值为 (            )

```
int a=4 , b=5 , x=3;
if(++a==b) x=x*a;
```

A、 3            B、 12            C、 15            D、 20

104、下列语句执行后 , z 的值为 (            )

```
int x=3,y=4,z=0;
switch(x%y+2)
{
case 0:z=x*y;break;
case 6:z=x/y;break;
case 12:z=x-y;break;
default:z=x*y-x;
}
```

A、 15            B、 0            C、 2            D 、 12

105、请看下面的程序代码 : (            )

```
if(x>0) { System.out.println(" first ");}
else if(x<20) { System.out.println(" second ");}
else { System.out.println(" third ") }
```

当程序输出 “ second ” 时 , x 的范围为 :

A、 x <= 0            B、 x < 20 && x > 0            C、 x > 0            D 、 x >= 20

106、请看下面的程序代码： ( )

```
switch(n) {
case 0: System.out.println(" first ");
case 1:
case 2: System.out.println(" second "); break;
default: System.out.println(" end ");
}
```

当 n 为何值时，程序段将输出字符串 second：

- A、 0                      B、 1                      C、 2                      D、 以上都可以

107、下列语句执行后， x 的值是 ( )

```
int x=2;
do
{x+=x;}while(x<17);
```

- A、 4                      B、 16                      C、 32                      D、 256

108、下列语句执行后， j 的值是 ( )

```
int j=3,i=3;
while(--i!=i/j)
j=j+2;
```

- A、 4                      B、 5                      C、 6                      D、 7

109、下列语句执行后， i , j 的值是 ( )

```
int i=1, j=8;
do
{if(i++>--j)
continue ; }while(i<4);
```

- A、 i=4 , j=5                      B、 i=5 , j=4                      C、 i=4 , j=5                      D、 i=5 , j=6

110、下列语句执行后， k 的值是 ( )

```
int j=4,i,k=10;
for(i=2;i!=j;i++)
k=k-i;
```

- A、 4                      B、 5                      C、 6                      D、 7

111、下列语句执行后， c 的值是 ( )

```
char c='\0';
for(c='a';c<'z';c+=3)
{
if(c>='e')
break;
}
```

- A、 ?e?                      B、 ?d?                      C、 ?f?                      D、 ?g?

112、下列语句执行后， k 的值是 ( )

```

int i,j,k=1;
lable:for(i=1;i<8;i++)
{
for(j=1;j<6;j++) {
k*=i+j;
if(j==3)
break lable;
}
}

```

A、 22                      B、 23                      C 、 24                      D、 25

113、 下列语句执行后， k 的值是 (                      )

```

int i,j,k=1;
lable:for(i=1;i<3;i++)
{
for(j=1;j<4;j++)
{
k+=i+j;
if(j==3)
continue lable;
}
}

```

A、 22                      B 、 23                      C 、 24                      D、 25

## 二、 填空题

- 1、 流程控制分为三种， 分别是 \_\_\_\_\_、 \_\_\_\_\_和 \_\_\_\_\_。
- 2、 Java 支持两种分支语句， 分别是 \_\_\_\_\_和 \_\_\_\_\_。
- 3 、 Java 支持三种循环语句， 分别是 \_\_\_\_\_， \_\_\_\_\_， \_\_\_\_\_。
- 4、 如果程序中的语句是以选择性语句来判断选择下一步操作， \_\_\_\_\_ 此语法结构就是 \_\_\_\_\_。
- 5、 如果程序中的语句是以重复性语句来做程序的结构称为 \_\_\_\_\_， 也叫重复结构。
- 6、 正如汉语中的“如果 \_\_\_\_\_， 那么就 \_\_\_\_\_， ”，“如果 \_\_\_\_\_， 那么就 \_\_\_\_\_， 否则就 \_\_\_\_\_， ”， 这样的语句我们称之为 \_\_\_\_\_， 也称为分支语句。
- 7、 分支语句提供了这样一种控制机制， \_\_\_\_\_ 它根据条件使得程序的执行可以跳过某些语句不执行， 而转去执行特定的语句。 分支语句主要有两种、 \_\_\_\_\_和 \_\_\_\_\_。
- 8、 if 语句是选择结构中最基本的语句， 它有两种基本形式： 与 \_\_\_\_\_语句。
- 9、 语法格式中的“布尔表达式”为关系表达式或逻辑表达式， 其值类

型为\_\_\_\_\_。

10、\_\_\_\_\_首先执行循环体，然后再判断循环条件。

11、\_\_\_\_\_的作用是中断当前循环的执行，如果是放在一个多重循环结构中，它只会中断当前的循环，而不会完全结束循环。

12、\_\_\_\_\_的作用是中断当次循环的执行，循环将直接进入下一次循环。

13、能保证循环体至少被执行 1 次的循环语句是\_\_\_\_\_。

14、If 语句中的表达式的值必须是\_\_\_\_\_的。

15、If 语句中也可以是布尔类型的\_\_\_\_\_、\_\_\_\_\_，或\_\_\_\_\_。如果是其他类型，则编译出错。

16、java 中的选择结构分为\_\_\_\_\_、\_\_\_\_\_及\_\_\_\_\_三种。

17、java 中的选择结构分别用\_\_\_\_\_和\_\_\_\_\_实现。

18、一种结构执行完毕后，就进入下一个结构。从宏观上看，任何算法结构都是\_\_\_\_\_。

19、java 语句的循环语句实现循环结构的操作，\_\_\_\_\_循环语句由\_\_\_\_\_和两部分构成，循环体是要被反复执行的语句，循环条件决定循环的开始、重复执行以及结束循环。

20、\_\_\_\_\_可以有条件或无条件地改变程序的执行顺序。

21、java 支持三种控制转移语句、\_\_\_\_\_和\_\_\_\_\_。

22、break 语句的作用是使程序的流程从一个语句块内部转移出去，\_\_\_\_\_只在\_\_\_\_\_和\_\_\_\_\_中使用，允许从 switch 语句的 case 子句中跳出，或从循环体内部跳出，不允许跳入任何语句块内。

23、if ( ) 语句括号中是个\_\_\_\_\_。

24、while 是关键字，循环条件结果是\_\_\_\_\_，当值为 true 时，执行循环体；当值为 false 时，循环结束。

25、while 语句在循环一开始就计算循环条件表达式，若表达式的值为\_\_\_\_\_，则循环体一次也不会执行

26、\_\_\_\_\_循环是先判断后执行，\_\_\_\_\_循环先执行后判断，先执行初始化部分，再执行循环条件的布尔表达式，为 false，则不执行循环体，退出循环，.如果为 true，就执行循环体 .接着执行迭代部分。

27、\_\_\_\_\_是多个单循环的嵌套使用，\_\_\_\_\_循环语句的循环体内又有循环语句，则构成多重循环结构。

28、常用的多重循环有\_\_\_\_\_和\_\_\_\_\_。

29、单循环结构只有一个 for 循环，而在双重循环中，在 for 循环中还嵌套了另一个 for 循环，这样就形成了一个\_\_\_\_\_。

30、for 循环中还可以嵌套 while 循环，同样，我们也可以在 while 循环中嵌套 for 循环。

有三个循环结构嵌套使用的叫\_\_\_\_\_。

31、当需要处理二维数据时，就会采用\_\_\_\_\_来处理。

- 32、break 语句的作用、 中断当前循环的执行 ,如果 break 语句是放在一个 \_\_\_\_\_ , 它只会中断当前层次的循环 , 而不是完全结束循环
- 33、 \_\_\_\_\_ 是多个单循环的嵌套使用。
- 34、循环语句的 \_\_\_\_\_ 又有循环语句 , 则构成多重循环结构 ;
- 35、 \_\_\_\_\_ 中 break 语句的作用、 中断当前循环的执行 ;
- 36、 \_\_\_\_\_ 的作用、 是 ( 中断 ) 当次循环的执行 , 循环将直接进入下一次循环 ;
- 37、 \_\_\_\_\_ 的作用、 中断当前所有 ( 循环 ) \_\_\_\_\_ , 并且直接跳出当前的方法。
- 38、如果循环体包含多条 \_\_\_\_\_ , 必须将其放在大括号内 , 若循环体只有一条语句 , 虽然不用写花括号表示也是可以的 , \_\_\_\_\_ 但作为代码规范 , 我们要求不能省略花括号。
- 39、 \_\_\_\_\_ 、 \_\_\_\_\_ 和 do-while 语句 的循环体可以为 \_\_\_\_\_ , 这是因为一个空语句 ( 仅由一个分号组成的语句 ) \_\_\_\_\_ 在语法上是合法的。
- 40、 while 循环语句中 , 循环条件的测试都是在循环开始前进行的 , 称为 \_\_\_\_\_ 。 如果一开始条件就不成立 , 则循环体语句可能一次也不执行。
- 41、do-while 循环 , 它对循环条件的测试是在执行循环语句之后进行的 , 称为 \_\_\_\_\_ , 这意味着一个 do-while 循环至少要执行一次。
- 42、在执行 for 语句时 , 先执行 \_\_\_\_\_ , 这部分只会被执行一次 , 接下来执行循环条件的布尔表达式 , 为 \_\_\_\_\_ , \_\_\_\_\_ 则不执行循环体 , 退出循环 , 如果为 \_\_\_\_\_ , 就执行循环体接着执行迭代部分 , \_\_\_\_\_ . 然后再计算作为循环条件的表达式
- 43、 \_\_\_\_\_ 语句是三个循环语句中功能最强 , 使用最广泛的一个
- 44、Java 语言没有提供 \_\_\_\_\_ 语句来控制程序的跳转 , \_\_\_\_\_ 这种做法提高了程序流程控制的可读性 , 但降低了程序流程控制的灵活性。
- 45、Java 提供了 \_\_\_\_\_ 和 \_\_\_\_\_ 来控制循环结构。除此之外 , \_\_\_\_\_ return 可以结束整个方法 , 当然也就结束了一次循环。
- 46、程序遇到一个 \_\_\_\_\_ 语句 , 这行代码将会导致结束 \_\_\_\_\_ outer 标签指定的循环 , \_\_\_\_\_ 不是结束 break 所在的循环 , 而是结束 break 循环的外层循环。
- 47、顺序结构是按照 \_\_\_\_\_ 的顺序执行的结构。
- 48、选择结构是以 \_\_\_\_\_ 来判断选择下一步操作的。
- 49、 \_\_\_\_\_ 三种循环语句都能解决循环次数已经确定的次数型循环 , 其中循环语句最适合。
- 50、循环结构是以 \_\_\_\_\_ 语句来做程序的结构。
- 51、在 java 中 , 执行控制的关键字包括 \_\_\_\_\_ 、 \_\_\_\_\_ 、 \_\_\_\_\_ 、 \_\_\_\_\_ 和 \_\_\_\_\_ 语句。
- 52、半段语句 ( if-else ) 中的 if 括号后是一个 \_\_\_\_\_ 值。



53、else 和 \_\_\_\_\_ 的 if 产生了匹配。

54、循环语句中 \_\_\_\_\_ 语句用于已知循环次数的循环中。

55、循环语句中 \_\_\_\_\_ 语句用于不知道循环次数的语句中。

56、do-while 也是迭代的一种，它和 \_\_\_\_\_ 语句大同小异。

57、\_\_\_\_\_ 语句也称开关语句。

58、下面是一个 java 应用程序 (Application)，它的功能是在屏幕上输出 26 个小写英文字母，其中每个字母相隔一个制表符，请完成程序填空。

```
public _____ Class1{
public static void main(String args[]){
char c='a';
for (int i=1;i<=26;i++)
{ System.out.print(_ _____); }
}
}
```

59、下面是一个 java 应用程序 (Application)，它的功能是计算  $s=1+2+3+\dots+10$ ，请完成程序填空。

```
public _____ Class1{
public static void main(String args[]){
int s=0;
for (int i=1;i<=10;i++){
s+=_____;
}
System.out.println("s="+s);
}
}
```

60、下面是一个 java Application 的主类的定义，其功能是输出乘法口诀表第一列，请完成程序填空。

```
import java.io.*;
public class MyClass{
public static void main(String[] args){
int j=1;
for(int i=1; _____; i++){
System.out.println(i+"*"+j+"="+_ _____);
}
}
}
```

### 三、判断题

1、在 JAVA 中，流程控制主要是由相应保留字和条件表达式所构成的。

( )

2、Java 支持两种分支语句，分别是 If 语句和 while 语句。( )

- 3、如果程序中的语句是以选择性语句来判断选择下一步操作，此语法结构就是选择结构，也叫条件结构。（ ）
- 4、选择结构是根据表达式和变量的不同状态下选择不同的执行分支进行运行。（ ）
- 5、如果程序中的语句是以重复性语句来做程序的结构称为循环结构，也叫重复结构。（ ）
- 选择结构使程序重复执行某个程序块或语句。（ ）
- 6、正如汉语中的“如果 ，， ，那么就 ，， ”，“如果 ，， 那么就 ，， 否则就 ，， ”，这样的语句我们称之为选择语句，也称为重复语句。（ ）
- 7、分支语句提供了这样一种控制机制，它根据条件使得程序的执行可以跳过某些语句不执行，而转去执行特定的语句。（ ）
- 8、if 语句是选择结构中最基本的语句，它有两种形式、if 及 else~if 语句。（ ）
- 9、语法格式中的“布尔表达式”为关系表达式或逻辑表达式，其值为布尔类型。（ ）
- 10、在 if 语句末可以加分号。（ ）
- 11、在 if 语句中，布尔表达式无需括号括起来。（ ）
- 12、else if 可以有任意多个，而且后一个 else if 能在前面 else if{} 后出现。（ ）
- 13、else 总是与离它最近的 if 想匹配，可以通过使用大括号 { } 来改变 if~else 的配对关系。（ ）
- 14、Java 语言提供 3 种循环语句、while 语句 ,do-while 语句和 if 语句。（ ）
- 15、在双重循环中，在 for 循环中还潜逃了另一个 for 循环，这样就形成了一个双重循环结构。（ ）
- 16、if 语句语法格式中的“布尔表达式”为关系表达式或逻辑表达式，其值为布尔类型（ ）
- 17、continue 语句只用于循环语句中，它起到终止本次循环的作用。（ ）
- 18、结构化程序设计提高了程序的清晰简单性，并且它只需使用三种形式的控制结构就足够了。（ ）
- 19、else 子句可以单独使用，同样 if 子句也可以单独使用（ ）
- 20、default 在 switch 选择结构中是必需的。（ ）
- 21、break 语句在 switch 选择结构是必需的。（ ）
- 22、while 循环中循环体至少执行一次。（ ）
- 23、if 结构和 if/else 结构所做的动作是相同。（ ）
- 24、break 语句只用于循环语句中，它起到终止本次循环的作用。（ ）
- 25、break 语句可以用在循环和 switch 语句中。（ ）
- 26、continue 语句用在循环结构中表示继续执行下一次循环。（ ）
- 四、程序阅读题

1、请答出下列程序的输出结果是：

```
public class Test {
 public static void main(String[] args) {
 int x=2;
 switch(x){
 case 1: System.out.println("case 1");
 case 2: System.out.println("case 2");
 case 3: System.out.println("case 3");break;
 case 4: System.out.println("case 4");
 default: System.out.println("default");
 }
 }
}
```

该程序的参考答案如下：

2、下面程序的输出结果是什么？

```
public static void main(String args[]){
 int a=10;
 int b=20;
 if(a>b)
 System.out.println("Not Equal");
 else
 System.out.println("Equal");
}
```

答：

3、下面程序的输出结果是什么？

```
class Happy{
 public static void mmin(String args[]){
 int i =1;
 int j =10;
 do{
 if(i++<j--)
 continue;
 }
 while(i<5);
 System.out.println(i+ "" +j);}
}
```

结果为：

4、下面代码的输出结果是什么？

```

class foo
{
 public static void main(String args[]){
 int x=4,j=0;
 switch(x){
 case 1:j++;
 case 2:j++;
 case 3:j++;
 case 4:j++;
 case 5:j++;
 default :j++;
 }
 System.out.println(j);}
}

```

结果为：

5、阅读程序，分析程序运行流程。

[Matrix.java]

```

class Matrix {
 public static void main(String[] args){
 for(int i=1; i<=3; i++) {
 for (int j=1; j<=5; j++)
 System.out.print(i*j + "\t"); //内循环体
 System.out.println(); //换行
 }
 }
}

```

本程序的输出结果是：

6、写出下面程序的运行结果，了解 while 语句的使用。

```

public class Useli3 {
 public static void main(String args[]){
 int n = 9;
 System.out.println("\n");
 while(n>6){
 n--;
 System.out.print(n);
 }
 }
}

```

本程序的输出结果是：

7、写出下面程序的运行结果，了解 do-while 语句的使用。

```
public class Useli4 {
 public static void main(String args[]){
 int x = 23;
 do{
 System.out.print(x- -);
 }while(x>20);
 }
}
```

本程序的输出结果是：

8、写出程序的运行结果，了解 for 语句的使用。

```
public class Useli5 {
 public static void main(String args[]){
 int i,sum = 0;
 for (i=1;i<=3;i++)
 sum += i;
 System.out.println("sum = " + sum);
 }
}
```

本程序的输出结果是：

9、下面程序段的输出结果是多少。

```
int i = 0,sum = 1;
do{
 sum += i++;
}while(i<6);
System.out.println("sum = " + sum);
```

本程序的输出结果是：

10、标有 `/**/` 的语句的执行次数是多少。

```
int y = 0,i;
for(i=0;i<20;i++){
 if(i%2==0)
 continue;
 y +=i; /* 考虑本条语句的执行次数 */
}
```

本程序的输出结果是：

11、下面的两个程序段功能相同吗？

```
(1) int i = 1,sum = 0;
 for(;i<=100;i++){
 sum +=i;
 }
```

```
(2) i = 1;sum = 0;
 for(;;){
 sum +=i;
 if (i==100)break;
 i++;
 }
```

本程序的输出结果是：

12、以下程序的输出结果是什么？

```
public class Useli6 {
 public static void main(String args[]){
 int a,b;
 for (a=1,b=1;a<=100;a++){
 if(b>=20) break;
 if(b%3==1){
 b += 3;
 continue;
 }
 b -= 5;
 }
 System.out.println("a = "+a);
 }
}
```

本程序的输出结果是：

13、以下程序的输出结果是什么？

```
public class Useli7 {
 public static void main(String args[]){
 int i;
 for (i=1;i<=5;i++){
 if(i%2==0)
 System.out.println("*");
 else
 continue;
 }
 }
}
```

```

 System.out.println("#");
 }
 System.out.println("$\n");
}
}

```

本程序的输出结果是：

14、执行下面的程序段后， B 的值为多少？

```

int x = 35;
 char z = 'A';
 boolean B;
 B = ((x<15)&&(z<'z'));
 System.out.println(B);

```

本程序的输出结果是：

## 五、编程题

1、求分段函数的值。

$$F(x)=\begin{cases} 0 & (x<0) \\ 2x+1 & (x\geq 0) \end{cases}$$

2、已知函数

$$y=\begin{cases} x+3 & (x>0) \\ 0 & (x=0) \\ x^2-1 & (x<0) \end{cases}$$

编程输入 x 的值，输出 y 的值。

3、用 switch 语句编写程序，给学生写评语，若学生成绩在 60-69 则打印“及格”，70-89“良好”，90-100“优秀”，60 分以下“不及格”。

4、编写程序，输出 1 到 1000 之间所有可以被 3 整除又可以被 7 整除的数。

5、编写程序，输出 1 到 100 之间，所有素数，每输出 5 个换行一次。

6、从键盘输入某个月份（1--12），并输出通过使用 if-else-if 阶梯来确定某个月是什么季节。

7、编写程序，计算邮局汇款的汇费。如果汇款金额小于 100 元，汇费为 1 元；如果汇费在 100 元与 5000 元之间，按 1%收取汇费；如果金额大于 5000 元，汇费为 50 元。汇款金额由命令行输入。

8、编写一个应用程序，由命令行输入年份，判断该年份是平年还是闰年，并打印出相应的结果。

9、由命令行输入一个 4 位整数，求将该数反转以后的数，如原数为 1234，反转以后的数为 4321。

10、编写程序，键盘输入两个数，计算并输出它们的积。

11、编写程序，接受用户输入 1~7 之间的整数，若不符合条件则重

新输入，利用 switch 语句输出星期几。

12、编写一段程序，实现求三个数中的最大数。

13、编写一个程序，从 1 开始，计算需要多少个连续的自然数相加，其和才会超过 1000，并打印出这个数字。

14、请编写一段代码输出如下图形、

```



```

15、请编写一段代码输出如下图形、

```
*

**


```

16、根据考试成绩确定分数等级，当考试成绩大于或等于 90 分，等级为 'A'，当成绩在 80~89 分之间，等级为 'B'；在 70~79 之间，为 'C' 级，60~69 之间为 'D' 级；低于 60 分为 'E'。

17、编写一个程序，找出大于 200 的最小的质数

18、整型变量 a 的值可能是 1、2、3 和 4，如果 a 为 1，则输出“进行加法运算”，a 为 2，则进行减法运算，a 为 3，则进行乘法运算，a 为 4，则进行除法运算。

19、山上有一口缸可以装 50 升水，现在有 15 升水。老和尚叫小和尚下山挑水，每次可以挑 5 升。问：小和尚要挑几次水才可以把水缸挑满？

20、分别用三种循环语句求前 100 个自然数的和。

21、求  $1+2+3+\dots+N < 1000$  的 N 的最大数。

22、编写一个程序段，输出一个三角形形式的九九乘法表。

```
1 × 1=1
2 × 1=2 2 × 2=4
3 × 1=3 3 × 2=6 3 × 3=9
4 × 1=4 4 × 2=8 4 × 3=12 4 × 4=16
5 × 1=5 5 × 2=10 5 × 3=15 5 × 4=20 5 × 5=25
6 × 1=6 6 × 2=12 6 × 3=18 6 × 4=24 6 × 5=30 6 × 6=36
7 × 1=7 7 × 2=14 7 × 3=21 7 × 4=28 7 × 5=35 7 × 6=42 7 × 7=49
8 × 1=8 8 × 2=16 8 × 3=24 8 × 4=32 8 × 5=40 8 × 6=48 8 × 7=56 8 × 8=64
9 × 1=9 9 × 2=18 9 × 3=27 9 × 4=36 9 × 5=45 9 × 6=54 9 × 7=63 9 × 8=72 9 × 9=81
```

23、编写一个程序段，当用户从输入一个参数 MONTH 为月份值，由此确定出这个月的天数 DAY 值，用 switch 语句执行该功能。提示：



不同的月份有不同的天数， 这需要判断后才能确定。 把月份归为三类， 30 天的月份有 4、6、9、11 月份， 28 或 29 天（由年份再确定）的月份是 2 月份，其余皆为 31 天。

24、编写一个程序段，输入一个自然数，判断该数是否为素数。

提示：先确认什么样的数为素数。根据数学定义知，一个自然数除了 1 和本身之外没有其他的因子， 该数便为素数。 2 专门规定为素数， 且是最小的素数。 对于自然数  $n$ ，我们只要用 2 到  $n-1$  去测试，如果它们中没有一个是  $n$  的因子，  $n$  就为素数，否则  $n$  不是素数。

25、编写一个程序段，求所有三位素数。分析：本题是上题的拓展。不是判断某一个数，而是判断所有三位数。这就需要用循环变量遍历三位数，因为所有的三位偶数都非素数，所以可直接遍历三位奇数。对于确定的三位数判断其是否为素数的方法同上。

26、编写一个程序段，输出如下菲波那契数列的前 20 项，并求其和。

1, 1, 2, 3, 5, 8, 13,,

提示：菲波那契数列的前两项均为 1，以后的每一项都是前两项之和。

这是一个重复求和处理的问题，可用循环解决。循环次数是所要求的项数 18。对所有项求和的问题可以设置一个累加器变量 `sum`，初值为前两项之和，后在循环中每求得一新项，即将该项加进 `sum`。

27、有父子二人， 已知父亲年龄不大于 90 岁，儿子年龄不大于 50 岁。10 年前父亲的年龄是儿子的 4 倍，10 年后父亲的年龄是儿子年龄的整数倍。求父子的年龄。

28、有面值为一元、二元、五元的人民币若干，从中取出 20 张使其总值为 60 元，问有多少种取法？每种取法的人民币各为多少张？

29、编写一个程序段，由键盘输入一个大写字母，要求输出其小写字母。

30、编写一个程序段，输入一个 3 位整数，将它反向输出。如输入 123，输出 321。

31、编写一个程序段，把 100 ~ 200 之间的不能被 3 整除的数输出。

32、编写一个程序段，输入两个正整数  $m$  和  $n$ ，求其最大公约数和最小公倍数。

33、设计个人纳税计算程序。 “税法”规定，工资、薪金所得、适用超额累进税率，以每月收入额减去 1200 元后的余额，为相应纳税所得额。

34、编写一个程序段，输入一个自然数，判断该数是否为素数。

### 35、幸运猜猜猜

需求说明：游戏随机给出一个 0~99（包括 0，99）的数字，然后你猜是什么数字。你可以随便猜一个数字，游戏会提示太小还是太大，从而缩小结果范围。经过几次猜测与提示后，最终推出答案。在游戏过程中，记录你最终猜对时所需要的次数，游戏结束后公布结果。

36、编写一个程序段，有 1、2、3、4 个数字，能组成多少个互不相同且无重复数字的三位数？都是多少？  
程序分析：可填在百位、十位、个位的数字都是 1、2、3、4。组成所有的排列后再去掉不满足条件的排列。

37、回文字符串判断。 所谓回文是指正读和反读都一样的数或文本段，如 abcba、1234321 等。编写程序，从键盘输入字符串，由程序判断是否为回文串，并给出提示。

38、编写一个程序段，打印出如下图形

\* \* \*

\* \* \* \* \*

\* \* \* \* \* \*

\* \* \* \* \* \*

\* \* \* \* \* \*

\* \* \* \* \* \*

\* \* \* \* \*

\* \* \* \* \*

\* \* \* \* \*

39、编程实现如下功能、请你从键盘上读入一个整数  $n$ ，然后打印  $n$  行。每行包括  $n$  个字符。每行的首个和最后一个字符是符号 ‘ # ’，中间的是符号 ‘ + ’。举例、如果你从键盘上读入的整数是 5，那么打印结果如下、

```
+ + +
+ + +
+ + +
```

# + + + #  
# + + + #

### 第三章 字符串

#### 一、选择题

- 1、 String str = new String("abc") 此代码中有几个对象 ( )  
A、 1 个                      B、 2 个                      C、 3 个                      D、 4 个
- 2、 下面哪个是对字符串 String 的正确定义 ( )  
A、 String s1= "t null"                      B、 String s2=?null?  
C、 String s3=(String) ,abc?                      D、 String s4=(String) , uface?
- 3、 给出下面一段程序，选择运行结果 ( )  

```
public class X{
 public static void main(String [] args){
 String names[]=new String[5];
 for(int x=0;x<args.length;x++) names[x]=args[x];
 System.out.println(names[2]);
 }
}
```

  
命令行执行、 java X a b 结果是下面哪一个？  
A、 names                      B、 b                      C、 null                      D、 运行出现异常
- 4、 如何将字符串转换为数值？ ( )  
A、 利用字符串对象的 toString 函数将字符串转换为数值  
B、 利用数值类的 parseInt,parseDouble 等函数将字符串直接转换为数值  
C、 利用数值类的静态函数 valueOf 将字符串转换为数值  
D、 利用数值类的静态函数 intValue 将字符串转换为数值
- 5、 阅读下面代码  

```
if(x==0){System.out.println(小明 "");}
else if(x<- 2){System.out.println(小华 " ");}
else{System.out.println(小军 "");}
```

  
若要求打印字符串 “小军 ”，则变量 x 的取值范围是 ( )。  
A、 x=0&x<=-2                      B、 x>0                      C、 x>= -2 && x!=0                      D、  
x<= -2 && x!=0
- 6、 下面那个选项是正确的包含 50 个字符串数组的声明？ ( )  
A、 String obj[50];                      B、 String[] obj;                      C、 Object obj[50];                      D、  
Char obj[50];
- 7、 用于存放创建后则不变的字符串常量是 ( )  
A、 String 类                      B、 StringBuffer 类                      C、 Character 类                      D、 以上都不对
- 8、 要为程序的子窗口设定一个由字符串 s 指定的名字，可以采用的代码是 ( )  
A、 JFrame(s)                      B、 window(s)                      C、 new JFrame(s)                      D、  
super(s)

- 9、Math 类的 ( ) 方法可用于计算所传递参数的平方根  
A、 squareRoot                      B、 root                      C 、 sqrt                      D 、 square
- 10、Math. ( ) 方法可返对某数求幂后的结果  
A、 power                      B、 exponent                      C、 pow                      D 、 exp
- 11、( ) 方法可返回一个 Integer 对象的 int 数据  
A、 intData                      B、 getValue                      C、 getData                      D、 intValue
- 12、( ) 中包含了一个 Random 类  
A、 java.awt                      B、 java.util                      C、 java.swing                      D、 java.util
- 13、Random 对象能够生成 ( ) 类型的随机数  
A、 int                      B、 string                      C、 double                      D 、 A 和 C
- 14、import 语句使应用程序能够从 Java 类库中访问到 ( )  
A、 包和类                      B 、 类和对象                      C、 对象和方法                      D、 方法和变量
- 15、Random 类的 nextInt 方法能够 ( )  
A 、 接受一个参数                      B、 不带参数                      C、 接收两个参数  
D、 A 和 B
- 16、语句 ( ) 是将一个 5 到 20 之间的随机数赋值给变量 value  
A 、 value=4+randomObject.nextInt(16)                      B 、 value=randomObject.nextInt(21)  
C 、 value=5+randomObject.nextInt(15)                      D 、 value=5+randomObject.nextInt(16)
- 17、Java.util 包中的 Random 类 ( )  
A、 能够产生正整数                      B、 能够产生正的双精度数  
C、 具有产生随机数的能力                      D、 以上答案都对
- 18、表达式 example.substring ( 3 , 4 ) 返回的子都为 ( )  
A、 起始于位置 3 的 4 个连续字符                      B、 起始于位置 3 终止于位置 4 之前的字符  
C、 位置 3 和位置 4 上的字符                      D、 位置 3 上的字符，并重复次
- 19、代码片段、 sample ( ) 可返回 String 型 sample 的长度  
A 、 .getLength                      B、 .getLength()                      C 、 .length  
D、 .length()
- 20、String 类的 ( ) 方法将返回该字符串的字符个数  
A 、 maxChars                      B、 length                      C、 characterCount                      D、 size

- 21、字符串是由 ( ) 组成  
A、 数字                      B 、 小写字母                      C、 特殊字符                      D、 以上答案都对
- 22、通过调用 ( ) 方法可在某个字符串中定位出一个字符串首次出现的位置  
A 、 indexOf              B、    firstIndexOf              C、    findFirst              D、    locate
- 23、在一个接收 String 和 int 作为其参数的 lastIndexOf 方法中，传递至 lastIndexOf 方法中的第二个参数表示 ( )  
A 、 所要查找的字符个数  
B 、 采用向前查找法开始进行查找时的起始位置处的索引  
C 、 需要定位的子字符串的长度  
D、 采用向后查找法开始进行查找时的起始位置处的索引
- 24、substring 方法 ( )  
A 、 可接收一个或两个参数  
B 、 返回一个新的 string 对象  
C、 通过复制现有 string 对象中的一部分从而创建一个新的 string 对象  
D、 以上答案都对
- 25、当调用带两个参数的 substring 方法时，第二个参数表示 ( )  
A 、 通过此处位置上的索引向后进行复制  
B 、 需要复制的字符串的长度  
C 、 超过最后一个所要复制的字符的下一个字符的索引  
D 、 所要复制的最后一个字符的索引
- 26、( ) 方法可删除某个字符串中的所有空格  
A、 removeSpaces                      B、 squeeze                      C 、 trim                      D、 truncate
- 27、如果某个字符串起始于参数字符串，那么调用 startsWith 方法时将返回 ( )  
A 、 true                      B、 false                      C、 1                      D、 子字符串的索引
- 28、假若 indexOf 方法未能找到所指定的子字符串，则返回 ( )  
A、 false                      B、 0                      C、 - 1                      D、 以上答案都不对
- 29、以下是关于字符串运算的代码，执行该代码段的输出结果是 ( )  

```
String s= "0123456789" , s1 , s2 ;
s1=s.substring(2) ;s2=s.substring(2 ,5) ;System.out.println(s1+s2) ;
```

  
A、 0122345      B、 234567892345      C、 012234      D、 23456789234
- 30、string 类允许开发人员 ( )

## B、从字符串中检索字符

D、以上答案都正确

A、 get                      B、 char                      C、 getAt                      D、 charAt

A 、 stringCopy      B、      substring      C、      copyString      D、  
copySubString

A 、 都拥有相同的长度                      B 、 总是彼此相等

34、作为接受两个参数的 indexOf 方法，将不会考虑任何出现在（ ）之前的所有字符

C、该字符串中最后一个字符                      D、以上答案都不对

A、 checkEnd      B、 stringEnd      C、 endsWith      D、 ends

A、 内部  
B 和 C

A、 abs      B、 absolute      C 、 positive      D、 positiveValue

A、 Calc      B、 Math      C、 Calculation      D、 Number

A    7+randomObject.nextInt(293)  
8+randomObject.nextInt(292)

40、语句 ( ) 返回的整数范围为 15 到 35

A 、 10+randomObject.nextInt(26)  
15+randomObject.nextInt(21)

C 、 10+randomObject.nextInt(25)  
15+randomObject.nextInt(35)

41、下面哪些语句能够正确地生成 5 个空字符串？ ( )

A、 `String []a=new String[5]; for( int i=0;i<5;a[i++]=null);`

B、String a[]={"","","","",""};

C、String a[5];

D、String[5]a;

42、 下列定义 String 变量 s 的语句中，错误的是 ( )

- A、 String s=new "abc@" ;                      B、 String s=new String( "abc@" );  
C、 String s;                                      D、 String s= "abc@" ;

43、 需要分析字符串，并将字符串分解成可被独立使用的单词，可使用 java.util 包中的类是 ( )

- A、 String              B、 StringBuffer              C、 StringTokenizer              D、 StringTokenizer

44、 已知 String 对象 s="abcdefg"，则 s.substring(2, 5)的返回值为 ( )

- A、 "bcde"              B、 "cde"              C、 "cdef"              D、 "def"

45、 请看下面的代码

```
String s = " people " ;
String t = " people " ;
String c[] = {,p?,?e?,?o?,?p?,?l? ,?e?};
```

下面哪一选项的语句返回值为真： ( )

- A、 s.equals(t);              B、 t.equals(c);              C、 s==t;              D、 t==c;

46、 已知 s 为一个 String 对象，s="abcdefg"，则 s.charAt(1)的返回值为 ( )

- A、 a              B、 b              C、 f              D、 g

47、 请看下面的代码

```
String s = " good " ;
```

下面选项语句书写正确的是： ( )

- A、 s= " student " ;              B、 char c = s[1];  
C、 int len = s.length;              D、 String t = s. LowerCase( );

48、 正确创建一个含有 5 个元素的字符串数组的选项是： ( )

- A、 String s[ ] = String[5];              B、 String s[ ]={ " " , " " , " " , " " , " " };  
for(int i=0;i<5;s[i++]= " " );  
C、 String s[5];              D、 String [5]s;

49、 请看下面的代码

```
public class class1{
String str = new String("Girl");
char ch[]={'d','b','c'};
public static void main(String args[]){
class1 x=new class1();
x.change(x.str,x.ch);
System.out.println(x.str+" and "+x.ch[0]+x.ch[1]+x.ch[2]);
}
public void change(String str,char ch[]){
str="Boy";
```



```
ch[0]='a';
}
```

} 该程序的输出结果是： ( )

A、 Boy and dbc      B、 Boy and abc      C、 Girl and dbc      D、 Girl and abc

50、 下面的程序段输出的结果是 ( )。

```
StringBuffer buf1=new StringBuffer(20);
```

```
System.out.println(buf1.length()+ " , " +buf1.capacity());
```

A 、 0,20              B 、 0,null              C、 20,20              D、 0,0

## 二、 填空题

1、 字符串分为两大类，一类是字符串常量，使用 \_\_\_\_\_ 类的对象表示；另一类是字符串变量，使用类 \_\_\_\_\_ 的对象表示。

2、 .对于字符串 String s1=new String("ok");String s2=new String("ok");  
表达式 s1==s2 的值是 \_\_\_\_\_ , s1.equals(s2)的值是 \_\_\_\_\_

3、 对于字符串 String s1="ok";String s2="ok"; 表达式 s1==s2 的值是 \_\_\_\_\_,s1.equals(s2)的值是 \_\_\_\_\_

4、 Math.min(-25,-9)=\_\_\_\_\_ ; Math.sqrt(16)=\_\_\_\_\_ ;

5、 创建字符串对象 a 的两种方式 \_\_\_\_\_ 和 \_\_\_\_\_ 。

6、 字符串的两大类 \_\_\_\_\_ 和 \_\_\_\_\_ 。

7 、 设 String 对象 s= “ Hello ” , 运行语句  
System.out.println(s.concat( “ World ” )) ;  
“ World ” 对象 s 的内容为 \_\_\_\_\_  
\_\_\_\_\_ , 所以语句输出为 \_\_\_\_\_。

1、 定义一个整型数组 y , 它有 5 个元素分别是 1 , 2 , 3 , 4 , 5。 用  
一个语句实现对数组 y  
的声明、创建和赋值： \_\_\_\_\_。

2、 设有整型数组的定义： int x [][]={{12,34},{-5},{3,2,6}}; , 则 x.length  
的值为 \_\_\_\_\_。

3、 求取二维数组 a [][] 第 i 行元素个数的语句是 \_\_\_\_\_。

4、 若有定义 int[ ] a=new int[8]; 则 a 的数组元素中第 8 个元素的下标是 \_\_\_\_\_ 。

5、 已知 String 对象 s=“ hello ” , 运行语句 System.out.println(s.concat  
(“ World! ” )); 后 s 的值为 \_\_\_\_\_。

6、 使用 += 将字符串 s2 添加到字符串 s1 后的语句是 \_\_\_\_\_。

7、 比较 s1 中的字符串和 s2 中的字符串的内容是否相等的语句是 \_\_\_\_\_。

8、 已知 sb 为 StringBuffer 的一个实例 , 且 sb= "abcde " , 则 sb  
reverse()后 sb 的值为 \_\_\_\_\_。

9、 已知 sb 为 StringBuffer 的一个实例 , 且 sb= "abcde " , 则 sb  
delete(1,2)后 sb 的值为 \_\_\_\_\_。

10、 用 for 循环求一维整型数组 a 数组的所有元素之和的语句段是 \_\_\_\_\_

\_\_\_\_\_。

11、下面程序的功能为计算数组各元素的和，完成程序填空。

```
import java.awt.Graphics;
import java.applet.Applet;
public class SumArray extends Applet {
 int a[] = { 1, 3, 5, 7, 9, 10 };
 int total;
 public void init() {
 total = 0;
 for (int i = 0; _____; i++)
 total+=_____ ;
 }
 public void paint(Graphics g) {
 g.drawString("Total of array elements: " +total, 25, 25);
 }
}
```

12、下面程序的功能为计算数组下标为奇数的各元素的和\_\_\_\_\_，完成程序填空。

```
import java.awt.Graphics;
import java.applet.Applet;
public class SumOfArray _____ Applet {
 public void paint(Graphics g) {
 int a[] = { 1, 3, 5, 7, 9, 10 };
 int total=0;
 for (int i = 1; i < A.length; _____)
 total+=a[i];
 g.drawString("Total of array elements: " + total, 25, 25);
 }
}
```

### 三、判断题

1、类 String 对象和类 StringBuffer 对象都是字符串变量，建立后都可以修改（ ）

2、字符串中的索引从 0 开始（ ）

3、连接字符串，当前字符串本身不改变（ ）

4、用 “+” 可以实现字符串的拼接，用 “-” 可以从一个字符串中去除一个字符串。（ ）

5、Java 中的 String 类的对象既可以是字符串常量，也可以是字符串变量。（ ）

6、用运算符 “==” 比较字符串对象时，只要两个字符串包含的是同一个值，结果便为 true。

( )

7、String 类字符串在创建后可以被修改。( )

8、方法 replace (String srt1, String srt2) 将当前字符串中所有 srt1 子串换成 srt2 子串。( )

9、方法 compareTo 在所比较的字符串相等时返回 0。( )

10、方法 IndexOf((char ch , -1)返回字符 ch 在字符串中最后一次出现的位置。( )

11、方法 startsWith 判断当前字符串的前缀是否和指定的字符串一致。( )

#### 四、编程题

1、怎么样获得字符串的第一个字符？怎么样获得字符串的最后一个字符？？怎么样删除第一个字符？？怎么样删除最后一个字符？？？

2、编写一个程序，给定一个字符串 “aa果4金5a”。要求：如果出现数字，把数字后面的字符替换成数字的个数个。比如这个字符串要求被替换成 “aa果金金金金aaaaa”。

3、编写一个程序，比较输入的两个字符串的长度大小。

4、编写一个程序，判断两个字符串是否相同：s1= " you are a student, " s2= " how are you; "

5、编写一个程序，判断字符串 " 22030219851022024的前缀、后缀是否和某个字符串 " 220302相同。

6、编写一个程序，按字典顺序比较两个字符串 '你'和'我'的大小关系。

7、将数字型字符串 " 100和 " 123.678转换为数字；

8、将字符串 " FEDCBA' 存放到数组中，调用 for 循环读出数组数据显示在屏幕上。

9、从键盘输入两个整数，在程序中计算这两个整数的和，并输出。

10、判断回文字符串。回文是一种 “从前向后读”和“从后向前读”都相同的字符串。如 “ rotor是”一个回文字符串。要求利用命令行参数输入一个字符串，判断是否为回文。

11、编写应用程序，把字符串 “ ABCDEFHIJKLM 逆序输出。

12、编写应用程序，实现从命令行参数中输入一字符串，统计该字符串中字符 ,e?出现的次数。

13、编写应用程序，实现从命令行参数中输入一字符串，屏蔽掉字符 ,s'后打印出来。

14、编写应用程序，实现从命令行参数中提取出所有的数字，然后将这些数字按出现的顺序转换成一个整数。

15、编写一个应用程序，设有字符串 "yekmaakkccekymbvb", 求出该字符串中有多少种字符 ,以及每个字符的个数 ？

16、编写一个截取字符串的方法，输入为一个字符串和字节数，输出为按字节数截取的字符串。但是要保证汉字不被截半个，如 "我ABC"4，应该截为 "我 AB"，输入 "我 ABC 汉 DEF"6，应该输出为 "我

ABC" 而不是 "我 ABC+ 汉的半个 "。

17、编写一个应用程序，对字符串 sbc, fds, des 按字母的升序输出。

18、编写自定义方法生成 k 个 50~100 之间的随机整数，再另写一个输出方法。在应用程序的 main ( ) 方法中调用这两个方法，生成一个整型数组并输出该数组的所有元素。

19、编写一个 Application 程序，比较命令行中给出的两个字符串是否相等，并输出比较的结果。

20、请编写一个 Application 实现如下功能：接受命令行中给出的一个字母串，先将该串原样输出，然后判断该串的第一个字母是否为大写，若是大写则统计该串中大写字母的个数，并将所有大写字母输出；否则输出信息串 " 第一个字母不是大写字母 ! "。

21、一个应用程序，接受用户输入的一行字符串，统计字符个数，然后反序输出。

22、给你一个字符串数组 String []strings=new

String{"tom","is","a","boy"};

然后让你输入一句话，如果你的输入和他给的一样的话，就原文不动的输出。如你输入 tom is a boy 就输出 tom is a boy。但是如果你输入 tom is an open，输出的时候要把一样的写下来，不一样的找有没有对应的字母一样的，如果有就在那个一样的字母后加一个 "?"

## 第四章 数组

### 一、选择题

- 1、int [] s=new int[3] ; s 数组中元素最大的下标值为 ( ) ;  
A、 0      B、 1      C、 2      D、 3
- 2、int a[]={45, 4, 67, 23, 657, 787, 34, 52, 56} ;数组中 a[5] 的值为 ( ) ;  
A、 23      B、 45      C、 657      D、 787
- 3、int a[]={45, 4, 67, 23, 657, 787, 34, 52, 56} 数组中 , a.length 的值为 ( ) ;  
A、 9      B、 8      C、 7      D、 10
- 4、int a[]=\_\_\_\_ int [4]; 空格中正确的填空为 ( ) ;  
A、 new      B、 try      C、 this      D、 throws
- 5、int a[]={13,45}, a=int b[], 其中 b.length 的值为 ( ) ;  
A、 3      B、 4      C、 2      D、 1
- 6、数组的下标最大为 ( ) ;  
A、 等于数组的长度      B、 等于数组的长度减一      C、 是随意定的  
D、 与长度无关
- 7、定义一个数组 , 但未赋初值 , 系统会自动赋值 , 其值为 ( ) ;  
A、 0 或 null      B、 1      C、 任何值      D、 9
- 8、int a=new int [\_\_\_\_];a[0]=1;a[1]=4;a[2]=2; 空格中最小能够填的数据为 ( )  
A、 2      B、 4      C、 3      D、 5
- 9、阅读后面的程序 : 该程序运行的结果为 ( )  
int a[]={13,45,67};  
int b[]=a;  
b[1]=23;  
for(int i=0;i<b.length;i++)  
System.out.print(b[i]);  
A、 0 0 0      B、 13 23 67      C、 13 45 67      D、 0 23 0
- 10、int a []={34,56,67,78,36}; \_\_\_\_=56; 请在下面选择空格处正确的填写为 ( )  
A、 a[0]      B、 a[1]      C、 a[2]      D、 a[4]
- 11、声明一个数组 , \_\_\_\_a[]=new String[3]; 空格处应该填写的内容为 ( )  
A、 int      B、 double      C、 String      D、 string
- 12、阅读本列代码 : int a[]={12,34, \_\_\_\_,34,56}; 下面的描述错误的是 ( )  
A、 a[1]=34      B、 a[0]=12      C、 a[2]=      D、 a[3]=null
- 13、阅读本列代码 : String a=new String[2], 下面表达式正确的是

- (            );
- A、 a[0]=null            B、 a[1]= " "            C、 a[0]=0            D、 a[2]=null
- 14、阅读本列代码： String[] a=new String[2] ;a.length 的值为 (            );
- A 、 1            B、 2            C、 3            D、 4
- 15、数组用什么方法进行比较 (            )
- A、 equals()            B、 length()            C 、 length            D、 没法比较
- 16、用来访问数组中某特定元素的数据被称为 (            )
- A 、 值            B、 大小            C、 带索引的数组名            D、 索引
- 17、 (            )\_表示一维数组 units 中的第三个元素。
- A、 units[2]            B、 units(2)            C、 units[0,2]            D、 units{2}
- 18、通过使用关键字 (            ) 可以创建一个数组
- A 、 declare            B、 create            C、 new            D、 make
- 19、数组元素之所以相关，是因为它们具有相同的 (            )
- A、 名字            B、 参数            C、 下标            D、 地址
- 20、数组的长度等于 (            )
- A 、 数组最后一个索引加 1            B、 数组最后一个索引减 1
- C、 数组最后一个索引            D、 以上答案都不对
- 21、将数组中的元素按照某种特定顺序来排列的过程被称为数组的 (            )
- A、 创建            B、 排序            C、 声明            D、 初始化
- 22、数组可以被声明为 (            ) 类型
- A、 double            B、 int            C 、 string            D、 任意
- 23、数组中的元素之所以是相互有联系的是，因为他们具有相同的名称和 (            )
- A、 常量值            B 、 下标            C、 类型            D、 值
- 24、表达式 (            ) 可以返回数组 arrayName 的最大索引。
- A 、 arrayName.getUpperBound            B 、 arrayName.getUpperLimit
- C 、 arrayName.length            D 、 arrayName.length-1
- 25、设有定义语句 int a[ ]={66,88,99}; 则以下对此语句的叙述错误的是 (            )
- A、定义了一个名为 a 的一维数组            B、a 数组有 3 个元素
- C、a 数组的元素的下标为 1 ~ 3            D、数组中的每个元素都是整型
- 26、数组 (            )
- A 、 是组件            B、 总是一维的            C、 任何时候都处于已排序的状态
- D、 是对象
- 27、使用两个索引的数组被称为 (            )
- A 、 单下标            B 、 二维            C、 正方            D 、

一维

28、能正确创建一个 2 行 5 列的 int 型数组的语句是 ( )

A、 new integer[2][5];

B、 new integer[5][2];

C、 new int[2][5];

D、 new int[1][4];

29、在一个  $m \times n$  的数组中，m 代表 ( )

A、 该数组的列数

B、 数组元素的总个数

C、 该数组的行数

D、 每一行元素的数目

30、语句 ( ) 将把一个 5 行 3 列的数组赋值给一个 int 型二维数组变量 myArray

A、 myArray=new int[5][3]

B、 myArray=new int[4][2]

C、 myArray=new int[3][5]

D、 myArray=new int[2][4]

31、二维数组通常用来表示 ( )

A、 饼分图

B、 位距

C、 线条

D、 表

32、数组的一个局限性是 ( )

A、 数组大小不能动态的发生改变  
类型值

B、 它们只能存储基本

C、 不能存放字符串

D、 以上答案都对

33、下面哪个语句正确地声明一个整型的二维数组？ ( )

A、 int a[][] = new int[][];

B、 int a[10][10] = new int[][];

C、 int a[][] = new int[10][10];

D、 int []a[] = new int[][10];

34、若已定义：int a[]={0,1,2,3,4,5,6,7,8,9}; 则对 a 数组元素正确的引用是 ( )

A、 a[-1]

B、 a[9]

C、 a[10]

D、 a(0)

35、在下述 Java 语言语句中，错误的创建数组的方法是 ( )

A、 int intArray [ ] ; intArray=new int[5] ;

B、 int intArray [ ]=new

int[5] ;

C、 int [ ] intArray = { 1,2,3,4,5} ;

D、 int intArray

[5]={ 1,2,3,4,5};

36、为了定义三个整型数组 a1 .a2 .a3,下面声明正确的语句是 ( )

A、 intArray [ ] a1,a2; int a3[ ]={1,2,3,4,5};

B、 int [ ] a1,a2; int a3[ ]={1,2,3,4,5};

C、 int a1,a2[ ]; int a3={1,2,3,4,5};

D、 int [ ] a1,a2; int a3=(1,2,3,4,5);

37、在创建数组时，下列创建的方法中合法的是 ( )

A、 int a[ ][ ]=new a[10][5];

B、 int a[]=new int[10];

C、 int [ ]a=new a[10];

D、 int a[]=new a[10];

38、下面哪条语句定义了 5 个元素的数组，其中正确的是 ( )

A、 int[] a={22,23,24,25,12};

B、 int a[]=new int(5);

C、 int[5] array;

D、 int[] arr;

39、执行完以下代码 int [ ] x = new int[25] ; 后，以下哪项说明是正

确的 ( )

A、x[24] 为 0                      B、x[24] 未定义                      C、x[25] 为 0                      D、x[0] 为空

40、设有定义 `int[] a=new int[4];` ,a 的所有数组元素是 ( )

A、 a0, a1, a2, a3                      B 、 a[0], a[1], a[2], a[3]  
C、 a[1], a[2], a[2], a[4]                      D、 a[0], a[1], a[2], a[3], a[4]

41、已知：`int[] a = new int[88];` 在下列给出的数组元素中，非法的是 ( )

A、 a[0]                      B、 a[1]                      C、 a[88]                      D、 a[87]

42、下面哪个选项正确地声明了一个字符串数组： ( )

A、 `char str[]`                      B、 `char str[][]`                      C、 `String str[]`                      D、 `String str[10]`

43、请看下面的程序段：

```
public class class1{
 public static void main(String a[]) {
 int x [] = new int[8];
 System.out.println(x [1]);
 }
}
```

当编译和执行这段代码时会出现： ( )

A、有一个编译错误为 “ possible reference before assignment ”  
B、有一个编译错误为 “ illegal array declaration syntax ”  
C、有异常出现为 “ Null Pointer Exception ”  
D、正确执行并且输出 0

## 二、判断题

- 1、数组的下标比它的长度少 1 ( ) ;
- 2、多维数组不是由简单的一维数组组成 ( ) ;
- 3、用 `length ( )` 方法求数组的长度 ( ) ;
- 4、数组的长度一旦定义了，就不能再改了 ( ) ;
- 5、数组作参数时，数组名后必须加方括号。 ( )
- 6、如果数组大小超过了原来的范围，那么他的范围可以自动的增长到所需的范围 ( ) ;
- 7、数组是具有相同数据类型的一系列数据元素的集合 ( ) ;
- 8、定义一个长度为 4 整型数组：`int a[]=new int [4];`( )
- 9、`int[] a={12,23,45,1,8,9,34}` 数组的长度是 8 ( ) ;
- 10、定义一个 `String` 数组 `String[] C={ “ 123 ” , “ 678 ” , “ 你好 ” , “ ” , “ 67 ” , “ Java ” }` ( ) ;
- 11、数组 `String [] C={"123" , "678" , "你好" , " " , "67" , "Java"}` 的长度是 6 ( ) ;
- 12、`int[][] a=new int [2 ][3]` 里的 2 表示 2 行 ,3 表示每行有三列 ( ) ;



- 13、int[][] a=new int [2 ][3] 里的 2 表示 2 维数组，3 表示每行有三列 ( )；
- 14、int[][] a=new int [2 ][3] 是 2 维数组 ( )；
- 15、int[] a=new int [2 ] 的长度为 3( )；
- 16、int[]a=new int [2 ] 的最大下标数是 1 ( )；
- 17、可以用 length ( ) 方法求字符串，所以也可以用它来求数组的长度 ( )；
- 18、判断 int a[]=new int[3]; a[0]=12; a[1]=0; a[2]=" 你好 "; 的对错 ( )；
- 19、判断数组 String [] a={123,345,adfg," 你好 "} 的对错 ( )；
- 20、数组可以跟数组比大小 ( )；
- 21: 数组可以跟数组比长度 ( )；
- 22、数组的长度定义后还可以改变 ( )；
- 23、在 int 型的数组里可以放字符型的值 ( )；
- 24、如果一个字符串未给他付初值，那么程序内部将自动赋值为 0 ( )；
- 25、定义一个 数组可以 不给他付初值，但一定要标明他的长度 ( )；
- 26、String a[] 的正确书写方式是 String a[]={1232,4354,"234" , " java " }( )；
- 27、表示一个长度为 6 的 int 型的一维数组，int a[]=new int [6];( )
- 28、一维数组是多维数组的基础，即多维数组是由 N 给一维数组组成 ( )；
- 29、要打印一个数组时，一定要人为的给他先付初值 ( )；
- 30、程序内部可以给基本数据类型的数组自动的付初值为 0 ( )；
- 31、程序内部可以给 String 数组自动的付初值为 0 ( )；
- 32、在 Java 里只有 int，String 二种类型的数组 ( )；
- 33、数组的下标等于他的长度 ( )；
- 34、数组属于基本数据类型 ( )；
- 35、数组属于类 ( )；
- 36、数组属于复合类型 ( )；
- 37、数组可以用关系运算符来比较大小 ( )；
- 38、可以用“=”来给数组赋值 ( )；
- 39、数组的长度大小规定后就不能在改变了 ( )
- 40、数组的大小可以自动增长 ( )；
- 41、数组与数组的内容可以用 length ( ) 进行比较 ( )；
- 42、在 Java 语言中，数组的定义形式为、数组元素类型 数组名 [], 或 数组元素类型 [] 数组名 [];( )

- 43、在 Java 中数组元素是通过访问下标来实现的 ( ) ;
- 44、java 有复合类型，复合类型也叫参考数据类型 ( )
- 45、下标是用于指出数组中某个元素位置的数字或变量。 ( )
- 46、同一个数组中可以存放多个不同类型的数据。 ( )
- 47、数组的下标可以是 int 型或 float 型。 ( )
- 48、数组可以声明为任何数据类型。 ( )
- 49、执行语句 int a[] 后数组元素的值为 0。 ( )
- 50、假定整形数组的某个元素被传递给一个方法并被该方法修改。当调用方法执行完毕时，这个元素中含有修改过的数值。 ( )
- 51、执行语句 int a[] = new int[50] 后数组元素的值为 0。 ( )
- 52、对于二维数组 s 来说，s[2].length 给出数组 s 第 2 行的元素个数。 ( )
- 53、定义完数组后，我们可以根据实际需要，再动态调整数组大小 ( )
- 54、数组、类和接口都是引用数据类型 ( )
- 55、int x=9; if(x>8 and x<10) System.out.println("true"); 以上语句运行的结果是显示 true ( )
- 56、String str; System.out.println(str.length()); 以上语句运行的结果是显示 0 ( )
- 57、float x=26f; int y=26; int z=x/y; 以上语句能正常编译和运行 ( )
- 58、if("Hunan".indexOf('n')==2) System.out.println("true"); 以上语句运行的结果是显示 true ( )
- 59、int[] a={1,2,3,4}; System.out.println(a.length()); 以上语句运行的结果是显示 4 ( )

### 三、填空题

- 1、int []a={12,45}; a[0]=12; a[1]=\_\_\_\_\_;
- 2、int []a=new int [\_\_\_\_\_]; a[0]=12; a[1]=23; a[2]=45;
- 3、String a={"12","34"," 顺水","23"} 的长度是 \_\_\_\_\_;
- 4、int a[][]=\_\_\_\_\_[2][2];
- 5、int a[]={12,23,45,\_\_\_\_\_,34,67}; a[3]=56;
- 6、String C={"123","23","a 你不好 ", " ", "他人同意 "} 的长度是 \_\_\_\_\_;
- 7、String a[2][2]={{"123","678"}, {" 你好 ", "书 Java"}}  
String[1][1]=\_\_\_\_\_;
- 8、数轴 String[23][3] 长度是 \_\_\_\_\_;
- 9、String[][][] 是由 \_\_\_\_\_个一维数组组成的 ;
- 10、数轴的下标是从 \_\_\_\_\_开始的 ;
- 11、阅读下列程序，把程序补充完整。  
public class Test{

```

 public static void main(String []args){
 int a[]=new int [5];
 for(int b=a.length-1;b>=0;b--){
 _____=b;
 System.out.println("a["+b+"]"+"="+b);
 }
 }
 }
}

```

12、阅读下列程序，把程序补充完整。

```

public class Test{
 public void man(){
 int a[]=new int[6];
 for(int b=_____;b>=0;b--){
 a[b]=_ _ _;
 System.out.println("a["+b+"]"+"="+b);
 }
 }
 public static void main(String []args){
 Test t= _____ Test();
 t.man();
 }
 System.out.println(t.man());
 }
}

```

13、 String b={"23","90","try","catch","finaly"},  
String b 的最大下标是 \_ \_ \_;

14、 String b={"23","90","try","catch","finaly"},  
b[4]=\_ \_ \_ \_ \_;

15、 String b={"23","90","try","catch","finaly"},  
b.length=\_\_\_\_\_;

16、 int a=new int [4]; 计算机自动的为它赋的初值是  
a[0]=\_ \_ \_; a[2]=\_ \_ \_;

17、 String a=new String[3]; 计算机自动的为它赋的初值是  
a[0]=\_ \_ \_; a[1]=\_ \_ \_;

#### 四、编程题

1、编写一个程序，求出数组中 20,10,50,40,30,70,95,80,90,100 的最小值。并查找该数组中是否有 95 这个数，如果有，打印出它的位置；如果没有，请给出提示。

2、编写一个程序，随机抽取 100 以内的 10 个数赋值给一个数组。求这个整数数组的最大值、最小值、平均值和所有数组元素的和。

- 3、编写一个长度为 10 的一维数组，并付初值；
- 4、编写一个二维数组，并付初值；
- 5、编写一个用 for 循环的一维数轴，并打印；
- 6、随机抽取 100 以内的 10 个数，把这个些数用数组表示出来，并打印。
- 8、 是一个计算天数的例子。已知日期是某年某月某日，计算这一天是这一年的第几天， 预先将一年前 11 个月的天数存放在 Montn-day 中，先将指定月份之前的各个月份天数累加，在添加上当月的天数，就得到这年的天数 total。在定义数轴的同时付初值，由于数组下表从 0 开始，第一个数可以是任意值， 下标为 1 的数组元素值为 1 月份的天数，当闰年是 2 月份的天数为 29 天。
- 9、编写一个程序，用选择法对数组 a[]={20,10,50,40,30,70,60,80,90,100} 进行从大到小的排序。
- 10、编写一个方法 halfArray()，要求该方法有一个元素类型为 int 的数组参数，方法返回一个新数组， 新数组的长度与参数数组的长度相同，而新数组元素的值是参数数组各元素的一半。
- 11、编写一个程序段，输出如下菲波那契数列的前 20 项，并求其和。  
1, 1, 2, 3, 5, 8, 13,,  
提示：菲波那契数列的前两项均为 1，以后的每一项都是前两项之和。这是一个重复求和处理的问题，可用循环解决。循环次数是所要求的项数 18。对所有项求和的问题可以设置一个累加器变量 sum，初值为前两项之和，后在循环中每求得一新项，即将该项加进 sum。
- 12、编写一个程序，把随机抽取 100 以内 10 个数构成的数组的顺序按相反的顺序打印出来。 例如数组 intArray{1,2,3,4,5,6,7,8,9,11} ；输出结果为 intArray{11,9,8,7,6,5,4,3,2,1} 。
- 13、选佳丽游戏  
舞台上 5 为佳丽，姓名分别为刘亦菲、佐藤爱、范冰冰、巩俐和宋祖英。现有 50 名评委参加投票选举，每人限投一票，不能弃权。得到票数最多者将被评委 2012 年年度大美女。 请编写一个程序，完成选佳丽投票系统。

## 第五章 类和对象

### 一、选择题

- 1、下面对方法的作用描述不正确的是：（ ）  
A、 使程序结构清晰                      B、 功能复用                      C、 代码简洁  
D、 重复代码
- 2、方法内定义的变量：（ ）  
A、 一定在方法内所有位置可见                      B、 可能在方法内的局部位置可见  
C、 在方法外可以使用                      D、 在方法外可见
- 3、方法的形参：（ ）  
A、 可以没有                      B、 至少有一个                      C、 必须定义多个形参  
D、 只能是简单变量
- 4、方法的调用：（ ）  
A、 必须是一条完整的语句                      B、 只能是一个表达式  
C、 可能是语句，也可能是表达式                      D、 必须提供实际参数
- 5、return 语句：（ ）  
A、 不能用来返回对象                      B、 只可以返回数值  
C、 方法都必须含有 return 语句                      D、 一个方法中可以有多 return 语句
- 6、void 的含义：（ ）  
A、 方法体为空                      B、 方法体没有意义  
C、 定义方法时必须使用                      D、 方法没有返回值
- 7、main() 方法的返回类型是：（ ）  
A、 boolean                      B、 int                      C、 void                      D、 static
- 8、方法重载所涉及的方法：（ ）  
A、 有相同的名字                      B、 参数个数必须不同                      C、 参数类型必须不同  
D、 返回类型必须不同
- 9、下面关于类和对象之间关系的描述，正确的是（ ）  
A、 联接关系                      B、 包含关系                      C、 具体与抽象的关系                      D、 类是对象的具体化
- 10、下面关于 java 中类的说法哪个是不正确的（ ）  
A、 类体中只能有变量定义和成员方法的定义，不能有其他语句。  
B、 构造方法是类中的特殊方法。  
C、 类一定要声明为 public 的，才可以执行。  
D、 一个 java 文件中可以有多个 class 定义。
- 11、下列哪个类声明是正确的（ ）  
A、 public void H1 {, }                      B、 public class Move() {, }  
C、 public class void number{}                      D、 public class Car {, }
- 12、下面的方法声明中，哪个是正确的（ ）

A、 public class methodName(){  
methodName(){}  
}

B、 public void int

C、 public void methodName(){  
methodName{}  
}

D、 public void

13、下述哪些说法是不正确的？ ( )

A、 实例变量是类的成员变量  
关键字声明的

B、 实例变量是用 static

C、 方法变量在方法执行时创建  
须初始化

D、 方法变量在使用之前必

14、下面对构造方法的描述不正确是 ( )。

A、 系统提供默认的构造方法  
也可以有返回值

B、 构造方法可以有参数，所以

C、 构造方法可以重载

D、 构造方法可以设置参数

15、定义类头时，不可能用到的关键字是 ( )。

A、 class

B、 private

C、 extends

D、 public

16、下列类头定义中，错误的是 ( )。

A、 public x extends y {...}  
y {...}

B、 public class x extends

C、 class x extends y implements y1 {...}

D、 class x {...}

17、设 A 为已定义类名，下列声明 A 类的对象 a 的语句中正确的是 ( )。

A、 float A a;

B、 public A a=A( );

C、 A a=new int( );

D、 static A a=new A( );

18、设 i, j 为类 X 中定义的 int 型变量名，下列 X 类的构造方法中不正确的是 ( )。

A、 void X(int k){ i=k; }

B、 X(int k){ i=k; }

C、 X(int m, int n){ i=m; j=n; }

D、 X(){ i=0; j=0; }

19、有一个类 A，以下为其构造方法的声明，其中正确的是 ( )。

A、 void A(int x){...}

B、 public A(int x){...}

C、 public a(int x){...}

D、 static A(int x){...}

20、下列方法定义中，不正确的是 ( )。

A、 float x( int a,int b ) { return (a-b); }  
{ return a-b; }

B、 int x( int a,int b)

C、 int x( int a,int b ); { return a\*b; }  
{ return 1.2\*(a+b); }

D、 int x(int a,int b)

21、设 i, j 为类 X 中定义的双精度型变量名，下列 X 类的构造方法中不正确的是 ( A )。

A、 double X(double k){ i=k; return i; }

B、 X(){ i=6; j=8; }

C、 X(double m, double n){ i=m; j=n; }

D、 X(double k){ i=k; }

22、设 ClassA 为已定义类名，下列声明 Class A 类的对象 ca 的语句

中正确的是 ( )。

A、 public ClassA ca=new ClassA( );                      B、 public ClassA  
ca=ClassA( );

C、 ClassA ca=new class( );                      D、 ca ClassA;

23、设 m, n 为类 A 中定义的 int 型变量名, 下列 A 类的构造方法中不正确的是 ( )。

A、 void A(int k){ m=k; }

B、 A(int k){ m=k; }

C、 A(int m, int n){ m=i; n=j; }

D、 A(){ m=0; n=0; }

24、设 i, j, k 为类 School 中定义的 int 型变量名, 下列类 School 的构造方法中不正确的是 ( )。

A、 School ( int m){ ... }

B、 void School ( int m){ ... }

C、 School ( int m, int n){ ... }

D、 School ( int h, int m, int

n){ ... }

25、下列选项中, 哪个是 java 语言所有类的父类 ( )

A、 String

B、 Vector

C、 Object

D、 KeyEvent

26、Java 中所有的类都是通过直接或间接地继承 ( ) 类得到的。  
( )

A、 java.lang.Object

B、 java.lang.Class

C、 任意类

D、 以上答

案都不对

27、对于任意一个类, 用户所能定义的构造方法的个数至多为 ( )。

A、 0 个

B、 1 个

C、 2 个

D、 任意个

28、构造方法何时被调用 ( )

A、 类定义时

B、 创建对象时

C、 调用对象方法时

D、

使用对象的变量时

29、类 Test1 定义如下

```
public class Test1{
```

```
public float aMethod (float a , float b) { }
```

```
()
```

```
}
```

将以下哪种方法插入 ( ) 是不合法的。 ( )

A、 public float aMethod ( float a , float b , float c ) { }

B、 public float aMethod ( float c , float d ) { }

C、 public int aMethod ( int a , int b ) { }

D、 public int aMethod ( int a , int b , int c ) { }

30、用来导入已定义好的类或包的语句是 ( )

A、 main

B、 import

C、 public class

D、 class

31、类是具有相同 ( ) 的集合, 是对对象的抽象描述。 ( )

A、 属性和方法

B、 变量和方法

C、 变量和数据

D、 对象和

属性

32、一个对象的生命周期分 ( ) 3 个阶段。 ( )

- A、使用、生成和清除                      B、清除、使用和生成  
C、生成、使用和清除                      D、生成、清除和使用
- 33、以下（ ）是专门用于创建对象的关键字。                      （                      ）  
A、 new                      B、 double                      C、 class                      D、 int
- 34、表达式（ ）将返回当前的系统时间和日期                      （                      ）  
A、 DateFormat.getDataTime                      B、 new Data  
C、 DateFormat.getDataTimeInstance()                      D、 new CurrentData
- 35、（ ）为程序员提供了一些可完成通用任务的现有类                      （                      ）  
A、 Java 类库                      B、 preExisting                      C、 java 代码                      D、 library  
关键字
- 36、程序员通常会使用（                      ）  
A、 程序员定义的方法                      B、 现有的方法  
C、 程序员定义的得带以及现有的方法                      D、 既不是程序员定义的方法也不是现有的方法
- 37、利用方法中的（                      ）语句可为调用方法返回一个值（                      ）  
A、 return                      B、 back                      C、 end                      D、 以上答案都不对
- 38、（                      ）方法用做返回两个参数中的较大值（                      ）  
A、 max                      B、 maximum                      C、 larger                      D、 greater
- 39、（                      ）将被传送至一个被调用的方法（                      ）  
A、 返回值                      B、 返回类型                      C、 参数表                      D、 参数列表
- 40、方法的参数可以是（                      ）  
A、 常量                      B、 表达式                      C、 变量                      D、 以上答案都对
- 41、（                      ）是位于方法头中的一个以逗号分隔的声明列表（                      ）  
A、 参数表                      B、 参数列表                      C、 值表                      D、 变量表
- 42、方法的定义是由（                      ）组成的（                      ）  
A、 一个方法                      B、 一个方法体                      C、 一对花括号                      D、 以上答案都对
- 43、一个（                      ）对象将包括 int 型数据（                      ）  
A、 IntegerObject                      B、 Int                      C、 IntData                      D、 Integer
- 44、在方法调用过程中。位于方法名之后圆括号的变量被称为（                      ）  
A、 变元                      B、 参数                      C、 语句                      D、 声明
- 45、一条 return 语句将给调用程序返回（                      ）个值（                      ）  
A、 0                      B、 1                      C、 任意数量                      D、 A 和 B
- 46、方法的第一行被称为方法的（                      ）  
A、 方法体                      B、 标题                      C、 调用者                      D、 方法头
- 47、实例变量的作用域整个的（                      ）  
A、 语句块                      B、 方法                      C、 类                      D、 以上答案都不对
- 48、在某个方法内部定义的变量被称为（                      ）



A、实例变量    B、局部变量    C、类变量    D、隐藏变量

49、在一个类的内部声明，但又处于该类方法外部的变量被称为（    ）

A、局部变量    B、隐藏变量    C、实例变量    D、常量

50、从定义局部变量语句的外部访问该局部变量，会导致（    ）出现（    ）

A、逻辑出错    B、编译错误    C、运行错误    D、以上答案都不对

51、实例变量（    ）

A、是在一个类的内部声明的变量    B、的作用域为整个类

C、可被同一类中的任何方法所访问    D、以上答案都对

52、当方法内传递一个参数时，将该参数值的一个拷贝传递给方法的传递方式称为（    ）

A、调用传递    B、值传递    C、引用传递    D、方法传递

53、当一个变量在应用程序的整个生命周期中被使用，且整个过程中的其他值也不能被改变时，那么将他声明为一个（    ）

A、局部变量    B、常量    C、实例变量    D、以上答案都不对

54、关键字（    ）表示一个类定义的开始（    ）

A、declare    B、new    C、class    D、以上答案都不对

55、类的定义必须包含在一对（    ）内（    ）

A、方括号    B、花括号    C、逗号    D、圆括号

56、实例变量的初始化（    ）

A、是在他们声明时完成    B、将被设置为默认值

C、是在一个构造方法中完成的    D、以上答案都对

57、某个方法，在保证参数值有效性的前提下设置其实例变量    number，该方法的方法名最好取（    ）

A、number    B、set    C、setNumber    D、setValue

58、某个方法，在保证参数值有效性的前提下获取其实例变量    number，该方法的方法名最好取（    ）

A、number    B、get    C、getNumber    D、getValue

59、通过使用关键字（    ）创建对象（    ）

A、object    B、instantiate    C、create    D、new

60、参照以下 Java 代码，以下四个叙述中最确切的是（    ）

```
class A{
int x; static int y;
void fac(String s){System. out. println(" 字符串：" +s);}
}
```

A、x、y 和 s 都是成员变量  
s 是局部变量

B、x 是实例变量、 y 是类变量、

C、x 和 y 是实例变量、 s 是参数

D、x、y 和 s 都是实例变量

61、下列关于变量作用域的描述中，不正确的一项是 ( )

A、变量属性是用来描述变量作用域的

B、局部变量作用域只能是它做在的方法代码段内

C、类变量能在类的方法中声明

D、类变量的作用域是整个类

62、下列关于变量作用域的说法中，正确的一项是 ( )

A、方法参数的作用域是方法外部代码段

B、异常处理参数的作用域是方法外部代码段

C、局部变量的作用域是整个类

D、类变量的作用域是类的某个方法

63、System 类在哪个包中 ? ( )

A 、 java.util

B 、 java.io

C 、 java.awt

D 、 java.lang

64、给出下面代码段

1) public class Test {

2) int m, n;

3) public Test() {}

4) public Test(int a) { m=a; }

5) public static void main(String arg[]) {

6) Test t1,t2;

7) int j,k;

8) j=0; k=0;

9) t1=new Test();

10) t2=new Test(j,k);

11) }

12) }

哪行将引起一个编译时错误 ? ( )

A 、 line 3

B、 line 5

C、 line 6

D、 line 10

65、对于下列代码、

1) class Person {

2) public void printValue(int i, int j) { //... }

3) public void printValue(int i){ //... }

4) }

5) public class Teacher extends Person {

6) public void printValue() { //... }

7) public void printValue(int i) { //... }

8) public static void main(String args[]){

9) Person t = new Teacher();

```
10) t.printValue(10);
11) }
```

第 10 行语句将调用哪行语句？ ( )

A 、 line 2                      B、 line 3                      C 、 line 6                      D、 line 7

66、下列关于构造方法的说话正确的是 ( )

- A、 类中的构造方法不可省
- B、 构造方法必须与类同名，但方法不能与 class 同名
- C 、 类的构造方法在一个对象被创建时自动执行
- D、 一个类只能定义一个构造方法

67、下列语句中，对构造方法的描述错误的是 ( )

- A、 构造方法的名称必须和类名相同
- B、 构造方法没有返回值，返回类型也不能用 void
- C、 构造方法在一个类中可以多次出现
- D、 当重写了带参数的构造方法，系统默认的不带参数的构造方法依然存在

68、下面关于 java 中类的说法哪个是不正确的 ( )

- A、 类体中只能有变量定义和成员方法的定义，不能有其他语句
- B、 构造方法是类中的特殊方法
- C、 类一定要声明为 public 的，才可以执行
- D、 一个 java 文件中可以有多个 class 定义

69、下列哪个 不是 面向对象的三大特性之一 ( )

- A、 封装性                      B、 继承性                      C、 多态性                      D、 重载

70、下面哪种语言 不是 面向对象语言 ? ( )

- A、 java                      B、 C++                      C 、 C                      D、 Smalltalk

71、在创建对象时 必须 ( )

- A、 先声明对象，然后才能使用对象
- B、 先声明对象，为对象分配内存空间，然后才能使用对象
- C、 先声明对象，为对象分配内存空间，对对象初始化，然后才能使用对象
- D、 上述说法都对

72、以下叙述不正确的是 ( )

- A、 面向对象方法追求的目标是、尽可能地运用人类自然的思维方式来建立问题空间的模型，构造尽可能直观、自然的表达求解方法的软件系统。
- B、 面向对象方法的优点有、易于维护，可重用性好，易于理解、扩充和修改。
- C、 面向对象 = 对象 + 分类 + 继承 + 消息通信。
- D、 面向对象的基本特征是封装性、继承性和并行性。

73、对于构造函数，下列叙述不正确的是 ( )。

A、构造函数是类的一种特殊函数，它的方法名必须与类名相同。

B、构造函数的返回类型只能是 void 型。

C、构造函数的主要作用是完成对类的对象的初始化工作。

D、一般在创建新对象时，系统会自动调用构造函数。

## 二、填空题

1、通常用关键字 \_\_\_\_\_ 来新建对象，通过类创建对象的基本格式为：\_\_\_\_\_。

2、通过类 MyClass 中的不含参数的构造方法，生成该类的一个对象 obj，可通过以下语句实现 \_\_\_\_\_。

3、面向对象程序设计的主要特点是 \_\_\_\_\_、\_\_\_\_\_和 \_\_\_\_\_。

4、构造方法是类中的一种特殊方法，它用来定义对象的 \_\_\_\_\_。

5、在 Java 程序中定义的类中包括有两种成员，分别是：\_\_\_\_\_、\_\_\_\_\_。

6、用 static 修饰的方法，称为静态方法。它们不是对象的方法，而是整个类的方法。静态方法只能处理用关键字 \_\_\_\_\_ 修饰的数据。

7、列举三个常用类库的包、\_\_\_\_\_、和 \_\_\_\_\_。

8、Java 中所有类都是类 \_\_\_\_\_ 的子类。

9、Java 中类成员的限定词有以下几种：private、public、friendly、protected 其中，\_\_\_\_\_的限定的范围最大。

10、被关键字 \_\_\_\_\_ 修饰的方法是不能被当前类的子类重新定义的方法。

11、JAVA 程序的基本元素是 \_\_\_\_\_。

12、\_\_\_\_\_是 JAVA 语言中定义类时必须使用的关键字。

13、定义在类中方法之外的变量称为 \_\_\_\_\_。

14、方法声明包括 \_\_\_\_\_和 \_\_\_\_\_两部分。

15、方法头确定方法的 \_\_\_\_\_，形式参数的名字和 \_\_\_\_\_、\_\_\_\_\_的类型和访问限制。

16、方法体 由包括在花括号内的声明部分和语句部分组成，描述方法的功能。

17、类头定义的基本格式要求为 \_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_和 \_\_\_\_\_。

18、\_\_\_\_\_是指 Java 程序通过规则引擎调用此规则包时，将数据存储在 Java 的类中，以对象的形式传入。

19、在 Java 类对象的 \_\_\_\_\_中直接输入需要导入的类的完全类名。

20、\_\_\_\_\_是 Java 的核心内容，是用来创建对象的模板。

21、类有两种基本成员是 \_\_\_\_\_和 \_\_\_\_\_。

22、一个 Java 源文件就是由若干个 \_\_\_\_\_构成的。

- 23、\_\_\_\_\_是抽象，而\_\_\_\_\_是具体。
- 24、对象的创建、创建对象的过程就是\_\_\_\_\_的过程
- 25、对象的创建的步骤\_\_\_\_\_、\_\_\_\_\_及\_\_\_\_\_。
- 26、使用 new 运算符实例化一个已经声明的\_\_\_\_\_
- 27、实例化一个对象时，会同时调用类的构造方法对它进行\_\_\_\_\_。
- 28、类的\_\_\_\_\_是特殊的类方法，它必须和类同名，并且不能有返回类型，也不能用 void 来标注。
- 29、构造方法是具有特殊地位的方法，\_\_\_\_\_不可以调用构造方法。
- 30、一个类中可以有多个\_\_\_\_\_
- 31、类里面如果没有\_\_\_\_\_，系统会调用默认的构造方法，默认的构造方法是不带任何参数的。
- 32、包主要来解决\_\_\_\_\_的问题。
- 33、\_\_\_\_\_既是命名机制，又是可见度控制机制。
- 34、java 语言的属性与方法的引用均使用\_\_\_\_\_运算符。
- 35、定义类时通常使用关键字\_\_\_\_\_，\_\_\_\_\_，\_\_\_\_\_。
- 36、写出成员变量的定义格式\_\_\_\_\_。
- 37、对象的生命周期包括三个阶段\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_。
- 38、Java 中类成员的访问控制符有以下几种、private, public , \_\_\_\_\_, \_\_\_\_\_。其中，\_\_\_\_\_的限定的范围最大。
- 39、Java 中所有类都是类\_\_\_\_\_的子类。Applet 类是系统类\_\_\_\_\_类的子类，存在于\_\_\_\_\_包中。
- 40、按作用域分类，变量有局部变量、\_\_\_\_\_、方法变量和异常处理参数。
- 41、JAVA 语言通过接口支持\_\_\_\_\_继承，使类继承具有更灵活的扩展性。
- 42、JAVA 语言中\_\_\_\_\_类是所有类的根。

### 三、程序填空

1、请写出下面程序的运行结果。

```
package com.gem.j2SE03;
abstract class A {
 private int x = 100;
 public A(){
 this.print();
 }
 public abstract void print();
}
class B extends A {
 private int x = 200;
 public B(int x){
```

```

 this.x = x;
 }
 public void print(){
 System.out.println("x=" + x);
 }
}
public class TestJava {
 private void mian() {
 A a = new B(20);
 }
}

```

程序运行结果为：

2、请写出下面程序的运行结果。

```

public class Test extends TT{
 public void main(String args[]){
 Test t = new Test("Tom");
 }
 public Test(String s){
 super(s);
 System.out.println("How do you do?");
 }
 public Test(){
 this("I am Tom");
 }
}
class TT{
 public TT(){
 System.out.println("What a pleasure!");
 }
 public TT(String s){
 this();
 System.out.println("I am "+s);
 }
}

```

程序运行结果为：

3、阅读下列程序，写出程序运行的结果：

```

public class Test {
 public static void main (String[] args) {

```

```

String s1=new String("Hello");
String s2="Hello";
String s3="Hello";
System.out.println(s1==s2);
System.out.println(s1.equals(s2));
System.out.println(s2==s3);
System.out.println(s2.equals(s3));
String s4=s1;
System.out.println(s1==s4);
System.out.println(s1.equals(s4));
}
}

```

程序运行结果为：

```

4、 public class Test {
 private static void printArray(int[] arr) {
 for(int i=0;i< arr.length;i++)
 System.out.print(arr[i]+",");
 System.out.println("\n");
 }
 private static void changeValue(int value) {
 value*=2;
 }
 private static void changeValue(int[] arr) {
 for(int i=0;i<arr.length;i++)
 arr[i]*=2;
 }
 public static void main (String[] args) {
 int[] arr={1,2,3,4,5};
 changeValue(arr[0]);
 printArray(arr);
 changeValue(arr);
 printArray(arr);
 }
}

```

程序运行结果为：

```

5、 class MyException extends Exception {
 public MyException(String message) {

```

```

 super(message);
 }
}
public class Test {
 private static void fun1() throws MyException {
 throw new MyException("An MyException object is thrown in fun1().");
 }
 private static void fun2() throws MyException {
 try {
 fun1();
 }
 catch(NullPointerException ex) {
 System.out.println("NullPointerException 、 "+ex.getMessage());
 }
 finally {
 System.out.println("Go through finally code in fun2().");
 }
 }
 public static void main (String[] args) {
 try {
 fun2();
 }
 catch(MyException ex) {
 System.out.println("MyException:"+ex.getMessage());
 }
 catch(Exception ex) {
 System.out.println("Exception:"+ex.getMessage());
 }
 }
}

```

程序运行结果为：

6、下面是一个类的定义，请完成程序填空。

```

public class ____-____
{
 int x, y;
 Myclass (int ____ , int ____) // 构造方法
 {

```



```

 x=i; y=j;
 }
}

```

程序运行结果为：

7、 下面是一个类的定义，请将其补充完整。

```

class _____ {
 String name;
 int age;
 Student(_____ s, int i) {
 name=s;
 age=i;
 }
}

```

程序运行结果为：

8、下面程序的功能是通过调用方法 `max( )` 求给定的三个数的最大值，请将其补充完整。

```

public class Class1{
 public static void main(String args[]){
 int i1=1234,i2=456,i3=-987;
 System.out.println(" 三个数的最大值、 "+ _____);
 }
 public static int max(int x,int y,int z)
 { int temp1,max_value;
 temp1=x>y?x:y;
 max_value=temp1>z?temp1:z;
 return max_value;
 }
}

```

程序运行结果为：

9、 写出下列程序的输出结果

```

public class Test{
 public static void main(String[] args){
 Count myCount = new Count();
 int times = 0;
 for(int i=0;i<100;i++)
 increment(myCount , times);
 System.out.println(" count is " + myCount.count);
 }
}

```

```

 System.out.println(" time is " + times);
 }
 public static void increment(Count c , int times){
 c.count++;
 times++;
 }
}
class Count{
 public int count;
 Count(int c){
 count =c;
 }
 Count(){
 count =1;
 }
}

```

程序运行结果为：

10、阅读下列程序，写出程序运行的结果：

```

class Circle {
 double radius;
 public Circle(double r){
 this.radius=r;
 }
}
public class Test{
 public static void main(String[] args){
 Circle circle1 = new Circle(1);
 Circle circle2 = new Circle(2);
 System.out.println("Before swap:circle1 =" +circle1.radius+" circle2
= "+circle2.radius);
 swap(circle1,circle2);
 System.out.println("After swap:circle1 = "+circle1.radius+" circle2
= "+circle2.radius);
 }
 public static void swap(Circle x , Circle y){
 System.out.println("Before swap: x = "+x.radius+" y =
"+y.radius);
 Circle temp = x;
 x = y;
 }
}

```

```

 y = temp;
 System.out.println("Before swap: x = "+x.radius+" y =
"+y.radius);
 }
}

```

程序运行结果为：

11、下列程序的输出结果是什么？

```

public class Foo{
 static int i=0;
 static int j=0;
 public static void main(String[] args){
 int i=2;
 int k=3;
 int j=3;
 System.out.println(" i + j is " + i + j);
 k = i +j;
 System.out.println(" k is " +k);
 System.out.println(" j is " + j);
 }
}

```

程序运行结果为：

12、运行类 C 的输出结果是什么？

```

class A{
 public A(){
 System.out.println("The default constructor of A is invoked");
 }
}
class B extends A{
 public B(){
 System.out.println("The default constructor of B is invoked");
 }
}
public class C{
 public static void main(String[] args){
 B b = new B();
 }
}

```

程序运行结果为：

13、阅读下列程序写出输出结果：

```
class A{
 String s="class A";
 void show(){
 System.out.println(s);
 }
}
class B extends A{
 String s="class B";
 void show() {
 System.out.println(s);
 }
}
public class TypeConvert{
 public static void main(String args[]){
 B b1;
 B b2=new B();
 A a1,a2;
 a1=(A)b2;
 a2=b2;
 System.out.println(a1.s);
 a1.show();
 System.out.println(a2.s);
 a2.show();
 b1=(B)a1;
 System.out.println(b1.s);
 b1.show();
 System.out.println(b2.s);
 b2.show();
 }
}
```

程序运行结果为：

14、阅读下列程序，写出程序运行的结果：

```
class A {
int i, j;
public A(){
this.i=0;
this.j=0;
```

```

}
public void print(){ System.out.println ("i="+i+" "+"j="+j);}
}
class B extends A{
int m;
public B (int i, int j, int m){
super ();
this.m=m;
}
public void print (){System.out.println ("i="+i +" m="+m);}
}
public class C{
public static void main(String args[]){
A a=new A();
B b=new B(1, 2, 3);
a.print ();
b.print ();
}
}}

```

程序运行结果为：

15、阅读下列程序，写出程序运行的结果：

```

class Demo {
public void test ()
{System.out.println ("NO");}
public void test (int i)
{System.out.println (i);}
public void test (int a, int b)
{System.out.println (a+b);}
}
public class Test {
public static void main(String args[]){
 Demo de=new Demo();
de.test();
de.test(5);
de.test(6,8);
}
}

```

程序运行结果为：

16、阅读下列程序，写出程序运行的结果：

```
class Cube{
 int width;
 int height;
 int depth;
 Cube(int x,int y,int z){
 this.width=x;
 this.height=y;
 this.depth=z;
 }
 public int vol(){
 return width*height*depth;}}
public class UseCube {
 public static void main(String[] args) {
 Cube a=new Cube(3,4,5);
 System.out.println(" 长度="+a. width);
 System.out.println(" 体积="+a.vol());
 }
}
```

程序运行结果为：

17、阅读下列程序，写出程序运行的结果：

```
class SuperClass{
 int x=10;
 int y=20;
 public void show(){
 System.out.println(" 我是父类的成员方法！ ");}
}
class SubClass extends SuperClass{
 int z=30;
}
public class JC {
 public static void main(String[] args) {
 SuperClass father=new SuperClass();
 SubClass child=new SubClass();
 System.out.println(" 父 类 SuperClass 成 员 变 量 、
"+father.x+", "+father.y);
 System.out.println(" 子 类 SubClass 成 员 变 量 、
"+child.x+", "+child.y+", "+child.z);
 }
}
```

```
}
```

程序运行结果为：

18、阅读下列程序，写出程序运行的结果：

```
public class Class431 {
 public static void main(String[] args) {
 int a = 3;
 char m='a';
 prt("m in main="+m);
 prt("a in main="+a);
 prt("return from test2 : "+test2());
 prt("m + a="+test1(m,a));
 }
 static float test1(char x, int y) {
 int z = x+y;
 return z;
 }
 static float test2() {
 int a = 60;
 for (int i = 8; i < 12; i++)
 a= a+i;
 prt("a in test2 is : "+a);
 return a;
 }
 static void prt(String s) {
 System.out.println(s);
 }
}
```

程序运行结果为：

19、阅读下列程序，写出程序运行的结果：

```
public class Class432 {
 static int i=123;
 public static void main (String[] args) {
 int i=456;
 System.out.println("main() 中的 i="+i);
 m1();
 m2();
 }
 static void m1() {
```

```

int i=789;
System.out.println("m1() 中的 i="+i);
}
static void m2() {
System.out.println(" 类变量 i="+i);
i+=6;
System.out.println(" 加 6 后的类变量 i="+i);
}
}

```

程序运行结果为：

20、下面程序的结果是什么？

```

class Tester {
 int var;
 Tester(double var) {
 this.var = (int)var;
 }
 Tester(int var) {
 this("hello");
 }
 Tester(String s) {
 this();
 System.out.println(s);
 }
 Tester(){
 System.out.println("good-bye");
 }
 public static void main(String[] args) {
 Tester t = new Tester(5);
 }
}

```

程序运行结果为：

21、以下代码的运行结果是

```

class Base{
 int i;
 Base(){
 add(1);
 System.out.println(i);
 }
}

```



```

 }
 void add(int v){
 i+=v;
 System.out.println(i);
 }
 void print(){
 System.out.println(i);
 }
}
class MyBase extends Base{
 MyBase(){
 add(2);
 }
 void add(int v){
 i+=v*2;
 System.out.println(i);
 }
}
public class TestClu {
 public static void main(String[] args) {
 go(new MyBase());
 //System.out.println();
 }
 static void go(Base b){
 b.add(8);
 //b.print();
 }
}

```

程序运行结果为：

22、以下代码的输出结果是下列哪个选项？

```

class Base{Base(){
 System.out.println("Base");
}
}
public class Checket extends Base{
 Checket(){
 System.out.println("Checket");
 }
}

```

```

 super();
 }
 public static void main(String argv[]){
 Checket c = new Checket();
 // super();
 }
}

```

程序运行结果为：

23、给定下面的代码，哪个选项在替代 "///Here" 后可以被编译并且改变变量 oak 的值？

```

class Base{
 static int oak=99;
}
public class Doverdale extends Base{
 public static void main(String argv[]){
 Doverdale d = new Doverdale();
 d.amethod();
 }
 public void amethod(){
 ///Here
 }
}

```

程序运行结果为：

#### 四、问答题

- 1、什么是继承？什么是父类？什么是子类？继承的特性可给面向对象编程带来什么好处？什么是单重继承？什么是多重继承？
- 2、什么是类成员，什么是实例成员？他们之间有什么区别？
- 3、抽象类和接口有什么区别？
- 4、类与对象的关系是什么？
- 5、Java 的访问限定修饰符有几种，各自的访问权限是什么？
- 6、谈谈 final, finally, finalize 的区别。
- 7、Anonymous Inner Class (匿名内部类) 是否可以 extends(继承) 其它类，是否可以 implements(实现) interface(接口)？
- 8、abstract class 和 interface 有什么区别？

#### 五、判断题

- 1、类体中 private 修饰的变量在本类中能访问，类生成的对象也能访

问。( )

2、一个类中，只能拥有一个构造器方法。( )

3、抽象类中不可以有非抽象的方法。( )

4、如果将一个对象强制类型转换为一个不兼容的类型，将引发  
NullPointerException  
异常。( )

5、Java 中 Object 类是最顶层的类。( )

6 程序在编译通过以后，仍然可能出现运行时错误和逻辑错误。( )

7 可以通过 javac 来运行一个 Java Application。( )

8 可以在子类构造函数中通过 this 关键字调用基类的构造函数。( )

9 Java 语言是 IBM 公司设计的一种面向对象程序设计语言。( )

10 一个 Unicode 字符占 4 个字节。( )

11 在循环语句中，使用 continue; 语句可以跳出循环。( )

12 在 Java Application 中如果没有定义 main() 方法，则程序不能编译。  
( )

13、Java 语言的模块只有类。( )

14、定义方法时方法名可以是任何标识符。( )

15、方法体中至少有一条 return 语句。( )

16、如果需要可以在方法中再声明一个方法。( )

17、方法定义语句 void x( int a,int b ){ return (a-b); } 是正确的。( )

18、方法内定义的变量是可以在方法外使用的。( )

19、只有直接调用自身的方法才叫递归方法。( )

20、Java 中允许两方法同名，只要它们参数的类型或个数不同就行。  
( )

21、每个应用程序都必须要有 main 方法。( )

22、方法定义语句 float x( int a,int b ){ return (a-b); } 是正确的。( )

23、对象可以赋值，只要使用赋值号（等号）即可，相当于生成了一个各属性与赋值对象相同的新对象。( )

24、有的类定义时可以不定义构造方法，所以构造方法不是必需的。  
( )

25、类及其属性、方法可以同时有一个以上的修饰符来修饰。( )

## 六、编程题

1、学生有姓名 (name)和成绩 (score)信息。成绩有科目 (course)和分数 (grade)信息。学生类的 getResult 方法显示输出成绩信息， setData 方法实现初始化学生信息。编写学生类 (Student)和成绩类 (Score)，并测试。

2、定义一个学生类，包含三个属性（学号，姓名，成绩）均为私有的，分别给这三个属性定义两个方法，一个设置它的值，另一个获得它的值。然后在一个测试类里试着调用。

- 3、编写程序，模拟银行账户功能。要求如下、属性、账号、储户姓名、地址、存款余额、最小余额。方法、存款、取款、查询。根据用户操作显示储户相关信息。如存款操作后，显示储户原有余额、今日存款数额及最终存款余额；取款时，若最后余额小于最小余额，拒绝收款，并显示“至少保留余额        XXX ”
- 4、编程实现、以电话 Phone 为父类（例、电话有本机号码、打电话、接电话等属性和功能，当然还有一些其它的特性），移动电话 Mobilephone 和固定电话 Fixedphone 为两个子类，并使移动电话实现接口、可移动 Moveable。固定电话又有子类、无绳电话 Cordlessphone。设计并定义这几个类，明确它们的继承关系，定义子类时给出子类有别于父类的新特性。
- 5、声明测试类、声明 Phone 类的数组（含 5 个元素），生成五个对象存入数组、其中二个 Phone 类的对象、一个 Mobilephone 类的对象、一个 Fixedphone 类的对象和一个 Cordlessphone 类的对象，打印输出每个对象的某个成员变量。将一个父类的引用指向一个子类对象，用这个塑型后的对象来调用某个方法实现多态性。
- 6、设计一个动物类，它包含动物的基本属性。例如名称、大小、重量等，并设计相应的动作，例如跑、跳、走等。
- 7、设计一个长方形类，成员变量包括长和宽。类中有计算面积和周长的方法，并有相应的 set 方法和 get 方法设置和获得长和宽。编写测试类测试是否达到预定功能。要求使用自定义的包。
- 8、创建一个 People 类，定义成员变量如编号、姓名、性别、年龄；定义成员方法“获得编号”、“获得姓名”、“获得年龄”等，再创建 People 类的对象。
- 9、设计 Point 类用来定义平面上的一个点，用构造函数传递坐标位置。编写测试类，在该类中实现 Point 类的对象。
- 10、编写程序说明静态成员和实例成员的区别。
- 11、设计雇员 Employee 类，记录雇员的情况，包括姓名、年薪、受雇时间，要求定义 MyDate 类作为受雇时间，其中包括工作的年、月、日，并用相应的方法对 Employee 类进行设置。编写测试类测试 Employee 类。要求使用自己的包。
- 12、编写一个学生的类，要求有姓名、年龄、兴趣、班级编号和一个教员类，要求有姓名、教授课程、专业方向、教龄，设计一个主方法，要求在主方法中调用这两个类。
- 13、.声明 Patient 类表示在门诊中的病人，此类对象应包括 name(a string),sex(a char),age(an integer),weight(a float),allergies(a boolean). 声明存取及修改方法.在一个单独的类中，声明测试方法，并生成两个 patient 对象，设置其状态并将信息显示在屏幕上。
- 14、声明并测试一个负数类，其方法包括 toString() 及负数的加，减，乘运算。

15、定义一 Student 类:包括姓名的属性值, Statics 类:

16、定义一个类,成员变量是 String 数组 s,成员方法有 s 的 setter 和 getter,显示 s 中使用字符串元素的方法。

17、构造一个类来描述屏幕上的一个点,该类的构成包括点的 x 和 y 两个坐标,以及一些对点进行的操作,包括、取得点的坐标值,对点的坐标进行赋值,编写应用程序生成该类的对象并对其进行操作。

18、用两个方法打印出 200 以内的全部素数(一个是 main() 方法,一个是专门判定一个数是否否为素数的方法)。

19、已知菲波那契数列的数学表达式为:

$$\text{fibonacci}(n)=n, \quad n=0,1;$$

$$\text{fibonacci}(n)=\text{fibonacci}(n-1)+\text{fibonacci}(n-2), \quad n \geq 2;$$

用递归方法计算 Fibonacci 序列,并打印出其前 15 项的值。

20、编写一个 Application,其中包含三个同名方法 mySqrt(),它们都只有一个参数,参数的类型分别为 int 型、float 型和 double 型,它们的功能均为返回参数的平方根,返回值的类型与参数的类型相同。

在方法 main() 中调用上面的三个方法并输出结果。

21、编程生产 100 个 1 到 6 之间的随机数,统计 1 到 6 每个数出现的概率。

## 第六章 面向对象程序设计

### 一、选择题

- 1、JAVA 语言中，在类定义时使用 `final` 关键字修饰，是指这个类 ( )  
A、不能被继承                      B、在子类方法中不能被调用  
C、能被别的程序自由调用              D、不能被子类的方法覆盖
- 2、下列选项中，表示数据或方法可以被同一包中的任何类或它的子类访问，即使子类在不同的包中也可以的修饰符是 ( )  
A、`public`              B、`protected`              C、`private`              D、`final`
- 3、下列选项中，表示数据或方法只能被本类访问的修饰符是 ( )  
A、`public`              B、`protected`              C、`private`              D、`final`
- 4、下列选项中，接口中方法的默认可见性修饰符是 ( )  
A、`public`              B、`protected`              C、`private`              D、`final`
- 5、方法的重载指多个方法可以使用相同的名字，但是参数的数量或类型必须不完全相同，即方法体有所不同，它实现了 java 编译时的 ( )  
A、多态性              B、接口              C、封装性              D、继承性
- 6、在某个类中存在一个方法、`void sort(int x)`，以下不能作为这个方法的重载的声明的是 ( )。  
A、`public float sort(float x)`                      B、`int sort(int y)`  
C、`double sort(int x,int y)`                      D、`void sort(double y)`
- 7、为了区分类中重载的同名的不同方法，要求 ( )。  
A、采用不同的形式参数列表                      B、返回值类型不同  
C、调用时用类名或对象名做前缀                      D、参数名不同
- 8、在类的定义中，通过使用 ( ) 关键字可创建一个现有类的子类。  
A、`extends`              B、`implements`              C、`inherits`              D、`modifies`
- 9、为了调用超类的方法，可以使用 ( ) 关键字后跟包含该超类所需参数的一对圆括号。  
A、`superclass`              B、`superconstructor`              C、`super`              D、以上答案都不对
- 10、关键字 ( ) 表示某个新类是由一个现有的类中继承的  
A、`interits`              B、`extends`              C、`reuses`              D、以上答案均不对
- 11、JAVA 语言中的类间的继承关系是 ( )  
A、多重              B、单重              C、线程              D、不能继承
- 12、下列选项中，用于定义子类时声明父类名的关键字是 ( )  
A、`interface`              B、`package`              C、`extends`              D、`class`
- 13、下列关键字中，用于声明类实现接口的关键字是 ( )  
A、`implements`              B、`package`              C、`extends`              D、`class`
- 14、下列修饰符可以用于说明类的是 ( )  
A、`private`              B、`static`              C、`abstract`              D、`protected`

- 15、用于定义类成员的访问控制权的一组关键字是 ( )
- A、 class, float, double, public      B、 float , boolean ,int , long
- C、 char , extends , float , double      D 、 public ,private ,protected
- 16、下列哪些说法是正确的 ( )
- A、 JAVA 语言只允许单继承
- B、 JAVA 语言中只允许实现一个接口
- C、 java 语言不允许同时继承一个类并实现一个接口
- D、 JAVA 语言的单一继承使得代码更加可靠
- 17、方法的重载是指 ( )
- A、 两个或两个以上的方法取相同的方法名，但是参数的个数或类型不同
- B、 两个以上的方法去相同的名字和具有相同参数个数，但形参的类型可以不同
- C、 两个以上的方法名字不同，但形参的个数或类型相同
- D、 两个以上的方法取相同的方法名，并方法的返回值类型相同
- 18、下列关于类说法正确的是 ( )
- A、 类是 JAVA 语言中的一种复合数据类型
- B、 类是包含数据变量和方法
- C、 类是对所有具有一定共性的对象的抽象
- D、 JAVA 语言的类支持多继承
- 19、下面哪个说法确 ( )
- A、 如果某公共的类被声明在一个源文件中，则该类必须与源文件具有相同的名字
- B、 JAVA 语言中，一个类只能是实现一个接口
- C、 如果某源文件包含 package 语句，则该语句必须是第一个非空、非注释行
- D、 接口中的方法必须是抽象的，但可以不用关键字 abstract 修饰
- 20、类的设计要求它的某个成员变量不能被外部类直接访问，应该使用下面的那些修饰符获得需要的访问控制 ( )
- A、 public      B 、 不加修饰符      C 、 protected      D、 private
- 21、指出下列哪个方法不是方法 public void add(int a){} 的合理重载 ( )
- A、 public int add(int a)      B、 public void add(long a)
- C、 public void add (int a,int b)      D、 public void add(float a)
- 22、下列说法正确的是 ( )
- A、 用 final 修饰一个类表明这个类不可以派生子类
- B、 用 final 修饰一个方法表明这个方法不能被覆盖
- C、 用 final 修饰一个变量会变成一个常量
- D、 用 final 修饰的类可以被继承
- 23、下列程序片断用的是什么机制？ ( )

```

class Student{
String name;
int age;
Student(){}
Student(String n, int a){name=n; age=a;}
Student(String n){name=n;}
}

```

A、方法的覆盖      B、构造方法重载      C、多态      D、这是错误的代码

24、关于抽象类下列哪个描述正确？ ( )

A、抽象类不能包含抽象方法。      B、接口和抽象类是一回事。

C、抽象类不能实例化，即不能生成对象。      D、抽象类可以实例化对象。

25、关于被私有保护访问控制符 private protected 修饰的成员变量，以下说法正确的是 ( )

A、可以被三种类所引用 :该类自身、与它在同一个包中的其他类、在其他包中的该类的子类

B、可以被两种类访问和引用 :该类本身、该类的所有子类

C、只能被该类自身所访问和修改

D、只能被同一个包中的类访问

26、编译并运行以下程序，以下描述哪个选项是正确的 ( )

```

class X{
protected String toString(){
return super.toString();}
}

```

A、编译通过运行无异常

B、编译通过但运行时出错、

C、行 2 出错，不能成功编译

D、不能成功编译，行 3 出错

27、下列关于修饰符混用的说法，错误的是 ( )

A、abstract 不能与 final 并列修饰同一个类

B、abstract 类中不可以有 private 的成员

C、abstract 方法必须在 abstract 类中

D、static 方法中能处理非 static 的属性

28、类 Test1 定义如下：

```

public class Test1{
public float aMethod (float a , float b) {}
}

```

将以下哪种方法插入行 3 是不合法的。 ( )

A、 public float aMethod ( float a , float b , float c ) {}



- B、 public float aMethod ( float c , float d ) { }
- C、 public int aMethod ( int a , int b ) { }
- D、 private float aMethod ( int a , int b , int c )

29、类 Test1 、 Test2 定义如下、

- ```
1 . public class Test1
2 . { public float aMethod ( float a , float b ) throws
3 . IOException { }
4 . }
5. public class Test2 extends Test1{
6 .
7 . }
```

将以下哪种方法插入行 6 是不合法的。()

- A、 float aMethod (float a , float b) { }
- B、 public int aMethod (int a , int b) throws Exception{ }
- C、 public float aMethod (float p , float q) { }
- D、 public int aMethod (int a , int b) throws IOException{ }

30、下列有关 Java 中接口的说法哪个是正确的？()

- A、接口中含有具体方法的实现代码
- B、若一个类要实现一个接口，则用到 “ implements ” 关键字
- C、若一个类要实现一个接口，则用到 “ extends ” 关键字
- D、接口不允许继承

31、如果有一个对象 myListener (其中 myListener 对象实现了 ActionListener 接口), 下列哪条语句使得 myListener 对象能够接受处理来自于 smallButton 按钮对象的动作事件 ? ()

- A、 smallButton.add(myListener);
- B、 smallButton.addListener(myListener);
- C、 smallButton.addActionListener(myListener);
- D、 smallButton.addItem(myListener);

32、不允许作为类及类成员的访问控制符的是 ()。

- A、 public B、 private C、 static D、 protected

33、对于构造方法，下列叙述不正确的是 ()

- A、构造方法是类的一种特殊方法，他的方法名必须与类名相同。
- B、构造方法的返回值类型只能是 void 型。
- C、构造方法的主要作用是完成对类的对象的初始化工作。
- D、一般在创建新对象时，系统会自动调节用构造方法。

34、在 JAVA 中，一个类可以同时定义许多同名的方法，这些方法的形式参数的个数、类型、顺序各不相同，返回值也可以不相同，这种面向对象程序特征称为 ()

- A、隐藏 B、覆盖 C、重载 D、JAVA 不支持此特性

35、ArrayList 类的 () 方法可用来检索 ArrayList 中一个特定位置上放的元素

A、get B、insert C、getAt D、retrieve

36、若已经使用 A A 类创建对象 a 和 BB 类创建对象 b , 则下面的哪一个方法调用是正确的 ()

A、 a.show B、 AA.show C、 AA.show D、 a.show
b.show BB.show b.show BB.show

37、若在某一个类定义中定义有如下的方法 abstract void performDial();

该方法属于 ()。

A、本地方法 B、最终方法 C、成员方法 D、抽象方法

38、设有下面两个类的定义、

```
class Person {  
    long id; // 身份证号  
    String name; // 姓名  
}
```

```
class Student extends Person {  
    int score; // 入学总分  
    int getScore(){  
        return score;  
    }  
}
```

问：类 Person 和类 Student 的关系是 () 。

A、包含关系 B、继承关系
C、关联关系 D、无关系，上述类定义有语法错误

39、如果子类中的方法 mymethod() 覆盖了父类中的方法 mymethod() , 假设父类方法头部定义如下： void mymethod(int a) , 则子类方法的定义不合法的是、 ()

A、 public void mymethod(int a) B、 protected void mymethod(int a)

C、 private void mymethod(int a) D、 void mymethod(int a)

40、ArrayList 对象的 () 方法将返回一个可一次访问 ArrayList 中元素的迭代器

A、 iterator B、 getIterator C、 returnTterator D、 以上答案都不对

41、为删除 ArrayList 中的一个特定索引所对应的元素。应使用 () 方法

A、 remove B、 removeAt C、 delete D、 deleteAt

42、用 private 声明的类成员，其最大的访问范围是 ()。

A、本类 B、本包中的所有类 C、所有包中的子类 D、所有包中的全部类

43、对于任意一个类，用户所能定义的构造方法的个数至多为 ()。

A、0 个 B、1 个 C、2 个 D、任意个

44、构造方法何时被调用 ()

A、类定义时 B、创建对象时 C、调用对象方法时
D、使用对象的变量时

45、如果创建的一个子类覆盖一个父类 public 方法，则对该方法可以使用哪些修饰符修饰？ ()

A、public B、public 和 protected
C、private D、任意修饰符都可以，由系统随机选取

46、关于被保护访问控制符 protected 修饰的成员变量，以下说法正确的是 ()

A、可以被三种类所引用 :该类自身、与它在同一个包中的其他类、在其他包中的该类的子类

B、可以被两种类访问和引用 :该类本身、该类的所有子类

C、只能被该类自身所访问和修改

D、只能被同一个包中的类访问

47、下面对 java 继承机制描述错误的是、 ()

A、声明子类时用 extends 关键字 B、继承是面向对象的一个重要特征

C、java 支持多重继承 D、继承时子类可以添加自己的方法

48、对关键字 super 的用法描述正确的是、 ()

A、用来调用父类的构造方法 B、用来调用父类中被重载的方法

C、用来调用父类中隐藏的成员变量 D、以上说法都正确

49、使用 ArrayList 类的 () 方法生成一个迭代器

A、getIterator B、makeIterator C、iterator D、iterate

50、当子类中的成员变量与父类中的成员变量相同时 : ()

A、子类中的成员变量隐藏父类中的变量 B、父类中的成员变量隐藏子类中的变量

C、子类中变量无效 D、该声明非法

51、类 class A 中方法定义为 protected void f()，则该方法的访问性是 ()

A、所有类都访问该方法

B、所有子类都访问该方法

C、同一目录下的类可以访问该方法，但需要用 import 语句装载该类

D、以上均不对

52、在 Java 中，关键字 () 使类不能派生出子类。

A、final B、public C、private D、protected

- 53、在 Java 中，类 Worker 是类 Person 的子类，Worker 的构造方法中有一句“super()”，该语句（ ）。
- A、调用类 Worker 中定义的 super() 方法； B、调用类 Person 中定义的 super() 方法；
- C、调用类 Person 的构造函数； D、调用类 Worker 的构造函数；
- 54、下列说法正确的是？（ ）
- A、一个子类可以有多个父类，一个父类也可以有多个子类
- B、一个子类可以有多个父类，但一个父类只能有一个子类
- C、一个子类只能有一个父类，但一个父类可以有多个子类
- D、上述说法都不对
- 55、已知类关系如下：
- ```
class Employee{, };
class Manager extends Employee{, };
```
- 则以下有关创建对象的语句不正确的是（ ）
- A、Employee e=new Manager();      B、Employee m=new Employee();
- C、Manager c=new Manager();      D、Manager d=new Employee();
- 56、在 Java 程序中，对象之间的相互通信通过（ ）
- A、继承实现      B、调用类的方法实现      C、封装实现
- D、方法重载实现
- 57、关于抽象类正确的是（ ）
- A、抽象类中不可以有非抽象方法。
- B、某个非抽象类的父类是抽象类，则这个子类必须重载父类所有的抽象方法。
- C、不能用抽象类去创建对象。
- D、接口和抽象类是同一个概念。
- 58、下面关于抽象类的描述错误的是（ ）
- A、继承抽象类可以创建子类      B、抽象类中一定有抽象方法
- C、抽象类中可以有成员方法      D、抽象方法只能存在抽象类或接口中
- 59、下面选项正确的是（ ）
- A、抽象类可以有构造方法。      B、接口可以有构造方法。
- C、可以用 new 操作符操作一个接口。      D、可以用 new 操作符操作一个抽象类。
- 60、（ ）访问修饰符可允许其他类对某个方法进行调用
- A、public      B、private      C、默认      D、以上答案都不对
- 61、使用（ ）方法可以获取得到某个实例变量的值。
- A、get 方法      B、return 语句      C、值的方法      D、set 方法

- 62、一个 private 实例变量不能 ( )  
 A、在声明时被初始化 B、在所声明类的外侧被初始化  
 C、在某个构造方法之内初始化 D、初始化为默认值
- 63、构造方法和普通方法之间的一个重要区别是 ( )  
 A、构造方法不能指定返回值类型 B、构造方法不能指定任何参数  
 C、同一文件中，构造方法位于其他方法之前 D、构造方法能够向实例变量赋值
- 64、一个类可得到许多 ( )，正如一个基本类型可得到许多变量一样  
 A、名称 B、对象 C、值 D、类型
- 65、set 方法能够为开发人员 ( )  
 A、提供校验的范围 B、修改数据  
 C、提供数据验证 D、以上答案都正确
- 66、声明为 private 的实例变量不能直接 ( ) 访问  
 A、在类的外侧 B、由同一个类的其他方法  
 C、由类的构造方法 D、在同一个类的内部
- 67、下列选项中，定义最终类的关键字是 ( )。  
 A、interface B、implements C、abstract D、final
- 68、Java 语言的类间的继承关系是 ( )  
 A、单继承 B、多继承 C、单继承和多继承 D、不能继承
- 69、使用 ( ) 语句可调用超类中无参的构造方法  
 A、super(); B、superclass(); C、superClass(); D、以上答案不对
- 70、当子类覆盖超类中的方法时，如果要在子类中完成对超类方法的访问，那么在调用时需要在对象的名字前添加一个 ( ) 关键字和一个 ( )  
 A、super 分号 ( ; ) B、super 逗号 ( , )  
 C、super 点 ( . ) D、以上答案都不对
- 71、Graphics 类的 ( ) 方法将绘制一个填充的矩形  
 A、filledRect B、rectangle C、solidRect D、fillRect
- 72、如果 Truck 类扩展 Vehicle 类那么，若现在希望从 Truck 类中调用 Vehicle 类的一个无参的构造方法，则需编写的代码应为 ( )  
 A、Truck() B、superClass() C、super() D、noArgument()
- 73、如果 Truck 类扩展了 Vehicle 类，那么称 Truck 类为 ( )，称 Vehicle 类为 ( )  
 A、超类，子类 B、子类，超类 C、超类，直接超类 D、

直接超类，超类

74、当子类覆盖其超类中的方法时，（ ）

A、超类的方法需要拥有和子类方法相同的方法头

B、超类的方法将被子类的方法所取代

C、应使用关键字 `super` 后跟一个点（`.`）来实现子类中超类方法的访问

D、以上答案都对

75、在（ ）关系中，子类的对象也可看作是其超类的一个对象

A、`is a`

B、`like a`

C、`has a`

D、以上答案都不对

76、在（ ）关系中，类中可包含其他类的对象的引用

A、`is a`

B、`like a`

C、`has a`

D、以上答案都不对

77、一个类在实现接口时，必须（ ）

A、额外的定义一个实例变量

B、实现接口中的所有方法

C、扩展该接口

D、以上答案都不对

78、匿名内部类（ ）

A、没有名字

B、在另一个类的内部定义

C、在定义外将创建一个实例

D、以上答案都对

79、利用多态，使用同一个（ ）可产生不同的操作，这取决于调用方法的参数类型。

A、方法返回类型

B、实例变量

C、局部变量

D、方法名

80、如果 `MyTruck` 类继承 `MyCar` 类，则（ ）

A、`MyTruck` 类的对象可以赋值给 `MyCar` 类型的变量

B、`MyCar` 类的对象可以赋值给 `MyTruck` 类型的变量

C、两个类型的对象可以相互赋值给对方类型的变量

D、A 和 B

81、多态能够让开发人员（ ）进行编程

A、以抽象的形式

B、以全局的形式

C、以特定的形式

D、A 和 B

82、对于使用多态的应用程序来说，某个对象的确切类型（ ）

A、执行时才可以知道

B、应用程序编译时能够判断

断

C、程序员编写程序时就已知道

D、永远不会知道

83、通过继承，使用（ ）类型的变量可以调用超类和子类的方法

A、基本

B、超类

C、子类

D、以上答案都不对

84、下面哪个语句是正确的？（ ）

- A、 Panel p = new Panel();                      B、 Button b=new Object("B");  
C、 Panel p = new Frame();                      D、 Frame f = new Panel();

85、指出下列程序的错误，描述正确的是： (            )

```
final class First {
 private int a = 1;
 int b = 2;
}
class Second extends First {
 public void method() {
 System.out.println(a + b);
 }
}
```

- A 、 println() 参数应为字符串，因此此处不能调用该方法。  
B 、 因为变量 a 是 private，所以在其他类中不能访问 a。  
C、 Second 类 能继承 First 类。  
D、 关键字 final 不能修饰类。

86、接口 A 的定义如下，指出下列哪些类实现了该接口？ (            )

```
interface A {
 int method1(int i);
 int method2(int j);
}
```

A、 class B implements A {  
 int method1() { }  
 int method2() { }  
}

B、 class B {  
 int method1(int i) { }  
 int method2(int j) { }  
}

C、 class B implements A {  
 int method1(int i) { }  
 int method2(int j) { }  
}

D、 class B extends A {  
 int method1(int i) { }  
 int method2(int j) { }  
}

87、下面的是关于类及其修饰符的一些描述， 不正确的是、 (            )。

- A、 abstract 类只能用来派生子类，不能用来创建 abstract 类的对象。

B、final 类不但可以用来派生子类，也可以用来创建 final 类的对象。

C、abstract 不能与 final 同时修饰一个类。

D、abstract 类定义中可以没有 abstract 方法。

88、对于构造函数，下列叙述不正确的是（ ）。

A、构造函数也允许重载。

B、子类无条件地继承父类的无参构造函数。

C、子类不允许调用父类的构造函数。

D、在同一个类中定义的重载构造函数可以相互调用。

89、下列选项中，定义抽象类的关键字是（ ）。

A、interface

B、implements

C、abstract

D、class

90、如果任何包中的子类都能访问超类中的成员，那么应使用哪个限定词？（ ）

A、public

B、private

C、protected

D、transient

91、下面的哪个选项是正确的？（ ）

```
class ExSuper{
 String name;
 String nick_name;
 public ExSuper(String s,String t){
 name = s;
 nick_name = t;
 }
 public String toString(){
 return name;
 }
}
public class Example extends ExSuper{
 public Example(String s,String t){
 super(s,t);
 }
 public String toString(){
 return name +"a.k.a"+nick_name;
 }
 public static void main(String args[]){
 ExSuper a = new ExSuper("First","1st");
 ExSuper b = new Example("Second","2nd");
 System.out.println("a is"+a.toString());
 System.out.println("b is"+b.toString());
 }
}
```



- A、编译时会出现例外。
- B、运行结果为、  
a is First  
b is second
- C、运行结果为、  
a is First  
b is Secong a.k.a 2nd
- D、运行结果为、  
a is First a.k.a 1nd  
b is Second a.k.a 2nd

92、运行下列程序的结果是哪个？ ( )

```
abstract class MineBase {
 abstract void amethod();
 static int i;
}
public class Mine extends MineBase {
 public static void main(String argv[]){
 int[] ar = new int[5];
 for(i = 0;i < ar.length;i++)
 System.out.println(ar[i]);
 }
}
```

- A、打印 5 个 0。
- B、编译出错，数组 ar[] 必须初始化。
- C、编译出错，Mine 应声明为 abstract。
- D、出现 IndexOutOfBounds 的例外。

93、有关类 Demo，哪句描述是正确的？ ( )

```
public class Demo extends Base{
 private int count;
 public Demo(){
 System.out.println("A Demo object has been created");
 }
 protected void addOne() {count++; }
}
```

- A、当创建一个 Demo 类的实例对象时，count 的值为 0。
- B、当创建一个 Demo 类的实例对象时，count 的值是不确定的。
- C、超类对象中可以包含改变 count 值的方法。
- D、Demo 的子类对象可以访问 count。

94、在以下供选择的概念中，不属于面向对象语言概念的是 ( )

- A、消息
- B、模块
- C、继承
- D、多态性

95、对静态成员（用 static 修饰的变量或方法）的不正确描述是 ( )。

- A、静态成员是类的共享成员；
- B、静态变量要在定义时就初始化；

C、调用静态方法时要通过类或对象激活；  
操作静态属性；

D、只有静态方法可以

96、下列说法正确的是 ( )

A、语句 `import java.applet.Applet` ;最后的 `Applet` 是代表类 (class)。

B、抽象方法必须在抽象类中，所以抽象类中的方法都必须是抽象方法。

C、如果 `p` 是父类 `Parent` 的对象，而 `c` 是子类 `Child` 的对象，则语句 `c = p` 是正确的。

D、使用方法 `length()` 可以获得字符串或数组的长度。

97、下列说法错误的是 ( )。

A、类及其属性、方法可以同时有一个以上的修饰符来修饰。

B、接口是特殊的类，所以接口也可以继承，子接口将继承父接口的所有常量和抽象方法。

C、一个类如果实现了某个接口，那么它必须重载该接口中的所有方法。

D、Java的源代码中定义几个类，编译结果就生成几个以 `.class` 为后缀的字节码文件。

98、创建对象的关键字是 ( )

A、`errupt`

B、`set`

C、`new`

D、`create`

99、接口可以继承多个父类接口，父接口之间使用 ( ) 符号隔开。

A、`.`

B、`,`

C、`;`

D、`/`

100、下列说法正确的一项是 ( )

A、`java.lang.Integer` 是接口  
可以创建对象

B、`java.lang.Runnable` 类

C、`Double` 对象存在 `java.lang` 包中  
`Lang.Object` 包中

D、`Double` 对象在 `java.`

101、下列关于类、包和源文件的描述中，不正确的一项是 ( )

A、一个包可以包含多个类  
中，可能有一个公共类

B、一个源文件

C、属于同一个包的类在默认情况下可以相互访问  
源文件创建默认的包

D、系统不会为

102、下列关于类、包和源文件的说法中，错误的一项是 ( )

A、一个文件可以属于一个包

B、一个包可包含多个文件

C、一个类可以属于一个包

D、一个包只能含有一个类

103、下列关于 `java.lang` 包封装的基本类的说法中，不正确上网一项是 ( )

A、`java.lang` 包封装了 `Object`、`Class`、`System`、`String`、`Thread` 等基本类

B、`System` 类提供了标准的 I/O 操作，可以实例化

- C、Class 类的对象包含所对应的类的基本信息，可以通过 Class 类的方法访问这些信息
- D、Object 类是所有类都必须继承的类
- 104、定义类时，不可能用到的保留字是 ( )
- A、private                      B、class                      C、extends                      D、implements
- 105、下列那个不是面向对象程序设计方法的特点？ ( )
- A、抽象                              B、继承                              C、多态                              D、结构化
- 106、在使用 interface 声明一个接口时，可以使用哪一个修饰符定义该接口？ ( )
- A、private                              B、protected                              C、private protected                              D、public
- 107、Java 中所有的类都是通过直接或间接地继承 ( ) 类得到的。
- A、java.lang.Object                      B、java.lang.Class                      C、任意类                      D、以上答案都不对
- 108、为了某个 ArrayList 的末尾添加一个元素，应该调用 ( ) 方法
- A、add                      B、addToEnd                      C、append                      D、insertAt
- 109、现有两个类 M、N，以下描述中表示 N 继承自 M 的是 ( )。
- A、class M extends N                              B、class N implements M
- C、class M implements N                              D、class N extends M
- 110、面向对象的一个重要特征 ( ) 它使代码可重用，降低程序复杂性。
- A、类                      B、内部类                      C、继承性                      D、多态
- 111、使用 Iterator 的 ( ) 方法返回集合中下一个元素的引用
- A、next                      B、getNext                              C、hasNext                      D、isNext
- 112、利用 Iterator 的 next 方法所返回的引用类型为 ( )
- A、int                      B、Object                              C、ArrayList                      D、以上答案都不对
- 113、下列选项中，用于定义接口的关键字是 ( )。
- A、interface                      B、implements                              C、abstract                              D、class
- 114、以下关于 Java 语言继承的说法错误的是 ( )。
- A、Java 中的类可以有多个直接父类                              B、抽象类可以有子类
- C、Java 中的接口支持多继承                              D、最终类不可以作为其它类的父类
- 115、下列关于构造方法的叙述中，错误的是 ( )
- A、Java 语言规定构造方法名与类名必须相同
- B、Java 语言规定构造方法没有返回值，但不用 void 声明
- C、Java 语言规定构造方法不可以重载
- D、Java 语言规定构造方法只能通过 new 自动调用

116、在一个类中定义成员变量和成员方法时，不允许（ ）。

- A、两个成员变量名相同                      B、成员变量与成员方法名相同  
C、两个成员方法名相同                      D、成员方法中的参数与成员变量名相同

117、一个子类继承父类，子类不可以继承父类中访问权限设定为（ ）的变量和方法。

- A、public                      B、protected                      C、private                      D、未修饰

118、当子类中的成员变量或方法与继承的父类的成员变量或方法相同时，子类的对象所使用的变量是（ ）

- A、是父类的变量，子类的方法                      B、父类中的变量或方法  
C、子类中的变量或方法                      D、子类的变量，父类的方法

119、下列叙述正确的是（ ）

- A、在类中的构造方法的名字与类的名字相同，而且不能重载  
B、在类中的构造方法的名字与类的名字相同，但没有返回类型  
C、在类中的构造方法的名字与类的名字相同，而且有返回类型  
D、在类中的构造方法的名字与类的名字相同，而且不能初始化

120、要想定义一个不能被实例化的抽象类，在类定义中必须加上修饰符（ ）。

- A、final                      B、public                      C、private                      D、abstract

121、有一个类 A，对于其构造函数的声明正确的是（ ）

- A、void A(int x){ ... }                      B、A(int x){ ... }                      C、A A(int x){ ... }  
D、int A(int x){ ... }

122、下面是有关子类调用父类构造函数的描述正确的是（ ）

- A、子类定义了自己的构造函数，就不会调用父类的构造函数  
B、子类必须通过 super 关键字调用父类没有参数的构造函数  
C、如果子类的构造函数没有通过 super 调用父类构造函数，那么子类会先调用父类不含参数的构造函数，再调用子类自己的构造函数  
D、创建子类对象时，先调用子类自己的构造函数，然后调用父类的构造函数

123、在 Java 中，要定义类的保护类型元素，则必须使用的关键字是（ ）

- A、default                      B、private                      C、public                      D、protected

124、若要说明一个抽象类，使用的关键字是、（ ）

- A、static                      B、implements                      C、extends                      D、abstract

125、方法（ ）能够将 ArrayList 转换成一个一维数组

A、toDArray      B、toOneDimensionalArray      C、fromArrayList      D、toArray

126、如果任何包中的子类都能访问超类中的成员，那么应使用哪个限定词？（      ）

A、public      B、.private      C、protected      D、transient

127、ArrayList 类位于（      ）包内

A、java.util      B、java.awt      C、javax.swing      D、java.awt.event

128、下列关于接口和抽象类的说法不正确的是？（      ）

A、接口也有构造方法      B、实现接口时，需对接口中的所有方法都要实现

C、抽象类也有构造方法      D、抽象类可以派生出子类

129、有关类的说法正确的是？（      ）

A、类具有封装性，所以类的数据是不能被访问的

B、类具有封装性，但可以通过类的公共接口访问类中的数据

C、声明一个类时，必须用 public 修饰符

D、每个类中，必须有 main 方法，否则程序无法运行

130、下述哪一组方法，是一个类中方法重载的正确写法？（      ）

A、int addValue( int a, int b ){return a+b;}  
float addValue ( float a, float b ) {return a+b;}

B、int addValue (int a, int b ){value=a+b; }  
float addValue ( int a, int b ) {return (float)(a+b);}

C、int addValue( int a, int b ){return a+1;}  
int addValue ( int a, int b ) {return a+b;}

D、int addValue( int a, int b ) {return a+b;}  
int addValue ( int x, int y ) {return x+y;}

131、下列程序的结果输出为：（      ）

```
class Strange{
int j;
Strange(){
this(4);
}
Strange(int j){
this.j=j;
}
}
class MainStrange{
public static void main(string[] args){
Strange funny=new Strange();
System.out.println(“ j= “ +funny.j);
}
```

}

A、 j=2                      B、 j=4                      C、 j=6                      D、 j=7

132、(                      )可专门用于存储由多个值所形成的分组

A、 集合                      B、 属性                      C、 访问器                      D、 以上答案都不对

133、迭代器提供了一个用以访问集合中每一个元素的(                      )

A、 计数器                      B、 元素                      C、 引用                      D、 实例变量

134、Java 语言的类间的继承的关键字是 (                      )。

A、 implements                      B、 extends                      C、 class                      D、 public

135、访问修饰符(                      ) 用于指明其成员变量只能在该成员变量所属内部来访问(                      )

A、 private                      B、 public                      C、 dimensional                      D、 以上答案都不对

136、给出下面代码

```
1) class Parent {
2) private String name;
3) public Parent(){}
4) }
5) public class Child extends Parent {
6) private String department;
7) public Child() {}
8) public String getValue(){ return name; }
9) public static void main(String arg[]) {
10) Parent p = new Parent();
11) }
12) }
```

那些行将引起错误？ (                      )

A、 第3行                      B、 第6行                      C、 第7行                      D、 第8行

137、下面关于构造方法的说法不正确的是(                      )

A、 构造方法也属于类的方法，用于创建对象的时候给成员变量赋值。

B、 构造方法不可以重载。

C、 构造方法没有返回值

D、 构造方法一定要和类名相同。

138、下面哪个修饰符修饰的变量是所有同一个类生成的对象共享的(                      )

A、 public                      B、 private                      C、 static                      D、 final

139、下面对对象概念描述错误的是： (                      )

A、 操作是对象的动态属性

B、 任何对象都必须有继承性

C、 对象间的通讯靠消息传递

D、 对象是属性和方法的封装体

140、编译并运行下面的程序，运行结果为： ( )

```
public class A {
 public static void main(String[] args) {
 A a=new A() ;
 a.method(8) ;
 }
 void method(int i) {
 System.out.println(" int : " +i) ;
 }
 void method(long i) {
 System.out.println(" long : " +i) ;
 }
}
```

- A、程序可以编译运行，输出结果为 int : 8
- B、程序可以编译运行，输出结果为 long : 8
- C、程序有编译错误，因为两个 method() 方法必须定义为静态 (static) 的
- D、程序可以编译运行，但没有输出

141、下面关于类的说法不正确的是： ( )

- A、类是 Java 语言中的一种复合数据类型
- B、类中不包含数据变量和方法
- C、类是对所有具有一定共性的对象的抽象类只支持单继承
- D、Java 语言的

142、能作为类及其成员的修饰符是： ( )

- A、interface
- B、class
- C、protected
- D、public

143、设 x、y 均为已定义的类型名，下列声明对象 x1 的语句不可能正确的是： ( )

- A、public x x1=new y();
- B、public y x1=new x();
- C、x x1=new x();
- D、x x1=x();

144、面向对象程序设计技术的特点是： ( )

- A、可重用性
- B、可维护性
- C、表示方法的一致性
- D、可重用性、可维护性、表示方法的一致性

145、设已声明了一个类 A 的两个对象 a1、a2，为了初始化 a1 和 a2，下面语句正确的是： ( )

- A、a1=new(); a2=new();
- B、a1=A new(); a2=A new();
- C、a1,a2=new A();
- D、a1=new A(); a2=new A();

146、构造方法何时被调用 ( )

- A、类定义时
- B、使用对象的变量时
- C、调用对象方法时
- D、创建对象时

147、下列哪个类声明是正确的 ( )

- A、 public abstract class Car{ ... } B、 abstract private move(){ ... }  
C、 protected private number; D、 abstract final class Hl{ ... }

148、下列不属于面向对象程序设计的基本特征的是： ( )

- A、抽象 B、封装 C、继承 D、静态

149、请看下面的程序段

```
class Person {
String name,department;
int age;
public Person(String n) { name = n; }
public Person(String n,int a) { name = n; age = a; }
public Person(String n,String d,int a) {
//doing the same as two arguments version of constructor
//including assignment name=n,age=a
}
}
```

下面那一选项可以添加到 “ //doing the same,, ” 处： ( )

- A、 Person(n,a) B、 this(Person(n,a)) C、 this(n,a) D、 this(name.age)

150、( ) 是一个特殊的方法，用于对类的实例变量进行初始化。

- A、终止函数 B、构造函数 C、重载函数 D、初始化函数

151、关于变量和它们范围的说法，错误的是： ( )

- A、实例变量是类的成员变量  
B、实例变量要用关键字 static 来声明  
C、在一个方法执行时，定义在该方法的局部变量才被创建  
D、局部变量必须在它们使用前初始化

152、请看下面的程序段

```
class Test{
private int m;
public static void fun(){
//some code ...
}
}
```

方法 fun() 如何来访问变量 m : ( )

- A、将 private int m 改成 protected int m B、将 private int m 改成 public int m  
C、将 private int m 改成 static int m D、将 private int m 改成 int m

153、请看下面的程序段



```

public class Test {
long a[] = new long[10];
public static void main(String arg[]) {
System .out .println(a[6]);
}
}

```

那一个选项是正确的：（ ）

- A、不输出任何内容                      B、输出 0  
C、当编译时有错误出现                  D、当运行时有错误出现

154、关键字（ ）表明一个对象或变量在初始化后不能修改。

- A、 extends                      B、 final                      C、 this                      D、 finalizer

155、声明为 static 的方法不能访问（ ）类成员。

- A 、超类                      B、 子类                      C、 非 static                      D、 用户自定义类

156、为了区分重载多态中同名的不同方法，要求：（ ）

- A、参数名不同                      B、返回值类型不同  
C、采用不同的形式参数列表                  D、选项 A B C 都对

157、在某个类中定义一个方法： void GetSort(int x)，以下关于能否作为这个方法的重载错误的：（ ）

- A、 void GetSort(float x){x\*=x;}  
B、 int GetSort(double y){ return(int)(2\*y); }  
C、 double GetSort(int x,int y){return x+y; }  
D、都不能

158、下列对继承的说法，哪个是正确的：（ ）

- A、子类只能继承父类 public 方法和状态                      B、子类能继承父类的非私有方法和状态  
C、子类能继承父类的所有方法和状态                      D、子类只能继承父类的方法，而不继承状态

159、定义类 A 如下：

```

class A
{ int a,b,c;
public void B(int x,int y,int z){a=x; b=y; c=z; }
}

```

下面对方法 B() 的重载哪些是正确的：（ ）

- A、 public void A(int x1,int y1,int z1){ a=x; b=y; c=z; }  
B、 public void B(int x1,int y1,int z1){ a=x; b=y; c=z;}  
C、 public void B(int x,int y){ a=x; b=y; c=0;}  
D、 public B(int x,int y,int z){a=x; b=y; c=z; }

160、下列说法正确的是：（ ）

- A、Java 中允许一个子类有多个父类
- B、某个类是一个类的子类，它仍有可能成为另一个类的父类
- C、一个父类只能有一个子类
- D、继承关系最多不能超过 4 层

161、下列说法正确的是：（ ）

- A、子类只能覆盖父类的方法，而不能重载
- B、子类只能重载父类的方法，而不能覆盖
- C、子类不能定义和父类同名同形参的方法，否则，系统将不知道使用哪个方法。
- D、重载就是一个类中有多个同名但不同形参和方法体的方法

162、关于构造函数的说法正确的是：（ ）

- A、一个类只能有一个构造函数
- B、一个类可以有多个不同名的构造函数
- C、构造函数与类同名
- D、构造函数必须自己定义，不能使用父类的所有构造函数

163、下面是有关子类调用父类构造函数的描述正确的是：（ ）

- A、子类定义了自己的构造函数，就不会调用父类的构造函数
- B、子类必须通过 super 关键字调用父类有参数的构造函数
- C、如果子类的构造没有通过 super 调用父类构造函数，那么子类会先调用子类自己的构造函数，再调用父类不含参数的构造函数。
- D、创建子类的对象时，先调用子类自己的构造函数，然后调用父类的构造函数

164、声明为 final 的变量不能在子类中覆盖，声明为（ ）和（ ）的方法是隐式的 final。（ ）

- A、public private
- B、private protected
- C、protected static
- D、static private

165、继承性使 成为可能，它不仅节省开发时间，而且也鼓励人们使用已经验证无误和调试过的高质量软件。（ ）

- A、节省时间
- B、软件复用
- C、软件管理
- D、延长软件生命周期

166、编译运行下面的程序，结果是（ ）。

```
Public class A {
public static void main(String[] args) {
B b=new B();
b.test();
}
void test() {
System.out.print(“ A ”);
}
}
```

```

class B extends A {
void test() {
super.test();
System.out.println(“ B ”);
}
}

```

A、产生编译错误  
并输出结果： AB

B、代码可以编译运行，

C、代码可以编译运行，但没有输出  
产生运行时异常

D、编译没有错误，但会

167、已知类关系如下：

```

class Employee;
class Manager extends Employee;
class Director extends Employee;

```

则以下语句正确的是： ( )

A、 Employee e=new Manager();

B、 Director d=new Manager();

C、 Director d=new Employee();

D、 Manager m=new Director();

168 . 一个 ( ) 类的对象可以当作相应的 ( ) 类对象看待。 ( )

A、 子 系统

B、 用户 系统

C、 子 超

D、 超 用户

169、如果集合中仍有元素，则 Iterator 的 ( ) 方法将返回 true

A、 next

B、 hasNext

C、 hasNext

D、 iterator.next

170、被声明为 private , protected 及 public 的类成员，在类外部：  
( )

A、只能访问到声明为 public 的成员  
protected 和 public 的成员

B、只可能访问到声明为

C、都可以访问

D、都不能访问

171、现有两个类 A、B，以下描述中表示 B 继承自 A 的是 ( )

A、 class A extends B

B、 class B implements A

C、 class A

implements D、 class B extends A

172、设类 B 是类 C 的父类，下列声明对象 x1 的语句中不正确的是  
( )

A、 C x1=new B() ;

B、 B x1=new C() ;

C、 C x1=new C() ;

D、 B x1=new B() ;

173、下面有关接口的说法，哪个是正确的 ( )

A、接口与抽象类是相同的概念

B、一个类不可实现多个接口

C、接口之间不能有继承关系

D、实现一个接口必须实现接口

的所有方法

174、下列哪个不是属于接口用途的 ( )

A、通过接口可以实现不相关类的相同行为  
多个类需要实现的方法

B、通过接口可以指明

C、通过接口可以了解对象的交互界面  
对象所对应的类

D、通过接口可以了解

175、定义字符串 `s: String s="Microsoft 公司"`；执行下面的语句，`char c=s.charAt(9);`

c 的值为：( )

A、null

B、司

C、产生数组下标越界异常

D、公

176、虽然接口和抽象类不能创建对象，但它们的对象引用仍可指向该类型的对象。

这种说法( )

A、正确

B、不正确

C、不能确定

D、接口和抽象类不能说明其对象引用

177、下面哪个类包括 `drawImage()` 方法、`drawString()` 方法和 `setColor()` 方法。( )

A、`java .awt .Applet`

B、`java .awt .Image`

C、`java .awt .Graphics`

D、`java .util .Date`

178、接口中说明了一组公共的抽象方法，其中( )包含公共的静态方法

A、可以

B、不可以

C、不能肯定

D、因静态方法不能被改写，故不能出现在接口中

179、接口是 Java 面向对象的实现机制之一，以下说法正确的是：( )

A、Java 支持多重继承，一个类可以实现多个接口

B、Java 只支持单重继承，一个类可以实现多个接口

C、Java 只支持单重继承，一个类只可以实现一个接口

D、Java 支持多重继承，但一个类只可以实现一个接口

180、关于接口的定义和实现，以下描述正确的是：( )

A、接口定义中的方法都只有定义没有实现

B、接口定义中的变量都必须写明 `final` 和 `static`

C、如果一个接口由多个类来实现，则这些类在实现该接口中的方法时应采用统一的代码

D、如果一个类实现一个接口，则必须实现该接口中的所有方法，但方法未必声明为 `public`

181、下面的说法正确的是：( )

A、接口是 Java 中的一个重要特点

B、在实现

接口函数的时候要用关键字 `public`

C、在继承一个类时，可以同时实现一个或者多个接口。

D、接口可以内嵌在一个类中

182、声明接口的保留字是( )

A、`interface`

B、`protect`

C、`final`

D、`static`

183、下面关于 Vector 类的说法不正确的是：（ ）

A、类 Vector 在 java.util 包中

B、一个向量 (Vector) 对象存放的是一组有序的对象

C、一个向量 (Vector) 对象大小可以根据存放的元素个数的增加而自动增加

D、一个向量 (Vector) 对象中每个元素必须是相同类型的对象

184、下面的接口 (类) 定义正确的是：（ ）

A、

```
public class A {
 private int x;
 public getx() { return x; }
}
```

B、

```
public abstract class A {
 private int x;
 public abstract int getx();
 public int amethod()
 { return 0; }
}
```

C、

```
public class A {
 private int x;
 public abstract int getx();
}
```

D、

```
public interface Interfacea {
 private int x;
 public int getx() {
 return x; }
}
```

185、接口具有下面的特性。（ ）

A、多重继承

B、不可继承

C、抽象性

D、嵌套

186、定义一个包的关键字是（ ）。

A、import

B、package

C、interface

D、protected

187、下列选项中，用于在程序开头引入包的关键字是（ ）。

A、import

B、implements

C、integer

D、protected

188、以下语句输出的结果是（ ）。

```
System.out.println(Math.floor(-0.8));
```

A、0

B、-1

C、-0.8

D、0.8

189、下面的程序段执行后，变量 a、b、c 的值分别是（ ）。

```
int a,b,c;
```

```
a=(int)Math.round(-4.51);
```

```
b=(int)Math.ceil(-4.51);
```

```
c=(int)Math.floor(-4.1);
```

A、-5, -4, -5

B、-4,-4,-5

C、-5,-5,-5

D、

-4,-4,-4

190、下面说法不正确的是 ( )

A、如果某公共的接口被声明在一个源文件中，则接口必须与源文件具有相同的名字

B、如果源文件包含 `import` 语句，则该语句必须是除空行和注释行外的第一个语句行

C、如果某源文件包含 `package` 语句，则该语句必须是第一个非空、非注释行

D、接口中的方法不一定是抽象方法，所以可以不用关键字 `abstract` 修饰

191、在类定义的外部，可以被访问的成员有 ( )。

A、所有类成员 B、`private` 或 `protected` 的类成员 C、`public` 的类成员 D、`public` 或 `private` 的类成员

192、现有类 A 和接口 B，以下描述中表示类 A 实现接口 B 的语句是 ( )。

A、`class A implements B` B、`class B implements A`

C、`class A extends B` D、`class B extends A`

193、针对 `Vector MyVector = new Vector(100,50)` 语句，下面语句中解释正确的是 ( )。

A、创建一个数组类对象 `MyVector`，有 100 个元素的空间，每个元素的初值为 50。

B、创建一个向量类对象 `MyVector`，有 100 个元素的空间，每个元素的初值为 50。

C、创建一个数组类对象 `MyVector`，有 100 个元素的空间，若空间使用完时，以 50 个元素空间单位递增。

D、创建一个向量类对象 `MyVector`，有 100 个元素的空间，若空间使用完时，以 50 个元素空间单位递增。

194、在 Java 中，存放字符串常量的对象属于 ( ) 类对象。

A、`Character` B、`String` C、`StringBuffer` D、`Vector`

195、下面语句对集合描述正确的是 ( )

A、可以使开发人员的注意力集中到数据存储器上 B、可以加速应用程序的开发

C、是 Java 中预先定义好的类 D、B 和 C

二、填空题

1、方法重载实现多态时，要求 \_\_\_\_\_ 、 \_\_\_\_\_ 、 \_\_\_\_\_ 三

个中的一个必须不同。

2、在 Java 语言中， 仅支持类间的 \_\_\_\_\_ 继承。

3、抽象方法只有 \_\_\_\_\_，没有 \_\_\_\_\_的方法。

4、面向对象程序设计语言的三大特征是 \_\_\_\_\_、  
和 \_\_\_\_\_。

5、用于创建类实例对象关键字是 \_\_\_\_\_。

6、JAVA 中类成员的限定词有以下几种、 private ,public , \_\_\_\_\_，  
其中， \_\_\_\_\_ 的限定范围最大。

7、被关键字 \_\_\_\_\_修饰符的方法是不能被当前类的子类重新定义的方法

8、创建类的对象时，使用运算符 \_\_\_\_\_ 给对象分配类存空间

9、JAVA 中所有类都是类 \_\_\_\_\_ 的子类

10、定义类的构造方法不能有返回值类型，其名称与 \_\_\_\_\_ 名相同

11、接口中的成员只有静态常量和 \_\_\_\_\_。

12、在子类中使用关键字 \_\_\_\_\_做前缀可以调用被子类覆盖的父类中的方法

13、关键字 \_\_\_\_\_用于导入包中的类到程序中，供程序使用

14、\_\_\_\_\_ 方法是一种仅有方法头，没有具体方法体和操作实现的方法，该方法必须在抽象类之中定义。\_\_\_\_\_ 方法是不能被当前类的子类重新定义的方法

15、在 Java 程序中，通过类的定义只能实现 \_\_\_\_\_ 重继承，但通过接口的定义可以实现 \_\_\_\_\_ 重继承关系。

16、当类未用访问权限修饰符（ public 或 private ）修饰时， java 语言默认此类的访问权限是 \_\_\_\_\_

17、类是一组同种对象集合与抽象。 即是具有相同 \_\_\_\_\_ 和 \_\_\_\_\_ 的一组对象集合。

18、一个类可以派生出 \_\_\_\_\_子类，但只允许有 \_\_\_\_\_ 个父类，而不允许存在 \_\_\_\_\_个父类。

19、在 Java 中有两种方法实现多态，一种是使用方法的 \_\_\_\_\_实现多态，另一种是使用方法的 \_\_\_\_\_实现多态。

20、Java 中定义类必须使用关键字 \_\_\_\_\_作为标志，用 \_\_\_\_\_关键字声明该类将要实现哪些接口。

21、在 java 语言中通过 \_\_\_\_\_方法实现多继承问题。

22、接口是一种特殊的类，它只能定义 \_\_\_\_\_。

23、调用应用包中类的两种方法是 \_\_\_\_\_、  
和 \_\_\_\_\_。

24、java 语言异常处理三种方式是 \_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_。

25、在 java 语言中中，用（ try\_catch\_final ）结构来捕获异常。在接口声明变量时，其变量在缺省状况下就是 \_\_\_\_\_、  
和 \_\_\_\_\_。

- 26、对于接口的实现，不可以构造一个接口对象，但可以声明一个接口 \_\_\_\_\_。
- 27、在 JAVA 语言中，所有的类都是直接或间接继承 \_\_\_\_\_ 类而得到的。
- 28、多态是指 \_\_\_\_\_。
- 29、如果子类中的某个变量的变量名与它的父类中的某个变量完全一样，则称子类中的这个变量 \_\_\_\_\_ 了父类的同名变量。
- 30、属性的隐藏是指子类重新定义从父类继承来的 \_\_\_\_\_。
- 31、如果子类中的某个方法的名字、返回值类型和 \_\_\_\_\_ 与它的父类中的某个方法完全一样，则称子类中的这个方法覆盖了父类的同名方法。
- 32、同一类中的不同对象具有相同的 \_\_\_\_\_ 和 \_\_\_\_\_。
- 33、java.lang.String 类的访问修饰符权限是 \_\_\_\_\_ 类型的。
- 34、java 语言的接口是特殊的类，其中包含 \_\_\_\_\_ 常量和 \_\_\_\_\_ 方法。
- 35、如果接口中定义了一个方法 methodA()，一个属性 attA，那么一个类 ClassA 要实现这个接口的话，就必须实现其中的方法。
- 36、一个类如果实现一个接口，那么它就必须实现接口中定义的所有方法，否则该类就必须定义成 \_\_\_\_\_ 的。
- 37、Java 仅支持类间的单重继承，接口可以弥补这个缺陷，支持 \_\_\_\_\_ 继承。
- 38、在方法头用 abstract 修饰符进行修饰的方法叫做 \_\_\_\_\_ 方法。
- 39、Java 语言中用于表示类间继承的关键字是 \_\_\_\_\_。
- 40、Java 语言中，表示一个类不能再被继承的关键字是 \_\_\_\_\_。
- 41、创建类的对象时，使用运算符 \_\_\_\_\_ 给对象分配内存空间。
- 42、同一个类中多个方法具有相同的方法名，不同的 \_\_\_\_\_ 称为方法的重载。
- 43、通过类 MyClass 中的不含参数的构造方法，生成该类的一个对象 obj，可通过以下语句实现 \_\_\_\_\_。
- 44、在 java 的接口中声明的成员变量全部是 \_\_\_\_\_，声明的方法是 \_\_\_\_\_。
- 45、类及其 \_\_\_\_\_、\_\_\_\_\_ 可以同时有一个以上的修饰符来修饰。
- 46、一个 Java 类可以有 \_\_\_\_\_ 个父类。
- 47、关于被私有保护访问控制符 private protected 修饰的成员变量，\_\_\_\_\_ 可以被两种类来引用 \_\_\_\_\_、\_\_\_\_\_。
- 48、abstract 类中不可以有 \_\_\_\_\_ 的成员。
- 49、abstract 方法必须在 \_\_\_\_\_ 类中。
- 50、int 是 java 的 \_\_\_\_\_，Integer 是 java 为 int 提供的封装类。



- 51、如果源文件包含 `import` 语句，则该语句必须是 \_\_\_\_\_ 和 \_\_\_\_\_ 外的第一个语句行。
- 52、如果某源文件包含 `package` 语句，则该语句必须是 \_\_\_\_\_ 非空、非注释行。
- 53、接口中的方法必定是 \_\_\_\_\_ 方法，但可以不用关键字 \_\_\_\_\_ 修饰。
- 54、在接口的定义中，定义的变量的修饰符是 \_\_\_\_\_，定义的方法修饰符是 \_\_\_\_\_。
- 55、接口中的成员只有静态常量和 \_\_\_\_\_。
- 56、为 AB 类的一个无形参数无返回值的方法 `method` 书写方法头，使得使用类名 AB 作为前缀就可以调用它，该方法头的形式为 \_\_\_\_\_。
- 57、定义接口的关键字是 \_\_\_\_\_，定义最终类的关键字是 \_\_\_\_\_，定义抽象类的关键字是 \_\_\_\_\_。
- 58、表示数据或方法可以被同一包中的任何类或它的子类访问，即使子类在不同的包中也可以的修饰符是 \_\_\_\_\_。
- 59、表示数据或方法只能被本类访问的修饰符是 \_\_\_\_\_。
- 60、接口中方法的默认可见性修饰符是 \_\_\_\_\_。
- 61、为了区分类中重载的同名的不同方法，要求 \_\_\_\_\_。
- 62、当浏览器终止 Applet 时，\_\_\_\_\_函数被调用。
- 63、有 `abstract` 方法的类一定是 \_\_\_\_\_ 或 \_\_\_\_\_。
- 64、带 \_\_\_\_\_ 的类不能继承。
- 65、一个类中可以使用不包含在任何方法体中的静态代码块 \_\_\_\_\_ 当类被载入时 静态代码块被执行 且只被执行 \_\_\_\_\_ 次。
- 66、引用父类中的成员变量用 \_\_\_\_\_；引用当前的类或对象用 \_\_\_\_\_。
- 67、Vector 类是 JAVA 中专门负责处理对象元素有序存储和任意增删的类，因此，用 Vector 可以快速实现 JAVA 的 \_\_\_\_\_ 类。
- 68、栈（Stack）是限制仅在表的一端进行插入和删除运算的 \_\_\_\_\_。
- 69、LinkedList 数据结构是一种 \_\_\_\_\_ 结构，每一个对象除了数据本身外，还有两个引用，分别指向前一个元素和后一个元素，和数组的顺序存储结构（如、ArrayList）相比，插入和删除比较方便，但速度会慢一些。
- 70、是数据结构中链表的 java 实现，相对于 List 来说，LinkedList 最主要的功能方面的增强是可以在 List 的头部和尾部 \_\_\_\_\_，取得元素，直接提供了这些方法的现实 \_\_\_\_\_。所以它可以非常方便的实现我们数据结构中的常见 Stack(栈),queue(队列)
- LinkedList 类的构造方法

---

### 三、判断题

- 1、一个接口不可以继承其他接口。 ( )
- 2、一个类可以实现抽象类的所有方法，也可以只实现部分方法，但实现部分方法的类仍然是一个抽象类。 ( )
- 3、在实现接口的时候，要实现接口的所有方法。 ( )
- 4、抽象方法必须在抽象类中，所以抽象类中的方法都必须是抽象方法。 ( )
- 5、final 类中的属性和方法都必须被 final 修饰符修饰。 ( )
- 6、最终类不能派生子类，最终方法不能被覆盖。 ( )
- 7、子类要调用父类的方法，必须使用 super 关键字。 ( )
- 8、一个 Java 类可以有多个父类。 ( )
- 9、如果 p 是父类 Parent 的对象，而 c 是子类 Child 的对象，则语句 c = p 是正确的。 ( )
- 10、一个类如果实现了某个接口，那么它必须覆盖该接口中的所有方法。 ( )
- 11、接口是特殊的类，所以接口也可以继承，子接口将继承父接口的所有常量和抽象方法。 ( )
- 12、接口其实是一个特殊的 abstract class。 ( )
- 13、所谓数据封装就是将一组数据和与这一组数据有关的操作组装在一起，形成一个实体，这个实体也就是对象 ( )。
- 14、静态初始化器是构造方法加上 static 修饰符。 ( )
- 15、在定义一个类的时候，若想让类中的成员不能在类的外部访问，则必须在成员前面加上 private 修饰符。 ( )
- 16、类的静态成员所属于类所有，实例成员属于对象所有。 ( )
- 17、外部类能访问 static 变量 ( )
- 18、抽象类中不可以有非抽象的方法。 ( )
- 19、程序在编译通过以后，仍然可能出现运行时错误和逻辑错误。 ( )
- 20、可以在子类构造方法中通过 this 关键字调用基类的构造方法。 ( )
- 21、由于 Java 中类只支持单重继承，所以一个类只能继承一个抽象类和实现一个接口。 ( )
- 22、内部类始终可以访问外部类。 ( )
- 23、对于一个抽象类，改变它的属性和方法一定会改变它所有子类的属性和方法。 ( )
- 24、对象可以赋值，只要使用赋值号（等号）即可，相当于生成了一个各属性与赋值对象相同的新对象 ( )
- 25、有的类定义时可以不定义构造方法，所以构造方法不是必需的 ( )

- 26、类及其属性、方法可以同时有一个以上的修饰符来修饰。 ( )
- 27、抽象方法必须在抽象类中，所以抽象类中的方法都必须是抽象方法 ( )
- 28、超类对象的集合通常比它的任何一个子类对象集合大。 ( )
- 29、子类可以从多个超类中派生出来。 ( )
- 30、Java 中的类和接口都只支持单重继承。 ( )
- 31、Java 程序里，创建新的类对象用关键字 new，回收无用的类对象使用关键字 free。 ( )
- 32、由继承性可知，程序中子类拥有的成员数目一定大于等于父类拥有的成员数目。 ( )
- 33、Java 中 Object 类是最顶层的类。 ( )
- 34、可以在子类构造函数中通过 this 关键字调用基类的构造函数。 ( )
- 35、子类只能继续父类的非私有属性，不能继续父类的方法 ( )
- 36、子类能直接继续父类所有的非私有属性，也可通过接口继续父类的私有属性 ( )
- 37、抽象，封装，继承，多态是面向对象程序设计中得四个特点 ( )
- 38、面向对象程序设计方法的优点是、可重用性，可扩展性，可管理性 ( )
- 39、class 前的修饰符分为访问控制符和非访问控制符两大类。访问控制符包括 public 和 private。 ( )
- 40、final 类是最终类，是能有子类的类，abstract 和 final 不能同时修饰一个类，因为抽象类本身没有具体对象 ( )
- 41、创建对象的格式为：类名 对象名 =new 构造方法 (参数);注意前面是类名后面是构造方法名。 ( )
- 42、构造方法没有返回类型，也不能写 void，主要用于完成类对象的初始化工作，一般不能直接编程 ( )
- 43、如果 class 前面有 public 修饰符，则默认构造方法的前面也应该有 public 修饰符。 ( )
- 44、类中有 static 修饰的域或方法，可用类名或对象名访问。 ( )
- 45、类变量的最本质的特点是，他们是类的域，不属于任何一个类的具体对象实例，不是保存在某个对象实例的内存空间中，而是保存在类的内存区域的公共存储单元中。 ( )
- 46、用 static 修饰的变量或方法都为类成员，类成员可以用类名或实例名访问，实例成员只能用实例名来访问。 ( )
- 47、如果一个类中含有抽象方法，则此类必须为抽象类，如果抽象类的子类不为抽象类，则子类必须实现父类的所有抽象方法。 ( )
- 48、this 变量用在一个方法的内部，指向当前对象，当前对象指的是调用当前正在执行的方法的那个对象。super 变量是直接指向父类的构

造方法，用来引用父类中的变量和方法。由于他们指的是对象，所以不能通过它来引用类变量和类方法（ ）

49、访问修饰符、类中限定为 public 的成员可以被所有的类访问。（ ）

50、类中限定为 protected 的成员可以被这个类本身， 它的子类（包括同一个包中和不同包中的子类 ），以及同一个包中的其他类访问。（ ）

51、abstract 和 private , static , final , native 能并列修饰同一个方法。（ ）

52、abstract 类中不能有 private 修饰的域和方法（ ）

53、abstract 修饰符可修饰字段、方法和类（ ）

54、在类方法中可用 this 来调用本类的类方法（ ）

55、在类方法中调用本类的类方法时可直接调用（ ）

56、在类方法中只能调用本类中的类方法（ ）

57、在类方法中绝对不能调用实例方法（ ）

58、Java 面向对象语言容许单独的过程与函数存在（ ）

59、Java 面向对象语言容许单独的方法存在（ ）

60、Java 语言中的方法必定隶属于某一类 （对象），调用方法与过程或函数相同（ ）

61、Java 程序中的起始类名称必须与存放该类的文件名相同。（ ）

62、数组是一种对象（ ）

63、数组属于一种原生类（ ）

64、constructor 必须与 class 同名，但方法不能与 class 同名（ ）

65、Java 的类分为两大量、系统定义的类和用户自定义的类（ ）

66、类的静态属性和全局变量的概念完全一样，只是表达形式不同（ ）

67、类的成员至少有一个属性和一个方法（ ）

68、类是对象的实例化（ ）

69、一个类的构造方法可以有多个（ ）

70、子类不能继续父类的私有属性（ ）

71、构造方法只能由对象中的其它方法调用（ ）

72、构造方法可以用于给类的 private 实例变量赋值。（ ）

73、对象一经声明就可以立即使用。（ ）

74、在定义变量和方法时， 必须清楚地在其面前标上访问权限修饰符。（ ）

75、Java 程序中的参数传递都是把参数值传递给方法定义中的参数。（ ）

76、在类定义中，成员访问权限修饰符不能多次出现。（ ）

77、new 操作符动态地为对象按其指定的类型分配内存，并返回该类型的一个引用。（ ）

78、类的方法通常设为 public，而类的实例变量一般也设为 public。

- ( )
- 79、构造方法在创建对象时被调用。 ( )
- 80、通过点运算符与类对象的引用相连， 可以访问此类的成员。 ( )
- 81、声明为 `protected` 的类成员只能被此类中的方法访问。 ( )
- 82、同一个类的对象使用不同的内存段，但静态成员共享相同的内存空间。( )
- 83、类的成员变量可以放在类体的任意位置。 ( )
- 84、声明为 `protected` 的类成员可以被此类作用域中的任何对象访问。( )
- 85、没有返回值的方法可以用 `void` 来表示，也可以不加。 ( )
- 86、不需要定义类，就能创建对象。 ( )
- 87、在子类中可以用完全相同的 `signature` 重新定义一个超类方法，这称为方法的覆盖  
( `override` )。( )
- 88、`final` 修饰的方法是不能被子类覆盖的，只能被覆盖。 ( )
- 89、声明为 `final` 的类不能是超类。 ( )
- 90、当一个超类成员不适合该子类时，子类会以恰当的方式重新定义它。( )
- 91、对子类实例化对象需要调用超类的构造函数进行初始化工作， 完  
成对子类中的超类实例变量 的初始化，则子类可以继承超类的构造函数。( )
- 92、在类的定义中如果不明确给出超类，那么默认超类就是 `Object` 类。( )
- 93、`protected` 类型的实例变量或方法只能由声明该变量或方法的类本身及其子类访问。 ( )
- 94、Java 的每个子类的构造函数都要显式的调用它的直接超类的构造函数。( )
- 95、抽象方法不能是静态方法和最终方法。 ( )
- 96、抽象方法只有方法头的声明，而用分号来替代方法体，没有大括号。( )
- 97、如果存储空间十分宝贵，应使用 `Vector` 类的 `trimToSize` 方法将 `Vector` 调整到准确的大小。 ( )
- 98、Java 的 `Vector` 类提供了能自动调整大小的类似数组的数据结构。( )
- 99、创建 `Vector` 对象时构造方法给定的是其中可以包容的元素个数，使用中应注意不能超越这个数值。 ( )
- 100、一个 `Applet` 编译后的类名是 `Test.class`，运行此小程序的命令是 `Java Test` ( )
- 101、接口中的所有方法都没有被实现 ( )
- 102、实现一个接口，则在类中一定要实现接口中的所有方法。

- ( )
- 103、类体中 `private` 修饰的变量在本类中能访问，类生成的对象也能访问。( )
- 104、抽象类不能实例化 ( )
- 105、一个方法最多能有一个 `return` 语句 ( )
- 106、在 Java 程序中，通过类的定义只能实现单重继承 ( )
- 107、类 A 和类 B 位于同一个包中，则除了私有成员，类 A 可以访问类 B 的所有其他成员 ( )
- 108、一个类中，只能拥有一个构造器方法 ( )
- 109、可以用 `new` 来创建一个类的实例，即 "对象"。( )
- 110、类的 `public` 类型的成员变量不可以被继承。( )
- 111、子类可以继承父类所有的成员变量及成员函数。( )
- 112、Java 类中不能存在同名的两个成员函数。( )
- 113、对象是类的一个实例。( )
- 114、类的 `public` 类型的成员变量不可以被继承。( )
- 115、用 `abstract` 修饰的类必须被继承。( )

#### 四、编程题

- 设计一个类 `Triangle`，从键盘里面输入三个数字作为三角形的三个边。设计相应的方法判断三角形是否成立；判断三角形的类型（等边三角形、等腰三角形和任意三角形）；与计算三角形的面积和周长。面积的计算公式为、 $s=(a+b+c)/2$ ; `area=Math.sqrt(s*(s-a)*(s-b)*(s-c))`;  
设计一个主类，调用 `Triangle` 类中相应的方法，并打印出相应的结果。
- 通过一个点类 (`Point`)，具有自己的坐标位置 (`x,y`)，和显示本类信息的 `show()` 方法；  
一个圆类 (`Circle`) 由点类派生，增加一个新的属性半径 (`rad`)，一个计算圆面积的 `area()` 方法，和一个同样显示本类信息的 `show()` 方法；一个圆柱体类 (`Cylinder`) 由圆类派生，增加一个属性高度 (`height`)，一个计算圆柱形体积的方法 `vol()`，一个计算圆柱形表面积的 `area()` 方法，和一个同样显示本类信息的 `show()` 方法。建立一个主类，调用相关类的方法，并打印出相应的信息。
- 定义一个名为 `Rectangle` 表示长方形的类，它包含 `double` 类型的长 `length` 和宽 `width` 两个私有成员变量，计算长方形面积的成员方法 `area()`，分别对数据成员添加 `setter` 和 `getter` 方法，在主类 `RectangleDemo` 的 `main()` 中，通过 `setter` 和 `getter` 方法来访问 `Rectangle` 类对象的数据成员
- 定义一个名为 `hhe` 表示长方体的类，它包含 `int` 类型的长 `length` 和宽 `width` 和高 `high`，计算长方体的体积的成员方法 `tiJi()`，分别对数据成员添加 `setter` 和 `getter` 方法，在主类 `qiuZhi` 的 `main()` 中，通过 `setter` 和 `getter` 方法来访问 `hhe` 类对象的数据成员
- 设计一个 `people` 类，它包含 `name,sex,age,high` 属性，有相应的构

- 造方法，分别对数据成员添加 setter 和 getter 方法，在主类 xianShi 的 main() 中，通过 setter 和 getter 方法来访问 people 类对象的数据成员
- 6、编写一个方法，接收一个整数作为参数，此方法会找出其最大值、最小值和平均值。
- 7、编写一个 Person，该类包括属性耳朵、眼睛、鼻子、嘴巴、手和腿。包括的方法是、可以听见，可以闻见，可以咬，可以拿东西，可以走路。
- 8、定义一个 changFangTi 的父类，里面有 chang, kuan 两个属性。和相应的构造方法和 get、set 方法。定义个 changFangTi2 的子类继承 changFangti，里面有一个 gao 属性，有相信的 get 和 set 方法。和求体积的 tiJi ( ) 方法。在 main ( ) 方法中调用父类的方法。并求长方体的体积。
- 9、定义一个 bus 类，里面有 speed 和 number 属性。以及构造方法和 get、set 方法。定义一个 bus2 的子类。并继承父类的方法。计算 bus 的速度和人数。
- 10、定义一个动物类，Animal( 动物类型 type 成员变量，一个发声的方法 sound()); 定义一个基于动物的狗类 Dog ( 覆盖父类的发声方法 ) ；定义一个基于动物的猫类 Cat( 覆盖父类的发声方法 ) ；定义 AnimalDemo 类，声明一个 Animal 对象，分别创建一个 Dog, Cat, 类的对象，然后通过访问 Animal 对象来访问 Dog Cat 类对象的 sound() 方法。
- 11、创建两个 Student 类的对象，比较二者年龄，输出其中年龄大的学生的姓名。
- 12、编程创建一个 Person 类，在其中包括域（姓名，性别，年龄）和方法（获得姓名，性别，修改年龄）。将 Person 类作为父类，创建子类 Student 和 Teacher。编写应用程序，创建 Student 和 Teacher 对象。
- 13、定义一个体育活动类 (Sports) 作为基类，它有一个进行活动的方法 play。足球 (Football) 和篮球 (Basketball) 都是体育活动类的衍生类。
- 14、定义一个类 BOX，计算长方体的体积、长、宽、高由键盘输入。
- 15、自定义方法，求三个数的最大值（要求、从键盘上输入）
- 16、定义一个抽象类，他的属性有 length 和 height，它的一个构造方法是设置子性质，另一个轴向方法是 area() 用于求面积，定义两个类分别是长方形 Rect 和三角形 Triangle 求他们的面积，设计一个主类，调用相应的方法。
- 17、定义一个接口 M，其中有两个抽象方法，max()、min()，在 P 类中实现和继承，同时实现抽象方法 max(), min() 求三个数中的最大值和最小值
- 18、设计一个接口 <interface> 照顾宠物 "喂食 ()" "陪他玩 ()" 工人 .. 农民 .... 干部 ... 来实现 ..
- 19、设计一个公共汽车类 bus，只要求有速度和人数两个属性，和一个不带参数的构建器，具有加速度，减速，上人，下人，停车，启动等



方法。创建一个对象，车上有 35人，开始发车，中间下了 13人，又上了25人，请问到站还有多少人 用程序实现。

20、设计一个表示二维平面上的圆的类 Circle, 还包含有表示圆半径的 protected 类型的成员变量 获取和设置 radius 的 public 方法, 计算圆面积 的 public 方法 设计一个表示圆柱体的类 Cylinder, 它继承自类 Circle, 还包含有表示圆柱体高的 protected 类型的成员变量 height, 获取和设置 height 值的 public 方法, 计算圆柱体体积的 public 方法. 建立若干个 Cylinder 对象, 输出其半径, 高及其体积的值. 要求: A. 每个类包含无参数和有参数的构造方法. 构造方法用于对成员变量初始化,

21、设计一个抽象类，并比较三个数的大小。

22、利用多态，设计一个程序，并计算出圆的面积、正方体体积、长方体体积、圆柱体、圆锥体体积

23、定义一个 Compute 接口，有抽象方法 add()、subtract()、multiply() 和 divide(); 定义实现类 Computer. 创建 Compute 对象完成加、减、乘、除方法的调用

24、定义一个抽象 Myintertace, 其中有两个抽象方法 max()、min(). 在 Operation 类中实现该抽象，同时实现抽象方法 max()、min() 并且要求三个数的最大最小值。

25、利用外部类，求圆的面积。

26、定义一个 ssanjian 类，用于实现扑克牌的正常发放。里面有 face 和 suit 属性。并且有构造方法和相应的 get、set 方法和一个 toString 方法。第一个 ssanjian2 类作为它的子类，并继承父类的方法，并实现此方法。

27、定义一个计算 xn 的类，然后用该类求 10\*10\*10 的值。

28、设计一个 Birthday 类，其成员变量有、year, month, day; 提供构造方法、输出 Birthday 对象值的方法和计算年龄的方法。编写程序测试这个类。(引入 java.util.Calendar 包，取得当年份 Calendar.getInstance().get(Calendar.YEAR))

29、定义一个类 Rectangle 代表矩形，其中包括计算面积的方法。再定义一个它的子类 Square[代表正方形]，其中也包含计算面积的方法。编写一程序，测试新建子类对象的属性和方法。

30、实作一个时间类 MyTime，包含一个构建器，3 个私有成员变量 hour, minute, second 以及它们相应的 get 和 set 方法，再加上一个转换成字符串的方法 toString。

31、实作一个日期类 MyDate，包含一个构建器，3 个私有成员变量 year, month, day 以及它们相应的 get 和 set 方法，再加上一个转换成字符串的方法 toString。

32、实作一个 FullTime 类，包含一个 MyTime 类型的是有成员变量 time 以及与之相对应的 get 与 set 方法，一个 MyDate 类型的成员变量 date 以及与之相对应的 get 与 set 方法，一个构建器，再加上一个转换成字



字符串的方法 toString。

33、假设当前时间为 2010 年 3 月 10 日 9 点 30 分 45 秒，编写一个程序，用 FullTime 类设置时间为以上时间，并打印。

34、定义一个体育活动类 (Sports)作为基类，它有一个进行活动的方法 play(方法名命名规范、 J2EE 方向、小写字母开头 . NET 方向:大写字母开头)。

足球 (Football) 和篮球 (Basketball) 都是体育活动类的衍生类。

请在测试类 (Test)中编写一个方法 howToPlay(Sports sp)，该方法要求传递一个 Sports 类型的参数。该方法的作用是、

1) 当传入的实例类型为 Football 时，控制台上应打印、足球是用脚踢的！

2) 当传入的实例类型为 Basketball 时，控制台上应打印、篮球是用手打的！

在 main 方法中调用 howToPlay() 验证你的代码是对的。

35、乐器 (Instrument) 分为、钢琴 (Piano)、小提琴 (Violin)

- \* 各种乐器的弹奏 (play) 方法各不相同。

- \* 编写一个测试类 InstrumentTest，要求、

- \* 编写方法 testPlay，对各种乐器进行弹奏测试。要依据乐器的不同，进行相应的弹奏。

- \* 在 main 方法中进行测试

36、声明一个包括各种车辆的抽象类，通过继承机制定义公共汽车和自行车两个类，构造对应的方法区分公共汽车和自行车，并统计各自的数量，完成公共汽车和自行车的实例化。

37、编写学生类 (class student)继承人类 (class preson)的程序，要求父类的构造函数为 (preson(string name),子类的构造函数要调用父类的构造函数.p()方法用于输出该对象的信息 (如 The person's name is Zhangheng),学生类有 study()方法，并分别创建一个对象并调用各个方法

38、设计一个学生类 Student，其数据成员有 name(姓名)、age(年龄)和 degree(学位)。由 Student 类派生出本科生类 Undergraduate 和研究生类 Graduate，本科生类 Undergraduate 增加成员 specialty(专业)，研究生类增加成员 direction(研究方向)。每个类都有 show()方法，用于输出数据成员信息

39、编写学生类 (class student)继承人类 (class preson)的程序，要求父类的构造函数为 (preson(string name),子类的构造函数要调用父类的构造函数.p()方法用于输出该对象的信息 (如 The person's name is Zhangheng),学生类有 study()方法，并分别创建一个对象并调用各个方法

40、在父类 Base 和子类 Sub 中都定义了成员变量 var 及成员方法 method()。在 Sub 类中，可通过 super.var和 super.method()来访问 Base 类的成员变量 var 及成员方法 method()。

41、创建一个类 Queue 代表队列 (其特点为、先进先出)，添加方法

add(Object obj) 以及 get(), 并添加 main() 方法进行效果验证。(要求、使用 LinkedList 实现队列、在向 LinkedList 中添加时, 使用 addLast() 方法, 在从 LinkedList 中取出时, 使用 removeFirst() 方法)

42、定义了一个学生的类 Student, 包括成员变量 “学号”、“班号”、“姓名”、“性别”、“年龄”, 成员方法: “获得学号”、“获得班号”、“获得性别”、“获得姓名”、“获得年龄”。请为 Student 类定义构造函数初始化所有的成员变量。增加一个方法 public String toString() 把 Student 类对象的所有信息组合成一个字符串。

43、构造函数是否可以被重载? 试举例。

44、构造一个类来描述屏幕上的一个点, 该类的构成包括点的 x 和 y 两个坐标, 以及一些对点 进行的操作, 包括: 取得点的坐标值, 对点的坐标进行赋值, 编写应用程序生成该类的对象并 对其进行操作。

```
origin.getPoint(); } }
```

45、编程创建一个 Box 类, 在其中定义三个变量表示一个立方体的长、宽、高, 再定义一个方法 setDemo 来对这 3 个变量进行初始化, 然后定义一个方法来求立方体的体积。 创建一个对象, 求给定尺寸的立方体的体积。

46、定义一个类实现银行帐户的概念, 包括的变量有 “帐号” 和 “存款余额”, 包括的方法有 “存款”、“取款” 和 “查询余额”。定义主类, 创建帐户类的对象, 并完成相应操作。

47、编写 Student 类的测试程序 程序要求实现: 学生信息 (学号、姓名性别、年龄、平均分、最高分、最低分) 在程序中设置, 要求通过运行程序, 在屏幕中输出如下的结果: 例如: 学号: 200501 姓名: 李丽 性别: 女 年龄: 20 平均分: 90.0 最高分: 95 分 最低分: 87。

48、定义 Point、Circle、Cylinder 三个类, 三个类都拥有 x,y 属性, 而 Circle 又增加了属性 radius, Cylinder 类又增加了属性 height。都有获得 x,y 坐标, 设置 x,y 坐标, 求面积、体积的方法。

49. 请编写程序由三个类定义组成: 类 A 中只定义了一个类型为 String 的对象 x 和带有一个参数的构造函数 (用来初始化 x); 类 B 是类 A 的子类, 其中定义了一个 int 类型的变量 x 和 29 一个带两个参数的构造函数 (用来为父类中的对象 x 和类 B 中的变量 x 做初始化), 还定义了一个方法 myPrint(), 功能为输出父类 A 中的 x 和其子类 B 中的 x 的值; 测试类中定义了 B 类的对象 b, 请通过创建 b 为上述两个 x 赋值 (类 A 中的 x 的值为 “这是父类的 x.”, 类 B 中的 x 的值为 68, 显示两个 x 的值。

50. 请编写程序, 它由三个类定义组成, 类名分别为 A、B 和 Applet1: 类 A 中定义了一个类型为 float 的属性 f 和一个类型为 byte 的私有属性 k; 类 A 还定义了一个带有一个参数的构造函数 (用于初始化 f), 和两个公共方法 setk() 和 getk(), 其中 setk() 的功能为给私

有属性 `k` 赋值, `getk()` 的功能为返回私有属性 `k` 的值。类 `B` 是类 `A` 的子类, 其中仅定义了一个类型为 `boolean` 的属性 `b` 和一个带两个参数的构造函数 (用来为父类中的变量 `f` 和本类的变量 `b` 做初始化)。Test 类中定义了一个初值为 123 的 `int` 型变量 `n` 和一个 `B` 类的对象 `x`, 请通过创建 `x` 为上述 `f` 和 `b` 赋值 (`f` 的值为 68.963, `b` 的值为 `true`); 在 `paint()` 方法中, 给 `A` 类的私有属性 `k` 赋值 66, 然后显示 `f`, `k`, `b`, `n` 的值。

51. 设有 1 个 `Professor` 类, 继承自 `Human` 类, 该类是教师的蓝图, 它除继承相应的属性和方法外, 还具有下面的成员属性:

讲授课程属性: `private String course`

教学效果属性: `private int result` 该属性的取值为 0—3, 1 表示效果良好, 2 表示可以

接受, 3 表示效果不佳, 0 表示未予评价 成员方法: `public String getDetails()` 将姓名、性别、年龄和讲授课程属性以字符串方式返回。

`public boolean set_reslut(int x)` 该方法用于设置教学效果, 必须考虑设置是否合理 (若不是 0-3 之间, 则不允许设置) 构造方法: 用于

设置姓名, 性别和年龄和讲授课程。 `public Professor(String name, boolean sex, int age, String course)`

52. 定义一个接口或多个接口, 并用一个类实现。

53. 定义一个抽象类, 并定义它的两个非抽象子类。

54. 定义接口 `A`, 接口内有两个方法: `method1(int x)` 和 `method2(int x, int y)`, 它们的返回值类型均为 `int`。请编写一个 `Applet`, 并使其实现接口 `A`: 令方法 `method1(int x)` 的功能是求 5 的 `x` 次方、方法 `method2(int x, int y)` 的功能是求两参数的最大值。在 `paint()` 方法中调用这两个方法, 显示 `method1(2)` 和 `method2(2, 8)` 的结果。

55. 定义名为 `VolumeArea` 的抽象类, 在其中定义圆周率  $\pi$  的值为 3.14159, 并定义两个抽象方法 `volume(double r)` 和 `area(double r)`, 它们的返回值类型均为 `float`。再定义以类 `VolumeArea` 为父类的子类 `VolumeAndArea`, 在该子类中实现父类中的抽象方法: 方法 `volume(double r)` 的功能是求半径为 `r` 的球的体积 (公式为:  $4 * \text{Mth.PI} * r * r * r / 3$ ); 方法 `area(double r)` 的功能是求半径为 `r` 的圆的面积 (公式为:  $\text{Mth.PI} * r * r$ )。请编写一个 `Application`, 在其主类中定义一个 `VolumeAndArea` 类的对象 `x`, 通过对象 `x` 求半径为 `r` (`r` 的值由命令行给定) 的球的体积及该球最大切面的圆的面积, 并输出计算结果。

56. 定义一个图形的抽象类, 具有求面积和画图形的方法, 再定义点、线、圆的类继承这个抽象类实现它的方法。

57. 请定义一个名为 `Card` 的扑克牌类, 该类有两个 `private` 访问权限的字符串变量 `face` 和 `suit`, 分别描述一张牌的牌面值和花色。定义 `Card` 类中的 `public` 访问权限的构造方法, 为类中的变量赋值; 定义 `protected` 访问权限的方法 `getFace()`, 得到扑克牌的牌面值; 定义 `protected` 访问

权限的方法 `getSuit()`，得到扑克牌的花色；定义方法 `toString()`，返回表示扑克牌的花色和牌面值字符串。

若应用程序的 `main` 方法中，定义字符串数组 `f` 和 `s`；分别表示扑克牌的牌面值和花色；定义 52 个元素的 `Card` 类型数组 `deck`，用来存放 4 个花色的 52 张牌，如下所示、  
`string f[]={ "A", "2", .....? string }  
s[]={ 黑桃 " 红桃 " 梅花 " 方块 " } Card[] deck=new Card[52];`  
使用 `Card` 类的构造方法给 `deck` 数组的 52 张牌赋值，要求数组中先存放黑桃花色的 A、2、3.....K；然后是红桃花色的 A、2、3.....K；梅花花色的 A、2、3.....K；方块花色的 A、2、3.....K；把所有的牌信息打印出来。

再编写模拟洗牌的程序段，即把数组 `deck` 中的扑克牌随机打乱存放程序，分发给 4 人。

58、在 CS 枪战游戏中，有 Ak47 和 USP 两种枪，每种枪弹夹的子弹数和发射子弹数都不一样。设计一个枪战游戏程序，定义一个 `Gun` 类，该类具有枪名称的属性 `name`，子弹数量的名称 `num`，枪手的编号 `id` 和记录当前枪手数量的属性 `count`(静态)，具有 `num` 和 `name` 属性的 `getter` 和 `setter` 方法和发射的方法 `fire()`；一个构造方法为属性赋初始值，  
`Gun.count++;// 计数器累加 this.id = Gun.count ;name = "Gun";num = 12;`  
设计一个 Ak47 类，由 `Gun` 类继承而来，每发射一次消耗 3 发子弹，每装一个弹夹，上 30 发子弹；设计一个 USP 类，由 `Gun` 类继承而来，每发射一次消耗 1 发子弹，每装一个弹夹，上 12 发子弹；设计一个主类，分别创建不同的 Ak47 和 USP 对象，对象调用类里面的方法，装子弹，联系发射 10 次，显示枪里面还有多少子弹？

59、编程实现矩形类，其中应包括计算矩形周长和面积的方法。

(1) 使用矩形类，编程统计若干块土地的相关信息。由用户输入每块土地的长与宽（自己确定），程序将计算其面积并显示出来。

(2) 从矩形类派生一个子类、长方体类。对长方体类的操作是求表面积和体积。编程检查、运行所编写的长方体类。

60、编写一个类 `Book`，代表教材、

(1) 具有属性、名称 (`title`) 页数 (`pageNum`)，其中页数不能少于 200 页，否则输出错误信息，并赋予默认值 200；

(2) 具有方法、`detail`，用来在控制台输出每本教材的名称和页数；

(3) 具有带参数的构造方法、用来完成对象的初始化工作，并在构造方法中完成对页数的最小值限制。

## 第七章 数据库的连接

## 一、选择题

1、使用 Class 类的 ( ) 方法可载入一个数据库驱动

A 、 getClass                      B、      getName                      C 、      forClass                      D 、  
forName

2、( )惟一的表示了关系型数据库中的一个值集

A、记录      B、域      C、行      D、主键

3、数据库中的主键用于（ ）

A、创建数据库中的行      B、标识数据库中的列      C、区分表中的不同行  
D、以上答案都不对

#### 4、Statement 对象实现 ( )

A 、 数据库的连接  
库驱动程序

C、执行检索或修改数据库内数据的 SQL D、 以上答案都不对

## 5、executeQuery 方法 ( )

A、 检索数据库中的信息

C、 建立同数据库之间的连接  
之间的连接

6、关系型数据库是由一个或多个( )组成

A、 图  
都不对

7、是由包含惟一的（ ）来区分不同的行

A、 主键  
不对

8、导入（ ）包能让开发人员使用针对数据库操作的类及其接口

A、 java.text      B、 java.db      C、 java.sql      D、 以上答案都不对

9、如果 DriverManager 无法实现同数据库之间的链接，那么 getConnection 方法会抛出一个（ ）异常

A、 SQLException                      B、 ClassNotFoundException

C 、 DatebaseNotFoundException                      D、 以上答案都不对

10、接口 Statement 中的 ( ) 方法用于从数据库标准提取数据

A、 executeQuery      B、 select      C、 executeUpdate      D、 以上答案都不对

11、 Statement的（ ）方法用于修改数据表中的数据

A、 executeQuery      B、 modifyTable      C、 executeUpdate      D、 以上  
答案都不对

12、在 select 语句中， select 关键字之后应跟（ ）

- A、 表名      B、 列名或 \*      C、 数据库名      D、 行所满足的标准
- 13、 SQL 中的 (      ) 语句可修改为数据库中的信息
- A、 select      B、 modify      C、 change      D、 update
- 14、 (      ) 是一个有组织的数据集合
- A、 行      B、 数据库      C、 主键      D、 以上答案都不对
- 15、 一旦数据录入数据库， (      )
- A、 拥有适当权限的用户和应用程序可以进行访问      B、 用户必须等待系统重新启动才能访问数据
- C、 使用数据库的应用程序必须手动更新      D、 以上答案都不对
- 16、 在关系型数据库中，数据组织在 (      ) 中
- A、 类      B、 表      C、 结果集      D、 以上答案都不对
- 17、 在编写访问数据库的 Java 程序时， Connection 对象的作用是 (      )
- A、 用来表示与数据库的连接      B、 存储查询结果
- C、 在指定的连接中处理 SQL 语句      D、 建立与数据库连接
- 18、 在编写访问数据库的 Java 程序时， Statement 对象的作用是 (      )
- A、 加载连接数据库的驱动      B、 存储查询结果
- C、 在指定的连接中处理 SQL 语句      D、 建立与数据库连接
- 19、 在编写访问数据库的 Java 程序时， ResultSet 对象的作用是 (      )
- A、 加载连接数据库的驱动      B、 存储查询结果
- C、 在指定的连接中处理 SQL 语句      D、 建立与数据库连接
- 20、 我们在用 JDBC 连接 SQLServer 2000 数据库时，需要加载相应的驱动，以下选项中，不属于我们要加载的驱动包是 (      )
- A、 msbase.jar      B、 mssqlserver.jar      C、 msutil.jar      D、 tools.jar
- 21、 数据库的表由 (      ) 组成
- A、 数组      B、 数据      C、 行和列      D、 方法
- 22、 SQL 关键字 (      ) 的后面是在查询中选择记录的条件。
- A、 WHERE      B、 ORDER BY      C、 INNER JOIN      D、 ON
- 23、 SQL 关键字 (      ) 指定记录的排序顺序。
- A、 WHERE      B、 ORDER BY      C、 INNER JOIN      D、 ON
- 24、 SQL 关键字 (      ) 用于合并两个或多个表。
- A、 WHERE      B、 ORDER BY      C、 INNER JOIN      D、 ON
- 25、 (      ) 同时是另一个表中的主关键字
- A、 HERE      B、 ORDER BY      C、 INNER JOIN      D、 外关键字
- 26、 一个 (      ) 对象用于向数据库提交查询。

A、 ResultSet                      B、 Connection                      C、 Statement                      D、 Applet

27、不是主要数据库模型的是 (                      )

A、 层次模型                      B、 网状模型                      C、 关系模型                      D、 结构模型

28、下面说服错误的是 (                      )

A、 当某个字段被指定为主关键字字段时，不对这个字段赋值将违反实现完整性规则，是错误的。

B、 当某个字段被指定为主关键字时，都个记录在这个字段使用相同的值是错误的。

C、 在不区分大小写的系统上， SQL 关键字按惯例全部使用大写字母。

D、 使用选择标准减少从数据库返回的记录数不能提高性能

29、 Java 通过 (                      ) 接口的方法提供事物处理

A、 ResultSet                      B、 Connections                      C、 Connection                      D、 Statement

30.下面语句书写有误的是 (                      )

A、 SELECT \* FROM 表名 WHERE 标准

B、 SELECT \* FROM 表名 ORDER BY 字段 ASC

C、 SELECT \* FROM 表名 WHERE 标准 ORDER BY 字段 ASC

D、 SELECT \* FROM 表 1 INNER JOIN 表 2

31、我们在用 JDBC 连接 SQLServer 2000 数据库时，需要加载相应的驱动，以下选项中，加载驱动代码正确的是 (                      )

A、 class.forName("com.microsoft.jdbc.sqlserver.SQLServerDriver")

B、 Class.forName("com.microsoft.jdbc.sqlserver.SQLServerDriver")

C、 class.forName("com.microsoft.jdbc.sqlserver.sqlserverDriver")

D、 Class.forName("com.microsoft.jdbc.sqlserver.sqlserverDriver")

32、某程序利用网址参数创建 URL 对象 url，接着希望利用对象 url 获得 URLConnection 对象，则这个 url 对象要调用的方法是 (                      )

A、 connection()                      B、 URLConnection()                      C、 getConnection()

D、 openConnection()

## 二、填空题

1、 java 在连接 SqlServer 数据库时，需要导入三个驱动工具包是、 \_\_\_\_\_ ， \_\_\_\_\_ ， \_\_\_\_\_ 。

2、 在遍历数据表内容时，可以通过 \_\_\_\_\_ 方法把当前的指针自动向下移动一位。

3、 在向数据库发送 sql 语句时，要使用 \_\_\_\_\_ 类声明一个发送 sql 语句对象。

4、 连接 SqlServer 时，所用的协议时 \_\_\_\_\_。

5、 在 java 源文件中进行数据库操作时，在文件头要导入 \_\_\_\_\_ 包。

6、 在连接数据库的时候，为了获取本机的 IP 地址，常用的代码是 \_\_\_\_\_ 或者 \_\_\_\_\_。

7、 在获取数据表内容的时候，将指针移动到此 \_\_\_\_\_ ResultSet 对象的指



定行编号的方法是\_\_\_\_\_。

8、在获取数据表内容的时候，将指针移动到此\_\_\_\_\_ ResultSet 对象的第一行的方法是\_\_\_\_\_。

9、在获取数据表内容的时候，获取当前行编号的方法是\_\_\_\_\_。

10、在获取数据表内容的时候，判断指针是否位于此\_\_\_\_\_ ResultSet 对象的第一行的方法是\_\_\_\_\_，判断指针是否位于此\_\_\_\_\_ ResultSet 对象的最后一行的方法是\_\_\_\_\_。

### 三、判断题

1、最流行的数据库查询语言是 SQL。( )

2、ava 将表可以作为 connection 对象来操作。( )

3、关键字唯一地表示表中的每个记录。( )

4、数据库是集中控制的数据的有机集合。( )

5、Applet 包中含有 Java 中操作关系数据库的类和接口。( )

6、sun .jdbc .odbc 类代表了 JDBC 到 ODBC 数据库驱动程序。( )

7、共享是数据库系统的一个关注焦点。( )

8、数据库系统的主要特点之一是数据的独立性，既应用程序无需关心数据的存储和访问。( )

9、宿主语言通常包括一种数据库子语言，用于处理数据库对象和操作的细节。( )

10、数据库的表一般都有主关键字，但主关键字并不是必需的。( )

### 四、编程题

1、写一个应用程序，显示一个 test.mdb 中表 test 的字段。

2、编写一个应用程序，显示一个 test.mdb 中表 test 的所有数据。

3、编写一个应用程序，为 test.mdb 中表 test 添加一条记录。

4、编写一个应用程序，为 test.mdb 中表 test 删除 id 为 100 的一条记录。

5、编写一个应用程序，为 test.mdb 中表 test 修改 id 为 101 的一条记录的 name 值为 "张 三"。

6、假设 SqlServer 数据库中有 test 数据库中的 student 表，该表中的字段为 id,name，试编写一个程序，读出表中所有记录。

7、假设某商店有一个商品库存系统。其中用到一张表，表名为\_\_\_\_\_为 phoneinfo(有商品名称和商品库存量两个字段，其中商品名称为主键，两个字段皆不为空)。设计一个程序用来处理添加更新库存。若用户输入的商品不存在，则往数据库中添加新产品；若用户输入的商品已经存在，则在数据库中商品原有的库存基础上增加库存。不管是增加还是更新，最后都要显示商品全部信息。



## 第八章 多线程

### 一、填空题

- 1、Java 语言是一种简单的、 面向对象的、 分布的、 解释的、 \_\_\_\_\_、安全的、 独立于平台的、 可移植可扩展的、 高性能的、 \_\_\_\_\_ 的动态程序设计语言。
- 2、线程体是指 \_\_\_\_\_ 方法中的执行代码， 该方法是程序的入口点。
- 3、创建一个多线程程序时可以通过实现类 \_\_\_\_\_， 也可以通过实现接口 \_\_\_\_\_。
- 4、线程调度时要根据线程的优先级进行， \_\_\_\_\_ 线程的优先级可分为 10 种， 其中 \_\_\_\_\_ 代表优先级最高， \_\_\_\_\_ 代表优先级最低。
- 5、在进行线程同步时常使用 \_\_\_\_\_、 \_\_\_\_\_、 \_\_\_\_\_ 等关键字或方法。
- 6、每个小应用程序在运行时都会顺序地执行方法、 \_\_\_\_\_、 \_\_\_\_\_、 \_\_\_\_\_、 \_\_\_\_\_ 和 \_\_\_\_\_。
- 7、多线程是 JAVA 程序的 \_\_\_\_\_ 机制， 它能同步共享数据、 处理不同事件。
- 8、\_\_\_\_\_ 本身不能独立运行， 必须在进程中执行， 使用进程的地址空间。
- 9、如果其他线程企图访问一个处于不可用状态的对象， \_\_\_\_\_ 该对象将不能正确响应从而产生无法预料的结果， 这种情况就是一种 \_\_\_\_\_。
- 10、\_\_\_\_\_ 指得是程序运行中， 多个线程竞争共享资源时可能出现的一种系统状态， 每个线程都被阻塞， 都不会结束， 进入一种永久等待状态。
- 11、ReentrantLock 锁必须在 \_\_\_\_\_ 块中释放。
- 12、基于锁的多线程设计更可能引发 \_\_\_\_\_、 \_\_\_\_\_、 \_\_\_\_\_ 等情况， 令一些线程无法继续其进行。
- 13、多线程运行时分析工具 ( MTRAT ) 是由 IBM 开发的一个即高效又准确的 动态 分析工具， 它可以查找 出多 线程程序 中的 潜在的 和死锁。
- 14、指定线程阻塞多长时间的方法是 \_\_\_\_\_。
- 15、在 Java 语言中， 实现线程的同步执行要用到关键字 \_\_\_\_\_。
- 16、.Java 的线程调度策略是一种基于优先级 \_\_\_\_\_。
- 17、当使用 Thread t=new Thread(r) 创建一个线程时， 表达式、 r instanceof Thread 的值是 \_\_\_\_\_。
- 18、java 中的线程由 CPU， \_\_\_\_\_， \_\_\_\_\_ 等三部分构成。
- 19、FileInputStream 是 字节流 ;BufferedWriter 是 字符

流;OutputStream 是\_\_\_\_\_。

20、对 Java 对象读、写的过程被称为\_\_\_\_\_。

21、Swing 的事件处理机制包括\_\_\_\_\_、事件和事件处理者。

22、程序中实现多线程的方法有两种、继承\_\_\_\_\_类和实现 Runnable 接口。

23、在 Java 应用程序启动时，Java 运行系统为该应用程序创建了一个称为\_\_\_\_\_的线程组。

24、在程序运行阶段，要给图像框控件装载图片，应调用\_\_\_\_\_函数来完成

25、如果一个线程调用\_\_\_\_\_方法，将使线程进入休眠状态。

26、Java 中，可以通过建立\_\_\_\_\_类或其子类的实例创建和控制线程。

27、线程由于调用 sleep()方法进入阻塞状态，当睡眠结束时，该线程将进入\_\_\_\_\_状态。

28、Java 中，线程必须属于一个进程。线程是程序运行中的一个\_\_\_\_\_。

29、线程的\_\_\_\_\_方法只会使具有与当前线程相同优先级的线程有运行的机会。

30、当实现 Runnable 接口时，要实现的方法\_\_\_\_\_。

31、在 Java 程序中，主线程一般具有\_\_\_\_\_优先级。

32、实现线程交互的\_\_\_\_\_和\_\_\_\_\_方法在 Object 类中定义。

33、按照 Java 的线程模型，代码和\_\_\_\_\_构成了线程体。

34、在多线程程序设计中，如果采用继承 Thread 类的方法创建线程，则需要重写 Thread 类的\_\_\_\_\_方法。

35、线程在生命周期中要经历 5 种状态，分别是\_\_\_\_\_、\_\_\_\_\_、\_\_\_\_\_、阻塞状态和\_\_\_\_\_。

## 二、选择题

1、实现线程的创建有( )方式。

A、一种                      B、两种                      C、三种                      D、四种

2、Thread 类定义于下列哪一个包中？( )

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

3、下列那一个包给出了 Runnable 接口的定义？( )

A、java.util                  B、java.io                  C、java.lang                  D、java.sql

4、线程生命周期中正确的状态是( )

A、新建状态、运行状态和终止状态

B、新建状态、运行状态、阻塞状态和终止状态

C、新建状态、可运行状态、运行状态、阻塞状态和终止状态

D、新建状态、可运行状态、运行状态、恢复状态和终止

5、阅读下面编程题

```

public class Test 2 _____{
public static void main(String args[]){
Thread t=new test2();
t.start();}
public void run(){
sysetem.out.println("How are you");}
}

```

在程序下画线处填入正确选项是 ( )

- A、 implements Thread    B、 extends Runnable    C、 implements Runnable    D、 extends Thread

6、 阅读下面编程题

```

public class Test implements Runnable{
public static void main(String args[]){
_____};
t.start();
public void run(){
sysetem.out.println("Hello");}
}

```

在程序下画线处填入正确选项是 ( )

- A、 Test t=new Test()    B、 Thread t=new Thread()  
C、 Thread t=new Thread(new Test() )    D、 Test t=new Thread()

7、 下列特点中，哪个是 Java 虚拟机执行的特点之一？ ( )

- A、 字节代码    B、 多进程    C、 静态连接    D、 编译

8、 当浏览器重新返回 Applet 所在页面时，将调用 Applet 类的方法是 ( )

- A、 start()    B、 init()    C、 stop()    D、 destroy()

9、 当使用 SomeThread t=new SomeThread()创建一个线程时，下列叙述中正确的是 ( )

- A、 SomeThread 类是包含 run() 方法的任意 java 类  
B、 SomeThread 类一定要实现 Runnable 接口  
C、 SomeThread 类是 Thread 类的子类  
D、 SomeThread 类是 Thread 类的子类并且要实现 Runnable 接口

10、 下列叙述，错误的是 ( )

- A、 Java 提供了丰富的类库    B、 Java 最大极限地利用网络资料  
C、 Java 支持多线程    D、 Java 不支持 TCP/IP 协议

11、 如果线程正处于阻塞状态，不能够使线程直接进入可运行状态的情况是 ( )

- A、 sleep()方法的时间到    B、 获得了对象的锁  
C、 线程在调用 t.join() 方法后，线程 t 结束    D、 wait() 方法结束

12、 Java 语言中，负责并发管理的机制是 ( )

A、垃圾回收                      B、虚拟机                      C、代码安全                      D、多线程

13、下列描述中，错误的是（                      ）

A、Java 要求编程者管理内存                      B、Java 的安全性体现在多个层次上

C、Applet 要求在支持 Java 的浏览器上运行                      D、Java 有多线程机制

14、Thread 类中能运行线程体的方法是（                      ）

A、start()                      B、resume()                      C、init()                      D、run()

15、可以使当前同级线程重新获得运行机会的方法是（                      ）

A、sleep( )                      B、Join( )                      C、yield( )                      D、interrupt( )

16、下列关于 Java 语言中线程的叙述找中，正确的是和一（                      ）

A、线程是由代码，数据，内核状态和一组寄存器组成

B、线程间的数据是不共享

C、用户只能通过创建 Thread 类的实例建立和控制自己的线程

D、因多线程并发执行而引起的执行顺序的不确定性可能造成执行结果不确定

17、阅读下面程序

```
public class Try extends Thread{
public static void main (String args[]){
Try t =new Try();
t.start();}
public void run(int j){
int i=0;
while(i<5){
system.out.println(" 祝你成功！ ");
i++;}}}
```

该程序要求打印 5 行“祝你成功！”，必须改正程序中的某行代码，程序才能完成。选择正确的修改是（                      ）

A、将第一行的 extends Thread 改为 implements Runnable

B、将第三行的 new Try() 改为 new Thread ()

C、将第四行 t.start() 改为 start(t)

D、将第七行 public void run(int j) 改为 public void run()

18、线程在生命周期中要经历 5 种状态。如果线程当前是新建状态，则它可到达的下一个状态是（                      ）

A、运行状态                      B、阻塞状态                      C、可运行状态                      D、终止状态

19、下列关于 Java 多线程并发控制机制的叙述中，错误的是（                      ）

A、Java 中对共享数据操作的并发控制是采用加锁技术

B、线程之间的交互，提倡采用 suspend( )/resume( )方法

C、共享数据的访问权限都必须定义为 private

D、Java 中没有提供检测与避免死锁的专门机制，但应用程序员可以采用某些策略防止死锁的发生

20、Java 程序的并发机制是 ( )

A、多线程                      B、多接口                      C、多平台                      D、多动态

21、Java 虚拟机 ( JVM )运行 Java 代码时，不会进行操作是 ( )

A、加载代码                      B、校验代码                      C、编译代码                      D、执行代码

22、使新创建的线程参与运行调度方法是 ( )

A、run ( )                      B、start ( )                      C、init ( )                      D、resume ( )

23、Java 中的线程模型由三部分组成，与线程模型组成无关的是 ( )

A、虚拟的 CPU                      B、程序代码                      C、操作系统的内核状态  
D、数据

24、阅读下列代码段

```
class Test implements Runnable {
public intrun(){
int i=0;
while(true){
i++;
system.out.println("i"+i);}}}}
```

上述代码的编译结果是 ( )

A、程序通过编译并且 run() 方法可以正常输入递增的 i 的值

B、程序通过编译，调用 run() 方法将不显示任何输入

C、程序不能通过编译，while 的循环控制条件不能为 " true"

D、程序不能通过编译，因为 run() 方法的返回值类型不是 void

25、为实现多线程之间的通信，需要使用下列那种流才合适？  
( )

A、Filter stream                      B、File stream                      C、Random access stream

D、Pipe stream

26、下列方法中可以用来创建一个新线程的是 ( )

A、实现 java.lang.Runnable 接口并重写 start() 方法

B、实现 java.lang.Runnable 接口并重写 run() 方法

C、实现 java.lang.Thread 类并重写 run() 方法

D、实现 java.lang.Thread 类并重写 start() 方法

27、下列关于线程优先级的说法中，正确的是 ( )

A、线程的优先级是不能改变的                      B、线程的优先级是在创建线程时设置的

C、在创建线程后的任何时候都可以设置                      D、B 和 C

28 ( ) 方法不可以用来暂时停止当前线程的运行。

A、stop ( )                      B、sleep ( )                      C、wait ( )                      D、suspend ( )

- 29、在多线程并发程序设计中，能够给对象 x 加锁的语句是 ( )  
 A、x.wait()                      B、synchronized(x)                      C、x.nosity()                      D、x.synchronized()
- 30、下列方法能够用来实现线程之间通信的是 ( )  
 A、notify ( )                      B、run ( )                      C、sleep ( )                      D、Join ( )
- 31、下列关于线程的说法中正确的是 ( )  
 A、一个线程一旦被创建，就立即开始运行  
 B、使用 start ( ) 方法可以使一个线程成为可运行的，但是它不一定立即开始运行  
 C、当运行状态的线程因为调用了 yield ( ) 方法而停止运行，它一定被放在可运行的线程队列的前面  
 D、挡因等待对象锁而被阻塞的线程获得锁后，将直接进入运行状态
- 32、下列有关线程的叙述哪个是正确的？ ( )  
 A、一旦一个线程被创建，它就立即开始运行  
 B、调用 start ( ) 方法可以使一个线程成为可运行的，但是它不一定立即开始运行  
 C、主线程不具有默认优先级  
 D、Java 中线程的优先级从低到高以整数 0 ~ 9 表示
- 33、resume ( ) 方法负责恢复被哪个方法挂起的线程？ ( )  
 A、stop ( )                      B、sleep ( )                      C、wait ( )                      D、suspend ( )
- 34、.Java 虚拟机的执行过程有多个特点，下列哪个特点不属于 JVM 执行特点？ ( )  
 A、多线程                      B、动态连接                      C、异常处理                      D、异步处理
- 35、下列叙述中，正确的是 ( )  
 A、线程与进程在概念上不相关的                      B、一个进程可包含多个进程  
 C、一个线程可包含多个线程                      D、Java 中线程没有优先级
- 36、.关于下列程序的功能，说法正确的是 ( )  

```
public class ko10_1 extends Thread {
 int n;
 ko10_1(){
 Thread td=new Thread(this);
 td.start();
 }
 public void run() {
 for (n=0;n<6;n++) {
 try {
 System.out.print(n);
```

```

Thread.sleep(500); }
catch(InterruptedException e)
{ System.out.println("Exception"); } }
}
public static void main(String args[]) {
new ko10_1(); }
}

```

- A、这段程序的功能是、每隔 500 秒打印 012345
- B、这段程序的功能是、每隔 1 秒打印 012345
- C、这段程序的功能是、每隔半秒打印 012345
- D、这段程序的功能是、每隔半分钟打印 012345

37、.一个 Java application 运行后，在系统中是作为一个 ( )

- A、线程
- B、进程
- C、进程或线程
- D、不可预知

38、下列关于 Java 多线程并发控制机制的叙述中， 错误的是 ( )

- A、Java 中对共享数据操作的并发控制是采用加锁技术
- B、线程之间的交互，提倡采用 suspend( )/resume( )方法
- C、共享数据的访问权限都必须定义为 private
- D、Java 中没有提供检测与避免死锁的专门机制，但应用程序员可以采用某些策略防止死锁的发生

39、关于以下程序段的执行结果，说法正确的是 ( )

```

public class Borley extends Thread{
public static void main(String argv[]){
Borley b = new Borley();
b.start();
}
public void run(){
System.out.println("Running");
}
}

```

- A、编译通过并执行，但无输出
- B、编译通过并执行，输出、Running
- C、产生错误，没有 Thread 类对象
- D、产生错误，没有通道到达 Thread 包

40、下面不属于线程生命周期的状态的是 : ( )

- A、新建状态
- B、可运行状态
- C、运行状态
- D、等待状态

41、使比其自身优先级低的线程运行的 Thread 类的方法是、 ( )

- A、sleep()
- B、yiele()
- C、join()
- D、interrupt()

- 42、下面关于对象加锁的叙述错误的是、 ( )
- A、 当一个线程获得了对象的锁后，其他任何线程不能对该对象进行任何操作
- B、 对象锁的使用保证了共享数据的一致性
- C、 Java 中的对象锁是共享锁
- D、 对象锁只对临界区操作才有意义
- 43、 为实现多 线程之间 的通 信，需要 使用 下列哪种 流才 合适？ ( )
- A、 Filter stream      B、 File stream      C、 Random access stream
- D、 Piped stream
- 44、 下面关于线程优先级的说法中，错误的是、 ( )
- A、 Java 中的线程的优先级有三个静态常量
- B、 新建线程的优先级默认为最低
- C、 优先级高的线程优先被执行
- D、 一个线程运行时，有可能被比它高优先级的线程抢占运行
- 45、 下列方法中哪个是执行线程的方法？ ( )
- A、 run ( )      B、 start ( )      C、 sleep ( )      D、 suspend ( )
- 46、 使创建的线程参与运行调度的方法是 ( )
- A、 run()      B、 start()      C、 init()      D、 resume()
- 47、 下列方法中可用于定义线程体的是 ( )
- A、 start ( )      B、 init ( )      C、 run()      D、 main ( )
- 48、 如果线程 调用下列 方法 ，不能保 证使 该线程停 止运 行的是 ( )
- A、 sleep ( )      B、 stop ( )      C、 yield ( )      D、 wait ( )
- 49、 下列方法中可以用来创建一个新线程的是 ( )。
- A、 实现 java.lang.Runnable 接口并重写 start() 方法      B、 实现 java.lang.Runnable 接口并重写 run() 方法
- C、 继承 java.lang.Thread 类并重写 run() 方法      D、 实现 java.lang.Thread 类并实现 start() 方法
- 50、 阅读下列程序
- ```
public class Test implements Runnable{
    private int    x=0;
    private int    y=0;
    boolean flag =true;
    public static void main (String args[]){
        Test r =new Test ();
        thread  t1= new  thread(r);
        thread  t2= new  thread(r);
```



```

t1.start();
t2.start();}
public void run(){
while(flag){
x++;
y++ ;
system.out.println("("+x_",","+y+")" );
if ( x>=10){
flag=false;}}}

```

下列对程序运行结果描述正确的是 ()

- A、每行的 (x,y) 中，可能有 x y，每一对 (x,y) 值都出现两次
- B、每行的 (x,y) 中，可能有 x y，每一对 (x,y) 值都出现一次
- C、每行的 (x,y) 中，可能有 x=y，每一对 (x,y) 值都出现两次
- D、每行的 (x,y) 中，可能有 x=y，每一对 (x,y) 值都出现一次

51、在一个线程中调用下列方法，不会改变线程运行状态的是 ()

- A、yield ()
- B、另一个线程的 join 方法
- C、sleep ()
- D、一个对象的 notify 方法

52、在一个线程中使用 sleep (1000) 方法，将使该线程在多少时间后获得 CPU 控制，假设睡眠过程中不会有其他事件唤醒该线程 ()

- A、正好 1000 毫秒
- B、1000 毫秒不到
- C、>=1000ms
- D、不一定，可能 >1000ms，可能 <1000ms

53、为实现多线程之间的通信，需要使用下列哪种流才合适 ()。

- A、Filter stream
- B、File stream
- C、Random access stream
- D、Pipe stream

54、以下哪个方法用于定义线程的执行体 ()

- A、start()
- B、init()
- C、run()
- D、main()
- E、synchronized()

55、要串行化某些类的对象，这些类就必须实现 ()

- A、Serializable 接口
- B、java.io.Externalizable 接口
- C、java.io.DataInput 接口
- D、DataOutput 接口

56、线程交互中不提倡使用的方法是、 ()

- A、wait()
- B、notify()
- C、stop()
- D、notifyAll()

57、对象状态的持久化是通过 ()

- A、文件
- B、管道
- C、串行行
- D、过滤器

58、在 Java applet 程序中，用户自定义的 Applet 子类常常覆盖父类的 () 方法来完成 applet 界面的初始化工作。

- A、start()
- B、stop()
- C、init()
- D、paint()

59、main 方法是 Java Application 程序执行的入口点，关于 main 方

- 法的方法头以下哪项是合法的 ()。
- A、public static void main()
main(String[] args)
C、public static int main(String[] args)
D、public void main(String arg[])
- 60、下面哪些不是线程的状态？ ()
- A、ready B、dead C、running D、nonRunnable
- 61、如果当前线程需要让出 CPU 的使用权时，则调用方法、 ()
- A、wait() B、notify() C、yield() D、sleep()
- 62、关于线程的同步，下面说法正确的是、 ()
- A、线程的同步完全由系统来实现，用户不需要定义线程间的同步
B、在对共享资源以互斥的方式访问时，需要用户定义线程间的同步
C、线程的同步与线程的优先级有关
D、线程的同步就是线程的调度
- 63、关于线程的死锁，下面的说法正确的是 ()
- A、若程序中存在线程的死锁问题，编译时不能通过
B、线程的死锁是一种逻辑运行错误，编译器无法检测
C、实现多线程时死锁不可避免
D、为了避免死锁，应解除对资源以互斥的方式进行访问
- 64、在浏览器中执行 applet 程序，将被最先执行的方法是 ()。
- A、init() B、start() C、destroy() D、stop()
- 65、编译 JavaApplet 源程序文件编译后产生的字节码文件扩展名为 ()。
- A、class B、java C、html D、exe
- 66、在编写 JavaApplet 程序时，一定要引入 (import) () 包。
- A、java.awt.* B、java.applet.* C、java.io.* D、java.awt.event.*
- 67、容器 Panel 和 Applet 缺省使用的布局编辑策略是 ()。
- A、FlowLayout B、BorderLayout C、GridLayout D、CardLayout
- 68、下列关于 Java 多线程并发控制机制的叙述中，错误的是 ()
- A、Java 中对共享数据操作的并发控制是采用加锁技术
B、线程之间的交互，提倡采用 suspend()/resume() 方法
C、共享数据的访问权限都必须定义为 private
D、Java 中没有提供检测与避免死锁的专门机制，但应用程序员可以采用某些策略防止死锁的
- 69、下列程序的功能是创建了一个显示 5 个 "Hello!" 的线程并启动运行，请将程序补充完整。

```
public class ThreadTest extends Thread{
    public static void main(String args[]){
```

```

        ThreadTest t=new __ThreadTest()____;
        t.start() ; }
    public void run(){int i=0;
        while(true){System.out.println("Hello!");
            if (i++==4)    break;
        }
    }
}

```

下面关于 Applet 的说法正确的是 ()。

A、Applet 也需要 main 方法 B、Applet 继承自 java.awt.Applet 或 javax.swing.JApplet

C、Applet 能访问本地文件 D、Applet 程序不需要编译

70、要串行化某些类的对象，这些类就必须实现 ()。

A、Serializable 接口 B、java.io.Externalizable 接口

C、java.io.DataInput 接口 D、DataOutput 接口

71、下列关于 Applet 的叙述中，错误的是 ()

A、Applet 是 Java 类，所以可以由 JDK 中的解释器 java.exe 直接解释运行

B、Applet 应该定义为 java.applet.Applet 类或 javax.swing.JApplet 类的子类

C、Applet 与 Application 的主要区别在执行方式上

D、通过在 HTML 文件中采用标记可以向 Applet 传递参数

72、调用线程的下列方法，不会改变该线程的生命周期中状态的方法是()。

A、yield() B、wait() C、sleep() D、

isAlive()

73、下列情况中，不会使线程返回所持有的对象锁的是 ()。

A、当 synchronized() 语句块执行完毕

B、当调用了线程的 suspend()方法

C、当在 synchronized() 语句块中出现异常 (exception)

D、当持有锁的线程调用该对象的 wait() 方法

74、当输入一个字节流时，要 ()

A、实现 DataInput 接口

B、实现 ObjectInput 接口

C、Serializable 接口

D、继承 Reader 抽象类

75、在匹配器 (Matcher) 类中，用于输入字符串与模式串比较的方法是()。

A、static boolean matches()

B、boolean matcher.find()

C、int matcher.start()

D、int matcher.end()

76、关于以下程序段的执行结果，说法正确的是 ()

```

public class Borley extends Thread{
    public static void main(String argv[]){

```

```

Borley b = new Borley();
b.start();
}
public void run(){
System.out.println("Running");
}
}

```

A、编译通过并执行，但无输出
Running

B、编译通过并执行，输出、

C、产生错误，没有 Thread 类对象
达 Thread 包

D、产生错误，没有通道到

77、 下列程序的功能是在监控台上每隔一秒钟显示一个字符串

“ Hello ”,能够填写在程序中下划线位置，使程序完整并能正确运行的语句是 ()

```

public class Test implements Runnable{
public static void main(String args[]){
Test t=new Test();
Thread tt=new Thread(t);
tt.start();
}
public void run(){
for(;;){
try{
;
}catch( e){}
System.out.println("Hello");
} } }

```

A、 sleep(1000)

B、 sleep(1000)

InterruptedException

InterruptedException

C、 Thread.sleep(1000)

D、 Thread.sleep(1000)

RuntimeException

InterruptedException

78、关于下列程序的功能，说法正确的是 ()

```

public class ko10_1 extends Thread {
int n;
ko10_1() {
Thread td=new Thread(this);
td.start();
}
public void run() {
for (n=0;n<6;n++) {

```

```

try {
System.out.print(n);
Thread.sleep(500);
}
catch(InterruptedException e)
{ System.out.println("Exception"); }
} }
public static void main(String args[]) {
new ko10_1();
} }

```

- A、这段程序的功能是、每隔 500 秒打印 012345
- B、这段程序的功能是、每隔 1 秒打印 012345
- C、这段程序的功能是、每隔半秒打印 012345 32.
- D、这段程序的功能是、每隔半分钟打印 012345

79、在编写 Java applet 程序时，若需要对发生事件作出响应和处理，一般需要在程序的开头写上的语句是（ ）

- A、import java.awt.*;
- B、import java.applet.*;
- C、import java.io.*;
- D、import java.awt.event.*;

80、写 Java application 程序时，若需要使用到标准输入输出语句，必须在程序的开头写上（ ）语句。

- A、import java.awt.*;
- B、import java.applet.applet;
- C、import java.io.*;
- D、import java.awt.Graphics;

81、下列不是 InputStream 子类的是（ ）。

- A、文件输入流 FileInputStream
- B、对象输入流 ObjectInputStream
- C、字符输入流 CharInputStream
- D、压缩文件输入流 ZipInputStream

82、下列关于 Java Application 与 Applet 的说法中，正确的是（ ）。

- A、都包含 main() 方法
- B、都通过“ appletviewer ”命令执行
- C、都通过“ javac ”命令编译
- D、都嵌入在 HTML 文件中执行

83、Java 中类 ObjectOutputStream 支持对象的写操作，这是一种字节流，它的直接父类是（ ）

- A、Writer
- B、DataOutput
- C、OutputStream
- D、ObjectOutput

84、在下列程序的空白处，应填入的正确选项是（ ）

```

Import java.io.*;

```

```

Public class ObjectOutputStreamTest{
Public static void main(string args[]) throws IOException{
ObjectOutputStream oos= new ObjectOutputStream
(new FileOutputStream(      “ serial.bin      ” ));
Java.util.Date d= new Java.util.Date();
Oos _____ (d);
ObjectInputStream ois=
new ObjectInputStream(new FileOutputStream(      “ serial.bin      ” ));
try{
java.util.date restoredDate =
(Java.util.Date) ois.readObject();
System.out.println
( “ read object back from serial.bin file:      ”
+ restoredDate);
}
Catch (ClassNotFoundException   cnf) {
System.out.println (      “ class not found      ” );
}
}
}

```

A、 WriterObject B、 writer C、 BufferedWriter D、 writerObject

85、Java 语言具有许多优点和特点，下列选项中，哪个反映了 Java 程序并行机制的特点（ ）

A、安全性 B、多线性 C、跨平台 D、可移植

86、以下关于继承的叙述正确的是（ ）。

A、在 Java 中类不允许多继承
B、在 Java 中一个类只能实现一个接口
C、在 Java 中一个类不能同时继承一个类和实现一个接口
D、在 Java 中接口只允许单一继承

87、下面哪些方法将会将一个正在执行的线程中断 （ ）

A、wait() B、notify() C、yield() D、suspend()

88、面有关构造方法的叙述，不正确的是 （ ）

A、类必须含有构造方法，其名字与类名相同，若没有，系统会自动加载，但其参数表为空

B、父类的构造方法不能被子类继承，若想调用，将 super()放在子类构造方法的第一条语句

C、构造方法的主要功能是初始化类中的数据成员

D、在线程编程中，构造方法可以被同步，也就是说可以被 synchronized 关键字修饰

89、有关下述关于线程的代码 ,说法正确的是（ ）

```

1.      class Test
2.      {
3.          public static void main(String [] args)
4.          {
5.              printAll(args);
6.          }
7.          public static void printAll(String[] lines)
8.          {
9.              for(int i=0;i<lines.length;i++)
10.             {
11.                 try
12.                 {
13.                     System.out.println(lines[i]);
14.
15. Thread.currentThread().sleep(3000);
16.                 }
17.             catch(Exception e)
18.             {}
19.         }
20.     }

```

A、从键盘读入的字符都被打印 ,时间间隔为 3 秒

B、从键盘读入的字符都被打印 ,但是没有时间间隔

C、 程序将会编译出错 ,在第 14 行

D、 去掉异常处理语句 try 和 catch,程序才会正常编译

90、已知有一个多线程类 myThread 除 run 方法外还包含一个方法 void disp()。以下说法正确的是 ()

A、 myThread mt=new myThread(); mt.start();mt.disp(); 运行完线程体 run 方法后再运行 disp() 方法

B、 myThread mt=new myThread();mt.disp();mt.start(); 运行完 disp() 后才能启动线程体 run

C、 myThread mt=new myThread();mt.start();mt.disp(); 线程体 run 一定会在 disp 结束前结束。

D 、 myThread mt=new myThread();mt.start();mt.disp();mt.start() 和 mt.disp() 两条命令 , 哪条放在前面哪条放在后面运行结果一样。

91、识别用线程创建 apple 的正确陈述。 ()

A、扩展 JApplet 和 Thread

B、扩展 JApplet 和重设

Thread 类的 run() 方法。

C、扩展 JApplet 和实现 Runnable。

D、扩展 JApplet 和创建

Thread 类的实例。

92、下面的哪个关键字通常用来对对象加锁，该标记使得对对象的访问是排他的？（ ）

A、 lock B、 synchronized C、 synclock D、 belocka

93、断点不能设置在（ ）中

A、 注释 B、 可执行的代码行 C、 赋值语句 D、 算数语句

94、当应用程序在某个断点处暂停执行时，下一条即执行的语句将位于该断点所在行语句（ ）的语句

A、 之前 B、 之后 C、 处 D、 以上答案均不对

95、调试程试中（ ）命令能够对应用程序的源代码中的一个可执行的代码行设置断点

A、 stop B、 run C、 print D、 clear

96、下列哪个类是由编译器自动生成对象的特殊类，是用于类操作？（ ）

A、 Class 类 B、 Thread 类 C、 System 类 D、 Object 类

97、在 Java 语言中，下列哪个包是编译器自动导入的？（ ）

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

98、下列容器中哪一个是从 java.awt.Window 继承的？（ ）

A、 Frame B、 Panel C、 Container D、 Applet

99、下列描述中，错误的是（ ）。

A、 Java 要求编程者管理 内存。 B、 Java 的安全性体现在多个层次上。

C、 Applet 要求在支持 Java 的浏览器上运行。 D、 Java 内含多线程机制。

100、字符输出流类都是 ____ 抽象类的子类。（ ）

A、 FilterWriter B、 FileWrite C、 Writer D、 OutputStreamWrite

101、用 Thread 子类实现多线程的步骤顺序是（ ）

A、 声明 Thread 类的子类，创建 Thread 子类的实例，让线程调用 start() 方法

B、 声明 Thread 类的子类，在子类中重新定义 run() 方法，创建 Thread 子类的实例

C、 创建 Thread 子类的实例，让线程调用 start() 方法

D、 声明 Thread 类的子类，在子类中重新定义 run() 方法，创建 Thread 子类的实例，让线程调用 start() 方法

102、在以下供选择的方法中，能让线程从阻塞状态恢复到就绪状态的方法是（ ）

A、 start() B、 init() C、 resume() D、 run()

103、在以下四个供选的整数中，能作为线程最高优先级的整数是（ ）

A、 0 B、 1 C、 10 D、 11

104、异常包含下列那些内容？（ ）

A、程序中的语法错误
编译错误

B、程序的

C、程序执行过程中遇到的事先没有预料到的情况
先定义好的可能出现的意外情况

D、程序事

105、Java 语言具有许多优点和特点，下列选项中，哪个反映了 Java
程序并行机制的特点？（ ）

A、安全性
移植

B、多线性

C、跨平台

D、可

106、Java 语言中提供了一个（ ）线程，自动回收动态分配的
内存。

A、异步
集

B、消费者

C、守护

D、垃圾收

107、进程独有的内容包括（ ）

A、数据

B、系统栈

C、寄存器

D、代码

108、不是导致线程不能运行的原因的是（ ）

A、等待
作而阻塞

B、阻塞

C、休眠

D、挂起及由于 I/O 操

109、当（ ）方法终止时，能使线程进入死亡状态。

A、run

B、setPriority

C、yield

D、sleep

110、用（ ）方法可以改变线程的优先级。

A、run

B、setPriority

C、yield

D、sleep

111、线程通过（ ）方法可以使具有相同优先级线程获得处理器。

A、run

B、setPriority

C、yield

D、sleep

112、线程通过（ ）方法可以休眠一段时间，然后恢复运行。

A、run

B、setPriority

C、yield

D、sleep

113、（ ）方法使对象等待队列的第一个线程进入就绪状态。

A、run

B、setPriority

C、yield

D、sleep

114、方法 resume()负责重新开始（ ）线程的执行。

A、被 stop()方法停止

B、被 sleep()方法停止

C、被 wait()方法

停止 D、被 suspend()方法停止

115 线程交互中不提倡使用的方法是（ ）

A、wait()

B、notify()

C、stop()

D、notifyall()

三、多项选择

1、线程的哪些方法可以使线程暂停（ ）

A、sleep

B、stop

C、join

D、suspend

2、解决线程干扰的方法有哪几种（ ）

A、同步

B、使用锁

C、使用不可变对象

D、使用 ThreadLocal

变量

- 3、JDK 提供的用于并发编程的同步器有哪些？（ ）
A、Semaphore B、CyclicBarrier C、CountDownLatch D、Counter
- 4、ReentrantLock 表示的锁，下面哪些方法可以获取锁？（ ）
A、lock B、tryLock C、unlock D、isLocked
- 5、Amino 采用的调度模式和算法有哪几种？（ ）
A、Master-Worker B、Map-reduce C、Divide and conquer D、fork-join
- 6、下面哪个方法可用于定义新线程类（ ）
A、implement the Runnable interface
B、add a run() method in the class
C、create an instance of Thread
D、extend the Thread class
- 7、浏览器禁止 Applet 执行下列哪些操作？（ ）
A、在运行时调用其它程序。
B、文件读写操作。
C、装载动态连接库和调用任何本地方法。
D、试图打开一个 socket 进行网络通信,但是所连接的主机并不是提供 Applet 的主机。

四、判断题

- 1、在并程序序中，当两个并行的线程，在没有任何约束的情况下，访问一个共享变量或者共享对象的一个域，而且至少要有一个操作是写操作，就可能发生数据竞争错误。（ ）
- 2、原语 Compare-and-swap(CAS) 是实现无锁数据结构的通用原语。（ ）
- 3、获得内部锁的唯一途径是、进入这个内部锁保护的同步块或方法。（ ）
- 4、volatile 变量具有 synchronized 的可见性特性，但是不具备原子特性。（ ）
- 5、减小竞争发生可能性的有效方式是尽可能缩短把持锁的时间。（ ）
- 6、一个线程对象的具体操作是由 run() 方法的内容确定的，但是 Thread 类的 run 方法是空的，其中没有内容；所以用户程序要么派生一个 Thread 的子类并在子类里重新定义 run() 方法，要么使一个类实现 Runnable 接口并书写其中 run() 方法的方法体实现（ ）
- 7、java.lang.Runnable 接口并重写 run() 方法可以用来创建一个新线程。（ ）
- 8、.线程的优先级是不能改变的。（ ）
- 9、要在程序中实现线程必须导入 java.io.Thread 类。（ ）
- 10、线程之间可以共享数据。（ ）

- 11、启动线程时可以调用 run() 方法或 start() 方法。
()
- 12、C 和 Java 都是多线程语言。()
- 13、如果线程死亡，它便不能运行。()
- 14、在 Java 中，高优先级的可运行线程会抢占低优先级线程。()
- 15、Windows 和 Windows NT 的 Java 系统使用分时的方法。因此
可以使某一线程抢占具有相同优先级的线程。()
- 16、线程不是能够独立运行的程序，是某个程序内部的一个顺序执行
流。()
- 17、程序开发者必须创建一个线程去管理内存的分配。()
- 18、一个线程在调用它的 start 方法，之前，该线程将一直处于出生
期。()
- 19、当调用一个正在进行线程的 stop() 方法时，该线程便会进入休眠
状态。()
- 20、如果线程的 run 方法执行结束或抛出一个不能捕获的例外，线程
便进入等待状态。
()
- 21、一个线程可以调用 yield 方法使其他线程有机会运行。()

五、程序分析题

1、写出下面程序的输出结果

```
import java.util.concurrent.CountDownLatch;
public class TestCountDownLatch {
    public static void execute() throws InterruptedException {
        final int n = 3;
        final CountDownLatch start = new CountDownLatch(1);
        final CountDownLatch end = new CountDownLatch(n);
        System.out.print("A");
        Thread t = new Thread() {
            public void run() {
                for (int i = 0; i < n; i++) {
                    try {
                        start.await(); // 此处填空
                        System.out.print("B");
                    } catch (Exception e) {
                    } finally {
                        end.countDown(); // 此处填空
                    }
                }
            }
        };
        t.start();
        System.out.print("C");
    }
}
```

```

        start.countDown();
        end.await();
        System.out.print("D");
    }
    public static void main(String[] args) throws InterruptedException {
        execute();
    }
}

```

该程序的运行结果为：

2、判断下面的程序片断是否正确。若有错，指出错在哪里并改正；若正确，打“ ”。

```

public class Test implements Runnable
{Thread t=new Thread(this);
t.start();
protected void run()
{System.out.println(        “ thread is running now        ” );}
}

```

该程序的运行结果为：

3、阅读以下程序，请写出该程序的输出结果。

```

class CurrentThreadDemo {
    public static void main(String args[]) {
        Thread t = new Thread();    System.out.println("Current
thread");
        t.setName("My Thread");    System.out.println("After name
change");
        try {for(int n = 3; n > 0; n--) {
            System.out.println(n);    Thread.sleep(1000);
        }
        } catch (InterruptedException e) {
            System.out.println("Main thread interrupted");
        }
    }
}

```

该程序的运行结果为：

六、程序设计题

1、有三个线程 ID 分别是 A、B、C,请有多线程编程实现， 在屏幕上循环打印 10 次 ABCABC

2、试着编写一个程序，创建两个新线程。

3、试着编写一个程序，实现 Runnable 接口，创建的一个线程。

- 4、实现通过四个售票点发售某日某次列车的 100 张车票，一个售票点用一个线程表示。
- 5、CrazyCoder 工作室为例，现在 6 个工作室成员因工作需要，工作室成员需要做一个项目，因为是多个人做，设计用多线程来做。首先定义成员类 `UserInfo` (表示成员信息)，还有一个方法 `Work()` 表示工作。
- 6、用 JAVA SOCKET 编程，实现简单的 Echo 功能。如、客户端从键盘输入 hi (当用户输入 exit 退出程序)，服务端响应为 hi (服务器要求为多线程)
- 7、提货单真实数据一个调用者在调用耗时操作，不能立即返回数据时，先返回一个提货单，然后在过一断时间后凭提货单来获取真正的数据。去蛋糕店买蛋糕，不需要等蛋糕做出来（假设现做要很长时间），只需要领个提货单就可以了（去干别的事情），等到蛋糕做好了，再拿提货单取蛋糕就可以了。
- 8、一个从 1 加到 10,第 2 个从 11 加到 20...线程同步执行，最后再把十个线程结果相加
- 9、一个多线程的摇奖器的例子，在第一个类中定义三个 `void` 方法用于产生随机数。
- 10、编写一个多线程程序实现如下功能、线程 A 和线程 B 分别在屏幕上显示信息“ A start ”“ B start ”后调用 `wait` 等待；线程 C 开始后调用 `sleep` 休眠一段时间，然后调用 `notifyall`，使线程 A 和线程 B 继续运行。线程 A 和线程 B 恢复运行后输出信息“ A END ”“ B END ”后结束，线程 C 在判断线程 A 线程 B 结束后，自己也结束运行。
- 11、设计 4 个线程，其中两个线程每次对 `j` 增加 1，另外两个线程对 `j` 每次减少 1。写出程序。
- 12、在 java 中实现同时写多张读卡器
- 13、要实现一个 socket 服务器。接受客户端 1 发来的消息，客户端 1 每隔一段时间就会发送消息给服务器，服务器将消息存储数据库，同时发送到客户端 2 用于实时显示，是通过两个端口分别与两个客户端进行通信，一收一发还是仅监听一个端口，收消息，然后再调用方法进行发送。
- 14、假定一个缓冲区只能保存一条信息。即缓冲区只有两个状态、有信息或没有信息。
- 15、模拟火车票联网售票系统、多个线程同时出票，保证每张出票的编号连续且不重复。
- 16、JAVA 多线程编程。用 JAVA 写一个多线程程序，如写四个线程，二个加 1，二个对一个变量减一，输出。
- 17、编写一个多线程程序实现如下功能、线程 A 和线程 B 分别在屏幕上显示“。。。start”后，调用 `wait` 等待；线程 C 开始后调用 `sleep` 休眠一段时间，然后调用 `notifyAll`，使线程 A 和线程 B 继续运行。线程 A 和线程 B 恢复运行后输出信息“。。。end”后结束，线程 C 判断线

程 A 和 B 结束后，C 自己也结束。

18、有一个南北向的桥，只能容纳一个人，现桥的两边分别有 10 人和 12 人，编制一个多线程程序让这些人到达对岸，每个人用一个线程表示，桥为共享资源。在过桥的过程中显示谁在过桥其走向。

19、利用多线程技术编写 Applet 程序，其中包含一个滚动的字符串。字符串从左向右运动，当所有的字符都从屏幕的右边消失后，字符串重新从左边出现并继续向右移动。

20、编写一个应用程序，在线程同步的情况下来实现“生产者 消费者”问题。

21、利用多线程设计一个程序，同时输出 50 以内的奇数和偶数，以及当前运行的线程数。

22、编写采用原子量类的多线程程序模拟银行用户存取钱的并发程序

23、用线程实现动画

第九章 异常处理

一、选择题

- 1、无论是否发生异常，都需要执行（
A、 try 语句块 B、 catch 语句块 C、 finally 语句块 D、 return 语句
- 2、异常处理变量（
A、应用 public 关键字 B、可以应用 protected 关键字
C、可以应用 private 关键字 D、只能在异常处理方法内使用。
- 3、通常的异常类是（
A、Exception B、exception C、CommonException D、ExceptionShare
- 4、异常产生的原因很多，常见的有（
A、程序设计本身的错误 B、程序运行环境改变 C、软、硬件设置错误 D、以上都是
- 5、（
A、 RuntimeException B、 ClassCastException
C、 AarihmetticException D、 ArrayIndexOutOfBoundsException
- 6、下列描述中，对使用异常处理的原因描述错误的有（
A、将错误处理程序与正常程序流分开，增加程序的可读性
B、可以容易地指出异常在何处处理
C、减轻程序员处理异常的任务
D、增加语言的复杂机制
- 7、读下面代码，哪个选项是正确的（

```
import java.io.*;
public class Test2{
    public static void main(String []args)throws IOException{
        if(args[0]== " hello " )
            throw new IOException();
    }
}
```


A、没有错误，程序编译正确
B、编译错误，不能够在 main 方法中抛出异常
C、编译错误， IOException 是一个系统异常，不能够由 application 程序产生
D、没有输出结果
- 8、当变异并且运行下面程序时会出现什么结果？（

```
public class ThrowsDemo{
    static void throwMethod() {
        System.out.print("Inside throwMethod");
    }
}
```

```

        throw new IOException("demo");
    }
    public static void main(String [] args){
    try{
        throwMethod();
    }catch(IOException e){
        System.out.println("Cauht"+e);
    }
    }
}

```

- A、编译错误 B、运行错误 C、编译成功，但不会打印出任何结果
D、没有输出结果

9、执行下面程序的结果是什么？其中 a=4，b=0 （ ）

```

public static void divide(int a,int b){
    try{    int    c = a / b;    }
    catch(Exception e){
        System.out.println("Exception");}
    finally{
        System.out.println("Finally");
    }
}

```

- A、打印 Exception finally B、打印 Finally
C、打印 Exception D、没有输出结果

10、假定一个方法会产生非 RuntimeException 异常，如果希望把异常交给调用该方法的方法处理，正确的声明方式是什么？ （ ）

- A、throw Exception B、throws Exception
C、new Exception D、不需要指明什么

11、阅读下面的代码段、

```

try{
tryThis();
return;
}catch(IOException x1){
    System.out.println(    “ exception 1    ” );
Return;
}catch(Exception x2){
    System.out.println(    “ exception 1    ” );
Return;
}finally{
    System.out.println(    “ finally    ” )
}

```

如果 tryThis（）抛出一个 IOException，那么屏幕的输出是什么？

- ()
- A、 "exception 1 "后面跟着 " finally " B、 " exception 2 "后面跟着 " finally "
- C、 "exception 1 " D、 "exception 2 ""
- 12、 下列哪些内容不是异常的含义？ ()
- A、 程序的语法错 B、 程序编译或运行中所发生的异常事件
- C、 程序预先定义好的异常事件 D、 程序编译错误
- 13、 自定义的异常类可从下列哪个类继承？ ()
- A、 Error 类 B、 AWTError C、 VirtualMachineError D、 Exception 及其子集
- 14、 当方法遇到异常又不知如何处理时，下列哪种做法是正确的？ ()
- A、 捕获异常 B、 抛出异常 C、 声明异常 D、 嵌套异常
- 15、 如要抛出异常，应用下列哪种子句？ ()
- A、 catch B、 throw C、 try D、 finally
- 16、 对于 catch 子句的排列，下列哪种是正确的？ ()
- A、 父类在先，子类在后 B、 子类在先，父类在后
- C、 有继承关系的异常不能在同一个 try 程序段内 D、 先有子类，其他如何排列都无关
- 17、 下列选项中属于异常的关键字是 ()
- A、 public B、 InputStream C、 Exception D、 abstract
- 18、 下列选项中属于 Exception 异常的是 ()
- A、 ArithmeticException B、 NullPointerException
- C、 classCastException D、 以上都是
- 19、 以下是异常的处理，哪个选项是正确的 ()
- A、 book()throws exception B、 book(int a)exception
- C、 book()throws D、 book()throw exception
- 20、 所有属于 () 子类的异常都是非检查型异常
- A、 RuntimeException B、 Exception C、 Error D、 以上答案都不对
- 21、 Java 编译程序对于 () 需要加强捕获或声明要求
- A、 异常 B、 错误 C、 非检查型错误 D、 检查型异常
- 22、 如果试图捕获多个错误，可在 () 语句块的后面使用多个 () 语句块

A、 try catch B、 catch try C、 finally try D、 以上答案都不对

23、将需要处理异常作为 () 语句块的一个参数来进行声明

A、 try B、 catch C、 finally D、 以上都不对

24、try 语句块可以 ()

A、 拥有惟一的一个 catch 语句块 B、 多个 finally 语句块

C、 一个或多个 catch 语句块 D、 以上答案都不对

25、所有 RuntimeException 子类的异常都属于 ()

A、 致命错误 B 、 逻辑错误 C 、 检查型错误 D、 非检查型错误

26、() 是所有 Exception 和 Error 类的共同超类

A、 Throwable B、 CheckedException C、 Catchable D、 RuntimeException

27、Error 和 Exception 是 Throwable 的两个直接子类，其中下列属于 Error 的直接子类的是 ()

A、 ThreadDeath B、 LinkageError C、 VirtualMachineError D、 以上都是

28、在 try 代码中声明的变量只能在该块的大括号 () 使用

A、 结束之前 B、 结束之后 C、 不清楚 D、 不用执行

29、假定一个方法可能会产生非 RuntimeException 异常,如果希望把异常交给调用该方法的方法处理，正确的声明方式是 ()

A、 throws Exception B、 throw Exception C、 new Exception D、 不需要指明什么

30、try 代码块中包含的是可能引起一个或多个异常代码 ,能够抛出异常的代码必须位于 () 代码块中。

A、 catch B、 finally C 、 try D 、 B 和 C

31、Java 一般不会检查 () 是否被处理，而本身在程序中也不能捕获该类型的异常，因为它一旦产生，该程序基本会处于需要终止的状态

A、 Exception B、 Error C、 catch D、 ClassCastException

32、当应用程序检测到异常时，会调用 ()

A、 异常代码 B、 异常处理器 C、 异常处理程序 D、 以上答案都不是

33、调用方法 Integer.parseInt("123.45 "); 将抛出 ()

A、 NumberFormatException B、 ParsingException

- 6、_____是公共的异常类，它适用于任何异常操作。
- 7、使用_____来提醒类的使用者应该进行异常处理。
- 8、如果 try 语句块中未产生异常，应用程序会忽略相应语句块。
- 9、生成异常对象并把它们提交给运行时系统的过程称为一个异常。
- 10、运行系统把该异常对象交给这个方法进行处理，这个过程成为一个异常。
- 11、抛出 Exception 类常用的方法有：_____、_____、_____。
- 12、有时如果声明了某个类的一个对象，_____但没有创建它（给它分配内存空间），将会出现_____异常。
- 13、_____语句可以有一个或多个，_____而且至少有一个_____语句或_____语句与 try 语句配对。
- 14、异常是_____类的子类的对象。
- 15、如果一个方法抛出的异常没有被捕获，_____并且也不能有 Error 类或者 RuntimeException 类的子类表示，那么读者必须在方法的定义中使用一个_____子句来识别异常。
- 16、如果想要在方法中处理异常，那么必须将可能产生异常的代码放在_____代码块中。一个方法可能包含_____try 代码块。
- 17、异常处理代码位于紧跟在可能抛出异常的_____try 代码块之后的代码块中。
- 18、一个 try 代码块能够包含_____catch 代码块，每个块处理一种不同类型的异常。
- 19、_____代码块用于包含必须在 try 代码块执行之后运行的代码。
- 20、不论 try 代码块的执行如何结束。_____一个_____代码块将总会在方法发运行结束前被执行。
- 21、用户自定义的异常类应该从_____类派生。
- 22、java 中的异常类为 Exception，它包括很多子类，用于处理各种具体的异常，如_____异常，RuntimeException 为_____，“运行异常”是设计程序时异常处理的终点，_____RuntimeException 又有他的子类，表示更具体的运行异常，常用的有_____（空指针异常），_____（除0异常），_____（类型转换异常），_____（数组元素下表越界异常）
- 23、“extends Exception”，它的意思是、除包括一个_____的全部含义以外，还有更多的含义。增加的代码数量非常少——实际只添加了一个构建器，对 MyException 的创建方式进行了定义。
- 24、在 JAVA 运行时，如获得一个异常对象，它会自动寻找处理该异

常的代码，它从生成异常对象的代码构件开始，沿着方法按层回溯寻找，直至找到处理该类异常方法为止。

- 25、异常对象从产生和被传递给 JAVA 运行系统的过程称为 _____ 异常。
- 26、public Throwable(String message) 构造函数包含 getMessage() 方法，用来返回带参数构造函数创建异常时的 _____，对无参数构造函数而言，用它返回的是 _____。
- 27、catch 子句都带一个参数，该参数是某个异常的类及其变量名（该异常对象的指针），catch 用该参数去与抛出异常对象的类进行 _____。
- 28、Java 虚拟机能自动处理运行 _____。
- 29、在 Java 的异常处理机制中，使用 _____ 语句来捕获并处理异常。
- 30、在 try-catch-finally 语句中，不管 try 是否抛出异常，catch 是否捕获到异常 _____ 语句块都要执行，因为它是异常处理机制的出口。
- 31、throwable 类是 Java 异常体系中的根类，它有 _____ 类和 _____ 类。
- 32、Java 中的异常是当程序中出现不正常情况时生成的一个 _____。
- 33、接受异常对象作为参数的代码称为 _____。
- 34、throw 子句用来 _____ 异常，而 throws 子句用来声明异常。
- 35、异常的处理分为 _____ 和 _____。
- 36、在一个程序里，当看到一个数组的下标大于这个数组的最大下标时，这时候是属于 _____ 异常。
- 37、只要一个方法可以抛出你的异常，_____ 都将检查它是否在该方法中被捕获。
- 38、被称为“可检测异常”或“捕获型异常”，而且该异常在编程过程中经常会遇到，它属于编译期异常，那么该异常是 _____。
- 39、RuntimeException 一般称为运行时异常，在程序正式运行的时候会碰到的偶发型异常，这种异常因为是在运行时抛出一般情况下不需要进行捕获操作。该异常继承于 _____ 类。
- 40、如果函数体里面存在 throw 语句，而且函数体本身没有进行捕捉的话，那么必须使用 _____ 在函数头里面添加对应的异常抛出语句，否则无法通过编译。
- 41、在 Java 里面，异常处理机制的编程部分需要使用到几个关键字、
- 42、如果在 finally 代码块中返回一值，则这个返回值会 try 代码块中执行的任何返回语句的值
- 43、自定义异常必须总是以 _____ 类作为超类，否则将不能定义

异常

- 44、Error 类代表 Java 应用程序中所发生的异常，该异常通常不需要在应用程序中 _____
- 45、当应用程序检测并抛出一个异常时，所执行的语句块位于语句块内
- 46、Throwable 类是所有 _____ 和 _____ 的超类
- 47、通过参数可以让 _____ 语句块与被捕获的对象之间进行交互
- 48、catch 声明的异常类型必须和 _____ 里面抛出的类型相同
- 49、通常嵌入式 try 代码块用于以 _____ 处理不同类型的错误。
- 50、Throwable 有两个直接子类 _____ 和 _____。
- 51、如果你想明确地抛出一个 RuntimeException，你必须用语句来声明它的类型。
- 52、_____ 关键字是用在方法声明中，用来列出从方法中发出的、非起源于 Error 或 RuntimeException 中的任何异常。能够主动引发异常的方法必须用 _____ 来声明。
- 53、_____ 子句中同时可以指明多个异常，说明该方法将不对这些异常进行处理，而是声明抛弃它们。
- 54、IOException 属于 _____ 包。

三、判断题

- 1、不能在 finally 块中执行 return,continue 等语句，否则会把异常“吃掉”()
- 2、Try 语句后面可以跟多个 catch 语句()
- 3、finally 语句必须执行。()
- 4、throws 和 throw 功能一样。()
- 5、try 语句后面必须跟 catch 语句()
- 6、try 语句后面只能跟一个 catch 语句()
- 7、当代码出现异常时，才执行 try/catch/finally 语句的 finally 部分代码()
- 8、Try/catch 语句不可以进行嵌套操作。()
- 9、异常就是程序运行过程中遇到的严重错误，使程序运行中止，或者即使程序能够继续运行，但得出错误的结果甚至导致严重的后果。()
- 10、事实上，异常以及异常处理是非常简单，所以程序员选择用异常而不选择自己处理错误。()
- 11、异常的处理分为异常抛出和异常捕获。异常抛出就是系统自动检测到特定位置出现的异常，从而可以进行实现编写好的异常处理程序。异常捕获是指用户自己抛出特定的异常来出发异常处理程序，以检查异常处理程序是否能正常工作。()

- 12、finally 子句不一定会执行的，要根据具体的情况。（ ）
- 13、因为 java 类库提供了十分丰富的异常类型，能够满足绝大多数编程需要，所以，在开发较大的程序时， 不需要建立自己的异常类。（ ）
- 14、 程序中抛出异常时（throw ， ），只能抛出自己定义的异常对象。（ ）
- 15、. 一个异常处理中 finally 语句块只能有一个或者可以没有。（ ）
- 16 程序产生的原因有程序设计本身的错误、程序运行环境改变、软硬件设置的错误 （ ）
- 17 finally 语句块只有在 try 语句块中检测到异常时才会执行里面的语句 （ ）
- 18 异常就是程序运行过程中遇到的严重错误，使程序运行终止，或者即使程序能够继续运行，但得出错误的结果甚至导致严重的后果（ ）
- 19 异常抛出就是系统自动检测到特定位置出现的异常，从而可以执行事先编写好的异常处理程序 （ ）
- 20 为了实现异常的声明，在产生异常的方法参数列表之后、方法体之前提供一个含检查型异常的 throw 语句 （ ）
- 21 异常处理机制允许开发人员将错误处理代码从应用程序的逻辑代码中分离出来，从而改善应用程序的清晰度和可修 （ ）
- 22 在 Java 处理机制中， try-catch-finally 语句来捕获异常而用 throws 来抛出异常 （ ）
- 23 Java 系统在产生异常后，从上到下依次对每个 catch 语句进行检测，直到找到和抛出的异常对象想匹配的异常类型为止（ ）
- 24 利用 try 语句将可能跑出 NumberFormatException 异常的语句进行封装并使用 catch 语句块实现对 NumberFormatException 异常的处理（ ）
- 25 非运行时异常就是编译器编译的时候发现不了， 只有在运行的时候才报错的异常；而运行时异常则是指编译的时候就能发现的异常（ ）
- 26 Exception 由程序处理的异常，分为运行时异常和非运行时异常（ ）
- 27 RuntimeException 异常需要区别对待并且编译器允许忽略它们的原因是，它们通常由代码中的严重错误产生 （ ）
- 28 CalssCastException 指的是试图把一个对象进行非法的类型强制转换，即这个对象不是指定的类，也不是指定的 子类或者超类（ ）
- 29 catch 代码块包含的代码不是用来处理相关的 try 代码块中可能抛出的某种具体类型的代码 （ ）
- 30 try 代码块只能抛出一个异常不能抛出多个异常， try 后面可以跟多个 cathc （ ）

- 31 如果需要关闭一个文件，或者需要释放一项重要资源，可以把相关处理的代码放入一个 finally 代码中以确保完成工作 ()
- 32、IOException 属于 Exception 包 ()
- 33、如果 try 语句块中未产生异常，应用程序会忽略相应 catch 语句块。()
- 34、程序中抛出异常时 (throw ,)，只能抛出自己定义的异常对象。()
- 35、一个异常处理中 finally 语句块只能有一个或者可以没有。()

四、程序阅读题

1、阅读下面代码，写出运行结果。

```
public class TestException{
    public TestException(){}
    boolean testEx() throws Exception{
        boolean ret = true;
        try{
            ret = testEx1();
        }
        catch (Exception e){
            System.out.println("testEx, catch exception");
            ret = false;
            throw e;
        }
        finally{
            System.out.println("testEx, finally; return value=" + ret);
            return ret;
        }
    }
    boolean testEx1() throws Exception{
        boolean ret = true;
        try{
            ret = testEx2();
            if (!ret){
                return false;
            }
            System.out.println("testEx1, at the end of try");
            return ret;
        }
        catch (Exception e){
            System.out.println("testEx1, catch exception");
            ret = false;
        }
    }
}
```

```

throw e;
}
finally{
System.out.println("testEx1, finally; return value=" + ret);
return ret;
}
}
boolean testEx2() throws Exception{
boolean ret = true;
try{
int b = 12;
int c;
for (int i = 2; i >= -2; i--){
c = b / i;
System.out.println("i=" + i);
}
return true;
}
catch (Exception e){
System.out.println("testEx2, catch exception");
ret = false;
throw e;
}
finally{
System.out.println("testEx2, finally; return value=" + ret);
return ret;
}
}
public static void main(String[] args){
TestException testException1 = new TestException();
try{
testException1.testEx();
}
catch (Exception e){
e.printStackTrace();
}
}
}

```

答案：

2、阅读下面代码，写出运行结果

```
class MultiNest {
static void procedure() {
try {
int a = 0;
int b = 42/a;
} catch(java.lang.ArithmeticException e) {
System.out.println("in procedure, catch ArithmeticException: " + e);
}
}
public static void main(String args[]) {
try {
procedure();
} catch(java.lang. Exception e) {
System.out.println("in main, catch Exception: " + e);
}
}
}
```

答案：

3、写出下面代码所捕获的异常类型

```
Public class Exception1{
Public static void main(String args[]){
Try{ int a[]={1,2,3,4,5}, sum=0;
For(int i=0;i<=5;i++)sum=sum+a[i];
System.out.println(" sum= " +sum);
}
Catch(ArrayIndexOutOfBoundsException e){
System.out.println(" ArrayIndexOutOfBoundsException detected " );
}
Finally{
System.out.println(" Programm Finished!! " )}}}
```

答案：

4、写出下面代码所捕获的异常类型

```
Public class Exception2
{
Public static void main(String args[])
{
Try{ int x,y;
```

```

X=15;
Y=0;
System.out.println(x/y);
System.out.println(      “ Computing successfully!      ” );
}
Catch(ArithmeticException e)
{
System.out.println(      “ ArithmeticException caught!      ” )
System.out.println(      “ Exception message:      ” +e.toString());
}
Finally
{
System.out.println(      “ Finally block.      ” )
}

}

```

答案：

- 5、在上例中要求输入文件名， 它只声明所输入文件名是否存在异常并没有去处理。请在划线处上填上适当的子句，使其能捕获异常并正确执行。

```

import java.io.* ;
public class BException{
    public static void main(String[] args){// 获得文件名
        BufferedReader stdin=new BufferedReader(new
InputSteramReader(System.in)) ;
        System.out.printl( 'Filename: ' ) ;
        String s=stdin.readLine;
        BufferedReader filein=null;// 为处理组织文件流
        try{
            Filein= new BufferedReader(new FileReader(s)) ;
            } _____{
                System.err.println(s+ “ 、 cannot be opened for reading ” ) ;
                System.exit(0) ;
            }
        int numerator=Integer.paresInt(filein.readLine( ) ) ; //提取值和计算
商
        int denominator=Integer.parseInt(filein.readLine( ) ) ;
        int quotient=numerator/denominator ;
        System.out.pintln( ) ;
    }
}

```

```

        System.out.println(numerator+ " / " +denominator+ " = " +quotient) ;
        return ;
    }
}

```

6、下列是异常的处理，请完成程序

```

public class number{
    public static void main ( String args[] ){
        int a=9;int b=0;
        try{
            int sum=a/b;
            System.out.println(sum);
        }catch( _____e){
            System.out.println(e.toString());
        }
    }
}

```

7、下面是一个类型转换的程序，请完成程

```

Public calss TryCatch{
    Public static void main(String args[]){
        String t1= '03121 ';
        Double t2=0;
        Try{
            Ti=Double.parseDouble(t2);
            System.out.println( '转换成功 ');
        }catch ( _____e) {
            System.out.println( '转换失败 ');
        }
    }
}

```

8、下面的代码段中 finally 语句块会被执行吗？

```

public class ExceptionExam3
{
    public static void main(String [] args)
    {
        try
        {
            int [] a=new int[3];
            System.exit(0);
        }
        catch(ArrayIndexOutOfBoundsException e)

```

```

        {System.out.println(" 发生了异常 ");}
    finally
    {System.out.println("Finally");}
    }
}

```

答案：

五、程序题

- 1、创建一个 `Exception` 类的子类 `DivideByZeroEx`，代表除数为 0 异常；编写一个 `TestDivideByZero` 类，该类包括一个方法 `div(double a,double b)`，实现两个参数的相除操作，如果 `b` 为 0，则生成异常对象，并抛出异常，否则得到 `a/b` 的结果。然后在 `main` 函数中调用 `div` 方法，然后处理异常。
- 2、编写一个程序，以说明 `catch (Exception e)` 如何捕获各种异常。
- 3、运行 `try-catch` 语句的实例，熟悉 `try-catch` 语句的语法形式和应用、
- 4、异常类型匹配的处理方式。程序功能、首先输出“这是一个异常处理的例子”，然后在你程序中主动地产生一个 `ArithmeticException` 类型被 0 除而产生的异常，并用 `catch` 语句捕获这个异常。最后通过 `ArithmeticException` 类的对象 `e` 的方法 `getMessage` 给出异常的具体类型并显示出来。
- 5、根据所给创建自定义异常类的使用格式，编写一个自定义异常的小程序。
- 6、编写一个程序，捕获两种不同类型的异常。
- 7、编写一个程序，嵌套 `try` 语句。
- 8、编写一个会产生 `NullPointerException`、`NegativeArraySizeException` 和 `IndexOutOfBoundsException` 类的异常程序。通过现实存储在异常对象以及栈的跟踪记录中的信息记录捕获的每个异常。
- 9、编写一个程序，使其在 `for` 循环中随机地调用一次抛出 `ArithmeticException` 类型异常的方法。在方法中捕获这个异常，并在出现异常的时候，通过试用自己定的异常类的对象将迭代的次数传递给该方法。
- 10、编写一个异常类 `MyException`，再编写一个类 `Student`，该类有一个产生异常的方 `public void speak(int m) throws MyException`，要求参数 `m` 的值大于 1000 时，方法抛出一个 `MyException` 对象，最后编写主类，在主类的 `main` 方法中用 `Student` 创建一个对象，让该对象调用 `speak` 方法。

第十章 文件操作

一、选择题

- 1、如要求读取文件中的中间一段内容，最方便的采用下列哪种流来操作？（ ）
A、File stream B、Piep stream C、Random stream D、Filter stream
- 2、当检索一个压缩文件时，首先要建立压缩文件输入流对象。该对象（ ）
A、以选中的压缩文件为参数 B、以 FileInputStream 对象为参数
C、以 InputStreamReader 为参数 D、以 BufferedReader 对象为参数
- 3、下列哪个是 FileReader 文件的创建语句（ ）
A、FileReader(File file) B、read() C、FileWreter(File file)
D、ready()
- 4、File 类的 mkdir 方法的返回值的类型是（ ）
A、boolean B、int C、String D、Integer
- 5、FileReader 类的 ready 方法的作用是（ ）
A、读入一个字节 B、读入一行字节
C、判断文件读入类是否到达文件尾 D、判断文件读入类是否到达文件头
- 6、FileWriter 类的 getEncoding 方法的作用是（ ）
A、取得文件读入类 B、取得文件读入类的字符串
C、取得文件读入类的编码机制 D、设置文件读入类的编码机制
- 7、FileReader 的父类是（ ）
A、BufferedReader B、LineReader C、FilterReader D、Reader
- 8、FileInputStream 类的 skip(long n) 方法的作用是（ ）
A、向上跳过流文件的 n 长度数据 B、跳过流文件的全部数据
C、向上跳过流文件的全部数据 D、跳过流文件的 n 长度数据
- 9、File 类的 delete 方法的返回值的类型是（ ）
A、boolean B、int C、String D、Integer
- 10、FileWriter 类的 write(int c) 方法的作用是（ ）
A、写入一个字符 B、写入一个字符串 C、写入一个整形数据
D、写出一个字符串
- 11、在（ ）情况下用户能使用 File 类。
A、改变当前的目录 B、返回根目录名

C、删除一个文件 D、查找一个文件是否包含文本或二进制信息

12、在 FilterOutputStream 类中，下面是合法的是 ()

A、File B、InputStream C、OutputStream D、FileOutputStream

13、RandomAccessFile 类的 () 方法可用于设置文件定位指针在文件中的位置。

A、readInt B、readLine C、seek D、close

14、RandomAccessFile 类的 () 方法可用于从指定流上读取字符串。

A、readInt B、readLine C、seek D、close

15、RandomAccessFile 类的 () 方法可用于从指定流上读取整数

A、readInt B、readLine C、seek D、close

16、以下哪个类没有实现 FilterInputStream.()

A、BufferedInputStream B、LineNumberInputStream
C、BufferedOutputStream D、DataInputStream

17、下面哪个文件操作类可以实现一次读入多个文件 ()

A、FileReader B、BufferedReader C、FileInputStream D、SequenceInputStream

18、文本文件读入类的创建方法是 ()

A、new BufferedReader(File) C、new FileOutputStream(File)
C、FileInputStream(File) D、new FileReader(File)

19、File 类的 public boolean delete() 方法的作用是 ()。

A、从文件系统中删除该文件。 B、程序顺利结束时从系统中删除文件。

C、判断文件是否存在。 D、当文件不存在时创建文件。

20、File 类的 public String toString() 方法有什么作用。 ()

A、将文件设置为只读。 B、将文件对象的路径转换为字符串返回。

C、返回文件和目录清单。(字符串对象)

D、判断该 File 对象所对应的是否是文件。

21、下面是从文件读取对象的代码，请将代码补充完整。 ()

File theFile = new File(" MyFile ")

//Perhaps check out the file ,

//Create the object output stream for the file

ObjectInputStream objectIn = null;

try{

 objectIn

 =new

 ObjectInputStream(new

 _____ (theFile));


```

    }catch(IOException e ){
    e.printStackTrace(System.err);
    System.exit(1);
    }

```

- A 、 FileOutputStream B 、 DataInputStream C 、 DataOutputStream D、 FileInputStream
- 22、 File 类中以字符串形式返回文件名称的方法是 ()。
- A、 public String getPath() B、 public String getParent()
- C、 public String getName() D、 public boolean isFile()
- 23、下列不是 InputStream 子类的是 ()
- A、 FileInputStream B、 ObjectInputStream C、 CharInputStream
- D、 ZipInputStream
- 24、在读字符文件 Post.dat 时，使用该文件作为参数的类 ()
- A、 BufferedReader B、 DataInputStream C、 DataOutputStream
- D、 FileInputStream
- 25、以下哪个是文本文件读入类 ()
- A、 FileReader B、 FileWriter C、 BufferedReader
- D、 BufferedWriter
- 26、以下哪个是 FileOutputStream 创建语法 ()
- A 、 close() B 、 FileOutputStream(File file) C 、 FileInputStream(File file)
- D、 write(int b)
- 27、文本文件读入类的创建方法是 ()
- A、 new BufferedReader(File) C、 new FileOutputStream(File)
- C、 FileInputStream(File) D、 new FileReader(File)
- 28、(D) 流类的 close 方法不可用于关闭文件。
- A、 FileOutStream B、 FileInputStream C、 RandomAccessFile
- D、 FileWrite
- 29、下面哪个不是属于读取和写入已存在的文件的方式。 ()
- A、 顺序存取 B、 随机存取 C、 直接存取
- D、 间接存取
- 30、字母、数字和特殊符号称为 (B)
- A 、 位 B 、 字节 C、 兆 D、 文件
- 31、File 类的 public File[] listFiles() 方法有什么作用 ()
- A、返回文件和目录清单。(File 对象)
- B、返回文件和目录清单。(字符串对象)
- C、在当前目录下生成指定的目录。
- D、判断该 File 对象所对应的是否是文件。
- 32、JAVA 程序是通过什么来完成输入输出的、 ()
- A、 流 B、 类 C、 main 方法 D、 键盘接收

- 33、计算机中的流是 ()
A、 流动的字节 B、 流动的对象 C、 流动的文件 D、 流动的数据缓冲区
- 34、当输入一个字节流时，要 ()
A、实现 DataInput 接口 B、实现 ObjectInput 接口 C、Serializable 接口 D、继承 Reader 抽象类
- 35、计算机处理的最小数据单元称为 ()
A、位 B、字节 C、兆 D、文件
- 36、一组相关记录称为 ()
A、位 B、字节 C、字符 D、数据库
- 37、为读取的内容进行处理后再输出，需要使用下列哪种流？
()
A、File stream B、Piped stream C、Random stream D、Filter stream
- 38、计算机处理的数据最终分解为 () 组合
A、0 和 1 B、数据包 C、字母 D、2
- 39、FileReader 的父类是 ()
A、BufferedReader B、LineNumberReader C、FilterReader D、Reader
- 40、BufferedReader 的父类是 ()
A、FileReader B、Reader C、PipedReader D、InputStreamReader
- 41、() 对象可显示一个供用户从磁盘中选择某个文件的对话框
A、JFileDialog B、JFileChooser C、JOptionPane D、JOptionPane
- 42、用来检索文件或目录信息的类为 ()
A、File B、FileDirectory C、FileInformation D、DirectoryInformation
- 43、数据流可以分为两种，它们是 ()
A、输入流 B、输出流 C、A 和 B D、不知道
- 44、Java 流被分为字节流、字符流两大流类，两者都作为 () 类的直接子类
A、Exception B、Object C、Throwable D、以上都不是
- 45、字节流继承于 InputStream 和 OutputStream，() 继承于 InputStreamReader 和 OutputStreamWriter
A、数据流 B、字符流 C、输入流 D、输出流

46、属于 Java 输入输出流的、且处理的是 char 类型的类是 ()
A、Reader类 B、InputStream 类 C、OutputStream 类 D、File 类

```
47、import java.io.*;
    public class Test2 {
    public static void main(String args[])throws Exception{
        int a=4;
        BufferedReader br=new BufferedReader(new
InputStreamReader(System.in));
        System.out.println(" 请输入一个数字 ");
        String input=br.readLine();
        int b=Integer.parseInt(input);
        if(b>a){
            int sum=b/a;
            System.out.println(sum);
        }else{
            System.out.println(" 输入错误 ");
        }
    }
}
```

当输入的数字是 8 时，那打印输出的结果是 ()
A、 4 B、 2 C、 6 D、 8

48、以下那个是 java.io 包中一个兼有输入输出功能的类 ()
A. Object B. Serializable
C. RandomAccessFile D. Java.io 中不存在这样的类

49、在与 FileWriter 类结构结合使用时， () 类能够将字符串和数值写入一个文件
A、 OutputFile B、 StreamWriter C、 PrintWriter D、 BuffereReader

50、Java 为了提高程序员的编程效率，定义了专门的类 PipedInputStream，它也像类 ByteArrayInputStream 一样，有一些数据成员，以下关于这些数据成员的描述， () 是错的。
A、 buffer(类型为 byte[]) 用来储存将要读取的字符
B、 in(类型为 int) 表示从自身读取的下一个字节在 buffer 中的位置
C、 out (类型为 int) 表示当从一个 OutputStream 读取下一个字节时，这个字节在 buffer 中存放的位置
D、 PIPE_SIZE(类型为 static int) 表示缓冲大小

51、文件打开过程中所产生的异常类为 ()
A IOException B OpenException C FileNotFoundException D InputOutputException

52、为 16 位 Unicode 字符流提供读和写操作支持的类分别是 ()

- A、FileInputStream 类和 FileOutputStream 类
- B、InputStream 类和 OutputStream 类
- C、FileReader 类和 FileWriter 类
- D、File_Reader 类和 File_Writer 类

53、在 JAVA 中会经常使用 Scanner 类实现从控制台读取用户输入， 下面的选项正确的是（ ）

- A、 Scanner in = new Scanner(System.in);
int x = in.nextInt();
- B、 Scanner in = Scanner(System.in);
int x = in.getInt();
- C、 Scanner in = new Scanner(System.in);
int str = in.next();
- D、 Scanner in = new Scanner(System.in);
int x = in.nextLine();

54、RandomAccessFile 类指针控制方法 public long getFilePointer() 的作用是

()

- A 、 获取当前指针指向文件的位置。
- B 、 将指针移动到参数指定的位置。。
- C 、 指针从当前位置向后移动 n 个字节位置，并返回指针实际移动的字节数。
- D 、 关闭输入输出流。

55、用来读取字符流的类是（ ）

- A、 InputSream B、 OutputStream C、 Reader
- D、 Writers

56、Writer 的 write(int ch) 的作用是（ ）

- A、 将相应于整型实参 ch 的 2 个低位字节的字符写入
- B 、 写入字符串 ch
- C、 将字符数组 ch 的内容写入
- D 、 这是一个抽象方法，从 ch 开始将方法中的 length 个字符写入

57、BufferedReader 类的（ ）方法能够读取文件中的一行

- A、 readLine B、 read C、 line D 、 close

58、字节通常是由（ ）位组成

- A 、 4 B、 8 C、 1 D、 2

59、BufferedReader 类的（ ）方法可关闭已经打开并进行读取操作的一个文件

- A、 flush B、 closeBufferedReader C、 closeFile D 、 close

60、以下属性或 FileReader 类的常用属性或方法（ ）

- A、 getEncoding() B、 close() C、 write(int c) D、

readLine()

61、以下那个是文本文件读入类 ()
A、 FileReader B、 FileWriter C、 BufferedReader D、
BufferWriter

62、数字、字母、特殊字符均属于 ()
A、 常量 B、 整数 C、 字符串 D、 字符

63、ByteArrayOutputStream 的含义是 ()
A、 从字节数组读取的输入流 B、 向字节数组写入
的输出流
C、 写文件的输出流 D、 缓冲输出流

64、当文件内无可读取的数据时， readLine 方法 ()
A、 返回 0 B、 抛出 EndOfFileException 异常
C、 返回 null D、 以上答案都不对

65、保存在文件中的数据被称为 ()
A、 永久数据 B、 位 C、 二级数据 D、 数据库

66、() 包提供了用来实现文件处理的类及其方法
A、 java.io B、 java.files C、 java.stream D、 以上答案都
不对

67、System.in 的作用是 ()
A、 控制台输入数据的读取 B、 控制带输出的数据读
取

C、 磁盘文件的顺序读写 D、 对文件的随机访问
68、JComboBox 类的 () 方法可删除 JComboBox 中的所有选项
A、 clear B、 removeAll C、 clearAll D、
removeAllItems

69、以下哪个流类是字符流 ()
A、 InputStream B、 OutputStream
C、 Writer D、 BufferdReader

70、为打开一个用于读取的文件，需使用一个 () 对象
A、 InputFile B、 StreamReader C、 BufferedResder D、
FileReader

71、下面程序的输出结果是： Enter a characters. 135abc ,空白处应
当填 ()。

```
import java.io.* ;  
class BRRead{  
public static void main(String[]args)throws IOException{  
char c;  
BufferedReader br = new BufferedReader(new  
InputStreamReader(System.in));  
System.out.println( "Enter a character");
```

```
System.out.println(c);  
    }  
}
```

A、 c = (char)br.read();

B 、 c=br.read();

C、 c =(char)br.readLine()

D、 c = br.readLine()

72、Java 将文件看做为一个字节 () 序列

A、 流 B、 循环 C、 字符串 D、 记录

73、File 类的常用方法 public boolean isFile() 的作用是。 ()

A、 判断该 File 对象所对应的是否是文件。

B、 判断该 File 对象所对应的是否是目录。

C、 返回文件的最后修改时间。

D、 在当前目录下生成指定的目录。

二、填空题

1、目录操作的类是 _____，所在的类包是 _____。

2、文件操作分为两种，分别是 _____和 _____操作。

3、读入与写出文本文件的类是 _____,所在类包是 _____。

4、逐行读入与写出文本文件的类是 _____，所在类包是 _____。

5、读入与写出流文件的类是 _____，所在类包是 _____。

6、FileReader 类读入一个字节的的方法是 _____。

7、目录的创建语法如下、

(1)、File.mkdir(), 根据 File 类定义的目录 _____创建目录，返回值是 _____。

(2)、File.mkdirs()，根据 File 类定义的目录字符串创建目录，返回值是 boolean，该函数可以创建目录字符串包括的所有目录。

8、目录的删除语法如下、

File.delete()，删除 File 类的目录或者 _____，返回值是 _____,如果要删除的目录 _____不能删除。

9、读入与写出文本文件的类是 _____类与 _____类，所在的类包是 _____。

10、public boolean mkdir() 的作用是根据当前对象的抽象路径生成一个 _____。

11、public String[]list() 的作用是列出路径下的文件名和 _____。

12、File 类中提供了三种用来生成文件或者目录的构造方法。

public File(String path): 通过给定的路径来创建 _____。

public File(String path, _____):使用父目录字符串和子抽象路径创建文件对象。

public File(File dir,String name): 使用 _____和子抽象路径创建文

件对象。

13、文件的读写中两个最常用的流类是和_____，它们生成与文件链接的字节流。

14、要向文件中写数据，需用 `FileOutputStream` 定义的_____方法。

15、文件输入/输出流采用了_____读写，又是需要有选择的读写文件，例如读写一行或几行。_____可实现这种操作，它能让你从文件的不同位置读写不同长度的数据。

16、`RandomAccessFile` 所实现的接口是_____和_____

17、读取和写入文件已存在的文件有两种不同的方式，分别是和_____，后一种方式也称为直接存取。

18、`InputStream` 类的 `public void close()` 方法作用是关闭_____，并释放流占用的系统资源。

19、`DataInputStream` 类和_____类是从过滤流类继承过来，这两个流的对象均不能独立地实现数据的输入和输出处理，必须与其他输入流与输出流对象一起使用，才能实现不同类型数据的读写。这两个类与_____类和 `FileOutputStream` 类相配合可以完成对格式数据的读写。

20、完成随机文件读写的类与_____Java 中大部分的输入输出流类不同，没有针对输入、输出对应出现两个相应的类，它是由_____直接继承而来的，仅使用一个_____类来完成。所以 `RandomAccessFile` 类的对象既可以用于数据输入，也可以用于_____。

21、`File` 对象一共有四种构造方式，它们分别是：

`public File(File parent, String child)` ;从 `parent` 表示的_____名和 `child` 表示的文件名创建对象。

`public File(String pathname)` ;将给定的路径名称字符串转换成一个抽象的路径名，构造一个新的文件对象。

`public File(String parent, String child)` 该构造函数将 `pathname` 分成两部分_____和_____，参数 `parent` 表示目录或文件所在_____，参数 `child` 表示目录或文件名称。

`public File(URI uri)` ;从一个 `URI` 创建 `File` 对象，有关 `URI` 的类容将在后面有介绍。

22、`InputStreamReader` 是_____和_____之间的桥梁，由于 `System.in` 是字节流，需要用它来包装之后变为字符流供给使用。

23、`File` 类并没有指定信息怎样从文件读取或向文件存储，而需要由像_____与_____等这些类来实现。

24、`DataInputStream` 和 `DataOutputStream` 分别实现了 `java.io` 包中的和_____接口，能够读写 Java 基本数据类型的数据和 `Unicode` 编码格式的字符串。

25、`System.in` 是_____类的对象，调用_____方法就可以实

现标准输入的读操作。

26、JAVA 中定义了 _____ 和 _____ 这两种类型的流。

27、JAVA 中，控制台输入由从 _____ 读取数据来完成。

28、从 BufferedReader 读取字符，我们一般用 _____ 方法。

29、文件操作分为两种，分别是 _____ 和 _____ 操作。

30、文本文件读入类的创建方法是 _____。

31、在 Java 中，类 InputStream 定义了以下的函数、
`public int read();`
`public int read(_____);`
`public int read(_____, int off, _____);`
这是 read()函数的三种形式，第一种形式实现的是从 _____ 读取一个字节，然后保存为一个 _____ 类型的整数保存起来。后两种形式返回的都是读取的 _____。区别是第二种方法保存在下标从 _____ 开始的数组中，而第三种不仅规定下标从 _____ 开始，而且规定了长度为 len。

32、字符流的类包括 StringBufferInputStream。它和 ByteArrayInputStream 类几乎没有区别，只不过后者使用一个作为缓冲。

33、现在的程序都离不开文档了，各种各样的记录都保存在文档里。程序启动时读取文档，退出时也将最新记录保存在文档里，Java 为储存文件提供了两个类 _____ 和 _____ 等。

34、路径分为 _____ 路径和 _____ 路径。_____ 路径是指一个完整的路径，系统不需要任何额外的信息都可以储存文件；_____ 路径就不同，系统需要从其他路径中获取信息才能确定那个位置。

35、InputStreamReader 是 _____ 和 _____ 之间的桥梁，由于 System.in 是字节流，需要用它来包装之后变为字符流供给使用。

36、Java 中的流类包含了 InputStream、_____、Reader、_____ 四类，前两者面向字节，称为 _____；后两者面向 _____，称为字符流。

37、printStream 只用于输出，新增了 _____ 和 println 方法，Reader 和 Writer 类是面向字符的，优点是可以直接 _____ 字符，而不用转换成 _____。RandomAccessFile 类既可以读又可以写，同时它还支持随机读写，因此是文件操作中功能最强大的类，也是首选类。

38、输入输出是指应用程序与外部设备及其他计算机进行 _____ 的操作。

39、java.io 包中提供了各种各样的输入输出流类，它们都是 _____ 的直接子类，每一个流类代表一种特定的 _____ 或 _____。

40、在这些类中，作为 Object 类的直接 _____，基本输入流（InputStream）和基本输出流（OutputStream）是处理以 _____ 为基本单位的字节流类，读写以 _____ 为单位进行。

41、File 类并没有指定信息怎样从文件读取或向文件存储，而需要由

像_____与_____等这些类来实现。

42、FileInputStream 和 FileOutputStream 直接从_____和继承而来

43、DataInputStream 和 DataOutputStream 分别实现了 java.io 包中的和_____接口，能够读写 Java 基本数据类型的数据和 Unicode 编码格式的字符串。

44、Java 系统事先定义的两个流对象分别与系统的标准输入和标准输出相连接。这两个流对象分别是_____和_____。

45、System.out 是 PrintStream 类的对象，PrintStream 类是类的子类。

46、流是输入/输出设备的一种_____表示，这些设备是数据源头或是数据终点。

47、当向一个流中写入数据的时候，这个流就被称为_____。

48、目录操作的类是_____，所在的类包是_____。

49、读入与写出文本文件的类是_____，所在类包是_____。

50、逐行读入与写出文本文件的类是_____，所在类包是_____。

51、读入与写出流文件的类是_____，所在类包是_____。

52、FileReader 类读入一个字节的的方法是_____。

53、字节流由_____类和_____类的子类来表示。

54、读取和写入文件已存在的文件有两种不同的的方式，分别是和_____，后一种方式也称为直接存取。

55、在 System 类中有三个预先定义好的成员变量、_____，和_____。

56、Reader 是一个抽象类，它的一个具体的子类是 InputStreamReader，该子类把_____转换成_____。

57、接口_____和_____提供了对对象直接读/写的一组方法，类 ObjectInputStream 和 ObjectOutputStream 分别实现了这两个接口。

58、StringBufferInputStream 用来将一个或多个_____转换为进行读取。

59、System.in 是_____类的对象，调用_____方法就可以实现标准输入的读操作。

60、System.out 是_____类的对象，PrintStream 类是_____类的子类，其中定义了可输出多种不同类型数据的方法 print() 和 println() 方法。

61、作为 Object 类的直接子类，基本输入流（InputStream）和基本输出流（OutputStream）是处理以_____字节为基本单位的字节流类，读写以_____为单位进行；Reader 和 Writer 类是专门处理_____位字符流的类，读写以_____（Unicode）为单位进行。

- 62、BufferedReader 类的含义是缓冲_____。
- 63、在 java.io 包中定义的大多数类是实行流式操作的，但文件操作类不是，它独立于 InputStream 类和 OutputStream 等类，包括_____、_____、_____、_____等类，它们是专门处理文件的。
- 64、InputStream 类的 public void close()方法作用是关闭_____，并释放流占用的系统资源。
- 65、DataInputStream 类和_____类是从过滤流类继承过来，这两个流的对象均不能独立地实现数据的输入和输出处理，必须与其他输入流与输出流对象一起使用，才能实现不同类型数据的读写。这两个类与_____类和 FileOutputStream 类相配合可以完成对格式数据的读写。
- 66、完成随机文件读写的类与 Java 中大部分的输入输出流类不同，没有针对输入、输出对应出现两个相应的类，它是由_____直接继承而来的，仅使用一个_____类来完成。所以 RandomAccessFile 类的对象既可以用于数据输入，也可以用于_____。
- 67、public int read()、该方法从输入流的当前位置读取一个的数据，并返回一_____型值，如果当前位置没有数据则返回 -1。该方法为 abstract，由子类来具体实现。
- 68、readInt() 方法读取一个_____型数，writeInt(int v)方法一个 int 型数。readUTF() 和 writeUTF(String str) 可以按 UTF-8 编码读写一个_____。
- 69、Sun 的文档对 ObjectOutputStream&ObjectOutputStream 类有详细的阐述。可以被当作流传送的对象自身必须实现_____ java.io.Serializable 接口或者 java.io.Externalizable 接口，它们被称为_____和外部化。_____(Serialization) 是 Java API 内建的，不需要再添加什么，它的缺点是占用的空间不大。
- 70、和其他输出流不同的是，PrintStream 类有一个方法 print，参数除了和 OutputStream 是同样的之外，多了 char 和_____。也就是说，这个类可以直接输出字符而不用经过手动类型的转换，虽然它是通过_____间接实现的。
- 71、Reader 类和 Writer 类增加了对字符 char 的支持，因此它是_____导向的 (char-oriented)。原来的 InputStream 类和 OutputStream 类不能直接存储的 char 变量，每次都要进行_____转换。

三、判断题

- 1、public boolean mkdir() 的作用是根据当前对象的抽象路径生成一个目录。()
- 2、File.delete() 方法可以删除不为空的目录。()
- 3、File.mkdirs() 方法可以创建目录字符串包括的所有目录。()
- 4、BufferedReader.readLine() 方法可以读入新行字符 “ \n ”。()

- 5、BufferWriter.newLine() 方法的作用是创建新行。 ()
- 6、FileInputStream.read() 方法的作用是读取一个 byte 字节数据。()
- 7、public String[]list() 的作用是列出路径下的文件名和生成文件。()
- 8、File 类中提供了两种种用来生成文件或者目录的构造方法，分别是 public File(String path) 和 public File(String path,String name)。()
- 9、FileOutputStream 值得是文件输入流。()
- 10、public String[]list() 的作用是列出路径下的文件名和目录名。()
- 11、目录的删除语法如 File.delete()，删除 File 类的目录或者文件，返回值是 boolean,且就算要删除的目录为空也能删除。()
- 12、FileReader 类一般有 FileReader(File file) 和 FileWriter(File file) 两种创建语法。()
- 13、FileWriter(File file)语句的作用是根据 File 类创建文件读入类。()
- 14、FileReader 类中的 ready()方法的作用是判断文件读入类是否到达文件尾。()
- 15、FileReader 类中 getEncoding() 方法的作用是取得文件读入类的编码机制。()
- 16、File 类的 delete 方法的返回值的类型是 String。()
- 17、FileReader 可以处理多个文件 ()
- 18、因为 InputStream、OutputStream、Reader、Writer 是基本输入输出流的抽象类，所以都能用于直接创建对象来完成输入输出操作 ()
- 19、InputStream 类是个抽象类，作为字节输出流的直接或间接的父类，它定义了许多有用的所有子类必须的方法 ()
- 20、File 类并没有指定信息怎样从文件读取或向文件存储，而需要由像 FileInputStream 与 FileOutputStream 等这些类来实现。()
- 21、File 类的常用方法 public String toString() 的作用是将文件对象的路径转换为字节返回。()
- 22、InputStream 类是个抽象类，作为字节输入流的直接或间接的父类，它定义了许多有用的、所有子类必须的方法，包括读取、移动指针、标记、复位、关闭等方法。()
- 23、由 InputStream 类和 OutputStream 类派生出来的一些常用的子类。子类中一般都重写上面说明的父类方法，并增加新的方法或实现 java.io 包中的某些接口，以适应读写各种不同格式数据流的需要，以提高输入输出的效率，将低层内容封装起来。()
- 24、File 类的常用方法 public String toString() 是将文件对象的路径转换为字符串返回。()
- 25、File 类的常用方法 public FileInputStream(String name) 的作用是为参数 name 所指定的文件名创建一个 FileInputStream 对象。()
- 26、RandomAccessFile 类实现随机读写，且可以在文件的任意位置执行数据读写。()

- 27、类 `InputStream` 类的 `void close()` 方法的作用是关闭流并且释放流的系统资源。()
- 28、`SequeceInputStream` 用来把两个或更多的 `InputStream` 输入流对象转换为单个 `InputStream` 输入流对象使用。()
- 29、可以使用从 `FileInputStream` 对象或 `RandonAccessFile` 对象活的的 `FileChannel` 对象来写出文件。()
- 30、`FileReader` 可以处理多个文件()
- 31、程序员必须创建 `System .in`,`System .out` 和 `System .err` 对象。()
- 32、`BufferedReader` 可以读入字符串信息()
- 33、`FileInputStream` 类可以读入流信息()
- 34、`FileOutputStream` 类可以写出流信息()
- 35、流可以理解为信息以串的形式传播，这个串的单位可以是比特，也可以是字符（实际上还是比特()）。
- 36、很多输入输出类都直接过间接的从两个基类派生而来，这两个基类分别是 `InputStream` 和 `OutputStream` 两个抽象类，但有些输入输出类比较特殊，不是由这两个基类继承()
- 37、在 JAVA 中，`InputStream` 定义了四种形式参数不同的 `read` 函数，分别是 `public int read();public int read(byte[]b);public int read(byte[]b,int len);public int read (byte[]b,int off,int len)`。()
- 38、Java 中，字符流不是字节流，它们之间有区别的。()
- 39、Java 为储存文件提供了两个类 `FileInputStream` 和 `FileOutputStream`。()
- 40、`FileInputStream` 类的功能在于把文件以直接的形式从磁盘读取出来，即使读的是字符流。()
- 41、在 Java 中，提供了数据输入输出流，它们独立于机器，而且可以读取基本的数据类型（`primitive`），这在以前的语言机制是做不到的。()
- 42、输入输出是指应用程序与外部设备及其他计算机进行数据交流的操作()
- 43、流式输入输出的特点是数据的获取和发送沿数据序列的顺序进行()
- 44、Java 语言中提供了比较完善的处理输入输出数据的功能，实现这些 I/O 操作的类和接口都在 `java.io` 包中。()
- 45、因为 `InputStream`、`OutputStream`、`Reader`、`Writer` 是基本输入输出流的抽象类，所以都能用于直接创建对象来完成输入输出操作()
- 46、`InputStream` 类是个抽象类，作为字节输出流的直接或间接的父类，它定义了许多有用的、所有子类必须的方法()
- 47、`File` 类并没有指定信息怎样从文件读取或向文件存储，而需要由像 `FileInputStream` 与 `FileOutStream` 等这些类来实现。()

- 48、由于 `FileInputStream` 和 `FileOutputStream` 直接从 `InputStream` 和 `OutputStream` 继承而来，并没有对其功能扩充，因此所有的读写操作只能以字节为单位对二进制数据进行处理。（ ）
- 49、`DataInputStream` 类和 `DataOutputStream` 类是从过滤流类继承过来，这两个流的对象均不能独立地实现数据的输入和输出处理，必须与其他输入流与输出流对象一起使用，才能实现不同类型数据的读写。（ ）
- 50、`RandomAccessFile` 类的对象仅可以用于数据输入，不能用于数据输出。（ ）
- 51、Java 系统事先定义的两个流对象分别与系统的标准输入和标准输出相连接。这两个流对象分别是 `System.in` 和 `System.out`。（ ）
- 52、`System` 类是 `java.lang` 包中的一个很重要的类，它的所有方法和属性都是非静态的（ ）
- 53、与大多数输入输出方法一样，`System.in.read()` 方法可能引发 `IOException` 异常，因此必须使用异常处理的 `try{ ... }catch(IOException e){..}` 语句块来捕获异常并进行异常处理。（ ）
- 54、`println()` 方法在向屏幕输出数据完毕时自动换行，光标停留在下一行的第一个字符位置。（ ）
- 55、流是输入 / 输出设备的一种抽象表示，这些设备是数据源头或是数据终点。（ ）
- 56、当向一个流中写入数据的时候，这个流就被称为输出流。（ ）
- 57、流文件读入与写出类可以操作其它格式的文件，如 `Word`、`Excel` 格式（ ）
- 58、`File.delete()` 方法可以删除不为空的目录。（ ）
- 59、`File.mkdirs()` 方法可以创建目录字符串包括的所有目录。（ ）
- 60、`BufferedReader.readLine()` 方法可以读入新行字符 “`\n`”。（ ）
- 61、`BufferWriter.newLine()` 方法的作用是创建新行。（ ）
- 62、`FileInputStream.read()` 方法的作用是读取一个 `byte` 字节数据。（ ）
- 63、文件流的 `getChannel()` 方法返回文件的 `FileChannel` 对象。（ ）
- 64、缓冲区包含将要写入文件或者从文件读取的数据。而且 `ByteBuffer` 对象不能被 `I/O` 操作直接引用。（ ）
- 65、可以使用从 `FileInputStream` 对象或 `RandomAccessFile` 对象活的 `FileChannel` 对象来写出文件。（ ）
- 66、`PrintStream` 中提供了方便的格式输出方法。（ ）
- 67、`PushbackInputStream` 不能预读数据中开始的一个字节。（ ）
- 68、`SequecelInputStream` 用来把两个或更多的 `InputStream` 输入流对象转换为单个 `InputStream` 输入流对象使用。（ ）
- 69、`int read()` 方法的用途是从输入流中读取下一个字符流数据，其返回值是 `0-255` 之间的一个整数。（ ）
- 70、类 `InputStream` 类的 `void close()` 方法的作用是条用这个函数可以关

闭流并且释放流的系统资源。()

71、RandomAccessFile 类实现随机读写，且可以在文件的任意位置执行数据读写。()

72、InputStream、OutputStream、Reader、Writer 是基本输入输出流的抽象类，可以用于直接创建对象来完成输入输出操作。()

73、File 类的常用方法 public FileInputStream(String name) 的作用是为参数 name 所指定的文件名创建一个 FileInputStream 对象。()

74、File 类的常用方法 public String toString() 是将文件对象的路径转换为字符串返回。()

75、由 InputStream 类和 OutputStream 类派生出来的一些常用的子类。子类中一般都重写上面说明的父类方法，并增加新的方法或实现 java.io 包中的某些接口，以适应读写各种不同格式数据流的需要，以提高输入输出的效率，将低层内容封装起来。()

76、.BufferedReader 是 Reader 的一个子类，它具有缓冲的作用，避免了频繁的从物理设备中读取信息。()

77、InputStream 类是个抽象类，作为字节输入流的直接或间接的父类，它定义了许多有用的、所有子类必须的方法，包括读取、移动指针、标记、复位、关闭等方法。()

78、File 类的常用方法 public String toString() 的作用是将文件对象的路径转换为字节返回。()

79、如果顺序文件中的文件指针不是指向文件头，那么必须先关闭文件，然后在再打开它才能从 文件头开始读。()

80、在随机存取中查找指定记录时不必检查每一条记录。()

81、随机存取文件中的记录必须等长。()

82、seek 方法必须以文件头为基准进行查找。()

83、Java 中的每个文件均以文件结束标记 (EOF)，或者以记录在系统管理数据结构中的一个特殊的字节编号结束。()

84、如果要在 Java 中进行文件处理，则必须使用 Java .swing 包。()

85、InputStream 和 OutputStream 都是抽象类。()

86、Java 语言中对文件没有结构要求，所以需要使用时随机存取的程序必须自己建立它们。()

87、当把一个 RandomAccessFile 流与一个文件相联是，读写数据的位置由文件位置指针来 指定，并且所有数据均以基本数据类型来读写。()

四、读程序题

1、下面程序的输出结果是： Enter a characters. 135abc ,将程序补充完整。

```
import java.io.* ;  
class BRRead{
```

```

        public static void main(String[]args) _____{
            char c;
            BufferedReader br = new BufferedReader(new
InputStreamReader(System.in));
            System.out.println( "Enter a character ");

            _____
            System.out.println(c);
        }
    }

```

2、下面是从文件读取对象的代码，请将代码补充完整。

```

        File theFile = new File( "MyFile " )
        //Perhaps check out the file ,
        //Create the object output stream for the file
        ObjectInputStream objectIn = null;
        try{
            objectIn _____ =new _____ ObjectInputStream(new
(theFile));
        }catch( _____ e ){
            e.printStackTrace(System.err);
            System.exit(1);
        }
    }

```

3、标准输入输出示例，请将代码补充完整。

```

import java.io.*;
public class StandardIO1{
    public static void main(String []args){
        int ch;
        System.out.println(" 请输入一行字符 ");
        try{
            while((ch= _____)!='\r')
                System.out.write(ch);
        }catch( _____ e){
            System.out.println(e.toString());
        }
        System.out.write('\n');
    }
}

```

4、请将下列键盘接收数据的代码补充完整。

```

import java.io.* ;

```

```

    class BRRead{

```



```

        public static void main(String[]args) _____{
            char c;
            BufferedReader br = new _____(new
InputStreamReader(System.in));
            c = _____;
            System.out.println(c);
        }
}

```

5、阅读以下代码，并将空处填写完整。

```

import java.io.*;
    public class Test2 {
        public static void main(String args[])throws Exception{
            int a=4;
            BufferedReader br=new BufferedReader(new _____
(System.in));
                System.out.println(" 请输入一个数字  ");
                String input= _____;
                int b=Integer.parseInt(input);
                if(b>a){
                    int sum=b/a;
                    System.out.println(sum);
                }else{
                    System.out.println(" 输入错误 ");
                }
            }
        }
}

```

当输入的数字是 8 时，那打印输出的结果是 _____。

6、以下是使用 FileInputStream 类与 FileOutputStream 类复制文件，请将空白处补充完整。

```

import java.io.*;
class ReadWriteFile{
    public static void main(String[] args) {
        String file1,file2 ;
        int ch = 0 ;
        file1 = "readme.txt" ;
        file2="readme.bak";
        try{
            FileInputStream fis = new _____;
            _____fos=new FileOutputStream(file2);
            int size= _____;
            System.out.println(" 字节有效数、 "+size);
        }
    }
}

```



```

while ((ch=fis.read())!=-1){
System.out.write(ch);
fos.write(ch);
}
fis.close();
fos.close();
}
catch (IOException e){
System.out.println(e.toString());
}
}
}

```

7、下面是从文件中读取字符数据的类，请补全空白处的代码。

FileReader:按照字符方式读取文件内容

```

import java.io.*;
public class Hello{
public static void main(String args[]){
try{
    BufferedReader br=new _____(new
FileReader("/home/wuxiaoxiao/1.txt"));
    String in;
    while(( _____)
System.out.println(in);
}catch(IOException e){
System.out.println(e);
    }
}
}
}

```

8、下面是从键盘上读取字符数据的类，请补全空白处的代码。

```

public class Hello{
public static void main(String args[]){
try{
    _____
    BufferedReader br=new _____
BufferedReader(new
(System.in));
    System.out.println(" 输入 :");
    _____
System.out.println(" 输入的数据是 :"+str);
}catch(IOException e){
System.out.println(e);
}
}
}

```

```
}
```

9、请将下列代码补充完整。

```
import java.io.*;
import java.util.StringTokenizer;
public class FileOperate {
    private String message;
    public FileOperate() {
    }
    public String readTxt(String filePathAndName,String encoding)
    {
        encoding = encoding.trim();
        StringBuffer str = new StringBuffer("");
        String st = "";
        try{
            FileInputStream fs = new FileInputStream(filePathAndName);
            InputStreamReader isr;
            if(encoding.equals("")){
                isr = new InputStreamReader(fs);
            }else{
                isr = new _____(fs,encoding);
            }
            BufferedReader br = new BufferedReader(isr);
            try{
                String data = "";
                while((data = br.readLine())!=null){
                    str.append(data+" ");
                }
            }catch(Exception e){
                str.append(e.toString());
            }
            st = str.toString();
        }catch(IOException es){
            st = "";
        }
        return st;
    }
}
```

10、请将下面代码补充完整。

```
public String createFolder(String folderPath) {
    String txt = folderPath;
    try {
```

```

        _____ myFilePath = new java.io.File(txt);
        txt = folderPath;
        if (!myFilePath.exists()) {
            myFilePath. _____;
        }
    }
    catch (Exception e) {
        message = "创建目录操作出错 ";
    }
    return _____;
}

```

11、下面是从键盘上读取字符数据的类，请补全空白处的代码。

```

public class Hello{
    public static void main(String args[]){
        try{
            _____
            BufferedReader _____ br=new _____ BufferedReader(new
(System.in));
            System.out.println(" 输入 :");

            _____
            System.out.println(" 输入的数据是 :"+str);
        }catch(IOException e){
            System.out.println(e);
        }
    }
}

```

12、下面是从文件中读取字符数据的类，请补全空白处的代码。

```

FileReader:按照字符方式读取文件内容
import java.io.*;
public class Hello{
    public static void main(String args[]){
        try{
            BufferedReader br=new _____(new
FileReader("/home/wuxiaoxiao/1.txt"));
            String in;
            while(( _____)
            System.out.println(in);
        }catch(IOException e){
            System.out.println(e);
        }
    }
}

```

13、以下是使用 `FileInputStream` 类与 `FileOutputStream` 类复制文件，请将空白处补充完整。

```
import java.io.*;
class ReadWriteFile{
public static void main(String[] args) {
String file1,file2 ;
int ch = 0 ;
file1 = "readme.txt" ;
file2="readme.bak";
try{
FileInputStream fis = new _____;
_____ fos=new FileOutputStream(file2);
int size= _____;
System.out.println(" 字节有效数、 "+size);
while ((ch=fis.read())!=-1){
System.out.write(ch);
fos.write(ch);
}
fis.close();
fos.close();
}
catch (IOException e){
System.out.println(e.toString());
}
}
}
```

五、问答题

- 1、流的概念？
- 2、简述 `FileInputStream` 和 `FileOutputStream`。
- 3、写出 `InputStream` 定义的 `read()`函数的三种形式，并简要的写出它们之间的区别。
- 4、简述 `FileReader` 类与 `FileWriter` 类的创建语法。
- 5、简述 `BufferedReader` 类与 `BufferWriter` 类的创建语法。
- 6、简述 `FileInputStream` 类和 `FileOutputStream` 类的创建语法。
- 7、已知 `java.lang` 这个包中有一个名为 `System` 的类，这个类中有三个预先定义好的成员变量 `in`,`our` 和 `err`。现在请你简要的写出他们的作用。
- 8、简述类 `OutputStream` 的几个方法及其作用。
- 9、JAVA 语言是否可以读入和写出文本格式的文件？如果可以，则是使用的什么类？
- 10、JAVA 语言是否可以读入和写出 `xls` 格式的文件？如果可以，则是

使用的什么类？

11、Serialization 和文件的读入、写出有什么关系？

12、是否可以利用 JAVA 设计分不同进程下载文件的程序？如果可以则是使用的什么方法？

13、文件操作类是否可以读入多个文件？什么类可以完成这个操作？

14、FileReader 和 BufferedReader 类有什么区别？

15、什么叫流？流式输入输出有什么特点？

16、Java 流被分为字节流、字符流两大流类，两者有什么区别？

17、File 类有哪些构造函数和常用方法？

18、RandomAccessFile 有哪些构造函数和读写数据的常用方法？

19、如果要完成将各种数据的写入，需要将一个文件输出流对象 `fout` 与一个格式输出流对象 `out` 相连，要用到什么关键代码？

20、简述 DataInputStream 类和 DataOutputStream 类。

21、简述 Java 语言中的 I/O 流。

22、简述 JAVA 中的随机文件流。

23、简述读写内存及其几个类。

24、什么是管道流。

六、编程题

1、编写一个应用程序，完成文件的拷贝功能，文件名从命令行得到。

2、请编写用 `RandomAccessFile` `oureFile`; `String s= " information to Append \n mon ! "`;

3、利用文件输入输出流编写一个实现文件拷贝的程序，源文件名和目标文件名通过命令行参数传入。

4、编写一个程序，在当前目录下创建一个子目录 `test`，在这个新创建的子目录下创建一个文件，并把这个文件设置成只读。

5、编写一个程序，从键盘输入一串字符，从屏幕输出并将其存入 `a.txt` 文件中。

6、随机读取文件中的字符信息。

7、从键盘敲入一系列字母，将其存储到文件中。

8、从键盘敲入一系列字母，将其存储到文件中，对其进行升序排序后，存到另一个文件中，并显示在屏幕上。

9、编写一段程序，将 `c:\jdk1.3\example\newFile.txt` 中保存的信息输出到屏幕上。

10、编写一段程序，实现从键盘输入以“？”结束的一串字符，然后将该串中“？”前面的所有字符顺序写入磁盘文件 `c:\jdk1.3\example\newFile.txt` 中。

11、编写程序，接收用户从键盘输入的数据，回车后保存到文件 `test.txt` 中。若用户输入符号 `#`，则退出程序。

12、某人在玩游戏的时候输入密码 `123456` 后成功进入游戏（输错 3 次则被强行退出），要求用程序实现密码验证的过程。

- 13、创建一个输入“你好！”的 JAVA 语言程序，在屏幕上显示出来。
- 14、编程、编写一个截取字符串的函数，输入为一个字符串和字节数，输出为按字节截取的字符串。但是要保证汉字不被截半个，如“我 ABC ” 4，应该截为“我 AB”，输入“我 ABC 汉 DEF”，6，应该输出为“我 ABC”而不是“我 ABC+ 汉的半个”。
- 15、编写一段程序，要求从键盘输入 123bc 并显示出来。
- 16、编写一段使用 FileInputStream 类与 FileOutputStream 类复制文件的代码。
- 17、编写一段代码，使其能够将读入的信息用字节数组存放，转换成字符串后再输出。
- 18、编写一段程序，将 c:\jdk1.3\example\newFile.txt 中保存的信息输出到屏幕上。
- 19、编写代码实现从键盘输入整型、单精度型、布尔型数据、然后将这些数据写入磁盘。
- 20、编写程序将磁盘文件 c:\jdk1.3\example\dataFile.txt 中保存的各类数据复制到磁盘文件 c:\jdk1.3\example\outFile.txt 中。
- 21、利用文件输入输出流编写一个实现文件拷贝的程序，源文件名和目标文件名通过命令行参数传入
- 22、编写一个程序，从键盘输入 10 个整数，并将这些数据排序后在标准输出上输出
- 23、编写一个程序，从键盘输入一串字符，从屏幕输出并将其存入 a.txt 文件中
- 24、编写一个程序，从键盘输入一串字符，统计这串字符中英文字母、数字、其他符号的字符数。
- 25、编写一段代码，使其实现从键盘输入 8，能得到一个值为 2 的作用。

第十一章 GUI 界面设计

一、选择题

- 1、利用 () 可以指定内容面板的背景颜色
A、setBackground B、setBackColor C、setRGB D、setColor
- 2、JComboBoxes ,JButton 和 JPanel 都属于 () 的实例
A 、 平台 B、 高级语言 C 、 方法 D、 GUI 组件
- 3、字体属性属于控制 JLabel 中所显示字体的 ()
A、 大小 B、 字形 C 、 名字 D、 以上都正确
- 4、范围属性用来控制 ()
A、 文本大小 B、 组件位置 C、 组件大小 D、 B 和 C
- 5、一个 JLabel 组件通过指定 () 现实文本
A、 setCaption B、 setData C、 setText D 、 setName
- 6、使用 () 可以对齐 JLabel 中的文本
A 、 setAilgnment B 、 setCenter C 、 setRight D 、 setHorizontalAlidnment
- 7、RGB 值可用于指定 ()
A 、 JLabel 的大小 B、 某个颜色组件 C 、 窗口的大小 D、 JFrame 中的组件
- 8、像素是 ()
A 、 图像上的一个元素 B 、 某种字体的测量法
C、 字体集 D、 决定某个组件位置的属性
- 9、JLabel 上的文本是通过 () 来指定的。
A 、 setLable B、 changeLable C、 setText D、 changeText
- 10、JLabel 的范围和大小是通过 () 来指定的
A 、 setSizeAndLocation B、 setBounds C、 以上都不对 D、 A 和 B
- 11、使用 () 设置 JButton 上的文本
A 、 setName B、 setText C、 setTile D、 setFace
- 12、将 JTextField 的()属性设置为 false 可防止用户对该 JTextField 进行编辑
A 、 文本 B、 大小 C 、 可编辑 D、 horizontalAlignment
- 13、为对齐 JTextField 中的文本 , horizontalAlignment 属性设置为 ()
A、 JTextField.Right B J、 TextFiel.RIGHT C、 JTextField.right D 、 以上答案都对
- 14、改变 () 属性将改变 JTextField 的位置大小
A、 文本 B、 大小 C、 范围 D、 位置

- 15、在自定义 JLabel 时，可指定该 JLabel 的 ()。
- A、 文本对齐方式 B、 文本 C、 大小 D、 以上答案均正确
- 16、改变 () 属性值将改变 JTextField 的大小和位置。
- A、 文本 B、 大小 C、 范围 D、 位置
- 17、 可通过使用 () 帮助用户理解某个组件的用途。
- A、 JButton B、 JTextField C、 JLabel D、 标题栏
- 18、 使用 () 组件允许用户通过键盘输入数据。
- A、 JButton B、 JTextField C、 JLabel D、 以上选项均不对
- 19、 若 totalJTextField 是一个 JTextField，可以使用 () 获取 totalJTextField 的文本属性值。
- A、 totalJTextField.getText() B、 totalJTextField.setText()
- C、 totalJTextField.getText() D、 totalJTextField.setText()
- 20、Java 语言可以用 javax.swing 包中的类 JFileChooser 来实现打开和保存文件对话框。用户通过文件对话框不可能获得的信息是 ()
- A . 文件名称 B . 文件路径 C . 文件内容 D . 文件对象
- 21、方法 () 用于修改文本属性。
- A、 changeText B、 getText C、 setText D、 modifyText
- 22、用于完成某个特定任务并且可能会返回一个值的代码部分被认为是 ()。
- A、 变量 B、 方法 C、 操作数 D、 标识符
- 23、在 Java 中，() 可由 " " 来表示。
- A、 空字符 B、 空字符串 C、 空值 D、 以上答案均不对
- 24、在 JTextField 中按下某个键可以产生 () 的事件。
- A、 textTyped B、 valueChanged C、 keyPressed D、 keyReleased
- 25、() 是根据所执行的操作以及这些操作的次序来解决某个问题的一个过程。
- A、 图表 B、 控制语句 C、 算法 D、 有序列表
- 26、() 是帮助程序员开发算法的一个人工的、非正式的语言。
- A、 伪代码 B、 Java 语言 C、 注释 D、 执行
- 27、double 类型可以用做存储 ()。
- A、 字母和数字 B、 带有小数点的数 C、 字符串 D、 以上答案均不对
- 28、利用一个 () 组件可隐藏用户输入的信息；默认情况下，用户输入的每一个字符将被回显为一个星号 (*)。
- A、 JTextField B、 JPasswordField C、 JPasswordField D、 JPasswordField
- 29、方法的定义是由 () 组成的。
- A、 一个方法头 B、 一个方法体 C、 一对花括号 D、 以上答案都正确

- 30、 在方法调用过程中，位于方法名之后圆括号之间的变量被称为（ ）。
A、 变元 B、 参数 C、 语句 D、 声明
- 31、 方法调用的结果是（ ）。
A、 方法运行完成以后，应用程序将继续执行该方法调用发生时所在位置之后的语句。
B、 方法运行完成以后，应用程序将继续执行被调用方法定义之后的语句。
C、 执行该方法调用之前的语句。
D、 应用程序中止执行。
- 32、 当执行某种特定的用户操作，如点击 JButton 时，所产生的一个事件对象将被传递至（ ）。
A、 事件驱动程序 B、 操作方法 C、 事件处理程序 D、 操作驱动程序
- 33、 ActionEvent 的对象会被传递给（ ）事件处理程序。
A、 addChangeListener B、 addActionListener C、 stateChanged D、 actionPerformed
- 34、 完成事件处理程序的注册，需要为其关联一个（ ）。
A、 JButton B、 事件对象 C、 应用程序 D、 组件
- 35、 实现下哪个接口可以对 TextField 对象的事件进行监听和处理？（ ）。
A、 ActionListener B、 FocusListener C、 MouseMotionListener D、 WindowListener
- 36、 Fram 的默认布局管理器是下列哪一个？（ ）。
A、 FlowLayout B、 BorderLayout C、 GridLayout D、 CardLayout
- 37、 在下列事件处理机制中哪个不是机制中的角色？（ ）。
A、 事件 B、 事件源 C、 事件接口 D、 事件处理器
- 38、 要表示表格的数据，需要继承下列哪个类？（ ）。
A、 AbstactTableModel B、 TableModel C、 JTable D、 TableModelable
- 39、 每个使用 Swing 构件的程序必须有一个、（ ）。
A、 按钮 B、 标签 C、 菜单 D、 容器
- 40、 构造方法何时被调用（ ）。
A、 类定义时 B、 创建对象时
C、 调用对象方法时 D、 使用对象的变量时
- 41、 在定义 JLabel 时，可指定该 JLabel 的（ ）。
A、 文本对齐方式 B、 文本 C、 大小 D、 以上答案都对

- 42、JTextField 中文本的位置可以通过 () 设置
A setAlignmentProperty B setAlignment C setHorizontalAlidnment
D 以上都不对
- 43、可通过使用 () 帮助用户理解某个组件的意图
A JButton B JTextField C JLabel D 标题栏
- 44、使用 () 组件允许用户键盘输入数据
A、 JButton B、 JTextField C、 JLabel D、 以上都不对
- 45、起描述性作用的 JLabel 应该使用 ()
A、 语句大小写形式 B、 书名大小写形式
C、 一个冒号 (在文本的末尾) D、 A 和 C
- 46、通过使用 () 可以设置 JButton 上的文本
A、 setText B、 setButtonText C、 setJButtonText D、
setText
- 47、若 totaTextField 是一个 JtextField, 可以使用 () 获取
totaTextField 的文本属性值
A、 totaTextField.getText() B、 totaJtextField.setText()
C、 totaJtextField.getText() D、 totaJtextField.text
- 48、() 可表示一个用户的操作, 若点击一个 JButton
A、 语句 B、 事件 C、 应用程序 D、 函数
- 49、方法 () 用于修改文本属性
A、 changeText B、 getText C、 setText D、 modifyText
- 50、在 JTextField 中按下某个键可以产生 () 的事件
A、 textType B、 valueChange C、 pressedaKey D、 keyPressed
- 51、利用 () 方法可以设置 JCheckBox
A、 setText B、 setValue C、 setLable D、 setChecked
- 52、利用 () 方法可以判定 JCheckBox 是否已被复选
A、 isChecked B、 usSelectde C、 isClicked D、 getClicked
- 53、在一个 GUI 中, 一次可以选取 () 个 JCheckBox
A、 0 B、 1 C、 4 D、 任意个
- 54、与 JCheckBox 一同出现的文本被认为是 ()
A、 JCheckBox 的值 B、 JCheckBox 的名字
C、 JCheckBox 的文本 D、 JCheckBox 的数据
- 55、传递给 JOptionPane.showMessageDialog 方法的第二个参数为
()
A、 对话框标题栏内所显示的文本
B、 代表对话框中所显示的 JButton 的一个常量
C、 对话框中所显示的文本
D、 代表对话框中所出现的图标的一个常量
- 56、() 常量可以用在一个消息对话框中所显示一条错误信息

- 81、为一个组件 () 事件处理程序，从而告知该组件将调用哪一个事件处理程序
A 、 声明 B 、 定义 C 、 存储 D 、 注册
- 82、方法 () 可为 JSpinner 设置一个指定值
A 、 setJSpinner B 、 setText C 、 setNumber D 、 setValue
- 83、当某个操作发生时所调用的方法被称为 ()
A 、 事件监听器 B 、 JButton 方法 C 、 事件处理程序 D 、 action 方法
- 84、当产生 ChangeEvent 事件时，会调用 () 事件处理程序
A 、 actionPerformed B 、 valueChanged
C 、 stateChanged D 、 changeValue
- 85、可产生 ChangeEvent 事件的组件有 ()
A 、 JTextFiled B 、 JButtin C 、 JTextArea D 、 JSpinner
- 86、当执行某种特定的用户操作，如点击 Button 时，所产生的一个事件对象将被传递至 ()
A 、 事件驱动程序 B 、 操作方法 C 、 事件处理程序 D 、 操作驱动程序
- 87、ActionEvent 的对象会被传递给 () 事件处理程序
A 、 addChangeListener B 、 addActionListener
C 、 stateChanged D 、 actionPerformed
- 88、ActionEvent 是一种 () 类型
A 、 事件处理程序 B 、 事件对象 C 、 方法 D 、 GUI 组件
- 89、为使 JPanel 中的组件能够重新显示，可调用 () 方法
A 、 paint B 、 update C 、 repaint D 、 redraw
- 90、为清除 JLabel 中的图片，应该将其 icon 属性设置为 ()
A 、 “ ” (双引号) B 、 null C 、 none D 、 o
- 91、方法 () 用于在 JLabel 中显示一幅图片
A 、 setImage B 、 setIcon C 、 setImageIcon D 、 以上的答案都不对
- 92、可以利用 () 方法来设置 TitledBorder 的标题
A 、 title B 、 setText C 、 setBorder D 、 setTile
- 93、() 方法需要接收一个代表索引值的 int 型参数并返回该索引位于 JConboBox 中的值
A 、 getValueAt B 、 getSource C 、 getIndexValue D 、 getItemAt
- 94、JConboBox 类的 () 方法可返回当前 JConboBox 中所选选项的索引值
A 、 getSelectedIndex B 、 getSelectedValue
C 、 getSelectedSubscript D 、 getSelectedItem

- 95、利用 () 方法可以设置 JConboBox 下拉列表中一次可以显示的选项数
A 、 getUpprebound B 、 getItemAt C 、 setMaximumRowGount
D 、 setBounds
- 96、() 方法用于返回 JConboBox 中所选择选项的索引
A 、 getUpprebound B 、 getSelectedIndex
C 、 setMaximumRowGount D 、 getItemAt
- 97、() 方法是用来确定 JRadioButton 的状态
A 、 isSelected B 、 getSelected C 、 isChecked D 、
getChecked
- 98、当某个 JRadioButton 被选取时，会相应的调用 () 事件处理程序。
A 、 checkedChanged B 、 actionPerformed
C 、 selectedChanged D 、 以上答案都不对
- 99、一个 ButtonGroup 中包含 () JRadionButton
A 、 两个 B 、 不多于一个 C 、 不多于三个 D 、
任意数量
- 100、当某个 ButtonGroup 中有一个 JRadionButton 被选取时，()
A 、 还可同时选取其他的 JRadionButton
B 、 还可同时选取两个 JRadionButton
C 、 其他所有已选取的 JRadionButton 将被调用
- 101、使用 () 方法可对 JRadionButton 组件进行选取或取消
A 、 setselected B 、 setChecked C 、 setDefault D 、 setEnabled
- 102、组件 () 是一种状态组件
A 、 JRadioButton B 、 JCKeckBox C 、 JButton D 、 A 和 B
- 103、JRadioButton 是一种 () 类型的按钮
A 、 复选 B 、 变化 C 、 状态 D 、 操作
- 104、通过使用一个 () 可将多个 JRadioButton 组织到相应的 JFrame 中
A 、 GroupBox B 、 ButtonBox C 、 ButtonGroup D 、 以上答
案都不对
- 105、通过使用 () 方法可将一个 JRadioButton 添加到某个 ButtonGroup 组中
A 、 add B 、 AddJButton C 、 addJRadionButton D 、 new
JRadionButton
- 106、当创建一个 JRadioButton 时，它 ()
A 、 处于已被选取状态 B 、 处于为选去状态
C 、 已被添加到内容面板中 D 、 已被添加到一个默认的
JRadioButton
- 107、GUI 组件左上角坐标为 ()

- A、 (500 , 500) B、 (x,y) C、 (0 , 0) D、 以上答案都不对
- 108、()对象用于管理绘图环境并实现文本及多种图形的绘制
- A、 Image B、 Coordinates C、 Graphics D、 以上答案都不对
- 109、Rectangle 类需接收 4 个参数，这个 4 个参数分别定义了 ()
- A、 矩形 4 个角的坐标对
- B、 矩形左上角及右下角的 x 坐标和 y 坐标
- C、 矩形右上角及左下角的 x 坐标和 y 坐标
- D、 矩形左上角的 x 坐标、 y 坐标、矩形的宽度和高度
- 110、类 () 中定义了一些 Java 应用程序中用来控制颜色的方法和常量
- A 、 Color B、 GraphicsColor C、 Colors D、 GraphicsColors
- 111、类 () 中定义了可用来绘制文本、直线、矩形及其他图形的一些方法
- A、 Pictures B、 Drawings C、 Graphics D、 Illustrations
- 112、通过调用 () 方法，可绘制一个填充的椭圆
- A、 fillEllipse B、 drawOval C、 drawEllipse D、 fillOval
- 113、() 的 getPoint 方法将返回一个 Point 对象，此对象拥有当前鼠标的 x 值和 y 值
- A、 Point B、 JPanel C、 MouseEvent D、 Graphics
- 114、当鼠标按键被释放时，会调用 () 事件处理程序
- A、 mouseReleased B、 mouseUp C、 mouseOff D、 mouseLetGo
- 115、当鼠标按键被按下并进行移动时会调用 () 事件处理程序
- A 、 mouseDragged B 、 mousePositionChanged
- C 、 mouseMoved D、 mouseChanged
- 116、调用 () 方法可对拖动的鼠标进行注册
- A、 addMouseListener B、 addMouseDraggedListener
- C、 addMouseMotionListener D、 addMouseMoveListener
- 117、MouseEvent 对象的 x 坐标和 y 坐标与 () 相关
- A、 屏幕 B、 应用程序 C、 产生事件的 G U I 组件 D、 以上答案都不对
- 118、传递到鼠标事件处理程序的 () 对象包含与所发生鼠标事件相关
- A、 EventHandler B、 MouseEventHander C、 MouseEvent D、 EventArgs
- 119、当按下鼠标键时，会调用 () 事件处理程序
- A、 mouseDown B、 mouseClicked C、 mousePressed D、 mouseButtonDown
- 120、mouseMotionListener 事件处理程序是在 () 接口中被定义的

- A、 MouseMotionListener B、 MouseListener
C、 MouseDraggedListener D、 ActionListener
- 121、使用 () 方法可获去鼠标的位置
A、 getPoint B、 getLocation C、 getMouse D、 getCursor
- 122、MouseEvent 类的 () 方法用于确定所按下的鼠标键
A、 isPressed B、 rightButton C、 leftButton D、 isMetaDown
- 123、() 事件处理程序并未定义在 MouseListener 接口中
A、 mouseEntered B、 mouseExited C、 mouseMoved D、
mouseClicked
- 124、() 的椭圆是一个圆
A、 高度是宽度的两倍 B、 宽度为 0
C、 高度是宽度的一半 D、 高度等于宽度
- 125、边框并未要求某个对象的 ()
A、 高度 B、 颜色 C、 宽度 D、 位置
- 126、当按下或放开键盘上的一个按键时会产生 () 事件
A、 Keyboard B、 KeyPressedEvent C、 KeyChar D、
KeyReleasedEvent
- 127、当释放一个按键时，会调用 () 事件处理程序
A、 KeyEventReleased B、 KeyUp C、 KeyReleased D、
以上答案都不对
- 128、JMenu 中包含了 ()
A、 供用户选择的命令 B、 子菜单 C、 分隔条 D、 以
上答案都对
- 129、子菜单是一个添加在一个 () 组件中的 JMenu
A、 JMenu B、 JMenuBar C、 JMenuItem D、 JSeparator
- 130、() 对话框允许用户选择颜色
A、 ColorChooser B、 JColorChooseer C、 ChooserColor
D、 JChooserColor
- 131、使用 () 方法可返回对应事件源
A、 getSource B、 getEventSource C、 eventSource D、
source
- 132、KeyEvent 中的常量 () 代表最大的一个虚键码
A、 MAXIMUM B、 KEY_FINAL
C、 VK_MAX D、 KEY_LAST
- 133、() 是一个菜单容器
A、 键盘事件 B、 菜单条 C、 分隔条 D、 菜单选项
- 134、添加到 JMenu 中以创建子菜单的 GUI 组件作为 ()
A、 JSubmenu B、 JMenuItem C、 JMenu D、 JMenuBar
- 135、assSeparator 方法能够为菜单添加一个分隔 ()
A、 条 B、 助记符 C、 子菜单 D、 菜单选项

- 136、() 能够分组管理应用程序中的相关命令
A 、 分隔条 B、 热键 C、 菜单 D、 边界指示条
- 137、KeyEvent 的 () 方法可返回用户所按按键
A 、 keys B、 KeyCode C、 add D 、 getKeyCode
- 138、当用户按下某隔间时，会调用 () 事件处理程序
A 、 keyDown B、 keyReleased C、 keyPressed D、 以上答案都不对
- 139、使用 JTextArea 的 () 方法来设置 JTextArea 中文本的颜色
A 、 setForeground B、 setTextColor
C、 setBackground D、 setColor
- 140、当用户只能选择惟一的一个菜单选项时，应考虑使用 ()
A 、 子菜单 B、 HRadioButtonMenuItem
C、 JColorChooser D、 JCheckBoxMenuItem
- 141、() 能够绘制一条垂直直线
A 、 drawLine(50,50,25,25) B、 drawLine(25,25,50,25)
C、 drawLine(50,25,50,25) D 、 drawLine(50,25,50,50)
- 142、下面 () 语句可绘制一条水平直线
A 、 drawLine(0,5,5,0) B 、 drawLine(0,5,5,5)
C 、 drawLine(5,5,5,0) D 、 drawLine(5,5,5,5)
- 143、Graphics 类的 () 方法可绘制一个圆
A、 fillOval B、 fillCircle C、 drawOval D 、 drawCircle
- 144、drawLine, fillOval 和 setColor 方法都属于 () 类
A 、 Draw B 、 Graphics C、 Drawing D、 Graphic
- 145、代码 () 能绘制一个实心圆
A、 drawCircle(50,50,25) B、 fillOval(50,25,50,25)
C、 fillOval(50,50,25,25) D 、 drawCircle(50,50,50,50)
- 146、drawLine 方法中的第一个参数和第二个参数代表直线的 () 坐标
A 、 左上角 B、 x C、 y D 、 以上答案都不对
- 147、诸如 drawOval 和 drawRect 的方法定义在 () 类中
A 、 Drawing B、 Paint C、 Graphics D 、 Images
- 148、要为程序的子窗口设定一个由字符串 s 指定的名字，可以采用的代码是 ()
A、 JFrame(s) B、 window(s) C、 new JFrame(s) D、 super(s)
- 149、以下关于菜单的叙述中，正确的是 ()
A、 菜单分三级定义，最高一级的是菜单条，菜单条中放菜单，菜单中放菜单项
B、 菜单分三级定义，最高一级的是菜单，菜单中放菜单条，菜单条

中放菜单项

C、菜单分两级定义，最高一级的是菜单，菜单中放菜单项

D、菜单分两级定义，最高一级的是菜单条，菜单条中放菜单项

150、以下所述内容全部是字型要素的是（ ）

A、字体、字形和大小

B、字体、风格和字号

C、字形、风格和字号

D、字形、颜色和字号

151、以下是关于 FlowLayout 布局组件排列顺序的叙述，正确的是（ ）

A、从上到下排列

B、从左到右排列

C、从下到上排列

D、从右到左排列

152、在 Swing 中，它的子类能用来创建框架窗口的类是（ ）

A、JWindow

B、JFrame

C、JDialog

D、JApplet

153、MouseListener 接口不能处理的鼠标事件是（ ）

A、按下鼠标左键

B、点击鼠标右键

C、鼠标进入

D、鼠标移动

154、以下不属于文字字型要素的是（ B ）

A、颜色

B、字体

C、风格

D、字号

二、填空题

1、在 Java.awt 包中共定义了 5 种布局类，分别是_____、_____、_____、_____和_____。

2、Java.swing 包中共定义了 4 种布局类，分别是 _____、_____、_____和_____。

3、CardLayout 构造新的 _____ 布局。

4、BoxLayout 是_____提供的布局管理器，它有两种排序方式，一种是 _____，另一种是 _____。

5、在 Java 中，当用户使用键盘进行操作时，则会出生 _____ 事件。

6、在 Java 中，当用户鼠标进行操作时，则会产生鼠标事件

MouseEvent。MouseEvent 事件的响应是实现 _____ 接口或 _____ 接口，或者是继承 _____ 类，来实现 MouseAdapter 提供的方法。

7、Mouse 有关的事件可以分为两类，一类是 _____ 接口，共提供了 5 种方法，主要针对鼠标的按键和位置进行检测；另一类是 _____ 接口，共提供了两种方法，主要针对鼠标的坐标和按键进行检测。

8、在 Java 中，一般菜单格式包含有菜单栏 _____ 类、菜单 _____ 类和菜单项 _____ 类对象组成。

9、制作菜单的一般步骤

1) _____

2) _____

3) _____

4) _____

- 10、AWT 中的 _____ 和 _____ 中的 JPanel 就是这样一类容器。
- 11、Panel 和 Panel 的区别在于； _____ 支持双缓冲区功能，在处理动画时较少发生画面闪烁现象。
- 12、对于窗口我们已经不陌生，使用 _____ 包的 JFrame 类就可以创建一个窗口，实际上对话框也是一种类型的窗口。
- 13、JPanel 是创建一个双缓冲区和 _____ 布局的 JPanel 对象。
- 14、常用的 awt 组件主要包括 _____、_____、_____、_____、_____、_____和 _____。
- 15、Swing 组件的基类是 _____。
- 16、图形用户界面 (Graphics User Interface)，简称 _____。
- 17、图形用户界面的包是 _____，简称抽象窗口工具包，即 AWT (Abstract Window Toolkit)。概念 - 设计 - 实现字体设计 - 界面设计
- 18、Java 基础类库，包含 AWT 和 Swing 组件，_____是 JFC 的核心。
- 19、基于 Java 的图形用户界面开发工具 (即组件集) 最主流的有三种 _____、_____、_____。
- 20、AWT 组件大致可以分为以下三类 _____、_____、_____。
- 21、_____提供的对话框是模式对话框
- 22、_____提供标准对话框的布局支持、图标、指定对话框的标题和文本、自定义按钮文本、允许自定义组件的对话框显示、指定对话框在屏幕上的显示位置等特性。
- 23、_____语言提供多种对话框类来支持多种形式的对话框。
- 24、_____类支持简单、标准的对话框。
- 25、_____类支持定制用户自己的对话框。
- 26、_____类支持文件打开、保存对话框。
- 27、_____类支持操作进度条控制对话框等。
- 28、菜单的格式分为 _____和 _____。
- 29、为了简化表单填写过程，通常为用户提供多种可供选择的选项，而无需用户写出他们的响应。常用于选择性输入的组件有 _____、_____、_____、_____。
- 30、对话框分为 _____和 _____两种。
- 31、_____是指基于组件和容器的事件，当一个组件上发生事件，如、鼠标的进入，点击，拖放等，或组件的窗口开关等，触发了组件事件。
- 32、AWT 事件共有 10 类，可以归为两大类 _____和 _____。
- 33、_____是 Abstract Window ToolKit 的缩写，是抽象窗口组件工具包，是 Java 语言中最早的用于编写图形界面应用程序的工具包。该工具包提供了一套与本地图形界面进行交互的接口。

- 34、_____集遵循最大公约数原则，即 AWT 只拥有所有平台上都存在的组件的公有集合
- 35、_____为了解决 AWT 存在的问题而新开发的包。 Swing 是在 AWT 的基础上构建的一套新的图形界面系统，它提供了 AWT 所能够提供的所有功能，并且用纯粹的 Java 代码对 AWT 的功能进行了大幅度的扩充。
- 36、_____组件包含在 java.awt 包里，如 Button、Checkbox、Scrollbar 等，都是 Component 类的子类。
- 37、_____中大多数组件其名称都是在原来 AWT 组件名称前加上 J，例如 JButton、JCheckBox、JScrollbar 等，都是 JComponent 类的子类。
- 38、_____程序一般选择 JFrame(或 Frame)作为顶层容器_____，Applet 程序一般选择 JApplet(或 Applet)作为顶层容器_____。
- 39、布局管理就是创建这种类型的一个对象，并采用此对象来安排其它容器和基本组件。常用的四种布局管理器_____、_____、_____、_____。
- 40、(框架) 是一种顶层容器，是用来设计类似于 Windows 系统中的窗口形式的应用程序。
- 41、Java GUI 发展的三个阶段：_____、_____、_____。
- 42、基本的控制组件是用来提供人机交互的基本控制界面，包括_____、_____、_____、_____、_____、_____、_____、_____、_____、_____。
- 43、_____是由事件源产生的，事件的产生者称为事件源。例如，在 Button 组件上点击鼠标会产生以这个 Button 为源的一个事件、ActionEvent。
- 44、Swing 组件存放在_____包中。
- 45、Container 有几个主要的子类_____、_____、_____。
- 46、Frame(框架)是 AWT 的强大特征，我们可以用_____方法增加组件到 Frame。
- 47、Image 类_____方法返回要作用为此窗体图显示的图像
- 48、_____方法将此窗体的标题设置为指定的字符串。
- 49、_____方法显示窗体。
- 50、_____方法返回此文本组件表示的组件。
- 51、_____方法用于回显的字符。
- 52、_____该方法用来创建一个没有显示内容的对象。
- 53、Label 的对齐方式有_____种。
- 54、Label 类的_____、_____和_____常量来分别表示左对齐、居中对齐和右对齐。

- 55、Button(String label) 创建一个 _____ 的按钮。
- 56、方法的定义是由 _____、_____、_____、_____组成的
- 57、利用一个 _____ 组件可隐藏用户输入的信息；默认情况下，用户输入的每一个字符将被回显为一个星号（*）。
- 58、列表框 _____ 可以使用用户选择多个选择。
- 59、Checkbox 类用于创建标签复选框，复选框有两部分——和 _____。
- 60、_____ 创建一个没有标签的复选框。
- 61、_____ 是组成图形用户界面的最小单位之一。
- 62、复选框有 _____ 和 _____ 两种状态。
- 63、单选按钮是一组 _____ 的集合。
- 64、List 构造函数的第一个参数表示列表 _____，也表示有几项内容
- 65、List 构造函数的第二个参数表示列表 _____。
- 66、List 构造函数的第二个参数用 _____ 类型表示。
- 67、调用 _____ 类的 add() 方法来增添菜单选项；
- 68、swing 建立在 _____ 基础之上。
- 69、awt 包的运行速度比 swing _____
- 70、使用 _____ 方法设置窗口的大小
- 71、_____ 类是按钮基类
- 72、Icon getIcon() 是获得此 _____
- 73、_____ 获得按钮上的文字
- 74、void _____ 设置按钮 _____
- 75、boolean 的 _____ 得到按钮的状态
- 76、JButton() 创建一个 _____

三、判断题

- 1、组件不能独立显示出来，必须将组件放在一定的容器中才可以显示出来。（ ）
- 2、Swing 组件存放在 javax.swing 包中（ ）
- 3、Dialog 是弹出式窗口用于接受用户输入。（ ）
- 4、Panel(面板)不是用作组织组件的。（ ）
- 5、Label() 该方法用来创建一个没有显示内容的对象。（ ）
- 6、Label 对象是一个在容器中放置文本的组件。（ ）
- 7、Label 类的 LEFT 常量表示右对齐。（ ）
- 8、Button 类不用于创建按钮。（ ）
- 9、Button() 创建一个没有标题的按钮。（ ）
- 10、Choice 类选择菜单用于显示可让用户选择的选择列表。（ ）
- 11、List 类构造函数的第一个参数指明可见项的数目而不是列表中项数的最大值。（ ）

- 12、Checkboxl 类用于创建标签复选框。 ()
- 13、容器是用来盛放其它部件的单元 ()
- 14、容器都有一定的范围 ()
- 15、容器有一定的位置 ,一般包含许多部件 ()
- 16、打开容器时,其内部的部件都能显示,关闭容器,其内部的部件就隐藏起来 ()
- 17、在容器内可以采用一定的规则来安排所包含的部件 ()
- 18、容器可以包含容器 ()
- 19、组件,其内不在包含任何其他成分 ()
- 20、Checkbox 是复选框 ()
- 21、CheckboxGroup 不是单选按钮 ()
- 22、每个 CheckboxGroupd 的对象对应着一种可能的取值情况, 即对应一个选项 ()
- 23、List 是列表 ()
- 24、创建列表时要把各项内容添加进去 ()
- 25、List 构造函数的第一个参数表示列表高度 ,也表示有几项内容 ()
- 26、在 JMenuBar 类中 JMenuBer() 是创建菜单栏 ()
- 27、在 JMenuI 类中 JMenu() 创建菜单 ()
- 28、JMenu(String str) 创建菜单栏 ()
- 29、在 JMenuI 类 JMenuI (String str,boolean b) 创建具有指定文字的菜单,通过布尔值确它是否有下拉式菜单 ()
- 30、JMenuItem () 创建一个菜单栏 ()
- 31、JMenuItem(String str ,int nmemonic) 创建一个指定标签和键盘设置快捷健的菜单项。 ()
- 32、JMenuItem(String str ,Icon icon) 创建具有指定文字和图形的菜单项 ()
- 33、在 Java 中,当用户使用鼠标进行操作时,则会产生鼠标事件 MouseEvenr ()
- 34、BorderLayout 是构造新的 BorderLayout 布局 ()
- 35、AWT 是 Java 语言进行 GUI 程序设计的基础 ()
- 36、Swing 包是 Java 基础类库 (Java Foundation Classes , JFC) 的一部分。 ()
- 37、Swing 组件是 Java 语言提供的第二代 GUI 设计工具包,它以 AWT 为基础。 ()
- 38、JOptionPane 提供的对话框是模式对话框 ()
- 39、下拉式菜单包含有一个菜单条 (JMenuBar),在菜单条上安排有若

干个菜单 (Menu) , 每个菜单又包含若干菜单项 (MenuItem) , 每个菜单项对应了一个命令或子菜单项。 它们构成一个应用程序的菜单系统。用鼠标或键盘选择对应一个命令的菜单项与选择一个按钮类似。使用菜单系统可方便地向程序分布命令。 ()

40、组件 (Component) 是构成 GUI 的基本元素 , 如按钮、标签、画布、复选框等。 ()

41、容器 (Container) 组件 , 简称容器 , 是一个可以包含组件和其他容器的组件 . 如 , JPanel, JTabbedPane JScrollPane 等。 ()

42、非容器组件 (原子组件) 则是不可以包含组件和其他容器的组件 , 如 JButton、JLabel、JTextField 等 ()

四、程序填空

1、阅读下面的程序 , 在空格处填上相应的代码

```
_____
_____
public class JFrameExample1 {
    public static void main(String args[]) {
        JFrameExample1 obj = new JFrameExample1();
        obj.myJFrame();
    }

    public void myJFrame() {
        JFrame jframe = new JFrame(" 一个框架应用 ");
        Container panel = jframe.getContentPane();
        panel.setLayout(new FlowLayout());
        panel.add(new JButton(" 欢迎使用 ! "));
        jframe.setSize(300, 150);
        jframe.show();
    }
}
```

2、阅读下面的程序 , 在空格处填上相应的代码

```
import java.awt.*;
import javax.swing.*;
public class myNullLayout _____ JFrame{
    JLabel 学号 ,姓名 ,性别 ;
    JTextField stdno,name,sex;
    int x=0,y=0,w,h;
    Container cp=getContentPane();
    public myNullLayout() {
        setLayout(null);
        学号 =new JLabel(" 学号、 ",JLabel.CENTER);
```

```
姓名 =new JLabel(" 姓名、 ",JLabel.CENTER);
性别 =new JLabel(" 性别 ",JLabel.CENTER);
```

```
sex=new JTextField();
x=80;y=60;
w=100;h=30;
cp.add(学号 ); cp.add(姓名 );cp.add(性别 );
cp.add(stdno);cp.add(name);cp.add(sex);
学号 .setBounds(0,y,w,h); stdno.setBounds(x,y,w,h);
姓名 .setBounds(0,2*y,w,h);name.setBounds(x,2*y,w,h);
性别 .setBounds(0,3*y,w,h);sex.setBounds(x,3*y,w,h);
setSize(300,500);
show();
```

```
}
public static void main(String args[]){
    myNullLayout obj=new myNullLayout();
}
```

```
}
```

3、 阅读下面的程序，在空格处填上相应的代码

```
import java.awt.*;
import javax.swing.*;
public class JpanelExample{
    private void fillJPanel(Container c){
        Button button1,button2,button3;
        button1=new Button(" 确定 ");
        button2=new Button(" 取消 ");
        button3=new Button(" 保存 ");
    }
    public static void main(String args[]){
        JpanelExample  obj=new JpanelExample();
        obj.myJPanel();
    }
    public void myJPanel(){
        JFrame jframe=new JFrame(" 一个面板应用 ");
        Container contentPane=jframe.getContentPane();
        contentPane.setLayout(new FlowLayout());
        JPanel jpanel1=new JPanel();
        fillJPanel(jpanel1);
```



```

        jpanel1.setBackground(Color.BLUE);
        jpanel1.setForeground(Color.RED);
        JPanel jpanel2=new JPanel();
        fillJPanel(jpanel2);
        jpanel2.setBackground(Color.WHITE);
        jpanel2.setForeground(Color.BLACK);
        contentPane.add(jpanel1);
        contentPane.add(jpanel2);
        contentPane.add(new JButton(" 我不在那些面板里  !!"));
        jframe.setSize(400,250);
        jframe.show();
    }
}
package xiazai;
import java.awt.*;
import javax.swing.*;
public class JpanelExample{
    private void fillJPanel(Container c){
        Button button1,button2,button3;
        button1=new Button(" 确定 ");
        button2=new Button(" 取消 ");
        button3=new Button(" 保存 ");
        c.add(button1);c.add(button2);c.add(button3);
    }
    public static void main(String args[]){
        JpanelExample  obj=new JpanelExample();
        obj.myJPanel();
    }
    public void myJPanel(){
        JFrame jframe=new JFrame(" 一个面板应用 ");
        Container contentPane=jframe.getContentPane();
        contentPane.setLayout(new FlowLayout());
        JPanel jpanel1=new JPanel();
        fillJPanel(jpanel1);
        jpanel1.setBackground(Color.BLUE);
        jpanel1.setForeground(Color.RED);
        JPanel jpanel2=new JPanel();
        fillJPanel(jpanel2);
        jpanel2.setBackground(Color.WHITE);
        jpanel2.setForeground(Color.BLACK);
    }
}

```

```

contentPane.add(jpanel1);
contentPane.add(jpanel2);
contentPane.add(new JButton("我不在那些面板里  !!"));
jframe.setSize(400,250);

```

```

}

```

```

}

```

4、阅读下面的程序，在空格处填上相应的代码

```

import java.awt.*;
import javax.swing.*;
public class myCardLayoutExample1{
    public static void main(String args[]){
        JFrame jframe=new JFrame(" 一个滚动列表的例子 ");
        Container cp=jframe.getContentPane();
        CardLayout card=new CardLayout(20,20);
        cp.setLayout(card);
        JButton jbt1=new JButton(" 足球 ");
        JButton jbt2=new JButton(" 篮球 ");
        JButton jbt3=new JButton(" 排球 ");
        JButton jbt4=new JButton(" 羽毛球 ");
        JButton jbt5=new JButton(" 乒乓球 ");

        cp.add("b",jbt2);
        cp.add("c",jbt3);

        cp.add("e",jbt5);

        jframe.setSize(150,200);
        jframe.show();
    }
}

```

5、阅读下面的程序，在空格处填上相应的代码

```

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
//对话框
public class myDialog extends _____{
    private JButton jbt;
    private JLabel jlabel;
    public myDialog(){

```

```

super("一个显示信息对话框的例子  ");
Container cp=getContentPane();
cp.setLayout(new GridLayout(2,2,10,10));
jbt=new JButton(" 显示信息按钮 ");
cp.add(jbt);
jbt.addActionListener(new handle());
jlabel=new JLabel("");
cp.add(jlabel);
setSize(500,450);

}

public class handle _____ActionListener{
    public void actionPerformed(ActionEvent e){
        String title=" 信息对话框 ";
        String content=" 请选择 (yes/no)?";
        int dialogtype=JOptionPane.YES_NO_OPTION;
        int choice=3;
        if(e.getSource()==jbt)

choice=JOptionPane.showConfirmDialog(null,content,title,dialogtype);
        if(choice==0)    jlabel.setText(" 您选择的是 'YES'");
        else    jlabel.setText(" 您选择的是 'NO'");
    }
}

public static void main(String args[]){
    myDialog app=new myDialog();
}
}

```

6、 阅读下面的程序，在空格处填上相应的代码

```

import java.awt.*;
import javax.swing.*;
public class myJScrollPaneExample{
    public static void main(String args[]){
        JFrame jframe=new JFrame(" 按钮 + 面板 + 滚动窗口 ");
        Container contentPane=jframe.getContentPane();
        contentPane.setLayout(new FlowLayout());
        JPanel jpanel=new JPanel();
        JScrollPane jscrollpane=new JScrollPane();
        JButton jbutton1=new JButton(" 确定 ");
        JButton jbutton2=new JButton(" 取消 ");

```

```

JButton jbutton3=new JButton(" 保存 ");
jpanel.add(jbutton1);

_____

jpanel.add(jbutton3);
jscrollpane.setViewportViewView (jpanel);
contentPane.add(jscrollpane);
jframe.setSize(800,650);

_____

```

```

}

```

```

}

```

7、阅读下面的程序，在空格处填上相应的代码

```

import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
public class myKeyEvent extends JFrame _____{
    private String line1="", line2 = "";
    private JTextArea textArea;
    public myKeyEvent() {
        super( "显示键盘事件演示程序  " );
        textArea=new JTextArea(10,15);
        textArea.setText(" 按任意键继续  ...");
        textArea.setEnabled(false);
        textArea.setFont(new
Font("TimesRoman",Font.BOLD+Font.ITALIC,18));
        getContentPane().add(textArea);

        _____

        setSize(300,100);
        show();
    }
    public void _____(KeyEvent e){
        line1=" 键盘按下、  "+e.getKeyText(e.getKeyCode());
        setString(e);
    }
    public void _____(KeyEvent e){
        line1=" 按键释放、  "+e.getKeyText(e.getKeyCode());
        setString(e);
    }
    public void _____ (KeyEvent e){
        line2=" 按键输入、  "+e.getKeyChar();
        setString(e);
    }
}

```

```

    }
    private void setString(KeyEvent e){
        textArea.setText(line1+"\n"+line2);
    }
    public static void main(String args[]){
        myKeyEvent app = new myKeyEvent();
        app.addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent e){
                System.exit(0 );
            }
        });
    }
}

```

8、阅读下面的程序，在空格处填上相应的代码

```

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
public class myMouseEvent extends JFrame{
    private JTextField text;
    public myMouseEvent(){
        super(" 鼠标事件测试例子 ");
        Container cp=getContentPane();
        cp.setLayout(new BorderLayout(5,5));
        JLabel label=new JLabel(" 移动拖动鼠标 ");
        Font g = new Font("Courier", Font.BOLD, 22);
        label.setFont(g);
        cp.add(label,BorderLayout.NORTH);
        text=new JTextField(20);
        cp.add(text,BorderLayout.SOUTH);
        cp.addMouseMotionListener(new MouseMotionHandler());
        _____
        setSize(300,450);
        _____
    }
    public class MouseMotionHandler extends MouseMotionAdapter {
        public void mouseDragged(MouseEvent e){
            String s=" 鼠标拖动、 X="+e.getX()+"Y="+e.getY();
            text.setText(s);
        }
    }
}

```

```

public class MouseEventHandler extends MouseAdapter {
    public void mouseEntered(MouseEvent e){
        String s=" 鼠标进入 !";
        _____
    }
    public void mouseExited(MouseEvent e){
        String s=" 鼠标离开 !";
        text.setText(s);
    }
}
public static void main(String args[]) {
    myMouseEvent app=new myMouseEvent();
    app.addWindowListener( new WindowAdapter(){
        public void windowClosing(WindowEvent e){
            System.exit( 0 );
        }
    });
}
}

```

9、阅读下面的程序，在空格处填上相应的代码

```

import java.awt.*;
import java.awt.event.*;
public class FlowLayoutTest {
    public static void main(String args[])
    {
        _____
        f.setLayout(new
                                FlowLayout(FlowLayout.RIGHT,40,20));
        Button button1 = new Button("Ok");
        Button button2 = new Button("Open");
        Button button3 = new Button("Close");
        f.add(button1);
        _____
        f.setSize(100,100);
        f.setVisible(true);
        f.addWindowListener(new WindowAdapter(){public void
windowClosing(WindowEvent e)
    {System.exit(0); }});
    }
}

```

```
}
```

10、阅读下面的程序，在空格处填上相应的代码

```
import java.awt.event.*;
import javax.swing.*;
public class TestJScrollPane extends WindowAdapter
{
    JFrame frame;
    JPanel cp=new JPanel();
    JScrollPane jsp=new
JScrollPane(JScrollPane.VERTICAL_SCROLLBAR_ALWAYS,
JScrollPane.HORIZONTAL_SCROLLBAR_AS_NEEDED);
    JLabel ImagL=new JLabel();
    JTextArea ta=new JTextArea();
    JSplitPane jsplitpane =new
JSplitPane(JSplitPane.HORIZONTAL_SPLIT);
    //分隔组件类对象声明
    public void Test() {
        _____
        ImageIcon icon=new ImageIcon("test.jpg");
        ImagL.setIcon(icon);
        jsp.setViewportView(ImagL);
        jsplitpane.setBorder(BorderFactory.createEtchedBorder());
        jsplitpane.setOneTouchExpandable(true);
        jsplitpane.setDividerLocation(100);
        jsplitpane.setDividerSize(20);
        jsplitpane.setLeftComponent(ta);
        jsplitpane.setRightComponent(jsp);
        cp.add(jsplitpane);
        frame.setSize(400,400);
        frame.add(cp);
        _____
    }
    public static void main(String [] args){
        TestJScrollPane JSP1 = new TestJScrollPane();
        JSP1.Test();
    }
    public void windowClosing(WindowEvent e)
{ System.exit(0); }
}
```

11、阅读下面的程序，在空格处填上相应的代码

```
import java.awt.*;
```

```

import java.awt.event.*;
import javax.swing.*;
public class JComboBoxDemo extends JPanel
{
    JLabel picture,text;
    public JComboBoxDemo()
    {
        String[] pStrings = {"cup","cat","boy","girl"};
        JComboBox pList = new JComboBox(pStrings);
        pList.setSelectedIndex(0);

        _____

        public void actionPerformed(ActionEvent e) {
            JComboBox cb = (JComboBox)e.getSource();
            String pName = (String)cb.getSelectedItem();
            picture.setIcon(new ImageIcon("images/"+pName +
".gif"));
            text.setText(pName);
            text.setHorizontalAlignment(JLabel.CENTER);
        }
    });
    picture = new JLabel(new ImageIcon("images/"
        +pStrings[pList.getSelectedIndex()] +".gif"));
    picture.setBorder(BorderFactory.
        createEmptyBorder(10,0,0,0));
    picture.setPreferredSize(new Dimension(180, 140));
    text = new
JLabel(pStrings[pList.getSelectedIndex()],JLabel.CENTER);
setLayout(new BorderLayout());
    add(pList,BorderLayout.NORTH);
    add(picture,BorderLayout.CENTER);
    add(text,BorderLayout.SOUTH);
    setBorder(BorderFactory.createEmptyBorder(20,20,20,20));
}
    public static void main(String args[]) {
        JFrame frame = new JFrame("使用 JComboBox");

frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
frame.setContentPane(new JComboBoxDemo());
frame.pack();

```

```
}  
}
```

12、阅读下面的程序，在空格处填上相应的代码

```
import java.awt.*;  
import javax.swing.*;  
import javax.swing.event.*;  
import javax.swing.border.*;  
public class JListDemo extends JFrame  
{public static void main(String[] args) {  
    new JListDemo();  
}  
    private JList jl;  
    private JTextField valueField;  
    public JListDemo()  
{  
    super("一个简单的 JList");  
    try {  
        UIManager.setLookAndFeel(UIManager.  
getSystemLookAndFeelClassName());  
    } catch(Exception e){  
        System.out.println(" 设置 native LAF 错误 : "+ e);  
    }  
    Container content = getContentPane();  
    String[] entries = { " 北京 ", " 上海 ", " 天津 ", " 重庆 ", " 武汉 "};  
    jl = new JList(entries);  
    jl.setVisibleRowCount(4);  
    Font displayFont = new Font("Serif", Font.BOLD, 18);  
    jl.setFont(displayFont);  
    jl.addListSelectionListener(new ValueReporter());  
    JScrollPane listPane = new JScrollPane(jl);  
    JPanel listPanel = new JPanel();  
    listPanel.setBackground(Color.white);  
    Border listPanelBorder = BorderFactory.  
createTitledBorder("JList 数据");  
    listPanel.setBorder(listPanelBorder);  
    listPanel.add(listPane);  
    content.add(listPanel, BorderLayout.CENTER);  
    JLabel valueLabel = new JLabel(" 选择为 :");  
    valueLabel.setFont(displayFont);
```

```

valueField = new JTextField("", 6);
valueField.setFont(displayFont);

valuePanel.setBackground(Color.white);
Border valuePanelBorder = BorderFactory.
createTitledBorder("JList 选择 ");
valuePanel.setBorder(valuePanelBorder);
valuePanel.add(valueLabel);
valuePanel.add(valueField);
content.add(valuePanel, BorderLayout.SOUTH);
pack();
}

```

```

private class ValueReporter implements ListSelectionListener{
public void valueChanged(ListSelectionEvent event){
if (!event.getValueIsAdjusting())
valueField.setText(
jl.getSelectedValue().toString());
}
}
}

```

13、阅读下面的程序，在空格处填上相应的代码

```

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
public class CelsiusConverter implements _____{
    JFrame frame;
    JPanel panel;
    JTextField tempCelsius;
    JLabel celsiusLabel, fahrenheitLabel;
    JButton convertTemp;
    public CelsiusConverter() { // 构造方法
        frame = new JFrame(" 温度变换 ");
        panel = new JPanel();
        panel.setLayout(new GridLayout(2, 2));
        tempCelsius = new JTextField();
        celsiusLabel = new JLabel(" 摄氏温度 = ");
        convertTemp = new JButton(" 变换 ");
        fahrenheitLabel = new JLabel(" ? 华氏温度 ");
        convertTemp.addActionListener(this);
    }
}

```

```

        panel.add(tempCelsius);
        panel.add(celsiusLabel);
        panel.add(convertTemp);
        panel.add(fahrenheitLabel);
        frame.getContentPane().add(panel, BorderLayout.CENTER);
        frame.setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE);
        frame.pack();
    }

    public void actionPerformed(ActionEvent event)
    {
        int tempF;
        tempF
        =(int)((Double.parseDouble( tempCelsius.getText()))*1.8+32);
        fahrenheitLabel.setText(tempF + " 华氏温度 ");
    }
    public static void main(String[] args) {
        CelsiusConverter converter = new CelsiusConverter() ;
    }
}

```

14、实现列表的程序代码

```

add(list);
list.add();
list.add( “ 中国 ” );
list.add( “德国 ”);

```

15、实现下拉表的程序代码

```

add(choice);
choice.add( “男 ”);
choice.add “ 女 ” );

```

16、实现标签的代码

```

add(label);
设置窗口的  ContentPane,创建按钮对象

```

```

but1=new JButton( “按钮 1 ” );

```

17、阅读下面的程序，在空格处填上相应的代码

```

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
public class JRadioButtonDemo extends _____
{
    static JFrame frame;
    static String metal= "Metal";
    static String motif = "Motif";
    static String windows = "Windows";
    JRadioButton    metalButton,motifButton,windowsButton;
    public JRadioButtonDemo(){
        JButton button = new JButton("Hello, world");
        button.setMnemonic('h');
        metalButton = new JRadioButton(metal);
        metalButton.setMnemonic('o');
        metalButton.setActionCommand(metal);
        motifButton = new JRadioButton(motif);
        motifButton.setMnemonic('m');
        motifButton.setActionCommand(motif);
        windowsButton = new JRadioButton(windows);
        windowsButton.setMnemonic('w');
        windowsButton.setActionCommand(windows);
        ButtonGroup group = new ButtonGroup();
        group.add(metalButton);
        group.add(motifButton);
        group.add(windowsButton);
        RadioListener myListener = new RadioListener();
        metalButton.addActionListener(myListener);
        motifButton.addActionListener(myListener);
        windowsButton.addActionListener(myListener);
        add(button);
        add(metalButton);
        add(motifButton);
        add(windowsButton);
    }
    class RadioListener implements ActionListener {
        public void actionPerformed(ActionEvent e) {
            if((e.getActionCommand()).equals(metal))
                SetLAF.setJavaLookAndFeel();
            else if((e.getActionCommand()).equals(motif))

```

```

        SetLAF.setMotifLookAndFeel();
    else SetLAF.setNativeLookAndFeel();
        SwingUtilities.updateComponentTreeUI(frame);
        frame.pack();
    }
}
public static void main(String s[]) {
    JRadioButtonDemo panel = new JRadioButtonDemo();

    frame = new JFrame(" 使用 JRadioButton 选择观感 ");
    frame.setDefaultCloseOperation(

JFrame.EXIT_ON_CLOSE);
    frame.getContentPane().add("Center", panel);
    frame.pack();

    _____
}
}

```

18、阅读下面的程序，在空格处填上相应的代码

```

import java.awt.*;
import java.awt.event.*;

import javax.swing.event.*;
public class JTabbedPaneDemo{
    public static void main(String args[]){
        new MyTabbedPane();
    }
}
class      MyTabbedPane      extends      JFrame      Implements
ChangeListener,ActionListener
{
    JTabbedPane jt;
    JButton jb[];
    int index = 0;
    MyTabbedPane(){
        super(" 使用标签面板容器  ");
        JTabbedPane jt = new JTabbedPane();
        JButton[] jb = new JButton[5];
        for(int i = 0;i<5;i++){
            jb[i] = new JButton(" 第 " + i + " 页面板 ");

```

```

        jb[i].addActionListener((java.awt.event.ActionListener) this);
        jt.addTab(" 页标签 " + i,jb[i]);
    }
    jt.addChangeListener((javax.swing.event.ChangeListener) this);
    getContentPane().add(jt,BorderLayout.CENTER);
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setSize(300,200);
}
public void stateChanged(ChangeEvent e){
    Object jt;
    if(e.getSource()==jt){
        int i = ((JTabbedPane)e.getSource())
                .getSelectedIndex();
        Object[] jb;
        ((Window) jb[index]).setVisible(false);
        ((Window) jb[i]).setVisible(true);
        index = i;
    }
}
public void actionPerformed(ActionEvent e){
    setTitle(" 响应单击 "+((JButton)e.getSource()).getText());
}
}

```

19、阅读下面的程序，在空格处填上相应的代码

```

import javax.swing.*;
import java.awt.event.*;
import java.awt.*;
public class JCheckBoxDemo extends _____ implements
{
    private JTextField t;
    private JCheckBox bold,italic;
    public void init()
    {t = new JTextField(" 观察这里文字字形    的变化 ",40);
      t.setFont(new Font("Serif",Font.PLAIN,20));
      getContentPane().add(t,BorderLayout.NORTH);
      bold = new JCheckBox(" 粗体  Bold");
      bold.addItemListener(this);
      getContentPane().add(bold,BorderLayout.CENTER);
      italic = new JCheckBox(" 斜体  Italic");
    }
}

```

```

        getContentPane().add(italic, BorderLayout.SOUTH);
    }
    public void itemStateChanged(ItemEvent e){
        int b = bold.isSelected()?Font.BOLD:Font.PLAIN;
        int i = italic.isSelected()?

```

```

Font.ITALIC:Font.PLAIN;
        t.setFont(new Font("Serif",b + i,20));
    }
}import javax.swing.*;
import java.awt.event.*;

```

```

public class JCheckBoxDemo extends JApplet implements
ItemListener{
    private JTextField t;
    private JCheckBox bold,italic;
    public void init()
    {
        t = new JTextField(" 观察这里文字字形的变化 ",40);
        t.setFont(new
Font("Serif",Font.PLAIN,20));

```

```

        bold = new JCheckBox(" 粗体 Bold");
        bold.addItemListener(this);
        getContentPane().add(bold, BorderLayout.CENTER);
        italic = new JCheckBox(" 斜体 Italic");
        italic.addItemListener(this);
        getContentPane().add(italic, BorderLayout.SOUTH);
    }
    public void itemStateChanged(ItemEvent e){
        int b = bold.isSelected()?Font.BOLD:Font.PLAIN;
        int i = italic.isSelected()?

```

```

Font.ITALIC:Font.PLAIN;
        t.setFont(new Font("Serif",b + i,20));
    }
}

```

五、简答题

1、设计和实现图形用户界面的主要工作分为那两个步骤？

- 2、与 awt 相比， Swing 带来了三个显著的进步是什么？
- 3、AWT 组件大致可以分为那三类、
- 4、Java Swing GUI 应用程序的开发过程有那些？
- 5、Java 事件编程一般包括那些步骤？
- 6、什么是事件？
- 7、利用 AWT 组件和 Swing 组件进行程序设计的基本步骤有那些？
- 8、对 JFrame 添加组件有两种方式、
- 9、制作菜单的一般步骤、
- 10、什么是模式和非模式对话框
- 11、Frame 类具有特点有那些？
- 12、Java 中设计和实现图形用户界面的步骤分为那些？

六、编程题

- 1、编写程序，包括一个标签、一个文本框和一个按钮，当单击按钮时把文本框中的内容复制到标签中显示。

```
package test;
import java.awt.Toolkit;
import javax.swing.SwingUtilities;
import javax.swing.UIManager;
import java.awt.Dimension;
public class Application {
    boolean packFrame = false;
    public Application() {
        Frame1 frame = new Frame1();
        if (packFrame) {
            frame.pack();
        } else {
            frame.validate();
        }
        Dimension screenSize =
Toolkit.getDefaultToolkit().getScreenSize();
        Dimension frameSize = frame.getSize();
        if (frameSize.height > screenSize.height) {
            frameSize.height = screenSize.height;
        }
        if (frameSize.width > screenSize.width) {
            frameSize.width = screenSize.width;
        }
        frame.setLocation((screenSize.width - frameSize.width) / 2,
                           (screenSize.height - frameSize.height) /
2);
    }
}
```



```

        frame.setVisible(true);
    }
    public static void main(String[] args) {
        SwingUtilities.invokeLater(new Runnable() {
            public void run() {
                try {
                    UIManager.setLookAndFeel(UIManager.
getSystemLookAndFeelClassName());
                } catch (Exception exception) {
                    exception.printStackTrace();
                }

                new Application();
            }
        });
    }
}

```

2、练习使用对话框。包括一个 JLabel 和两个 JButton，按任何一个 JButton 都会产生一个输入对话框，按确定后将输入内容在 JLabel 中显示出来

```

import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
public class dTest{
    public static void main(String args[]){
        Form form=new Form();
        form.setSize(300,200);
        form.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        form.setVisible(true);
    }
}
class Form extends JFrame implements ActionListener{
    private Dial dia;
    JLabel lab=new JLabel(" 请输入 ");
    public Form(){
        JButton btn1=new JButton("Yes");
        JButton btn2=new JButton("No");
        getContentPane().setLayout(new GridLayout(3,1));
        add(lab);
    }
}

```

```

add(btn1);
add(btn2);
setTitle("inputprac");
btn1.addActionListener(this);
btn2.addActionListener(this);
}
public void actionPerformed(ActionEvent event){
if(event.getActionCommand()==" 输入"){
dia=new Dial(this," 请输入内容 ",true);
dia.setVisible(true);
lab.setText(dia.getSt());
}
else{
System.exit(0);
}
}
}
class Dial extends JDialog implements ActionListener{
private JTextField text=new JTextField("change");
private JButton btn1=new JButton(" 确定 ");
private JPanel panel=new JPanel();

public Dial(JFrame frame,String title,boolean b){
super(frame,title,b);
getContentPane().setLayout(new FlowLayout());
panel.add(text);
panel.add(btn1);
btn1.addActionListener(this);
getContentPane().add(panel);
pack();
}
public void actionPerformed(ActionEvent event){
this.dispose();
}
public String getSt(){
return text.getText();
}
}

```

3、.练习使用 JTable。包括姓名、学号、语文成绩、数学成绩、总分共 5 项

```

import java.awt.*;
import javax.swing.*;

public class table{
public static void main(String args[]){
JFrame form=new JFrame("Table 试验");
final Object[] row={" 姓名 "," 学号 "," 语文成绩 "," 数学成绩 "," 总分 "};
Object[][] rowData = {
{" 张三 ", "01", "81", "80", "160"},
{" 李四 ", "02", "93", "72", "165"},
{" 王麻子 ", "03", "99", "97", "196"},
};
JTable tb=new JTable(rowData,row);
JScrollPane pane= new JScrollPane (tb);
form.getContentPane().add(tb);
form.setVisible(true);
form.pack();
}
}

```

4、一个简单的 Swing GUI 应用程序。 在一个框架窗口中显示两个标签和一个按钮、上面的标签显示一串固定的文字信息，选择按钮后在下方的标签上显示系统现在的时间。

```

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.util.*;
public class SwingDemo extends JFrame
                                implements ActionListener{

    JButton b1;
    JLabel l1,l2;
    SwingDemo(){
        super("Swing 应用程序的例 ");
        l1=new JLabel(" 一个 GUI 应用程序的例子 ",JLabel.CENTER);
l2 =new    JLabel(" ");
        b1=new JButton(" 现在时间 [T]");
        b1.setMnemonic(KeyEvent.VK_T);
        b1.setActionCommand("time");
        b1.addActionListener(this);
        getContentPane().add(l1,BorderLayout.NORTH);
        getContentPane().add(l2,BorderLayout.CENTER);
    }
}

```

```

        getContentPane().add(b1, BorderLayout.SOUTH);
    }
    public void actionPerformed(ActionEvent e)
    {
        Calendar c1 = Calendar.getInstance();
        if(e.getActionCommand().equals("time")){
            l2.setText(" 现在是 "
                +c1.get(Calendar.HOUR_OF_DAY)
                +" 时 "+c1.get(Calendar.MINUTE)+" 分");
            l2.setHorizontalAlignment(JLabel.CENTER);
        }else System.exit(0);
    }

```

```

    public static void main(String args[]){
        JFrame.setDefaultLookAndFeelDecorated(true);
        JFrame frame = new SwingDemo();
        frame.setDefaultCloseOperation(
            JFrame.EXIT_ON_CLOSE);
        frame.pack();
        frame.setVisible(true);
    }
}

```

5、创建一个界面质量管理器，并显示课程建设、一般课程、精品课程，并在精品课程张显示子菜单。

```

import java.awt.*;
import javax.swing.*;
import javax.swing.tree.*;
import java.awt.event.*;
public class TestJtree extends JFrame {
    JPanel cp=new JPanel();
    JTree jtree;
    DefaultMutableTreeNode root;
    public TestJtree() {
        this.setSize(300,300);
        this.setTitle("JTree 组件 ");
        cp=(JPanel)this.getContentPane();
        cp.setLayout(new BorderLayout());
        root=new DefaultMutableTreeNode(" 质量管理 ");
        createTree(root);
        jtree=new JTree(root);
    }
}

```

```

        cp.add(jtree, BorderLayout.CENTER);
    }
    public static void main(String[] args) {
        TestJtree TestJtree = new TestJtree();
        TestJtree.setVisible(true);
    }
    private void createTree(DefaultMutableTreeNode root)    {
        DefaultMutableTreeNode classroom=null;
        DefaultMutableTreeNode number=null;
        classroom=new DefaultMutableTreeNode(" 课程建设 ");
        root.add(classroom);
        number=new DefaultMutableTreeNode(" 一般课程 ");
        classroom.add(number);
        number=new DefaultMutableTreeNode(" 重点课程 ");
        classroom.add(number);
        number=new DefaultMutableTreeNode(" 精品课程 ");
        classroom.add(number);
        number.add(new DefaultMutableTreeNode(" 国家精品课程 "));
        number.add(new DefaultMutableTreeNode(" 市级精品课程 "));
        number.add(new DefaultMutableTreeNode(" 校级精品课程 "));
        number=new DefaultMutableTreeNode(" 其他课程 ");
        classroom.add(number);    }
    protected void processWindowEvent(WindowEvent e)    {
        if(e.getID()==WindowEvent.WINDOW_CLOSING)
            System.exit(0);
    }
}

```

6、使用文件打开、关闭对话框（ JFileChooser ）。将选 择的文件名显示到文本区域中。

```

import java.io.*;
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import javax.swing.filechooser.*;
public class JFileChooserDemo extends JFrame {
    public JFileChooserDemo() {
        super("使用 JFileChooser");
        final JTextArea ta = new JTextArea(5,20);
        ta.setMargin(new Insets(5,5,5,5));
        ta.setEditable(false);
    }
}

```

```

JScrollPane sp = new JScrollPane(ta);
final JFileChooser fc = new JFileChooser();
JButton openBtn = new JButton(" 打开文件 ...");
openBtn.addActionListener(new ActionListener(){
    public void actionPerformed(ActionEvent e) {
        int returnVal = fc.showOpenDialog(
            JFileChooserDemo.this);
        if (returnVal == JFileChooser.APPROVE_OPTION) {
            File file = fc.getSelectedFile();
            ta.append("打开 : " + file.getName() + ".\n");
        } else ta.append("取消打开命令  .\n");
    }
});
JButton saveBtn = new JButton(" 保存文件 ...");
saveBtn.addActionListener(new ActionListener(){
    public void actionPerformed(ActionEvent e){
        int returnVal = fc.showSaveDialog(
            JFileChooserDemo.this);
        if (returnVal == JFileChooser.APPROVE_OPTION)
        {
            File file = fc.getSelectedFile();
            ta.append("Saving: " + file.getName() + ".\n");
        } else ta.append("取消保存命令。  \n");
    }
});
JPanel buttonPanel = new JPanel();
buttonPanel.add(openBtn);
buttonPanel.add(saveBtn);
Container c = getContentPane();
c.add(buttonPanel, BorderLayout.NORTH);
c.add(sp, BorderLayout.CENTER);
}

public static void main(String[] args) {
    JFrame frame = new JFileChooserDemo();
    frame.setDefaultCloseOperation(EXIT_ON_CLOSE);
    frame.pack();
    frame.setVisible(true);
}
}

```

7、编写一个对话框程序。

```

import javax.swing.*;

import java.awt.*;
import java.awt.event.*;

public class JOptionPaneDemo{
    public JFrame aFrame;
    public JButton aButton;
    public JOptionPane anOptionPane;
    static JLabel aLabel;
    public JOptionPaneDemo(){
        aFrame = new JFrame("使用 JOptionPane");
        aButton = new JButton(" 显示对话框 ");
        aLabel = new JLabel(" 您作出的选择是、 ",
            JLabel.CENTER);
        aButton.addActionListener(new OptionListener());
        anOptionPane = new JOptionPane();
        aFrame.getContentPane().add(aButton,
            BorderLayout.NORTH);
        aFrame.getContentPane().add(aLabel,
            BorderLayout.SOUTH);
        aFrame.setVisible(true);
        aFrame.setDefaultCloseOperation(
            JFrame.EXIT_ON_CLOSE);
    }
    public class OptionListener implements ActionListener{
        public void actionPerformed(ActionEvent e){
            String[] choices = {" 喜欢 ", " 不喜欢 "};
            int n = anOptionPane.showOptionDialog(aFrame,
                "您喜欢音乐吗？ ", " 请选择 ",
                JOptionPane.YES_NO_OPTION,
                JOptionPane.QUESTION_MESSAGE, null,
                choices,choices[0]);
            if(n == 0)
                JOptionPaneDemo.aLabel.setText(" 喜欢音乐 ");
            else JOptionPaneDemo.aLabel.setText("不喜欢音乐 ");
        }
    }
    public static void main( String[] args){
        new JOptionPaneDemo();
    }
}

```

```
}
```

8、编写一个功能选择程序，方法自定。

```
import javax.swing.*;
```

```
import java.awt.*;
```

```
import java.awt.event.*;
```

```
public class JMenuDemo extends JFrame implements ActionListener{
```

```
    JLabel j1 = new JLabel(" 请选择菜单、  ",JLabel.CENTER);
```

```
    JMenuItem aaMenuItem,baMenuItem;
```

```
    JMenuDemo(){
```

```
        super("使用 JMenu");
```

```
        JMenuBar aMenuBar = new JMenuBar();
```

```
        Icon i1 =    new ImageIcon("images/girl.gif");
```

```
        Icon i2 =    new ImageIcon("images/boy.gif");
```

```
        JMenu aMenu = new JMenu(" 菜单  A");
```

```
        JMenu bMenu = new JMenu(" 菜单  B");
```

```
        JMenuItem aaMenuItem = new JMenuItem(" 菜单项  AA",i1);
```

```
        JMenuItem abMenuItem = new JMenuItem(" 菜单项  AB",i2);
```

```
        JMenuItem baMenuItem = new JMenuItem(" 菜单项  BA");
```

```
        aMenuBar.add(aMenu);
```

```
        aMenuBar.add(bMenu);
```

```
        aMenu.add(aaMenuItem);
```

```
        aMenu.addSeparator();
```

```
        aMenu.add(abMenuItem);
```

```
        bMenu.add(baMenuItem);
```

```
        aaMenuItem.addActionListener(this);
```

```
        abMenuItem.addActionListener(this);
```

```
        baMenuItem.addActionListener(this);
```

```
        setJMenuBar(aMenuBar);
```

```
        getContentPane().add(j1,BorderLayout.CENTER);
```

```
    }
```

```
    public void actionPerformed(ActionEvent e)
```

```
    {
```

```
        JMenuItem source = (JMenuItem)(e.getSource());
```

```
        j1.setText(" 选择了菜单  :"+source.getText());
```

```
        j1.setHorizontalAlignment(JLabel.CENTER);
```

```
    }
```

```
    public static void main(String args[])
```

```
    {
```

```
        JFrame frame = new JMenuDemo();
```

```
        frame.setDefaultCloseOperation(
```



```

JFrame.EXIT_ON_CLOSE);
    frame.setVisible(true);
    frame.pack();
}
}

```

9、编辑一个登录界面，并用密码进行验证。

```

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
public class JPasswordFieldDemo extends JApplet
    implements ActionListener{

    JLabel jl1,jl2;
    JPasswordField jp1,jp2;
    JButton jb1,jb2;
    public void init(){
        Container c = getContentPane();
        jl1 = new JLabel(
            "<html><h3> 请输入您的密码 :",JLabel.CENTER);
        jl2 = new JLabel(
            "<html><h3> 请再次输入密码 :",JLabel.CENTER);
        jp1 = new JPasswordField(8);
        jp2 = new JPasswordField(8);
        jb1 = new JButton("<html><h3> 提交 ");
        jb2 = new JButton("<html><h3> 取消 ");
        c.setLayout(new GridLayout(3,2));
        c.add(jl1);
        c.add(jp1);
        c.add(jl2);
        c.add(jp2);
        c.add(jb1);
        c.add(jb2);
        jb1.addActionListener(this);
        jb2.addActionListener(this);
    }
    public void actionPerformed(ActionEvent e){
        if(e.getSource()==jb1)
            if(jp1.getPassword().length>0)
                if(String.valueOf(jp1.getPassword()).equals
                    (String.valueOf(jp2.getPassword())))
                    showStatus("密码输入成功！ ");
    }
}

```

```

        else{
            showStatus("两次输入的密码不同，请重新输入！");
            jp1.setText("");
            jp2.setText("");
        }
    else showStatus("密码不能为空！");
    if(e.getSource()==jb2){
        jp1.setText("");
        jp2.setText("");
        showStatus("");
    }
}
}
}

```

10、工资额计算器应用程序

需求分析、某公司需要一个能计算出每位员工在工作一周后的全部薪水的应用程序。每位员工的周薪是根据员工工作的小时数和每小时的工资额来计算出来的。一个标准工作周应该是 40 个小时，而一周以内任何超过 40 个小时的工作时间则被认为是属于加班时间，因而员工可获取相当于原每小时工资额的 1.5 倍加班费。

开发一个图形界面应用程序，接收员工工作的小时数和每小时的工资额并最终计算出该员工工作一周后的工资总额。同时设计一个对文本框信息的清除按钮。并且显示工资额数目的文本框是不可编译的和以红色字体显示给用户。如果工资管理员输入任何一个文本框的数据不能参加运算时，则系统会给出一条提示错误的信息。

```
package jiemian;
```

```

import java.awt.Color;
import java.awt.Font;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.*;
import org.dyno.visual.swing.layouts.Constraints;
import org.dyno.visual.swing.layouts.GroupLayout;
import org.dyno.visual.swing.layouts.Leading;

```

```

public class Frame1 extends JFrame {
    private static final long serialVersionUID = 1L;
    private JLabel jLabel0;
    private JLabel jLabel1;
    private JLabel jLabel2;
    private JLabel jLabel3;

```

```

private JTextField jTextField0;
private JTextField jTextField1;
private JLabel jLabel4;
private JButton jButton0;
private JToggleButton jToggleButton0;
private static final String PREFERRED_LOOK_AND_FEEL =
"javax.swing.plaf.metal.MetalLookAndFeel";

public Frame1() {
    initComponents();
}

private void initComponents() {
    setTitle(" 工资额计算器 ");
    setBackground(new Color(64, 128, 128));
    setLayout(new GroupLayout());
    add(getJLabel1(), new Constraints(new Leading(69, 10, 10), new
Leading(
    78, 10, 10)));
    add(getJLabel0(), new Constraints(new Leading(99, 169, 10, 10),
    new Leading(12, 35, 12, 12)));
    add(getJLabel2(), new Constraints(new Leading(57, 10, 10), new
Leading(
    72, 10, 10)));
    add(getJTextField0(), new Constraints(new Leading(134, 157, 10,
10),
    new Leading(69, 28, 12, 12)));
    add(getJLabel3(), new Constraints(new Leading(68, 12, 12), new
Leading(
    122, 10, 10)));
    add(getJTextField1(), new Constraints(new Leading(132, 160, 12,
12),
    new Leading(119, 28, 12, 12)));
    add(getJButton0(), new Constraints(new Leading(80, 77, 10, 10),
    new Leading(251, 12, 12)));
    add(getJToggleButton0(), new Constraints(new Leading(216, 82, 10,
10),
    new Leading(251, 10, 10)));
    add(getJLabel4(), new Constraints(new Leading(102, 162, 12, 12),
    new Leading(182, 34, 10, 10)));
}

```

```

        setSize(343, 300);
    }

    private JToggleButton getJToggleButton0() {
        if (jToggleButton0 == null) {
            jToggleButton0 = new JToggleButton();
            jToggleButton0.setFont(new Font("Dialog", Font.BOLD, 14));
            jToggleButton0.setText(" 取消 ");
        }
        jToggleButton0.addActionListener(new c(this));
        return jToggleButton0;
    }

    private JButton getJButton0() {
        if (jButton0 == null) {
            jButton0 = new JButton();
            jButton0.setFont(new Font("Dialog", Font.BOLD, 14));
            jButton0.setText(" 确定 ");
        }
        jButton0.addActionListener(new a(this));
        return jButton0;
    }

    private JLabel getJLabel4() {
        if (jLabel4 == null) {
            jLabel4 = new JLabel();
            jLabel4.setFont(new Font("Dialog", Font.BOLD, 14));
            jLabel4.setForeground(Color.red);
        }
        return jLabel4;
    }

    private JTextField getJTextField1() {
        if (jTextField1 == null) {
            jTextField1 = new JTextField();
            jTextField1.setText("");
        }
        return jTextField1;
    }
}

```

```
private JTextField getJTextField0() {
    if (jTextField0 == null) {
        jTextField0 = new JTextField();
        jTextField0.setText("");
    }
    return jTextField0;
}
```

```
private JLabel getJLabel3() {
    if (jLabel3 == null) {
        jLabel3 = new JLabel();
        jLabel3.setFont(new Font("Dialog", Font.BOLD, 14));
        jLabel3.setText(" 工资额、 ");
    }
    return jLabel3;
}
```

```
private JLabel getJLabel2() {
    if (jLabel2 == null) {
        jLabel2 = new JLabel();
        jLabel2.setFont(new Font("Dialog", Font.BOLD, 14));
        jLabel2.setText(" 工作时间、 ");
    }
    return jLabel2;
}
```

```
private JLabel getJLabel1() {
    if (jLabel1 == null) {
        jLabel1 = new JLabel();
    }
    return jLabel1;
}
```

```
private JLabel getJLabel0() {
    if (jLabel0 == null) {
        jLabel0 = new JLabel();
        jLabel0.setFont(new Font("Dialog", Font.BOLD, 26));
        jLabel0.setForeground(new Color(128, 0, 0));
        jLabel0.setText(" 工资额计算器 ");
    }
}
```

```

        return jLabel0;
    }

class a implements ActionListener {
    private Frame1 adaptee;

    a(Frame1 adaptee) {
        this.adaptee = adaptee;
    }

    public void actionPerformed(ActionEvent e) {
        adaptee.b(e);
    }
}

public void b(ActionEvent l) {
    try {
        String a = jTextField0.getText();
        String b = jTextField1.getText();
        double m = Double.parseDouble(a);
        double n = Double.parseDouble(b);
        if (m < 0) {
            JOptionPane.showMessageDialog(null, " 工作时间不能小于
0 ! ", " 消息信息 ",
JOptionPane.INFORMATION_MESSAGE);
        } else {
            if (m == 0 || m <= 40) {
                jLabel4.setText(" 总工资为、 " + String.valueOf(m * n));
            }
            if (m > 40) {
                jLabel4.setText(" 总工资为、 "
+ String.valueOf(40 * n + 1.5 * (m - 40) * n));
            }
        }
    } catch (Exception ee) {
        JOptionPane.showMessageDialog(null, " 请输入正确的数字信
息 ! ", " 消息信息 ",
JOptionPane.INFORMATION_MESSAGE);
    }
}
}

```

```

class c implements ActionListener {
    private Frame1 adaptee;

    c(Frame1 adaptee) {
        this.adaptee = adaptee;
    }

    public void actionPerformed(ActionEvent e) {
        adaptee.d(e);
    }
}

public void d(ActionEvent l) {
    jTextField1.setText("");
    jTextField0.setText("");
    jLabel4.setText("");
}

private static void installLnF() {
    try {
        String InfClassname = PREFERRED_LOOK_AND_FEEL;
        if (InfClassname == null)
            InfClassname =
UIManager.getCrossPlatformLookAndFeelClassName();
        UIManager.setLookAndFeel(InfClassname);
    } catch (Exception e) {
        System.err.println("Cannot install " +
PREFERRED_LOOK_AND_FEEL
+ " on this platform:" + e.getMessage());
    }
}

public static void main(String[] args) {
    installLnF();
    SwingUtilities.invokeLater(new Runnable() {
        @Override
        public void run() {
            Frame1 frame = new Frame1();

```

```

frame.setDefaultCloseOperation(Frame1.EXIT_ON_CLOSE);
    frame.setTitle(" 工资额计算器 ");
    frame.getContentPane().setPreferredSize(frame.getSize());
    frame.pack();
    frame.setLocationRelativeTo(null);
    frame.setVisible(true);
}
});
}
}

```

11、一个 Java 小应用程序，界面有三个文本框和两个按钮，前两个文本框用于输入整数，第三个文本框用于输出结果。两个按钮分别表示加和减运算。当前两个文本框输入整数后，点击相应按钮，程序就将计算结果输出在第三个文本框中。以下是程序的大部分，其中监视按钮事件的方法未给出，试编写处理按钮事件的方法。

12、编写程序的界面有一个按钮 button、一个文本区 textA 和一个文本框 textF。程序运行时，在文本区中输入数字序列，单击按钮，则在文本框中显示数字序列的和。

参考答案

第一章 java 基础

一、填空题

- 1、标识符
- 2、大小写
- 3、字母 下划线 美元符号
- 4、关键字
- 5、基本
- 6、j2se j2me j2ee
- 7、java 虚拟机 java 开发工具
- 8、布尔类型 boolean
- 9、false true
- 10、byte short int long
- 11、8 16 32 64
- 12、八 十六
- 13、float double
- 14、32 64
- 15、数字 小数点
- 16 浮点数
- 17 加法 减法 乘法 除法
- 18、+
- 19、%
- 20、++ -- =
- 21 、前
- 22、new .
- 23、boolean
- 24、true false
- 25、大于 小于 等于 大于等于 小于等于 不等于
- 26、等于 不等于
- 27、逻辑
- 28、实际
- 29、&& || !
- 30、逻辑与 逻辑或 逻辑非
- 31、包 import 语句 类 方法 语句
- 32、逻辑与和逻辑或
- 33、path1.path2.path3
- 34、开始，应用，结束
- 35、/** */ /* */ //
- 36、double

37、简单数据类型 符合数据类型
38、 true
39、 0---1 double
40、 true false
41、 N
42、 真
43、 true
44、 false
45、 false true
46、 boolean
47、 二
48、 & | ^
49、 逻辑
50、 false true
51、 1 0
52、 1
53、 0
54、 0 1
55、 << >>
56、 5 7
57、 3 -3
58、 3 class
59、 16
60、 double
61、 等于
62、 右 左
63、 变量
64、 变量
65、 main
66、 编译字节码
67、 java Application java Applet
69、 public static void main (String [] args)
70、 自动转换
71、 编译、 解释、 执行
72、 false true
73、 20
74、 十进制整数 十六进制 八进制
75、 public
76、 第一行
77、 package class

二、判断题		
1—10	FFFTTFFFFT	11-21 FFFFFFFTTTTT
三、选择题		
1-5	BADDA	6-10 CABDD
16-20	BACDB	11-15 CCDCC
21-25	BAABC	26-30 DACBD
36-40	CBACC	31-35 BBCAA
41-45	BADBB	46-50 ACCBD
56-60	CBAAC	51-55 ABBCD
61-65	ADDDC	66-70 BCDAA
76-80	ABDAD	71-75 DCBBD
81-85	DBADB	86-90 BDCCB
96-100	CBDDB	91-95 CADCC
101-105	DADAB	106-110 ABCDA
116-120	DDCDC	111-115 CDBBD
121-125	ABCCA	126-130 BBABA
136-140	DADDB	131-135 CDDBB
141-145	BDAAD	146-150 BCDBA
156-160	BBACA	151-155 ADABB
161-165	BBDAA	166-170 CDBCB
176-180	BDBCC	171-175 DCABA
181-185	DDABB	186-190 DBCCC
196-200	DCDCD	191-195 BBDDD
201-205	BCDCC	206-207 AD

四、程序阅读题		
1、	10	11
	12	12
2、	true	false false true false
3、Integer Arithmetic		
	a=2	
	b=6	
	c=1	
	d = -1	
	e=1	
	Floating Point Arithmetic	
	da = 2.0	
	db = 6.0	
	dc = 1.5	

- dd = -0.5
de = 0.5
- 4、 x mod 10 = 2
y mod 10 = 2.25
- 5、 a=6
b=8
c=3
- 6、 a = 2
b = 3
c = 4
d = 1
- 7、 a = 0011
b = 0110
a|b = 0111
a&b = 0010
a^b = 0101
~a&b|a&~b = 0101
~a = 1100
- 8、 Original value of a: 64
i and b: 256 0
- 9、 b = 0xf1
- 10、 a=3
b=1
c=6
- 11、 a = true
b = false
a|b = true
a&b = false
a^b = true
a&b|a&!b = true
!a = false
- 12、 Absolute value of 10 is 10
Absolute value of -10 is 10
- 13、 (1) -- a % b++ = 1
(2) (a >= 1 && a <= 12 ? a : b) = 2
(3) f ^ (a > b) = false
(4) (-- a) < < a = 4
(5) (double)(a+b)/5+a/b=-0.4

一、选择题

1-5 CBCBA	6-10 BADCB	11-15 CCDDD
16-20 DCCCB		
21-25 CDCBC	26-30 CCDCD	31-35 DDCBD
36-40 BADBB		
41-45 BBDBD	46-50 CCBDA	51-55 BDCDB
56-60 CDBCA		
61-65 BCAAC	66-70 BBCDD	71-75 CCCBA
76-80 CDCCB		
81-85 DADCA	86-90 ADABA	91-95 CCDDC
96-100 BDBAD		
101-105 CCDA	106-110 DCDAB	111-113 ADA

二、填空题

- 1、顺序结构 选择结构 循环结构
- 2、if 语句 switch 语句
- 3、for 语句 while 语句 do-while 语句
- 4、选择结构
- 5、循环结构
- 6、选择语句
- 7、条件语句 多分支语句
- 8、if 语句 if-else
- 9、布尔类型
- 10、do-while 语句
- 11、break 语句
- 12、continue 语句
- 13、do-while 语句
- 14、布尔类型
- 15、常量或变量 关系表达式 逻辑表达式
- 16、单选 双选 多选
- 17、if 选择语句 switch 选择语句
- 18、顺序结构
- 19、循环体 循环条件
- 20、控制转移
- 21、break continue return
- 22、开关语句 循环语句
- 23、逻辑表达式
- 24、布尔值
- 25、false
- 26、while do-while for 循环

- 27、多重循环
- 28、二重循环 三重循环
- 29、双重循环
- 30、三重循环
- 31、双重循环结构
- 32、多重循环结构
- 33、多重循环
- 34、循环体内
- 35、跳转语句
- 36、continue 语句
- 37、return 语句
- 38、语句
- 39、while 语句 for 语句 空
- 40、先判断后执行
- 41、先执行后判断
- 42、初始化部分 false true
- 43、for 循环
- 44、goto
- 45、continue break
- 46、break outer
- 47、从上到下
- 48、选择性语句
- 49、for
- 50、重复
- 51、if-else for while do-while switch
- 52、boolean
- 53、最近
- 54、for
- 55、while
- 56、while
- 57、switch
- 58、class (c++) + ?t?
- 59、class i
- 60、i<=9 j*i

三、判断题

1-10 TFTTTFFTF

11-20 FFTFTTFTFF

21-26 FFFFTT

四、程序阅读题

1、 case 2 , case 3

2、 equal

3、 56

4、 3

5、	1	2	3	4	5
	2	4	6	8	10
	3	6	9	12	15

6、 876

7、 232221

8、 sum = 6

9、 sum = 16

10、 10 次

11、 相同， sum 的计算结果均为 5050

12、 a = 8

13、 * # * # \$

14、 false

五、编程题

```
1、 public class IfAndElse{
public static void main(String args[]){
int x,y;
x=6;
if(x<0)
y=0;
else
y=2*x+1;
System.out.print(      “ F(x)= ” +y);
}
}
```

```
2、 import java.io.*;
public class Question15 {
    public static void main(String args[])throws IOException{
        int x,y;
        String str;
        BufferedReader br;
        br = new BufferedReader(new InputStreamReader(System.in));
        System.out.print("Input one integer numbers(x): ");
        str = br.readLine();
        x = Integer.parseInt(str);
        if(x>0)
```

```

        y = x+3;
    else if (x==0)
        y = 0;
    else
        y = x * x-1;
    System.out.println("y = " + y);
}
}

```

```

3、 public class Score{
public static void main(String args[]){
int score,temp;
score =88;
if(score==100)
temp=9;
else
temp=9;
switch (temp){
case 0:case 1:case 2:case 3:case 4:case 5:
System.out.print( 不及格 " ); break;
case 6:
System.out.print( 及格 " ); break;
case 7:
case 8:
System.out.print( 良好 " ); break;
case 9:
System.out.print( 优秀 " )break;
default:
System.out.print( 成绩输入错误 " ); break;
}
}
}

```

```

4、 public class ForExample
{ public static void main (String args[])
{ int n,num1,num2;
System.out.println(" 在 1~1000 可被 3 与 7 整除的数为 : ");
for (n=1;n<=1000;n++)
{ num1 =n%3;
num2=n%7;

```



```

if (num1==0)
{ if (num2==0)
System.out.print(n+" ");
}
}
System.out.println(" ");
}
}

```

```

5、 public class Prime{
public static void main(String args[])
{
int i,n;
int k=0;
float f=1.0;
for(n=1;n<=100;n++){
for(i=2;i<N;i++){
if(n%i==0)break;
}
if(i==n)
{ System.out.print(n+" ");
k++;
if(k%5==0)
System.out.println();
}
}
}
}

```

```

6、 import java.io.*;
public class IfElse
{
public static void main(String args[]) throws IOException
{
int month = 4; // April
String season; String str;
BufferedReader buf;
buf=new BufferedReader(new InputStreamReader(System.in));
System.out.print(" 输入月份 :");
str=buf.readLine();

```

```

month=Integer.parseInt(str);
if(month == 12 || month == 1 || month == 2)
season = "Winter";
else if(month == 3 || month == 4 || month == 5)
season = "Spring";
else if(month == 6 || month == 7 || month == 8)
season = "Summer";
else if(month == 9 || month == 10 || month == 11)
season = "Autumn";
else
season = "Bogus Month";
System.out.println(" 你输入的月份是    " + season + ".");
}
}

```

```

7、 import java.lang.Double;
public class HuiKuan {
public static void main(String args[]) {
double remittance, result;
if(args.length != 1) // 处理参数数目不正确情况    {
System.out.println("Please Input a number!");
return; }
remittance = Double.parseDouble(args[0]);
if (remittance < 0) { System.out.println("Please Input a positive number!");
return; }
if (remittance < 100) { result = 1.0; }
else if (remittance > 5000) { result = 50.0; }
else { result = remittance / 100.0; }
System.out.println("remittance=" + remittance + " result=" + result);
}
}

```

```

8、 public class Text {
public static void main(String args[]) {
int n; boolean result;
if(args.length != 1) // 处理参数数目不正确情况    {
System.out.println("Please Input a number!");
return; }
n = Integer.parseInt(args[0]);
if (n < 0) { System.out.println("Illegal Year!");

```

```

return; }
if ((n % 100) == 0) { if ((n % 400) == 0) result = true;
else result = false; }
else { if ((n % 4) == 0) result = true;
else result = false; }
System.out.println("Year:" + n + " " + result);
}
}

```

```

9、 public class Text {
public static void main(String args[]) {
int n, result, i, temp;
if(args.length != 1) // 处理参数数目不正确情况
{ System.out.println("Please Input a number!");
return; }
if (args[0].length() != 4) {
System.out.println("Illegal number!");
return; }
n = Integer.parseInt(args[0]);
if (n < 0) {
System.out.println("Illegal number!"); return; }
result = 0;
for (i = 1; i <= 4; i++) {
switch (i) {
case 1: {
temp = n / 1000; // 获得千位数
result += temp; // 转成个位数
break; }
case 2: {
temp = (n % 1000) / 100; // 获得百位数
result += 10 * temp; // 转成十位数
break; }
case 3: {
temp = (n % 100) / 10; // 获得十位数
result += 100 * temp; // 转成百位数
break; }
case 4: {
temp = (n % 10); // 获得个位数
result += 1000 * temp; // 转成千位数
break; }
}
}
}
}

```

```

default:break; }
}
System.out.println( n + " --> " + result);
}
}

```

```

10、 public class BaseText {
    public static void main(String[] args) {
        if(args != null && args.length == 2)        {

            try {
                double num1 = Double.parseDouble(args[0]);
                double num2 = Double.parseDouble(args[1]);

                System.out.println(" 得到的乘积是  :"+num1*num2);
            } catch (NumberFormatException e) {
                e.printStackTrace();
            }

        }else {
            System.out.println(" 你输入的数据不正确    ,输入数据的格
式样例为 :12.3 4");
        }
    }
}

```

```

11、 public class BaseText {
    public static void main(String[] args) {
        if(args != null && args.length == 1        )        {
            try {
                double num = Double.parseDouble(args[0]);

                int intValue = (int)num;

                switch (intValue) {
                    case 1:
                        System.out.println(" 今天是星期一  ");
                        break;
                    case 2:
                        System.out.println(" 今天是星期二  ");
                        break;
                }
            }
        }
    }
}

```

```

        case 3:
            System.out.println(" 今天是星期三 ");
            break;
        case 4:
            System.out.println(" 今天是星期四 ");
            break;
        case 5:
            System.out.println(" 今天是星期五 ");
            break;
        case 6:
            System.out.println(" 今天是星期六 ");
            break;
        case 7:
            System.out.println(" 今天是星期天 ");
            break;
        default:
            args = null;
            System.out.println(" 请重新输入 ");
            break;
    }
} catch (NumberFormatException e) {
    e.printStackTrace();
}

} else {
    System.out.println(" 你输入的数据不合法 ");
}

}
}

```

```

12、 import java.io.*;
    public class MaxDemo{
    public static void main(String []asd){
    int i =1,j=2,k=3;
    int max;
    System.out.println( “ 这三个数分别是、 ” );
    System.out.println(i+ “ ” +j+ “ ” +k+ “ ” );
    max = i;
    if(j>max)max = j;
    if(k> max)max = k;
    System.out.println( “ 最大数为、 ” +max);

```

```

    }
}
13、 public class Lesson1{
public static void main(String []as){
int n =0;    //初始化部分
int t=0;
do{
n++;
t=t+n;
}
while(t<=10000);
System.out.println( “ 由于、 1+,,    +n ” +n+ ” = ” +t+ ” \n 故、 需、 ” +n+ ”
个连续的自然数相加 , 其和才会超过    1000. ” );
}}
14、 public class test1{
public static void main(String []as){
for(int i=0;i<4;i++){
for(int j=0;j<12;j++){
Sysrem.out.print( “ * ” );}
System.out.println();
}
}
}
15、 public class test2{
public static void main(String []as){
for(int i=0;i<5;i++){
for(int j=0;j<=i;j++){
System.out.print( “ * ” );}
System.out.println();
}
}
16、 public class lianxi1 {
    public static void main(String []ads){
        int grade=97;
        if (grade>=90)
            System.out.println( "A" );
        else if (grade>=80)
            System.out.println( "B" );
        else if (grade>=70)
            System.out.println( "C" );
    }
}

```

```

        else if (grade>=60)
            System.out.println( "D" );
        else if (grade<60)
            System.out.println( "E" );
    }
}

```

```

17、 public class ZhiShu {
public static void main(String args[]) {
int i, j, result;
boolean found;
for (i = 200; ; i++) {
found = true;
for (j = 2; j < i; j++) {
if ((i % j) ==0) {
found = false; break; }
}
if (!found) continue;
else {
result = i; break; }
}
System.out.print("The minimum prime number ");
System.out.println("great than 200 is : " + result);
}
}

```

```

18、 public class IfElseTest {
    public static void main(String[] args) {
        int a=3;
        if(a==1)
            System.out.println(" 进行加法运算 ");
        else if(a==2)
            System.out.println(" 进行减法运算 ");
        else if(a==3)
            System.out.println(" 进行乘法运算 ");
        else
            System.out.println(" 进行除法运算 ");
    }
}

```

```

19、 import java.io.BufferedReader;
import java.io.IOException;
public class TT {
    public static void main(String[] args) {
        int water = 15;
        int i = 0;
        while (water < 50) {
            water += 5;
            i++;
        }
        System.out.println(" 和尚共挑了 "+i+" 次水 ");
    }
}

```

```

20、 [TestWhile.java]
    class TestWhile{
        public static void main(String[] args){
            int sum = 0, num = 1;
            while(num<=100){
                sum += num;
                num ++;
            }
            System.out.println("sum: " + sum);
        }
    }

```

```

[TestDoWhile.java]
    class TestDoWhile{
        public static void main(String[] args){
            int sum=0,num=1;
            do{
                sum += num;
                num ++;
            }while(num<=100);           //while 后的分号不可缺
            System.out.println("sum: " + sum);
        }
    }

```

```

[TestFor.java]
    class TestFor{
        public static void main(String[] args){
            int sum,num;
            for(sum=0, num=1; num<=100; num++)
                sum += num;
        }
    }

```



```

        System.out.println("sum: "+sum);
    }
}
21、 class TestBreak{
    public static void main(String[] args){
        int sum = 0, num = 1;
        while( true ){
            sum += num;
            if( sum>1000 ) break;
            num ++ ;
        }
        System.out.println("N: " + (num-1));
    }
}
22、 class MutiTable{
    public static void main( String[] args ){
        for( int i=1; i<=9; i++ ) {
            for ( int j=1; j<=i; j++)
                System.out.print( i + "*" +j+ "=" + i*j + "\t");
            System.out.println();
        }
    }
}
23、 import java.util.Scanner;
    class Monthdays{
        public static void main(String[] args){
            int month,days;
            Scanner sc = new Scanner(System.in);
            System.out.print("Month? ");
            month = sc.nextInt();
            if(month<1 || month>12){           //若输入数据非月份数
                System.out.println("error data");
                System.exit(0);                //退出运行
            }
            switch(month){
                case 4: case 6: case 9: case 11: days = 30;break;
                case 2:           //2 月份需要输入年份进一步判断
                    System.out.print("Year? ");
                    int year = sc.nextInt();
                    if(year%400==0 || year%4==0 && year%100!=0)

```

```

        days = 29;
        else days = 28;
        break;
        default: days = 31;
    }
    System.out.println("days: "+days);
}
}
24、 import java.util.Scanner;
    class PrimeTest{
        public static void main(String[] args){
            Scanner sc = new Scanner(System.in);
            System.out.print("number? ");
            int n = sc.nextInt();    //n 为被测数
            int i;                  //i 为 1 和本身外可能的因子
            for(i=2;i<n;i++)
                if(n%i == 0)break;    //i 是 n 的因子 , n 非素数
            if(i == n)                //所有可能的因子全非 n 的因子
                System.out.println( n + " is a prime number." );
            else
                System.out.println(n + " is not a prime number");
        }
    }
25、 class PrimeNumbers{
        public static void main(String[] args){
            int i,n;
            for( n = 101; n<1000; n+=2 ){    //n 遍历所有三位奇数
                for( i=2; i<n; i++)
                    if( n%i == 0 )break;
                if( i==n )System.out.print( n + "\t");
            }
        }
    }
26、 class Fei{
        public static void main(String[] args){
            int a,b,c,sum;
            a = b = 1;                //a,b 为前两项 , 均为 1
            sum = a + b;                //累加器变量 , 设初值
            System.out.print( a + "\t" + b );
            for( int i=1; i<=18; i++ ){

```

```

        c=a+b;           //生成的新项置放在变量    c 中
        System.out.print("\t" + c);
        sum += c;
        a = b;           //前两项后移一个数
        b = c;
    }
    System.out.println("\tsum=" + sum);
}
}

```

27、

```

public class Guess{
public static void main(String[] args){
int a,b;
for(a=14;a<=90;a++){
for(b=11;b<=50;b++){
if(((a-10)==(b-10)*4)&&((a+10)%(b+10)==0)){
System.out.println(a);
System.out.println(b);
}}}}

```

28、

```

public class Rmb{
public static void main(String[] args){
int count=0;
for(int a=0;a<20;a++){
for(int b=0;b<20;b++){
for(int c=0;c<20;c++){
if((a+b+c==20)&&(1*a+2*b+c*5==60)){
count++;
System.out.print(" 一元面钞张数 :  "+a);
System.out.print(" 二元面钞张数 :  "+b);
System.out.println(" 五元面钞张数 :  "+c);
}}}}
System.out.println(" 一共的取法 :  "+count);
}}

```

29、 import java.io.*;

```

public class Question17 {

```

```

    public Question17() { /* constructor method */
    }

```

```

    public static void main(String args[])throws IOException{

```

```

        int ch;
        System.out.print(" 请输入大写字符、 ");
        ch=System.in.read();
        System.out.println((char)(ch+32));
    }
}

```

30、 import java.io.*;

```

public class Question18 {

    public Question18() { /* constructor method */
    }

    public static void main(String args[])throws IOException{
        int m,b;
        String str;
        BufferedReader br;
        br = new BufferedReader(new InputStreamReader(System.in));
        System.out.print(" 请输入一个三位整数 (m)、 ");
        str = br.readLine();
        m = Integer.parseInt(str);
        b = (m%10)*100 + ((m%100)/10)*10 + m/100;
        System.out.println(" 原输入的三位整数是 (m)、 " + m);
        System.out.println(" 变换后的三位整数是 (b)、 " + b);
    }
}

```

31、 public class Question20{

```

    public Question20(){ /* constructor method */
    }

    public static void main(String args[]){
        int m,cnt = 0;
        for(m=100;m<=200;m++){
            if(m%3==0)
                continue;
            else{
                cnt++;
                System.out.print(" " + m);
                if(cnt%10==0) /* 以每行 10 个数输出 */

```

```

        System.out.print("\n");
    }
}
System.out.println("\n    100-200 之间不能被  3 整除的数共有    " +
cnt + "    个。 ");
}
}

```

32、 import java.io.*;

```
public class Question21 {
```

```

    public Question21() { /* constructor method */
    }

```

```

    public static void main(String args[])throws IOException{
        int m,n,gys,gbs,temp,mult;
        String str;
        BufferedReader br;
        br = new BufferedReader(new InputStreamReader(System.in));
        System.out.print("Input first integer numbers(m): ");
        str = br.readLine();
        m = Integer.parseInt(str);
        System.out.print("Input second integer numbers(n): ");
        str = br.readLine();
        n = Integer.parseInt(str);
        mult = m * n;
        if(m<n){
            temp = m;
            m = n;
            n = temp;
        }
        temp = m % n;
        while(temp!= 0){
            m = n;
            n = temp;
            temp = m % n;
        }
        gys = n;
        gbs = mult / gys;
        System.out.println(" 最大公约数是、    " + gys);
    }
}

```

```

        System.out.println(" 最小公倍数是、    " + gbs);
    }
}

```

33、 import java.io.*;

```

public class Question22 {

```

```

    public Question22() {
    }

```

```

    public static void main(String args[])throws IOException{
        double allIncome,incomeForTax,individualIncomeTax = 0;
        String str;
        BufferedReader br;
        br = new BufferedReader(new InputStreamReader(System.in));
        System.out.print("Input personal income(allIncome): ");
        str = br.readLine();
        allIncome = Double.parseDouble(str);
        incomeForTax = allIncome - 1200;
        if (incomeForTax <= 0)
            individualIncomeTax = 0;
        if ((incomeForTax > 0)&&(incomeForTax <= 500))
            individualIncomeTax = incomeForTax * 0.05;
        if ((incomeForTax > 500)&&(incomeForTax <= 2000))
            individualIncomeTax = 25 + (incomeForTax - 500) * 0.1;
        if ((incomeForTax > 2000)&&(incomeForTax <= 5000))
            individualIncomeTax = 175 + (incomeForTax - 2000) * 0.15;
        if ((incomeForTax > 5000)&&(incomeForTax <= 20000))
            individualIncomeTax = 625 + (incomeForTax - 5000) * 0.2;
        if ((incomeForTax > 20000)&&(incomeForTax <= 40000))
            individualIncomeTax = 3625 + (incomeForTax - 20000) * 0.25;
        if ((incomeForTax > 40000)&&(incomeForTax <= 60000))
            individualIncomeTax = 8625 + (incomeForTax - 40000) * 0.3;
        if ((incomeForTax > 60000)&&(incomeForTax <= 80000))
            individualIncomeTax = 14625 + (incomeForTax - 60000) * 0.35;
        if ((incomeForTax > 80000)&&(incomeForTax <= 100000))
            individualIncomeTax = 21625 + (incomeForTax - 80000) * 0.4;
        if (incomeForTax > 100000)
            individualIncomeTax  = 29625 + (incomeForTax - 100000) *

```

```

0.45;
        System.out.println(" 您本月个人收入总计为      " + allIncome + "
元人民币 !");
        System.out.println(" 您 本 月 个 人 需 要 纳 税 为      " +
individualIncomeTax + " 元人民币 !  ");
    }
}

```

```

34、 import java.util.Scanner;
    public      class PrimeTest{
        public static void main(String[] args){
            Scanner sc = new Scanner(System.in);
            System.out.print("number? ");
            int n = sc.nextInt();
            int i;
            for(i=2;i<n;i++)
                if(n%i == 0)break;
            if(i == n)
                System.out.println( n + " is a prime number." );
            else
                System.out.println(n + " is not a prime number");
        }
    }

```

```

35、 import java.util.*;
public class java5_1 {
    public static void main(String[] args) {
        System.out.println(" 我心里有一个 0 到 99 的整数，你来猜猜看、      ");
        int number=(int)(Math.random()*100); // 随机产生一个数字
        Scanner input=new Scanner(System.in);
        for(int i=0;i<20;) {
            int answer=input.nextInt();
            i++;
            if(answer!=number&& i==20){
                System.out.println(" 很可惜，你已经没机会了！      ");break;
            }
            if(answer>number) {
                System.out.println(" 大了点，再猜  ");
            } else if(answer<number) {
                System.out.println(" 小了点，再试  ");
            } else{
                System.out.println(" 猜对了！ "+" 共用 "+i+" 次。 ");
            }
        }
    }
}

```

```

switch(i){
case 1:System.out.println(" 你太有才了！  ");break;
case 2:System.out.println(" 这么快就猜出来了，不错！  ");break;
case 3:System.out.println(" 这么快就猜出来了，不错！  ");break;
case 4:System.out.println(" 这么快就猜出来了，不错！  ");break;
case 5:System.out.println(" 这么快就猜出来了，不错！  ");break;
case 6:System.out.println(" 这么快就猜出来了，不错！  ");break;
default :System.out.println(" 孩子还需继续努力啊！  ");break;
} } } }

```

36、 Public static void main()

```

{
int i,j,k;
system.out.println("\n");
for(i=1;i<5;i++) /* 以下为三重循环 */
    for(j=1;j<5;j++)
        for (k=1;k<5;k++)
            {
                if (i!=k&& i!=j&& j!=k) /* 确保 i、 j、 k 三位互不相同 */
                    System.out.println("%d,%d,%d\n",i,j,k);
            }
}

```

37、 import java.util.Scanner;

```

public class HuiWen {
public static void main(String[] args) {
    // TODO Auto-generated method stub
    Scanner scan = new Scanner(System.in);
    String text = null;
    while(true) {
        System.out.print(" 请输入字符串、  ");
        text = scan.nextLine().trim();
        if(text.equals("")) {
            System.out.println(" 警告、输入不能为空 ");
            continue;
        }
        boolean is = true;
        //检查对称位置的字符是否相同，如果找到一对位置不相同，说明
        不是回文字符串
        //把 is 变为 false,跳出 for 循环
        for(int i=0;i<text.length();i++) {
            char c1 = text.charAt(i);

```



```

        char c2 = text.charAt(text.length() - 1 - i);
        if(!(c1 == c2)) {
            is = false;
            break;
        }
    }
    if(is)
        System.out.println(" 字符串  :"+text+"  是回文字符串  ");
    else
        System.out.println(" 字符串  :"+text+"  不是回文字符串  ");
    System.out.print(" 是否继续 ( Y ) ? ");
    String yes = scan.nextLine().trim();
    if(yes.equalsIgnoreCase("Y"))
        continue;
    else
        break;
}
System.out.println(" 测试退出 ");
}
}

```

```

38、 public static void main(String[] args) {
    for (int i = 0; i < 5; ++i){
        for (int j = i; j < 5; ++j){
            System.out.print(" ");
        }
        for (int j = 0; j < 3 + 2 * i; ++j){
            System.out.print("*");
        }
        System.out.println();
        for (int i = 0; i < 4; ++i){
            for (int j = 0; j < 4; ++j)
                {System.out.print(" ");}
            for (int j = 0; j < 5; ++j){
                System.out.print("*");
            }
            System.out.println();
        }
    }
}

```

```

39、 import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
public class A1 {
    static void eqs(String str)
    {

```

```

int sun = Integer.parseInt(str);

for(int i=0;i<sun;i++)// 循环行数 假如你输入的是 5 那么就循环 5 次
{
//那么第一首先是你说的 # 那么我们就输出 #
System.out.print("# ");// 不换行的输出
for(int j=0;j<sun-2;j++)// 虚幻后面的 应该是输入进来的 N-1 但是最后后面还要保留一个 #
//所以 N-2 空留 2 个#的位置
{
System.out.print("+ ");
}
System.out.println("#");// 最后记得输出最后一个 #
}
}

public static void main(String[] args) {
String str = "";// 定义一个 str 接受字符串
BufferedReader br = new BufferedReader(new
InputStreamReader(System.in));
try {
while (true)
{
System.out.println(" 请输入一个数字 ");
str = br.readLine();// 接受从键盘输入进来的字符
eqs(str);
}
} catch (IOException e) {
// TODO Auto-generated catch block
e.printStackTrace();
} } }

```

第三章 字符串

一、选择题

1-5 CACCC	6-10 CACCC	11-15 DDDAD	16-20 DDBDB
21-25 DADDC	26-30 CACDD	31-35 DBDAC	36-40 DABCB
41-45 BACBA	46-50 BABDA		

二、填空题

- 1、String StringBuffer
- 2、true false
- 3、true true
- 4、-25 4
- 5、String a= "a "; String a =new String("a ");
- 6、String 类 StringBuffer 类
- 7、'Hello ' "Hello word ! "
- 8、int [] y ={1,2,3,4,5}
- 9、3
- 10、a.[i].length
- 11、7
- 12、'hello word "
- 13、s1+=s2
- 14、sq.equal(s2)
- 15、"edcba "
- 16、"acde "
- 17、int sum=0; for(int x=0;x<a.length; x++)
- 18、i<a.length total+a[i]
- 19、extends i+=2

三、判断题

1-11 F T F F T F F F T F T

四、编程题

```
1、 public class StringTest {
public static void main(String[] args) {
    String a = "this is String test";
    System.out.println(" 获得字符串的第一个字符、    "+a.charAt(0));
    System.out.println(" 获 得 字 符 串 的 最 后 一 个 字 符 、
    "+a.charAt(a.length()-1));
    int m = 0;
    if(a.length()>=2){
        m = 1;
        System.out.println(" 删除第一个字符、    "+a.substring(m, a.length()));
        System.out.println(" 删 除 最 后 一 个 字 符 、    "+a.substring(0,
a.length()-1));
    }else{
        System.out.println(" 删除第一个字符、    null");
        System.out.println(" 删除最后一个字符、    null");
    }
}
```

```
}  
}
```

```
}
```

```
2、 public class Test {  
    public static void main(String[] args) {  
        String input = "aaa 果 4 金 5a";  
        StringBuffer sb = new StringBuffer();  
        char[] c = input.toCharArray();  
        for (int i = 0; i < c.length; i++) {  
            if (c[i] >= '0' && c[i] <= '9') {  
                int tmp = Integer.valueOf(String.valueOf(c[i]));  
                for (int j = 0; j < tmp; j++) {  
                    sb.append(c[i + 1]);  
                }  
                i++;  
            } else {  
                sb.append(c[i]);  
            }  
        }  
  
        System.out.println(sb.toString());  
    }  
}
```

```
3、 import java.util.*;  
public class Test {  
    public static void main(String[] args) {  
        Map<Integer,String> map=new Hashtable<Integer,String>();  
        System.out.println(" 请输入字符串 #最后以空格结束输入！  
");  
  
        Scanner input=new Scanner(System.in);  
        String str=input.nextLine();  
        while(!str.equals(" "))  
        {  
            map.put(str.length(), str);  
            str=input.nextLine();  
        }  
        Set<Integer> key=map.keySet();  
        int[] sort=new int[key.size()];  
        int i=0;
```

```

        for(int num:key)
        {
            sort[i]=num;
            i++;
        }
        Arrays.sort(sort);
        System.out.println(" 最长字符串是、
"+map.get(sort[sort.length-1]));
    }

```

4、 public class Test {

```

    public static void main(String[] args) {

```

```

        }else
        {
            System.out.println(s1+" 和 "+s2+" 不同 ");
        }

```

```

        String s1="you are a student";
        String s2="how are you";
        if(s1.equals(s2))
        {
            System.out.println(s1+" 和 "+s2+" 相同 ");

```

5、 public class Test {

```

    public static void main(String[] args) {

```

```

        String s3="22030219851022024";

```

```

        String s4="220302";

```

```

        if(s3.startsWith(s4))
        {
            System.out.println(s3+" 的前缀为 "+s4);

```

```

        }else

```

```

        {
            System.out.println(s3+" 的前缀不为 "+s4);
        }

```

```

        if(s3.endsWith(s4))

```

```

        {
            System.out.println(s3+" 的后缀为 "+s4);

```

```

        }else

```

```

        {

```

```

        System.out.println(s3+" 的后缀不为 "+s4);
    }
}
}
6、 public class Test {
    public static void main(String[] args) {
        String s5="你";
        String s6="我";
        if(s5.compareTo(s6)<0)
        {
            System.out.println("\""+s5+"\" 比 \""+s6+"\" 大");
        }else if(s5.compareTo(s6)==0){
            System.out.println("\""+s5+"\" 和 \""+s6+"\" 等");
        }else
        {
            System.out.println("\""+s5+"\" 比 \""+s6+"\" 小");
        }
    }
}
7、 public class Test {
    public static void main(String[] args) {
String s7="100";
        String s8="123.678";
        int num1=Integer.parseInt(s7);
        Double num2=Double.parseDouble(s8);
        System.out.println(num1);
        System.out.println(num2);
    }
}
8、 public class Test {
    public static void main(String[] args) {
String s9="FEDCBA";
        char [] arr=s9.toCharArray();
        for (int i = 0; i < arr.length; i++) {
            System.out.println(arr[i]);
        }
    }
}
9、 import java.io.*;
public class InAndOut

```

```

{
public static void main(String args[]) throws IOException
{
int num1,num2,sum;
String str;
BufferedReader buf;
buf=new BufferedReader(new InputStreamReader(System.in));
System.out.print("Input the first integer:");
str=buf.readLine();
num1=Integer.parseInt(str);
System.out.print("Input the second integer:");
str=buf.readLine();
num2=Integer.parseInt(str);
sum=num1+num2;
System.out.println("The sum is "+sum);
}
}

```

```

10、 public class Rotor{
public static void main(String args[]){
String str,sub1;
int i;
if(args.length<0){
System.out.println(" 请输入一个命令行参数字符串！  ");
System.exit(0);
}
str=args[0];
String temp="";
for(i=0;i<STR.LENGTH();i++){
sub1=str.substring(i,i+1);
temp=sub1+temp;
}
if(str.equals(temp))
System.out.println(str+" 是回文！ ");
else
System.out.println(str+" 不是回文！ ");
}
}

```

```

11、 public class UseString{
public static void main(String args[]){

```

```
String str="ABCDEFGHJKLM";
System.out.println(str);
for(int i=0;i<STR.LENGTH();i++){
System.out.println(str.charAt(str.length()-1-i));
}
}
}
```

```
12、 public class Text{
public static void main(String args[]) {
if (args.length == 0) {
System.out.println("There are no string inputted");
return; }
String s = args[0];
int i, times = 0;
for (i = 0; i < s.length(); i++) {
if (s.charAt(i) == 'e') times++; }
System.out.println("There are " + times + " 'e' in string " + s);
}
}
```

```
13、 public class Text{
public static void main(String args[]) {
if (args.length == 0) {
System.out.println("There are no string inputted");
return; }
StringBuffer sb = new StringBuffer(args[0]);
int i = 0;
while (i < sb.length()) {
if (sb.charAt(i) == 's') {
sb.delete(i, i+1); }
else i++; }
System.out.println("original string : " + args[0]);
System.out.println("result string without 's' : " + sb);
}
}
```

```
14、 public class Text{
public static void main(String args[]) {
if (args.length == 0) {
System.out.println("There are no string inputted");
return; }
StringBuffer sb1 = new StringBuffer(args[0]);
```



```

StringBuffer sb2 = new StringBuffer();
int i, result = 0;
char ch;
for (i = 0; i < sb1.length(); i++) {
    ch = sb1.charAt(i);
    if ((ch >= '0') && (ch <= '9')) {
        sb2.append(ch); }
    }
    if (sb2.length() > 0) result = Integer.parseInt(sb2.toString());
    System.out.println("The result is : " + result);
}

```

```

15、 import java.util.HashMap;
import java.util.Map;
public class CountCharNum {
    public static void main(String[] args) {
        String aString = "yekmaakkccekymbvb";
        int count = 0;
        String singleString = "";
        Map stringMap = new HashMap();
        for (int i = 0; i < aString.length(); i++) {
            singleString = String.valueOf(aString.charAt(i));
            System.out.println(singleString);
            CountCharNum ccn = new CountCharNum();
            // 第一次出现的字符
            if (stringMap.get(singleString) == null) {
                count = 0;
            } else {
                count = (Integer)stringMap.get(singleString);
            }
            stringMap.put(singleString, count + 1);
        }
        System.out.println(stringMap);
    }
}

```

```

16、 public class StringSplit {
    String splitString;
    int splitBytes;
    public StringSplit(String splitString, int splitBytes) {
        super();
        this.splitString = splitString;
    }
}

```

```

        this.splitBytes = splitBytes;
        System.out.println("The string is : " + splitString
            + ",the splitbytes is : " + splitBytes);
    }
    public void splitIt() {
        int num = 0;
        StringBuffer sb = new StringBuffer();
        int i = 0;
        if (splitBytes >= 2) {
            while (num < splitBytes) {
                if (splitString.substring(i, i + 1).matches("[\u4e00-\u9fa5]+")) {
                    num += 2;
                } else {
                    num++;
                }
                sb.append(splitString.substring(i, i + 1));
                i++;
            }
            System.out.println("final string :" + sb.toString());
        }
    }
}
public static void main(String[] args) {
    StringSplit ss = new StringSplit(" 我 ABCD 妈 E 爸 F", 5);
    ss.splitIt();
}
17、 package com.searchkiller;
import java.util.ArrayList;
import java.util.Collections;
import java.util.List;
public class AlphaAsc {
    public static void main(String[] args) {
        String[] aString = {"sbc","fds","des"};
        List list = new ArrayList();
        for (int i = 0; i < aString.length; i++) {
            list.add(aString[i]);
        }
        Collections.sort(list);
        System.out.println(list);
    }
}

```

```

}
18、 public class xt5_01
{
static int n;
public static void main (String[] args)
{
if(args.length > 0)
{
n=Integer.parseInt(args[0]);
int[ ] a=new int[n];
a=myArray(n);
printA(a);
}
else {    System.out.println(" 缺命令行参数 !");    System.exit(-1);    }
}
static void printA(int[ ] x)
{
for(int i=0; i< x.length; i++)
System.out.print(x[i]+"    ");
System.out.println( );
return;
}
static int[ ]    myArray(int k)
{
int x[ ]=new int[k];
for(int i=0; i< k; i++)
x[i]=50+(int)(Math.random( )*50);
return    x;
}
}
19、 public class xt5_02
{
public static void main (String[] args)
{
if( args.length < 2 )
77 {
System.out.println( "Usage: string1    string2");
System.exit(-1);
}
if(args[0].equals(args[1]))

```

```

System.out.println( args[0]+" 和 "+args[1]+" 相等 ");
else
System.out.println( args[0]+" 和 "+args[1]+" 不等 ");
}
}
20、 public class xt5_03
{
public static void main(String args[])
{
int count=0;
if(args.length!=0)
{
System.out.println(args[0]);
if((int)args[0].charAt(0)>=65 && (int)args[0].charAt(0)<=90)
{
for(int i=0;i<args[0].length( );i++)
if((int)args[0].charAt(i)>=65 && (int)args[0].charAt(i)<=90)
{ System.out.print(args[0].charAt(i));
count++;      }
System.out.println( );
System.out.println(" 共有 "+count+" 个大写字母。 ");
}
else System.out.println(" 第一个字母不是大写字母  !");
}
else System.out.println(" 请给出命令行参数  !");
}
}
21、 public class xt5_04 {
public static void main (String args[])
{
String str;
str=args[0];
int j=str.length();
System.out.println(str+" "+j);
for(int i=j-1;i>=0;i--){
System.out.print(str.charAt(i)+" ");
}
}
}
}

```

```
22、 import java.util.Scanner;
public class test{
public static void main(String[] args) {
String [] strings=new String[]{"tom","is","a","boy"};
System.out.println(" 请输入一段文字  ");
Scanner input = new Scanner(System.in);
StringBuffer sb = new StringBuffer();
for(int i=0;i<strings.length;i++){
String temp = input.next();
if(temp.equals(strings[i]))
sb.append(strings[i]+" ");
else{
sb.append("?");
for(int j=0;j<temp.length();j++){
if(strings[i].indexOf(temp.charAt(j))!=-1)
sb.append(temp.charAt(j));
}
sb.append("? ");
}
}
System.out.println(sb);
}
```

第四章 数组

一、 选择题

- | | | | |
|-------------|-------------|-------------|-------|
| 1-5 CDAAC | 6-10 BACBB | 11-15 CDABD | 16-20 |
| DACAA | | | |
| 21-25 BDCDC | 26-30 ABCCA | 31-35 DACBD | 36-40 |
| BBAAB | | | |
| 41-43 CCD | | | |

二、 判断题

- | | |
|---------------------------|---------------------------|
| 1-10 T T F T F F T T F T | 11-20 T T F T F T F T F F |
| 21-30 F F F T F F T T T T | 31-40 F F F F F T F T T F |
| 41-50 F T T T T F F T F F | 51-59 T T F F F F T T T |

三、 填空题

- 1、 45
- 2、 3
- 3、 4
- 4、 new int
- 5、 56

6、 5
7、 书 java
8、 69
9、 3
10、 0
11、 a[b]
12、 a.length-1 b new
13、 4
14、 finally
15、 5
16、 0 0
17、 null null

四、编程题

1、 calss Array2

```
{
public static void main(String args[]){
final int N=10;
int i,j,temp;
int a[]={20,10,50,40,30,70,60,80,90,100};
System.out.println("The source data:");
    int min=a[0];
for(i=0;i<N;i++)
System.out.println(a[i]+"    ");
System.out.println();
//sorting
for(i=0;i<N-1;i++){
    if(min>a[i]){
temp = a[i];
a[i]=a[min];
a[min]=temp;
}
System.out.println("The sorting result:");
System.out.print(min+" ");

}
```

2、 public static void main(String [] args){
int number=100;// 数组的个数
int a[]=new int[number];

```
int min=a[0];// 最小值
int max=a[0];// 最大值
float average=0;// 平均值
int sum=0;// 数组元素的和
```

```
for(int i=0;i<number;i++){
if(min>a[i])min=a[i];
if(max<a[i])max=a[i];
sum+=a[i];
}
```

```
average=sum*1.0/number;
}
```

```
3、 public class Test{
public static void main(String[] args){
int a[]=new int[10];
```

```
for(int b=a.length-1,b<a.length;b--){
a[b]=b;
}
```

```
System.out.println( "a[b]= "+b);
```

```
}
}
```

```
4、 public class Test{
public static void main(String [] args){
int a[][]=new int[2][6];
for(int c=0;c<2,c++){
    for(int d=0;d<6;d++){
        a[c][d]=d;
```

```
}
}
```

```
System.out.println(a[c][d]);
```

```
}
}
```

```
5、 public class Te {
    public static void main(String[] args) {
        int []a={12,34,67,789,78};
        int []b={34,78};
        if(a.length>b.length){
```

```

        System.out.println(a.length);

    }else{
        System.out.println(b.length);
    }
}

```

```

}

```

```

6、 public class Test{
    public static void main(String[]args){
        int [] c=new int [10];
        for(int w=0;w<c.length;w++){
            int a=(int)(Math.random()*100+1) ;
            c[w]=a;
        }
        System.out.println(c[w]+ "    =    " +a);
    }
}

```

```

7、 public class Test{
    public static void main(String [] args){
        int a[]=new int[10];
        for(int w=a.length-1  ; w<a.length;w--){
            a[w]=w;
        }
        System.out.println(a[w]+ "    =    "+w);
    }
}

```

```

8、 public class T0taDay {
    public static void main(String[] args) {

```

```

        int year=2010,month,day=26;
        int[] MonthDay={0,31,28,31,30,31,30,31,31,30,31,30,31};
        int total;
        for(month=1;month<13;month++){
            total=0;
            if(year%4==0){
                MonthDay[2]++;
            }
            for(int m=1;m<=month-1;m++){
                total+=MonthDay[m];
            }
        }
    }
}

```



```

        total+=day;
    }
    System.out.print("        Year="+year);
    System.out.print("        Month =" +month);
    System.out.print("        Day="+day);
    System.out.print("        T0tal="+total);
}
}
}

```

```

}
9、 public class ArraySort{
    public static void main(String args[]){
        int array[]={20,10,50,40,30,70,60,80,90,100};
        int i,j,k,t;
        int l=array.length;
        for(i=0;i<l-1;i++)
        {
            k=i;
            for(j=i+1;j<l;j++)
            if(array[j]<array[k]) k=j;
            t=array[k];array[k]=array[i];array[i]=t;
        }
        for(i=0;i<l;i++)
        System.out.println("array["+i+"]="+array[i]);
    }
}

```

10、（略）

第五章 类和对象

一、选择题

1-5 DBACD	6-10 DCABC	11-15 DCBBB	16-20
ADABD			
21-25 AAABC	26-30 ADBDB	31-35 ACCCA	36-40
CAADD			
41-45 BDDBD	46-50 DCBCA	51-55 DBBCB	56-60 DCCDB
61-65 CBDDC	66-70 BDADC	71-73 DDB	

二、填空题

- 1、 new 类名 对象名 =new 构造方法名 ()
- 2、 MyClass obj = new MyClass()

- 3、封装 继承 多态
- 4、初始化
- 5、成员变量 成员方法
- 6、static
- 7、java.io java.awt java.applet
- 8、Object
- 9、public
- 10、private
- 11、类
- 12、class
- 13、成员变量
- 14、方法头 方法体
- 15、名字 类型 返回值
- 16、方法体
- 17、修饰符 关键字 类名 花括号
- 18、对象
- 19、编辑窗口
- 20、类
- 21、成员变量 方法
- 22、类
- 23、类 对象
- 24、实例化类
- 25、对象的声明 对象的实例化 对象的初始化
- 26、对象
- 27、初始化
- 28、构造方法
- 29、对象
- 30、构造方法
- 31、构造方法
- 32、类同名
- 33、包
- 34、.
- 35、public abstract final
- 36、[修饰符] [static][final][transient][volatile]< 变量类型 >< 变量名 >
- 37、创建 使用 清除
- 38、默认 protected public
- 39、Object java.awt.Panel java.applet
- 40、类变量
- 41、多重
- 42、Object

三、程序填空

1、 x=0

2、 What a pleasure!
I am Tom
How do you do?

3、 false

true

true

true

true

true

4、 1 , 2 , 3 , 4 , 5

2 , 4 , 6 , 8 , 10

5、 go through finally code in fun2().

catch MyException object. Message:An MyException object is thrown
in fun1().

6、 Myclass i j

7、 Student String

8、 max(1234,456,-987)

9、 count is101

time is0

10、 Before swap: circle1 = 1.0 circle2 = 2.0

Before swap: x = 1.0 y = 2.0

Before swap: x = 2.0 y = 1.0

After swap: circle1 = 1.0 circle2 = 2.0

11、 i + j is 23

k is 2

j is 0

12、 The default constructor of A is invoked

The default constructor of B is invoked

13、 class A

class B

class A

class B

class B

class B

class B

14、 i=0j=0

i=0 m=3

15、 NO

- 5
- 14
- 16、长度 =3
体积 =60
- 17、父类 SuperClass 成员变量、 10,20
子类 SubClass 成员变量、 10,20,null
- 18、m in main=a
a in main=3
a in test2 is : 98
return from test2 : 98.0
m + a=100.0
- 19、main() 中的 i=456
m1() 中的 i=789
类变量 i=123
加 6 后的类变量 i=129
- 20、good-bye
hello
- 21、22
- 22、Compile time error
- 23、super.oak=1;
oak=33;
Base.oak=22;

四、问答题

- 1、继承是面向对象程序设计中两个类之间的一种关系。 当一个类获取另一个类中所有非私有的数据和操作的定义作为它自己的部分或全部成分时，就称这两个类之间具有继承关系。被继承的类称为父类或超类，继承了父类或超类的所有数据和操作的类称为子类。面向对象程序设计中的继承，可使得程序结构更加清晰，降低了编码和维护的工作量。如果一个类只能有一个父类，则这种继承关系称为单重继承；如果一个类可以有多个父类，则这种继承关系称为多重继承。
- 2、Java 中使用 static 修饰的变量和方法称为类变量（或称静态变量）和类方法（或称静态方法），没用 static 修饰符的变量和方法称为实例变量和实例方法。类成员是指该成员属于这个类，由这个类所创建的所有对象共同拥有，该成员仅在类的存储单元中存在，这个类所创建的所有对象中仅存储一个指向该成员的 引用。任何一个该类的对象改变了类成员，其他对象同样会发生改变。对于类成员既可以使用对象访问，也可以使用类名直接进行访问， 并且在类方法中只能访问类 变 量，而不能访问实例变量。实例成员是指该成员由每一个对象个体独有，对象的存储空间中的确有一块空间用来存储该成员，不同对象的

实例成员不同，任何一个对象改变了实例成员，只会影响这个对象本身，其他对象中的实例成员不会发生改变。对于实例成员只能通过对象来访问，不能通过类名进行访问，在实例方法中既可以访问实例变量，也可以访问类变量。

3、(1)接口可以被多重 implements, 抽象类只能被单一 extends

(2)接口只有定义，抽象类可以有定义和实现

(3)接口的字段定义默认为 `:public static final`，抽象类字段默认是 "friendly"(本包可见) 当功能需要累积时用抽象类，不需要累积时用接口。

4、类给出了属于该类的全部对象的抽象定义，而对象则是符合这种定义的一个实体。可以把类与对象之间的关系看成是抽象与具体的关系。

在面向对象的程序设计中，对象被称作类的一个实例 (Instance)，而类是对象的模板 (Template)。类是多个实例的综合抽象，而实例又是类的个体实物。由于对象是类的实例，因此在定义对象之前应先定义类。在定义了类之后，才可以在类的基础上创建对象。

5、Java 的访问限定修饰符有 4 种，分别使 `public`、`protected`、`private` 和默认修饰符。用 `public` 修饰的类或成员拥有公共作用域，表明此类或类的成员可以被任何 Java 中的类所访问，是最广泛的作用范围。用 `protected` 修饰的变量或方法拥有受保护作用域，可以被同一个包中所有的类及其他包中该类的子类所访问。用 `private` 修饰的变量或方法拥有私有作用域，只能在此类中访问，在其他类中，包括该类的子类也是不允许访问的，`private` 是最保守的作用范围。没有任何修饰符拥有默认访问权限 (友好访问权限)，表明此类或类的成员可以被同一个包中的其它类访问。

6、`final` — 修饰符 (关键字) 如果一个类被声明为 `final`，意味着它不能再派生出新的子类，不能作为父类被继承。因此一个类不能既被声明为 `abstract` 的，又被声明为 `final` 的。将变量或方法声明为 `final`，可以保证它们在使用中不被改变。被声明为 `final` 的变量必须在声明时给定初值，而在以后的引用中只能读取，不可修改。被声明为 `final` 的方法也同样只能使用，不能重载

`finally` — 再异常处理时提供 `finally` 块来执行任何清除操作。如果抛出一个异常，那么相匹配的 `catch` 子句就会执行，然后控制就会进入 `finally` 块 (如果有的话)。

`finalize` — 方法名。Java 技术允许使用 `finalize()` 方法在垃圾收集器将对象从内存中清除出去之前做必要的清理工作。这个方法是由垃圾收集器在确定这个对象没有被引用时对这个对象调用的。它是在 `Object` 类中定义的，因此所有的类都继承了它。子类覆盖 `finalize()` 方法以整理系统资源或者执行其他清理工作。`finalize()` 方法是在垃圾收集器删除对象之前对这个对象调用的。

7、匿名的内部类是没有名字的内部类。不能 extends(继承) 其它类，

但一个内部类可以作为一个接口，由另一个内部类实现。

8、声明方法的存在而不去实现它的类被叫做抽象类（abstract class），它用于要创建一个体现某些基本行为的类，并为该类声明方法，但不能在该类中实现该类的情况。不能创建 abstract 类的实例。然而可以创建一个变量，其类型是一个抽象类，并让它指向具体子类的一个实例。不能有抽象构造函数或抽象静态方法。Abstract 类的子类为它们父类中的所有抽象方法提供实现，否则它们也是抽象类。取而代之，在子类中实现该方法。知道其行为的其它类可以在类中实现这些方法。

接口（interface）是抽象类的变体。在接口中，所有方法都是抽象的。多继承性可通过实现这样的接口而获得。接口中的所有方法都是抽象的，没有一个有程序体。接口只可以定义 static final 成员变量。接口的实现与子类相似，除了该实现类不能从接口定义中继承行为。当类实现特殊接口时，它定义（即将程序体给予）所有这种接口的方法。然后，它可以在实现了该接口的类的任何对象上调用接口的方法。由于有抽象类，它允许使用接口名作为引用变量的类型。通常的动态联编将生效。引用可以转换到接口类型或从接口类型转换， instanceof 运算符可以用来决定某对象的类是否实现了接口。

五、判断题

1-10 F F F F T T F F F F
F F F T

11-20 F F F T F F F F F T

21-25 T

六、编程题

1、public class Student {
private String name;
private Score score;

```
public void setData(String name,Score score){  
    this.name=name;  
    this.score=score;  
}
```

```
public void getResult(){  
    System.out.println("Courese:"+score.getCourse());  
    System.out.println("Grade:"+score.getGrade());  
}
```

```
public String getName() {  
    return name;  
}
```

```
public Score getScore() {  
    return score;  
}
```

```

    }
    public static void main(String[] args) {
        Student student = new Student();
        Score score = new Score();
        score.setCourse("java");
        score.setGrade(100);
        student.setData("mike", score);
        student.getResult();
    }
}
public class Score {
    private String course;
    private int grade;

    public String getCourse() {
        return course;
    }
    public void setCourse(String course) {
        this.course = course;
    }
    public int getGrade() {
        return grade;
    }
    public void setGrade(int grade) {
        this.grade = grade;
    }
}
2、 public class StudentBeanBean {
    private String ST_NUMBER;
    private String ST_NAME;
    private String ST_RESULT;
    public StudentBeanBean(){
    }
    public String getST_NUMBER(){
        return this.ST_NUMBER;
    }
    public String getST_NAME(){
        return this.ST_NAME;
    }
    public String getST_RESULT(){

```

```

    return this.ST_RESULT;
}
public void setST_NUMBER(String ST_NUMBER){
    this.ST_NUMBER=ST_NUMBER;
}
public void setST_NAME(String ST_NAME){
    this.ST_NAME=ST_NAME;
}
public void setST_RESULT(String ST_RESULT){
    this.ST_RESULT=ST_RESULT;
}
}

```

```

public class ParentClass {
    private String name; // 姓名
    private int age; // 年龄
    private String sex; // 性别

```

```

    public ParentClass() {
        super();
    }

```

```

    public ParentClass(String name, int age, String sex) {
        super();
        this.name = name;
        this.age = age;
        this.sex = sex;
    }

```

```

    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public int getAge() {
        return age;
    }
    public void setAge(int age) {

```



```

this.age = age;
}
public String getSex() {
return sex;
}
public void setSex(String sex) {
this.sex = sex;
}

public void f(){
System.out.println(" 我是 ParentClass");
}
}

public class SubClass extends ParentClass {
private String grade; // 年级

public SubClass() {
super();
}

public SubClass(String name,int age,String sex,String grade) {
super(name,age,sex); //调用父类带参构造
this.grade = grade;
}

public String getGrade() {
return grade;
}

public void setGrade(String grade) {
this.grade = grade;
}

public void myMethod(){
System.out.println(" 这是 SubClass 中的 myMethod 方法 ");
}

public void newMethod(){
System.out.println(" 这是 SubClass 中的 newMethod 方法 ");
}

```

```
}  
}
```

```
public class Test {  
    public static void main(String[] args) {  
        ParentClass pClass = new ParentClass("sky",21,"男");  
        ParentClass subClass = new SubClass("天空 ",22,"男","1 年级");  
        subClass.f();  
  
        System.out.println(" 姓名 :"+pClass.getName()+"\t"+" 年  
        龄:"+pClass.getAge());  
        System.out.println(" 姓名 :"+((SubClass)subClass).getName()+"\t"+" 年  
        龄:"+((SubClass)subClass).getAge()+"\t"+" 年  
        级:"+((SubClass)subClass).getGrade());  
    }  
}
```

```
3、 package com.sorce.bankSys;  
import java.util.Scanner;  
import sun.security.action.GetBooleanAction;  
public class BankSys {  
    private int    uld;    //这写好像用不上，你自己看着加吧  
    private String uName;  
    private String address;  
    private double minMon=0.50;  
    private double money=1000.50;  
  
    public static void main(String[] args) {  
        String answer;  
        // TODO Auto-generated method stub  
        System.out.println(" 欢迎进入存取款系统！    ");  
        System.out.println("-----");  
        do{  
            System.out.println("1. 取款  \t2.查询余额  \t3.存款 ");  
            Scanner sc=new Scanner(System.in);  
            BankSys bank=new BankSys();  
            int num=sc.nextInt();  
            switch (num) {  
            case 1:  
                bank.get();
```

```

        break;
    case 2:
        bank.search();
        break;
    case 3:
        bank.deposit();
        break;
    default:
        break;
}

    System.out.println(" 是否继续？ ( y/n ) ");
    answer=sc.next();
}while(answer.equals("y"));
System.out.println(" 程序结束 ");
}

public void deposit(){
    Scanner sc=new Scanner(System.in);
    System.out.println(" 请输入存款金额  :");
    double  input=sc.nextInt();
    System.out.println(" 您的帐户原有余额为  :"+money+" 元");
    System.out.println(" 已存入 :"+input+" 元");
    money+=input;
    System.out.println(" 您的帐户最终存款余额为  :"+money+" 元");
}

public void search(){
    System.out.println(" 您的帐户余额为、  "+money+" 元");
}

public void get(){
    Scanner sc=new Scanner(System.in);
    System.out.println(" 请输入取款金额  :");
    double input=sc.nextInt();
    if (money<input){
        System.out.println(" 您的帐户余额不足！  ");
    }else{
        System.out.println(" 您本次取出 :"+input+" 元");
        money-=input;
        System.out.println(" 剩余金额为 :"+money+" 元");
    }
}

```

```
}  
}
```

```
}
```

4、(略)

5、class Phone{

private String phonenumber;

public void setPhonenumber(String phonenumber){

this.phonenumber=phonenumber;

}

public String getPhonenumber(){

return phonenumber;

}

public void recCall(){

System.out.println(" 接到一个电话 ");

}

public void telCall(){

System.out.println(" 拨出一个电话 ");

}

}

class Fixedphone extends Phone{

private String phonenumber;// 号码是私有，设置为 private，不可继承

public void recCall(){

System.out.println(" 以 "+this.phonenumber+" 呼出了一个电话 "); // 重载了父类的 recCall

}

}

class Cordlessphone extends Fixedphone{

private String phonenumber;

public void info(){

System.out.println(" 这是无绳电话的信息 ");

}

}

interface Moveable{

public void moveinfo();

}

class Mobilephone extends Phone implements Moveable{

private String phonenumber;

public void moveinfo(){

System.out.println(" 我实现了可移动性 ");

```

}
}
public class PhoneTest{
public static void main(String a[]){
Phone[] p=new Phone[5];
Phone p1=new Phone();
p1.setPhonenumber("123456789");
p[0]=p1;
Phone p2=new Phone();
p2.setPhonenumber("987654321");
p[1]=p2;
Mobilephone mp=new Mobilephone();
mp.setPhonenumber("11111");
p[2]=mp;
Fixedphone fp=new Fixedphone();
fp.setPhonenumber("22222");
p[3]=fp;
Cordlessphone cp=new Cordlessphone();
cp.setPhonenumber("33333");
p[4]=cp;

for(int i=0;i<p.length;i++){
System.out.println(p[i].getPhonenumber());
}
p[4]=p[1];
System.out.println(p[4].getPhonenumber());

}
}

```

6、

```

public class Animal{
    private String name;
    private double weight;
    private int length;
    privateString sex;
    public void run(){
        System.out.println("the animal is running!"); 来源、考试大
    }
    public void jump(){
        System.out.println("the animal is jumping!");
    }
}

```

```

public void walk(){
    System.out.println("the animal is walking!");
}
}

```

7、package mypackage.math;

```

class Rectangle{ 来源、 www.examda.com

```

```

    private int length;

```

```

    private int width;

```

```

    public Rectangle(int length,int width){

```

```

        this.length=length;

```

```

        this.width=width;

```

```

    }

```

```

    public void set(int length,int width){

```

```

        this.length=length;

```

```

        this.width=width;

```

```

    }

```

```

    public void get(){

```

```

        System.out.println(" 此长方形的长为    "+length+" 宽为 "+width); 采集

```

者退散

```

    }

```

```

    public int getLength(){

```

```

        return length;

```

```

    }

```

```

    public int getWidth(){

```

```

        return width;

```

```

    }

```

```

    public int calculatePerimeter(){ 考试大论坛

```

```

        return 2*(length+width);

```

```

    }

```

```

    public int calculateArea(){

```

```

        return (length*width);

```

```

    }

```

```

    public void print(){

```

```

        System.out.println(" 此长方形的周长为    "+calculatePerimeter()+" 面

```

积为 "+calculateArea());

```

    }

```

```

}

```

```

class Test{

```

```

    public static void main(String [ ] args){

```

```

        Rectangle myrectangle=new Rectangle(15,10);

```

```

    myrectangle.get();
    myrectangle.print();
    myrectangle.set(20,15);
    System.out.println(" 长方形的长是  "+myrectangle.getLength()+"  宽是
"+ myrectangle .getWidth());
    myrectangle.print();
}
}

```

```

8、 public class People {
    private int number;
    private String name;
    private String sex;
    private int age;
    public People(){
    public People(int number,String name,String sex,int age){
        this.name=name;
        this.age=age;
        this.number=number;
        this.sex=sex;

    }
    public int getNumber() {
        return number;
    }
    public void setNumber(int number) {
        this.number = number;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public String getSex() {
        return sex;
    }
    public void setSex(String sex) {
        this.sex = sex;
    }
    public int getAge() {

```

```

        return age;
    }
    public void setAge(int age) {
        this.age = age;
    }
    public static void main(String []ad){
        People pp= new People(001,"张三 ","男 ",23);
        System.out.println( "编号 : "+pp.getNumber());
        System.out.println( "名字 : "+pp.getName());
        System.out.println( "年龄 : "+pp.getAge());
        System.out.println( "性别 : "+pp.getSex());

    }
}

```

}

9、 class Point

```

    { int x,y;
    Point(int a,int b)
    { x=a;
    y=b;
    }
    }
    public class A
    { public static void main(String [ ] args)
    { Point p1,p2;
    p1=new Point(10,10);
    p2=new Point(23,35);
    }
    }

```

10、 class StaticTest{

```

    int i=5;
    static int j=5;
    void print(){
        System.out.println("i="+i);
        System.out.println("j="+j);
    }
    static void classprint(){
        //System.out.println("i="+i);
        System.out.println("j="+j);
    }
}

```



```

}
}
class Test{
public static void main(String [] args){
//System.out.println(StaticTest.i);
System.out.println(StaticTest.j);
//StaticTest.print();
StaticTest.classprint();
StaticTest st=new StaticTest();
System.out.println(st.j);
System.out.println(st.j);
st.print();
st.classprint();
}
}

```

```

11、 package mypackage.employee;
class MyDate{
private int year;
private int month;
private int day;
public MyDate(int year,int month,int day){
this.year=year;
this.month=month;
this.day=day;
}
public int getYear(){
return year;
}
}
public class Employee{
private String name;
private double salary;
private MyDate hireDay;
public Employee(String name,double salary,MyDate day){
this.name=name;
this.salary=salary;
hireDay=day;
}
public void print(){
System.out.println(name+ ""+salary+""+hireYear());
}
}

```

```

    }
    public int hireYear(){
        return hireDay.getYear();
    }
}
class TestEmployee{
    public static void main(String [] args){
        Employee a=new Employee(" 李明 ",20000,new MyDate(2000,10,1));
        Employee b=new Employee(" 王强 ",24000,new MyDate(1995,8,12));
        a.print();
        b.print();
    }
}

```

12、 package s1java.sg.chap6;

* 学生类 P135--136

```

public class Student {
    String name ;           //姓名
    int age ;               //年龄
    String hobby ;          //兴趣
    String classNo ;        //班级编号

    public String toString() {
        return name + "\n 年龄、 " + age + "\n 爱好、 " + hobby + "\n 就
读于 " + classNo ;
    }
}

```

package s1java.sg.chap6;

```

public class Teacher {
    String name ;    //姓名
    String course ; //教授课程
    String dep ;     //专业方向
    int teaYear ;    //教龄
    public String toString() {
        return name + "\n 教授课程、 " + course + "\n 主攻方向、
" + dep + "\n 教龄、 " + teaYear ;
    }
}

```

package s1java.sg.chap6;

测试类 P136--137

```

/public class Test {

```

```

    public static void main(String[] args) {
        /* 创建一个学生对象 */
        Student stu = new Student() ;
        stu.name = "张三 " ;
        stu.age = 20 ;
        stu.hobby = " 网球 " ;
        stu.classNo = "ACCP0791" ;
        System.out.println(stu) ;
        /* 创建一个教员对象 */
        Teacher tea = new Teacher() ;
        tea.name = "陈浩然 " ;
        tea.course = "使用 Java 语言理解程序逻辑 " ;
        tea.dep = "Java/J2EE" ;
        tea.teaYear = 8 ;
        System.out.println("\n" + tea) ;
    }
}

```

13、 class Complex {

```

    private double realPart;
    private double imaginaryPart;

```

```

    public Complex(double real, double imaginary) {
        this.realPart = real;
        this.imaginaryPart = imaginary;
    }

```

```

    public Complex add(Complex other) {
        return new Complex(this.realPart + other.realPart,
            this.imaginaryPart + other.imaginaryPart);
    }

```

```

    public Complex subtract(Complex other) {
        return new Complex(this.realPart - other.realPart,
            this.imaginaryPart - other.imaginaryPart);
    }

```

```

    public Complex multiply(Complex other) {
        return new Complex(this.realPart * other.realPart -
            this.imaginaryPart * other.imaginaryPart,
            this.realPart * other.imaginaryPart +

```

```

        this.imaginaryPart * other.realPart);
    }

```

```

public Complex divide(Complex other) {
    if (other.realPart == 0 && other.imaginaryPart == 0) {
        throw new ArithmeticException("Divided By 0");
    }
    if (this.realPart == 0 && this.imaginaryPart == 0) {
        return this;
    }
    Complex numerator = this.multiply(new Complex(other.realPart,
        -other.imaginaryPart));
    double denominator = other.realPart * other.realPart +
        other.imaginaryPart * other.imaginaryPart;
    return new Complex(numerator.realPart / denominator,
        numerator.imaginaryPart / denominator);
}

```

```

public String toString() {
    StringBuilder sb = new StringBuilder();
    if (this.realPart != 0) {
        sb.append(this.realPart);
    }
    if (this.imaginaryPart > 0) {
        sb.append(" + ");
        sb.append(this.imaginaryPart);
        sb.append("i");
    } else if (this.imaginaryPart < 0) {
        sb.append(" - ");
        sb.append(Math.abs(this.imaginaryPart));
        sb.append("i");
    }
    return sb.toString();
}

```

```

14、 public class Test {
    public static void main(String[] args) {
        Patient april=new Patient();
        april.setName("ZhangLi");
        april.setSex('f');
    }
}

```

```

    april.setAge(33);
    april.setWeight(154.72f);
    april.setAllergies(true);
    System.out.println("Name: "+april.getName());
    System.out.println("Sex: "+april.getSex());
    System.out.println("Age: "+april.getAge());
    System.out.println("Weight: "+april.getWeight());
    System.out.println("Allergies: "+april.getAllergies());
    Complex c1 = new Complex(1, 2);
    Complex c2 = new Complex(3, 4);
    System.out.println(c1.add(c2));
    System.out.println(c1.subtract(c2));
    System.out.println(c1.multiply(c2));
    System.out.println(c1.divide(c2));
}
}
class Patient {
    private String name;
    private char sex;
    private int age;
    private float weight;
    private boolean allergies;
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public char getSex() {
        return sex;
    }
    public void setSex(char sex) {
        this.sex = sex == 'M' ? sex : 'F';
    }
    public int getAge() {
        return age;
    }
    public void setAge(int age) {
        this.age = Math.max(1, age);
    }
}

```

```

public float getWeight() {
    return weight;
}
public void setWeight(float weight) {
    this.weight = Math.max(0.0f, weight);
}
public boolean getAllergies() {
    return allergies;
}
public void setAllergies(boolean allergies) {
    this.allergies = allergies;
}
}

```

15、 Student 类：

```

public class Student {
    String name =" 小强 ";
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
}

```

Statics 类：

```

public class Statics {
    String str;
    public Statics() {
        this.str = new Student().getName();
    }
}

```

16、 class TestString{

```

    String s[] ;
    public TestString(int length){
        s=new String [lenth];
    }

    private void setter(){

    }
}

```

```

private void getter(){

}

private void dispS(){
    if(s==null){
        System.out.println("s is null ! ");
    }else{
        for(int i=0 ;i<s.length;++i)
            System.out.println(this.s[i]);
        }
    }
}

```

```

17、 import java.io.*;
public class Point
{
    int x,y;

    public Point(int x,int y)
    {
        this.x = x;
        this.y = y;
    }

    public Point getPoint()
    {
        Point tempPoint = new Point(x,y);
        return tempPoint;
    }

    public void setPoint(Point point)
    {
        this.x = point.x;
        this.y = point.y;
    }

    public static void main(String args[])
    {
        Point Point1 = new Point(3,4);
    }
}

```

```
System.out.println("Point1:"+"("+Point1.x+", "+Point1.y+")");
```

```
Point Point2 = Point1.getPoint();
```

```
System.out.println("Point2:"+"("+Point2.x+", "+Point2.y+")");
```

```
Point Point3 = new Point(5,6);
```

```
Point1.setPoint(Point3);
```

```
System.out.println("Point1:"+"("+Point1.x+", "+Point1.y+")");
```

```
}
```

```
}
```

```
18、 public class xt4_01
```

```
{
```

```
public static void main( String args[] )
```

```
{
```

```
70 for(int x=2;x<=200;x++)
```

```
ss(x) ;
```

```
System.out.println();
```

```
}
```

```
public static void ss(int x)
```

```
{
```

```
int i,j,flag;
```

```
flag=1;
```

```
i=(int)(Math.sqrt(x));
```

```
for(j=2;j<=i;j++)
```

```
if((x%j)==0)
```

```
{flag=0; break;}
```

```
if(flag==1)
```

```
System.out.print(x+"\t");
```

```
}
```

```
}
```

```
19、 public class xt4_02{
```

```
public static void main (String args[ ]){
```

```
for (int i=0;i<=15;i++) {
```

```
long n=i;
```

```
System.out.print(fib(n)+"    ");
```

```
}
```

```
System.out.println();
```

```
}
```

```
static long fib (long n){
```

```
long sum;
```



```

if (n==0||n==1){
sum=n;      }
else{
sum=fib (n-1)+fib(n-2);
}
return  sum;
}
}
20、 public class xt4_03
{
static int n;
public static void main (String[] args)
{
if(args.length >= 3)
{
int n;
float f;
double d;
n=Integer.parseInt(args[0]);
f=Float.valueOf(args[1]).floatValue( );
d=Double.valueOf(args[2]).doubleValue( );
System.out.println(n+" 的平方根为 : "+mySqrt(n));
System.out.println(f+" 的平方根为 : "+mySqrt(f));
System.out.println(d+" 的平方根为 : "+mySqrt(d));
}
else {      System.out.println(" 缺命令行参数  !");      System.exit(-1);      }
}
static int mySqrt(int x)
{ return (int)Math.sqrt(x); }
static float  mySqrt(float x)
{ return (float)Math.sqrt(x); }
static double  mySqrt(double x)
{ return Math.sqrt(x); }
}
21、 public class xt4_04
{
public static void main(String args[])
{
final int NUMBER = 100;
int count = 10;

```

```
int randomNum = 0;
int probability[] = new int[6];
for(int i=0;i<6; i++)
probability[i] = 0;
    for(int i=0; i<NUMBER;i++)
    {
randomNum = (int)(Math.random()*6) + 1;
probability[randomNum-1]++;
System.out.print(randomNum + "\t");
if(i%count == 9)
System.out.println();
}
for(int i=0;i<6;i++)
System.out.println("\n" + (i+1) + ":\t" + probability[i]);
}
}
```

第六章 面向对象程序设计

一、 选择题

1—5	AACAA	6—10	BAACB	11—15	BCACD
16—20	AADCD				
21—25	ADBCB	26—30	CDBAB	31—35	CCBCA
36—40	BDBCA				
41—45	AADBA	46—50	ACDCA	51—55	CACCD
56—60	BCBAA				
61—65	ABABB	66—70	ADAAC	71—75	DCBBA
76—80	CBDDDB				
81—85	ACCAB	86—90	CBCCC	91—95	CCABB
96—100	ABCBC				
101—105	DDBAD	106—110	DAADC	111—115	ACAAC
116—120	ACDBD				
121—125	BCDDD	126—130	AAABA	131—135	BACBA
136—140	DBCBA				
141—145	BDDDD	146—150	DADCB	151—155	BCBBC
156—160	CDBCBC				
161—165	DCBCB	166—170	BACCB	171—175	DADCB
176—180	BCDBA				
181—185	CADAA	186—190	BABAD	191—195	CADBD

二、 填空题

- 1、 参数的个数
- 、 参数的类型
- 、 参数的顺序
- 2、 单

- 3、方法头 , 方法体
- 4、封装 、 继承 多态
- 5、 new
- 6、protected , finally public
- 7、 private
- 8、 new
- 9、 java.lang.Object
- 10、 类
- 11、 抽象方法
- 12、 super
- 13、 import
- 14、 abstract final
- 15、 单 多
- 16、 default
- 17、 属性 方法
- 18、 多个 一 多
- 19、 重载 覆盖
- 20、 class implements
- 21、 implements
- 22、 一些抽象方法和常量
- 23、 使用包名、类名做前缀 使用 import 语句
- 24、 捕获异常 、 声明异常 、 自定义异常
- 25、 public 、 static final 。
- 26、 对象
- 27、 object
- 28、 一个程序中有多多个同名的不同方法
- 29、 隐藏
- 30、 变量
- 31、 参数列表
- 32、 属性 方法
- 33、 final
- 34、 final 抽象
- 35、 methodA()
- 36、 抽象
- 37、 多
- 38、 抽象
- 39、 extends
- 40、 final
- 41、 new
- 42、 参数类型个数

- 43、 MyClass obj=new obj()
- 44、 常量 抽象方法
- 45、 属性 、 方法
- 46、 —
- 47、 该类本身 、 该类的所有子类
- 48、 private
- 49、 abstract
- 50、 简单数据类型
- 51、 除空行 注释行
- 52、 第一个
- 53、 抽象 abstract
- 54、 public static final public abstract
- 55、 抽象方法
- 56、 static void method()
- 57、 interface final abstract
- 58、 public
- 59、 private
- 60、 public
- 61、 采用不同的形式参数列表
- 62、 stop()
- 63、 抽象类 接口
- 64、 final 修饰符
- 65、 —
- 66、 super this
- 67、 队列
- 68、 线性表
- 69、 双向的链式
- 70、 添加,删除
- LinkedList 变量名 =new LinkedList();
- LinkedList 变量名 =new LinkedList(Collection c);

三、 判断题

- | | |
|--|---|
| 1—10 F T T F F T F F F T
F
21—30 F T T F F T F T F F
41—50 T T T T T T T T T T
61—70 T T F F T F F F T T
81—90 F T F F F F T F T T
101—110 T T T T F T T F T F | 11---20 T T T T T T T F F T

31—40 F F T F F F T T T F
51—60 F T F F T F F F F T
71—80 F T F T F F T F T T
91—100 F T T F T T T T F T
111—115 F F T F T |
|--|---|

四、 编程题

- 1、 设计一个类 Triangle ，从键盘里面输入三个数字作为三角形的三个

边。设计相应的方法判断三角形是否成立；判断三角形的类型（等边三角形、等腰三角形和任意三角形）；与计算三角形的面积和周长。面积的计算公式为、 $s=(a+b+c)/2$; $area=Math.sqrt(s*(s-a)*(s-b)*(s-c))$; 设计一个主类，调用 Tirangle 类中相应的方法，并打印出相应的结果。

```
package xsc;
```

```
public class Triangle {
    private double a = 0;
    private double b = 0;
    private double c = 0;
    private String message = "";

    public Triangle() {
    }
    public Triangle(double a, double b, double c) {
        this.a = a;
        this.b = b;
        this.c = c;
    }
    public String getMessage() {
        return message;
    }
    public void setMessage(String message) {
        this.message = message;
    }
    public double getA() {
        return a;
    }
    public void setA(double a) {
        this.a = a;
    }
    public double getB() {
        return b;
    }
    public void setB(double b) {
        this.b = b;
    }
    public double getC() {
        return c;
    }
}
```

```

public void setC(double c) {
    this.c = c;
}
public double area() { // 计算面积的方法
    double s = (a + b + c) / 2;
    return Math.round(Math.sqrt(s * (s - a) * (s - b) * (s - c)));
}
public double perimeter() { // 计算 周长的方法
    return a + b + c;
}
public boolean isTriangle() { // 判断三角形是否成立的方法
    boolean temp = false;
    if (a + b > c && a + c > b && b + c > a)
        temp = true;
    return temp;
}
public String toString() { // 判断三角形形状的方法
    if (a == b && b == c)
        message = "此三角形是等边三角形 ";
    if (a != b && b != c)
        message = "此三角形是任意三角形 ";
    if ((a == b && b != c) || (a == c && b != c) || (c == b && b != a))
        message = "此三角形是等腰三角形 ";
    return message;
}
}
package xsc;
import javax.swing.*;
public class GoTriangle {
    public static void main(String[] args) {
        Triangle triangle = null;
        String a = JOptionPane.showInputDialog(" 请输入三角形的任意一边", "");
        String b = JOptionPane.showInputDialog(" 请输入三角形的任意一边", "");
        String c = JOptionPane.showInputDialog(" 请输入三角形的任意一边", "");
        try {
            int ba = Integer.parseInt(a);
            int bb = Integer.parseInt(b);

```

```

        int bc = Integer.parseInt(c);
        triangle = new Triangle(ba, bb, bc);
    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, " 请输入三角形边的数
据格式有误 ");
    }
    if (triangle.isTriangle()) {
        JOptionPane.showMessageDialog(null, " 你输入的三边可以构
成一个三角形 ");
        triangle.toString();
        JOptionPane.showMessageDialog(null, " 你的三角形的类型为  "
            + triangle.getMessage());
        JOptionPane.showMessageDialog(null, " 你的三角形的面积为  "
+ triangle.area());
        JOptionPane.showMessageDialog(null, " 你的三角形的周长为  "
            + triangle.perimeter());
    } else {
        JOptionPane.showMessageDialog(null, " 你输入的三边不可以
构成一个三角形 ");
    }
}
}

```

2、通过一个点类 (Point) , 具有自己的坐标位置 (x,y), 和显示本类信息的 show() 方法 ;

一个圆类 (Circle) 由点类派生 , 增加一个新的属性半径 (rad), 一个计算圆面积的 area() 方法 , 和一个同样显示本类信息的 show() 方法 ; 一个圆柱体类 (Cylinder) 由圆类派生 , 增加一个属性高度 (height), 一个计算圆柱形体积的方法 vol(), 一个计算圆柱形表面积的 area() 方法 , 和一个同样显示本类信息的 show() 方法。建立一个主类 , 调用相关类的方法 , 并打印出相应的信息。

```

class Point {
    int x;
    int y;
    public Point() {
        x = 0;
        y = 0;
    }
    public Point(int x, int y) {
        this.x = x;
        this.y = y;
    }
}

```

```

    }
    public void show() {
        System.out.println( "我是点，我的位置在：纵坐标为      " + x + "横
坐标为      " + y);
    }
}
class Circle extends Point {
    int rad;
    public Circle() {
        super();
        rad = 0;
    }
    public Circle( int x, int y, int rad) {
        super(x, y);
        this.rad = rad;
    }
    public double area() {
        return Math.PI * rad * rad;
    }
    public void show() {
        System.out.println( "我是圆，我的位置在：纵坐标为      " + x + "横
坐标为      " + y);
    }
}
class Cylinder extends Circle {
    int height;
    public Cylinder() {
        super();
        height = 0;
    }
    public Cylinder( int x, int y, int rad, int height) {
        super(x, y, rad);
        this.height = height;
    }
    public double vol() {
        return Math.PI * rad * rad * height;
    }
    public double area() {
        return 2 * Math.PI * rad * rad + 2 * Math.PI * rad * height;
    }
}

```



```

        public void show() {
            System.out.println( "我是圆柱，我的位置在：纵坐标为      " + x + "
横坐标为      " + y);
        }
    }
}

```

3、定义一个名为 Rectangle 表示长方形的类，它包含 double 类型的长 length 和宽 width 两个私有成员变量，计算长方形面积的成员方法 area ()，分别对数据成员添加 setter 和 getter 方法，在主类 RectangleDemo 的 main() 中，通过 setter 和 getter 方法来访问 Rectance 类对象的数据成员

```

public class Rectangle {
    private double length;
    private double width;
    public double getLength() {
        return length;
    }
    public void setLength(double length) {
        this.length = length;
    }
    public double getWidth() {
        return width;
    }
    public void setWidth(double width) {
        this.width = width;
    }
    public Rectangle (){}
    public Rectangle (double length){
        this.length=length;
    }
    public Rectangle (double length,double width){
        this.length=length;
        this.width=width;
    }
    public double area(){
        return length*width;
    }
}

```

```

public class RectangleDemo {
    public static void main(String[] args) {
        Rectangle r=new Rectangle ();
        r.setLength(12.0);
        r.setWidth(30.0);
        System.out.println(" 长方形的面积为、    "+r.area());
    }
}

```

4、定义一个名为 hhe 表示长方体的类，它包含 int 类型的长 length 和宽 width 和高 high，计算长方体的体积的成员方法 tiJi（），分别对数据成员添加 setter 和 getter 方法，在主类 qiuZhi 的 main() 中，通过 setter 和 getter 方法来访问 hhe 类对象的数据成员

```

public class hhe {
    int length;
    int width;
    int high;

    public int getLength() {
        return length;
    }
    public void setLength(int length) {
        this.length = length;
    }
    public int getWidth() {
        return width;
    }
    public void setWidth(int width) {
        this.width = width;
    }
    public int getHigh() {
        return high;
    }
    public void setHigh(int high) {
        this.high = high;
    }
    public hhe(){}
    public hhe (int length,int width,int high){
        this.length=length;
    }
}

```

```

        this.high=high;
        this.width=width;
    }
    public int tiJi(){
        return length*high*width;
    }
}
import javax.swing.JOptionPane;

public class qiuZhi {
    public static void main(String[] args) {
        JOptionPane.showMessageDialog(null," 创建一个求长方体体积的
对象！ ");
        hhe as=new hhe();
        as.setLength(5);
        System.out.println(" 您输入的长是、    "+as.getLength());
        as.setWidth(12);
        System.out.println(" 您输入的宽是、    "+as.getWidth());
        as.setHigh(10);
        System.out.println(" 您输入的高是、    "+as.getHigh());
        System.out.println(" 长方体的体积为、    "+as.tiJi());

    }
}

```

5、设计一个 people 类，它包含 name,sex,.age,high 属性，有相应的构造方法，分别对数据成员添加 setter 和 getter 方法，在主类 xianShi 的 main() 中，通过 setter 和 getter 方法来访问 people 类对象的数据成员

```

public class people {
    String name="";
    String sex="";
    int age;
    int high;
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public String getSex() {

```

```

        return sex;
    }
    public void setSex(String sex) {
        this.sex = sex;
    }
    public int getAge() {
        return age;
    }
    public void setAge(int age) {
        this.age = age;
    }
    public int getHigh() {
        return high;
    }
    public void setHigh(int high) {
        this.high = high;
    }
    public people (){}
    public people (String name,String sex,int age,int high){
        this.name=name;
        this.age=age;
        this.sex=sex;
        this.high=high;
    }
    public void shuXing(){
        System.out.println(" 所以我们都叫他、    ");

        System.out.println(" 瘦猴子！哈哈哈  ");
        System.out.println(" 头发很乱，就像犀利哥一样！    ");
    }
}

public class xianShi {
    public static void main(String[] args) {
        people ss = new people ();
        ss.setName("小丑 ");
        ss.setSex("不男不女，传说中的人妖    !");
        ss.setAge(150);
        ss.setHigh(170);
    }
}

```

```

        System.out.println(" 此人的名字、    " + ss.getName() + '\n' + " 性别、
" + ss.getSex()
        + '\n' + " 年龄、    " + ss.getAge() + '\n' + " 岁 " + " 身高、    "
        + ss.getHigh() + "cm");
        ss.shuXing();
    }
}

```

6、编写一个方法，接收一个整数作为参数，此方法会找出其最大值、最小值和平均值。

```

public class e {
    int a;
    int b;
    int c;
    public e() {}
    public e(int a) {
        this.a = a;
    }
    public e(int a, int b) {
        this.a = a;
        this.b = b;}
    public e(int a, int b, int c) {
        this.a = a;
        this.b = b;
        this.c = c;}
    public int max() {

        int x = a > b ? a : b;
        int w = c > x ? c : x;
        System.out.println(" 最大的一个数是、    " + w);
        return w;
    }
    public int getA() {return a;}
    public void setA(int a) {    this.a = a;}
    public int getB() {    return b;}
    public void setB(int b) {    this.b = b;}
    public int getC() {    return c;}
    public void setC(int c) {    this.c = c;}
    public int min() {
        int x = a < b ? a : b;

```

```

        int w = c < x ? c : x;
        System.out.println(" 最小的一个数是、    " + w);
        return w;}
public int Avg() {
    int x = (a + b + c) / 3;
    System.out.println(" 平均数是、    " + x);
    return x;
}
public static void main(String[] saf) {
    e sd = new e();
    sd.setA(4);
    sd.setB(5);
    sd.setC(6);
    sd.max();
    sd.min();
    sd.Avg();
}
}

```

7、编写一个 Person，该类包括属性耳朵、 眼睛、 鼻子、 嘴巴、 手和腿。包括的方法是、可以听见，可以闻见，可以咬，可以拿东西，可以走路。

```

public class f {
    String erDuo;
    String yanJing;
    String biZi;
    String zuiBa;
    String shou;
    String tui;
    public f() {
    }
    public f(String erDuo, String yanJing, String biZi, String zuiBa,
        String shou, String tui) {
        this.erDuo = erDuo;
        this.yanJing = yanJing;
        this.biZi = biZi;
        this.zuiBa = zuiBa;
        this.shou = shou;
        this.tui = tui;
    }
    public String getErDuo() {    return "可以听见！！";}
}

```

```

public void setErDuo(String erDuo) {    this.erDuo = erDuo;}
public String getYanJing() {    return "可以看见 ";}
public void setYanJing(String yanJing) {    this.yanJing = yanJing;}
public String getBiZi() {    return biZi;}
public void setBiZi(String biZi) {    this.biZi = biZi;}
public String getZuiBa() {    return "可以说话 ";}
public void setZuiBa(String zuiBa) {this.zuiBa = zuiBa;}
public String getShou() {    return "可以拿东西 ";}
public void setShou(String shou) {    this.shou = shou;}
public String getTui() {    return "可以走路！！";}
public void setTui(String tui) {
    this.tui = tui;
}
public static void main(String[] gds) {
    f sd = new f();
    sd.setBiZi("可以闻到");
    System.out.println(sd.getBiZi());
    System.out.println(sd.getErDuo());
    System.out.println(sd.getShou());
    System.out.println(sd.getTui());
    System.out.println(sd.getYanJing());
    System.out.println(sd.getZuiBa());
}
}

```

8、定义一个 changFangTi的父类，里面有 chang，kuan两个属性。和相应的构造方法和 get、set方法。定义个 changFangTi2的子类继承 changFangti，里面有一个 gao属性，有相信的 get和set方法。和求体积的tiJi（）方法。在 main（）方法中调用父类的方法。并求长方体的体积。

```

public class changFangTi {
    public int chang;
    public int kuan;
    public int getChang() {
        return chang;
    }
    public void setChang(int chang) {
        this.chang = chang;
    }
    public int getKuan() {
        return kuan;
    }
}

```

```

    }
    public void setKuan(int kuan) {
        this.kuan = kuan;
    }
}
public class changFangTi2 extends changFangTi {
    public int gao;
    public changFangTi2() {
    }
    public changFangTi2(int gao) {
        this.gao = gao;
    }
    public int getGao() {
        return gao;
    }
    public void setGao(int gao) {
        this.gao = gao;
    }
    public int jiShu() {
        return chang * kuan * gao;
    }
    public static void main(String[] ag) {
        changFangTi2 s = new changFangTi2();
        s.setChang(12);
        System.out.println(" 长方体的长是、    " + s.getChang());
        s.setKuan(5);
        System.out.println(" 长方体的宽是、    " + s.getKuan());
        s.setGao(6);
        System.out.println(" 长方体的高是、    " + s.getGao());
        System.out.println(" 长方体的体积为、    " + s.jiShu());
    }
}

```

9、定义一个 bus类，里面有 speed和number属性。以及构造方法和 get、set方法。定义一个 bus2的子类。并继承父类的方法。计算 bus的速度和人数。

```

public class bus {
    int speed;
    int number;
    public bus(){}
    public bus(int speed){

```



```

        this.speed=speed;
    }
    public bus(int speed,int number){
        this.speed=speed;
        this.number=number;
    }
    public int getSpeed() {
        return speed;
    }
    public void setSpeed(int speed) {
        this.speed = speed;
    }
    public int getNumber() {
        return number;
    }
    public void setNumber(int number) {
        this.number = number;
    }
    public void addA(int s) {
        if (s > 120) {
            speed = 120;
        } else {
            speed = speed + s;
        }
    }
    public void downA(int n) {
        if (speed < 0) {
            speed = 0;
        } else {
            speed = speed - n;
        }
    }
    public void addRen(int ren){
        if(ren>45){
            number=45;
        }else{
            number=number+ren;
        }
    }
    public void downRen(int ren){

```

```

        if(ren<0){
            number=0;
        }else{
            number=number-ren;
        }
    }
}

public class bus2 extends bus{
    public static void main(String[] args) {
        bus2 zi=new bus2();
        zi.addA(14);
        zi.addRen(15);
        System.out.println(" 车的速度为、 "+zi.getSpeed()+" "+" 车上
的人数为、 "+zi.getNumber());
    }
}

```

10、定义一个动物类， Animal(动物类型 type 成员变量， 一个发声的方法 sound());定义一个基于动物的狗类 Dog (覆盖父类的发声方法) ；定义一个基于动物的猫类 Cat(覆盖父类的发声方法) ；定义 AnimalDemo 类，声明一个 Animal 对象，分别创建一个 Dog,Cat,类的对象，然后通过访问 Animal 对象来访问 Dog Cat 类对象的 sound() 方法。

```

public class Animal {
    String type="";
    public Animal(){
    }
    public Animal(String type ){
        this.type=type;
    }
    public void sound(){
    }
}

public class Dog extends Animal {
    String color;
    public Dog (){
        super();
    }
    public Dog (String type,String color){
        this.type=type;
        this.color=color;
    }
    public void sound(){
        System.out.println(" 我的宠物 :"+type+"\t"+" 颜色 :"+color);}
}

public class Cat extends Animal {

```

```

String color;
public Cat(){super();}
public Cat(String type ,String color){
this.type=type;
this.color=color;}
public void sound(){
System.out.println(" 我的宠物 :"+type+"\t"+" 颜色 :"+color);}
public class AnimalDemo {
    public static void main(String[] args) {
        Dog dog =new Dog(" 狗"," 红色 ");
        dog.sound();
        Cat cat =new Cat(" 猫"," 黑色 ");
        cat.sound();
    }
}

```

11、创建两个 Student 类的对象， 比较二者年龄， 输出其中年龄大的学生的姓名。

```

public class xuesheng {
    String name;
    int age;
    String sex;
    public xuesheng(){
    public xuesheng(String name,int age,String sex){
        this.name=name;
        this.age=age;
        this.sex=sex;}
    public String toString(){
        return " 名字、 "+name+"          "+" 年龄、 "+age+"          "+" 性
别、 "+sex;
    }
    public String getName() {
        return name;}
    public void setName(String name) {
        this.name = name;}
    public int getAge() {
        return age;}
    public void setAge(int age) {
        this.age = age;}
    public String getSex() {
        return sex;}
}

```

```

        public void setSex(String sex) {
            this.sex = sex;
        }
    }
    public class xuesheng2 {
        public static void main(String[] args) {
            xuesheng x=new xuesheng("张三 ",59," 男 ");
            xuesheng d=new xuesheng("小红 ",56," 女 ");
            int w=x.getAge();
            int i=d.getAge();
            System.out.println(x.toString());
            System.out.println(d.toString());
            if(w>i){
                System.out.println(" 年龄较大的一个、    "+w+" 岁");
            }else{
                System.out.println(" 年龄较大的一个、    "+i+" 岁 ");
            }
        }
    }
}

```

12、编程创建一个 Person类，其中包括域（姓名，性别，年龄）和方法（获得姓名，性别，修改年龄）。将Person类作为父类，创建子类Student和Teacher .编写应用程序，创建 Student和Teacher对象。

```

public class chuangujian {
    private String name;
    private String sex;
    private int age;
    public chuangujian() {
    }
    public chuangujian(String name, String sex, int age) {
        this.name = name;
        this.sex = sex;
        this.age = age;
    }
    public int setAge() {
        if (age > 23 && age < 50) {
            age = age + 1;
            return 0;
        } else {
            return 1;
        }
    }
}

```

```

    public String getName() {
        return name;
    }
    public String getSex() {
        return sex;
    }
    public int getAge() {
        return age;
    }
    public String toString() {
        return "姓名、 " + name + "      " + " 性别、 " + sex + "      " + " 年
龄、 " + age;
    }
}

public class chuangjian2 extends chuangjian {
    private String banJi;
    public chuangjian2() {
    }
    public chuangjian2(String name, String sex, int age, String banJi) {
        super(name, sex, age);
        this.banJi = banJi;
    }
    public String toString() {
        return super.toString() + " 班级、 " + banJi;
    }
    public String getBanJi() {
        return banJi;
    }
    public void setBanJi(String banJi) {
        this.banJi = banJi;
    }
}

public class chuangjian3 extends chuangjian {
    private int jiaoShiHao;
    public chuangjian3() {
    }
    public chuangjian3(String name, String sex, int age, int jiaoShiHao) {
        super(name, sex, age);
        this.jiaoShiHao = jiaoShiHao;
    }
}

```

```

    public String fangfa() {
        return super.toString() + " 号码、 " + jiaoShiHao;
    }
    public int getJiaoShiHao() {
        return jiaoShiHao;
    }
    public void setJiaoShiHao(int jiaoShiHao) {
        this.jiaoShiHao = jiaoShiHao;
    }
}
public class chuangjian4 {
    public static void main(String[] args) {
        chuangjian2 c=new chuangjian2(" 张三 "," 男",8,"0920181");
        System.out.println(c.toString());
        chuangjian3 d=new chuangjian3(" 小红 "," 女",45,12);
        System.out.println(d.fangfa());
    }
}

```

13、定义一个体育活动类 (Sports)作为基类，它有一个进行活动的方法 play。足球 (Football) 和篮球 (Basketball) 都是体育活动类的衍生类。

```

public class gd {
    String name="";
    public gd(){}
    public gd(String name){
        this.name=name;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
}
public class gd2 extends gd {
    String ball;
    public String getBall() {
        return ball;
    }
    public void setBall(String ball) {
        this.ball = ball;
    }
}

```

```

    }
    public gd2(String s) {
        super(s);
    }
    public gd2(String name, String ball) {
        this.name = name;
        this.ball = ball;
    }
}
public class gd3 extends gd {
    String ball;
    public gd3() {
        super();
    }
    public gd3(String name, String ball) {
        this.name = name;
        this.ball = ball;
    }
    public String getBall() {
        return ball;
    }
    public void setBall(String ball) {
        this.ball = ball;
    }
    public static void main(String[] args) {
        gd g = new gd();
        gd2 gd = new gd2("");
        gd3 sd = new gd3();
        g.setName("坏人");
        gd.setBall("在踢足球！！");
        sd.setBall("在打篮球！！");
        System.out.println(gd.getName() + gd.getBall());
        System.out.println(sd.getName() + sd.getBall());
    }
}

```

14、定义一个类 BOX，计算长方体的体积、长、宽、高由键盘输入。

```

import java.io.*;
public class box {
    double l;
    double w;

```

```

double h;
public void main(String[] sdf) throws IOException {
    InputStreamReader ir = new InputStreamReader(System.in);
    BufferedReader br = new BufferedReader(ir);
    String str;
    box mybox;
    mybox = new box();
    double vol;
    System.out.println( "请输入长、  ");
    str = br.readLine();
    mybox.l = Double.parseDouble(str);
    System.out.println( "请输入宽、  ");
    str = br.readLine();
    mybox.w = Double.parseDouble(str);
    System.out.println( "请输入高、  ");
    str = br.readLine();
    mybox.h = Double.parseDouble(str);
    vol = mybox.l * mybox.h * mybox.w;
    System.out.println( "长方体的体积为、  " + vol);
}
}

```

15、自定义方法，求三个数的最大值（要求、从键盘上输入）

```

import java.io.*;
public class max {
    static int maxFrom(int x,int y,int z){
        int max;max=Math.max(x, y);
        max=Math.max(max, z);
        return max;
    }
    public static void main(String[] sad) throws IOException{
        BufferedReader br=new BufferedReader( new
InputStreamReader(System.in));
        String str;
        int max;
        System.out.println( "请输入三个数、  ");
        str=br.readLine();
        int a=Integer.parseInt(str);
        str=br.readLine();
        int b=Integer.parseInt(str);
        str=br.readLine();
    }
}

```



```

        int c=Integer.parseInt(str);
        max=maxFrom(a,b,c);
        System.out.println( "最大的一个数是、    "+max);
    }
}

```

16、定义一个抽象类， 他的属性有 length和 heigth，它的一个构造方法是设置子性质，另一个轴向方法是 area()用于求面积， 定义两个类分别是长方形 Rect和三角形 Triangle 求他们的面积，设计一个主类，调用相应的方法。

```

public abstract class chouXiang {
    int length;
    int heigth;
    abstract int sd();
    abstract int area(int length,int heigth);
}

public class chouXiang2 extends chouXiang {
    int area(int length, int heigth) {
        return length*heigth;
    }
    int sd() {
        return 0;
    }
}

public class chouXiang3 extends chouXiang {
    int area(int length, int heigth) {
        return (length * heigth) / 2;
    }
    int sd() {
        return 0;
    }
    public static void main(String[] args) {
        chouXiang3 c = new chouXiang3();
        System.out.println( "三角形的面积为、    " + c.area(10, 6));
        chouXiang2 d = new chouXiang2();
        System.out.println( "长方形的面积为、    " + d.area(8, 9));
    }
}

```

17、定义一个接口 M，其中有两个抽象方法， max()、 min()，在 P类中实现和继承，同时实现抽象方法 max(),min() 求三个数中的最大值和最

小值

```
Public abstract class MP{
Abstract int max (int a,int b,int c);
Abstract int min(int a,int b,int c);
}
Public class P extends M {
Int max(int a ,int b,int c){
Int x=a>b?a:b;
Return  x>c?x:c;}
Int min(int a,int b,int c){
Int x=a<b?a:b;
Return x<c?x:c;}
Public static void main(String [] ag){
P s=new P();
System.out.println(s.max(3,7,9);
System.out.println(s.min(3,7,9);
}
}
```

18、设计一个接口 <interface> 照顾宠物 "喂食 ()"" 陪他玩 ()", 工人 ..农民 干部 ...来实现 ..

```
interface PatCarel {
    public void raise();
    public void play();
}
```

```
class Worker implements PatCarel {
    String name;
    String sex;
    Worker(String name,String sex) {
        this.name = name;
        this.sex = sex;
    }
    public void raise() {
        System.out.println(" 喂养小猫 ");
    }
    public void play() {
        System.out.println("no time");
    }
}
```

```

class Farmer implements PatCarel {
    int age;
    int income;
    Farmer (int age,int income) {
        this.age = age;
        this.income = income;
    }
    public void raise() {
        System.out.println(" 喂养小狗 ");
    }
    public void play() {
        System.out.println("no money");
    }
}

```

```

class Officer implements PatCarel {
    int income;
    int age;
    Officer(int income,int age) {
        this.income = income;
        this.age = age;
    }
    public void raise() {
        System.out.println(" 喂养小鸟 ");
    }
    public void play() {
        System.out.println("have money");
    }
}

```

```

public class TestInterface {
    public static void main(String[] args) {
        PatCarel p = new Worker("Tom","male");
        PatCarel m = new Farmer(66,10000);
        PatCarel q = new Officer (100000,60);
        p.raise(); p.play() ;
        m.play() ;
        q.raise(); q.play() ;
    }
}

```

19、设计一个公共汽车类 bus，只要求有速度和人数两个属性，和一个不带参数的构建器，具有加速度，减速，上人，下人，停车，启动等方法。创建一个对象，车上有 35人，开始发车，中间下了 13人，又上了25人，请问到站还有多少人 用程序实现

```
public class bus {
    private int number;// 私有属性，只有通过方法才能获取
    全局变量
    int speed;
    public int getNumber() { // 获取 number的值
        return number;
    }
    public void setNumber(int number) { // 改变 number的值
        this.number = number;
    }
    public int getSpeed(int speed) { // 获取 speed的值
        return speed;
    }
    public void setSpeed(int speed) { // 改变 speed的值
        this.speed = speed;
    }
    public bus() { // 不带参数的构造方法
    }
    public bus(int number) { // 带一个参数的构造方法
        this.number = number;
    }
    public bus(int number, int speed) { // 带两个参数的构造方法
        this.number = number;
        this.speed = speed;
    }
    public void add(int number) {
        if (number > this.number) {
            this.number = 0;
        } else {
            this.number = this.number - number;
        }
    }
}

import javax.swing.*;
public class bus2 {
```

```

    public static void main(String[] args) {
        JOptionPane.showMessageDialog(null, " 创建一个 35人的对象！
");// 显示窗口
        bus bus =new bus(35,50);
        String st =JOptionPane.showInputDialog(" 要求下车人数、 ");// 输入
        要下车的人数
        try{int s =Integer.parseInt(st);
            if(s<0){JOptionPane.showMessageDialog(null," 兄弟，不要乱来！
");}
            else{bus.add(s);}// 调用前面的 add方法
        }
        catch(Exception e){
            JOptionPane.showMessageDialog(null," 兄弟，你在乱输哦！ ");
        }
        JOptionPane.showMessageDialog(null," 车上还有
"+bus.getNumber()+" 人");
    }
}

```

20、设计一个表示二维平面上的圆的类 Circle, 还包含有表示圆半径的 protected 类型的成员变量 radius, 获取和设置 radius 的 public 方法, 计算圆面积。 public 方法设计一个表示圆柱体的类 Cylinder, 它继承自类 Circle, 还包含有表示圆柱体高的 protected 类型的成员变量 height, 获取和设置 height 值的 public 方法, 计算圆柱体体积的 public 方法。建立若干个 Cylinder 对象, 输出其半径, 高及其体积的值。要求: A. 每个类包含无参数和有参数的构造方法。构造方法用于对成员变量初始化。

```

public class Circle {
    protected double radius;
    public Circle(){
    }
    public void fangFa(){
        System.out.println(" 圆的面积为、 "+Math.PI*radius*radius);
    }
    public double getRadius() {
        return radius;
    }
    public void setRadius(int radius) {
        this.radius = radius;
    }
}

```

```

    public double getR() {
        return radius;
    }
    public void setR(int r) {
        this.radius = r;
    }
}
public class Cylinder extends Circle {
    protected double height;
    public void gao(){
    }
    public void gao(int height){
        this.height=height;
    }
    public void tiJi(){
        System.out.println(" 圆柱体的体积为、
"+Math.PI*(radius*radius)*height);
    }
    public double getHeight() {
        return height;
    }
    public void setHeight(int height) {
        this.height = height;
    }
    public double getR() {
        return radius;
    }
    public void setR(int r) {
        this.radius= r;
    }
}
public class Cylinder2 {
    public static void main(String[] args) {
        Circle c=new Circle();
        Cylinder cc=new Cylinder();
        cc.gao(12);
        cc.setR(5);
        cc.tiJi();
        c.setRadius(1);
        c.setR(4);
    }
}

```

```

        c.fangFa();
        System.out.println(" 圆柱体的半径为  :"+cc.getRadius());
        System.out.println(" 圆柱体的高为  :"+cc.getHeight());
    }
}

```

21、设计一个抽象类，并比较三个数的大小。

```

public abstract class shu {
    abstract int max(int a,int b,int c);
    abstract double min(double a,double b,double c);
}

public class Operation extends shu {
    @Override
    int max(int a, int b, int c) {
        // TODO Auto-generated method stub
        int x=a>b?a:b;
        return x>c?x:c;
    }
    double min(double a, double b, double c) {
        double x=(a<b?a:b);
        return  x<c?x:c;
    }
    public static void main(String[] args) {
        Operation o=new Operation();
        System.out.println(" 最大的一个数是、    "+o.max(5, 7, 2));
        System.out.println(" 最小的一个数是、    "+o.min(4.6,44,5.6));
    }
}

```

22、利用多态，设计一个程序，并计算出圆的面积、正方体体积、长方体体积、圆柱体、圆锥体体积

```

public class zhengfatiti {
    int r;
    double l;
    int w;
    int h;
    public zhengfatiti(){}
    public double zhengfatiti(int l){// 正方体、长 * 长 * 长
        return l*l*l;
    }
    public double zhengfatiti(int r,int h){// 圆柱体、 3.14* 半径 * 高
        return Math.PI*r*h;
    }
}

```

```

    }
    public int zhengfatiti(int l,int w,int h ){//    长方体、长 * 宽 * 高
        return l*w*h;
    }
    public double zhengfatitit(double r,double h){//    圆锥体、 ( 3.14* 半径 *
高 ) /3
        return (Math.PI*r*h)/3;
    }
    public double zhengfatiti(double l){
        return 4.0/3*Math.PI*l*l*l;
    }
    public double zhengfatiti1(double r,int l){//    圆的面积
        return Math.PI*4*r*r;
    }
    public static void main(String [] ag){
        zhengfatiti z=new zhengfatiti();
        System.out.println(z.zhengfatiti(3,4,5));
    }
}

```

23、定义个 Compute接口，有抽象方法 add()、subtract()\multiply()、和 divide(); 定义实现类 Computer. 创建 Compute对象完成加、减、乘、除方法的调用

```

public interface Computer {
    public static final int a=6;
    public static final int w=2;
    public abstract int add();
    public abstract int subtract();
    public abstract int multiply();
    public abstract int divide();
}
public class Compute implements Computer {
    public int add(){
        return a+w;
    }
    public int subtract() {
        return a-w;
    }
    public int multiply() {
        return a*w;
    }
}

```



```

public int divide() {
    return a/w;
}
public static void main(String[] args) {
    Compute c=new Compute();
    System.out.println(" 该算式的加法是、 "+c.add());
    System.out.println(" 该算式的减法是、 "+c.subtract());
    System.out.println(" 该算式的乘法是、 "+c.multiply());
    System.out.println(" 该算式的除法是、 "+c.divide());
}
}

```

24、定义一个抽象 Myintertace, 其中有两个抽象方法 max()、 min(). 在 Operation 类中实现该抽象, 同时实现抽象方法 max()、 min() 并且要求三个数的最大最小值。

```

public interface Myinterface {
    public static final int a=3;
    public static final int b=4;
    public static final int c=5;
    public abstract int max();
    public abstract int min();
}
public class Operation implements Myinterface {
    public int max(){
        int x=a>b?a:b;
        int k=x>c?x:c;
        return k;
    }
    public int min(){
        int x=a<b?a:b;
        int k=x<c?x:c;
        return k;
    }
    public static void main(String[] args) {
        Operation o=new Operation();
        System.out.println(" 最大的一个数是、 "+o.max());
        System.out.println(" 最小的一个数是、 "+o.min());
    }
}

```

25、利用外部类, 求圆的面积。

```

public class lizi {// 定义一个外部类

```

```

int rad = 0;
public int mianji(int a) {
    return a * a;// 外部类的成员方法
}
class leibu {
    int m = 6;// 圆的半径
    public int mianji() {
        return (int)(Math.PI * m * m);// 内部类的成员方法
    }
}

public String toString() {
    leibu l = new leibu();
    int k = l.mianji();
    return k + "";
}

public static void main(String[] da) {
    lizi d = new lizi();
    System.out.println(d.toString());
}
}

```

26、定义一个 ssanjian类，用于实现扑克牌的正常发放。里面有 face和 suit属性。并且有构造方法和相应的 get、set方法和一个 toString 方法。第一个 ssanjian2类作为它的子类，并继承父类的方法，并实现此方法。

```

public class ssanjian {
    private String face;
    private String suit;
    public ssanjian(){}
    public ssanjian(String face){
        this.face=face;
    }
    public ssanjian(String face,String suit){
        this.face=face;
        this.suit=suit;
    }
    public String toString(){
        return suit+face;
    }
}

```

```

public class ssanjian2 extends ssanjian {
    public static void main(String [] sf){
        String f[]={"A","2","3","4","5","6","7","8","9","10","J","Q","K"};
        String s[]={" 黑桃 "," 红桃 "," 梅花 "," 方块 "};
        ssanjian[] de=new ssanjian[52];
        for(int i =0;i<de.length;i++){
            de[i]=new ssanjian(f[i%13],s[i/13]);
            System.out.println(de[i].toString());
            if(i==12||i==25||i==38){
                System.out.println();
            }
        }
        for(int i=0;i<de.length;i++){
            for(int j=0;j<de.length-i;j++){
                int se=(int)(Math.random()*(de.length-i));
                ssanjian er;
                er=de[j];
                de[j]=de[se];
                de[se]=er;
            }
        }
        for(int i =0;i<de.length;i++){

            System.out.print(de[i].toString()+"    ");
            if(i==12||i==25||i==38){
                System.out.println();
            }
        }
    }
}

```

27、定义一个计算 x^n 的类，然后用该类求 $10*10*10$ 的值。

```

import java.io.*;
public class Class1 {
    public static void main( String args[ ] )
    { SubClass a = new SubClass( 10,3 );
      System.out.println( a.exp( ));
    }
}
class SuperClass {
    float x;
    int n;
}

```

```

    SuperClass( float xx , int nn ) { x = xx ; n = nn; }
}
class SubClass extends SuperClass {
SubClass( float xx , int nn ) { super( xx , nn ); }
    float exp( ) {
float s = 1;
for ( int i = 1; i<=n; i++ ) s = s*x;
return s;
}
}

```

28、设计一个 Birthday 类，其成员变量有、 year,month,day；提供构造方法、输出 Birthday 对象值的方法和计算年龄的方法。编写程序测试这个类。（引入 java.util.Calendar 包，取得当年份 Calendar.getInstance().get(Calendar.YEAR))

```

public class a {
    int y;
    int m;
    int d;
    public a(){}
    public a(int y){
        this.y=y;

    }
    public a(int y,int m){
        this.y=y;
        this.m=m;
    }
    public a(int y,int m,int d){
        this.y=y;
        this.m=m;
        this.d=d;
    }
    public int fangFa(){
        return y-y;
    }
    public int getY() {
        return y;
    }
    public void setY(int y) {
        this.y = y;
    }
}

```

```

    }
    public int getM() {
        return m;
    }
    public void setM(int m) {
        this.m = m;
    }
    public int getD() {
        return d;
    }
    public void setD(int d) {
        this.d = d;
    }
}
import java.util.*;
public class a1 extends a {
    public static void main(String[] args) {

        a as = new a();
        Calendar c = Calendar.getInstance();
        int s = Calendar.getInstance().get(Calendar.YEAR);
        System.out.println(" 现在是、  " + s + " 年");
        as.setY(1990);
        System.out.println(" 您输入的年份是、  " + as.y + " 年");
        System.out.println(" 您现在的岁数是、  " + (s - as.getY()) + " 岁");
    }

}

```

29、定义一个类 Rectangle 代表矩形，其中包括计算面积的方法。再定义一个它的子类 Square[代表正方形]，其中也包含计算面积的方法。编写一程序，测试新建子类对象的属性和方法。

```

public class b {
    int a;
    int c;
    public b(){}
    public b(int a){
        this.a=a;
    }
    public b(int a,int c){
        this.a=a;
        this.c=c;
    }
}

```

```

    }
    public int mianJi(){
        return a*c;
    }
    public int zMianJi(){
        return a*a;
    }
    public int getA() {
        return a;
    }
    public void setA(int a) {
        this.a = a;
    }
    public int getC() {
        return c;
    }
    public void setC(int c) {
        this.c = c;
    }
}

import javax.swing.*;

public class b1 extends b {
    public static void main(String [] ad){
        JOptionPane.showMessageDialog(null, " 创建一个计算面积的对
象！ ");
        b1 u=new b1();
        u.setA(12);
        u.setC(3);
        System.out.println(" 矩形的面积为、 "+u.mianJi());
        System.out.println(" 正方形的面积为、 "+u.zMianJi());
    }
}

```

30、 实作一个时间类 MyTime ，包含一个构建器 ， 3 个私有成员变量 hour ， minute ， second 以及它们相应的 get 和 set 方法 ，再加上一个转换成字符串的方法 toString。

```

public class MyTime {
    private int hour;

```

```

private int minute;
private int second;
public MyTime(int h,int m,int s){
    hour=h;
    minute=m;
    second=s;
}
public int getHour() {
return hour;
}
public void setHour(int hour) {
    this.hour = hour;
}
public int getMinute() {
    return minute;
}
public void setMinute(int minute) {
    this.minute = minute;
}
public int getSecond() {
    return second;
}
public void setSecond(int second) {
    this.second = second;
}
public String toString(){
    return hour+":"+minute+":"+second;
}
}

```

31、实作一个日期类 `MyDate` ,包含一个构建器 , 3 个私有成员变量 `year` , `month` , `day` 以及它们相应的 `get` 和 `set` 方法 , 再加上一个转换成字符串的方法 `toString`。

```

public class MyDate {
    private int year;
    private int month;
    private int day;
    public MyDate(int y,int m,int d){
        year=y;
        month=m;
        day=d;
    }
}

```

```

    }
    public int getYear() {
        return year;
    }
    public void setYear(int year) {
        this.year = year;
    }
    public int getMonth() {
        return month;
    }
    public void setMonth(int month) {
        this.month = month;
    }
    public int getDay() {
        return day;
    }
    public void setDay(int day) {
        this.day = day;
    }
    public String toString(){

        return year+"-"+month+"-"+day;
    }
}

```

32、实作一个 FullTime 类 ,包含一个 MyTime 类型的是有成员变量 time 以及与之相对应的 get 与 set 方法 , 一个 MyDate 类型的成员变量 date 以及与之相对应的 get 与 set 方法 , 一个构建器 , 再加上一个转换成字符串的方法 toString。

33、假设当前时间为 2010 年 3 月 10 日 9 点 30 分 45 秒 , 编写一个程序 , 用 FullTime 类设置时间为以上时间 , 并打印。

```

public class FullTime {
    MyTime time;// 成员变量
    MyDate date;
    public FullTime (int y,int m,int d,int h,int minute,int s){
        time=new MyTime (h,minute,s);
        date=new MyDate (y,m,d);
    }
    public lianxi getTime() {

```



```

        return time;
    }
    public void setTime(lianxi time) {
        this.time = time;
    }
    public lianxi2 getDate() {
        return date;
    }
    public void setDate(lianxi2 date) {
        this.date = date;
    }
    public String toString(){
        return date.toString()+"    "+time.toString();
    }
    public static void main(String [] ad){
        FullTime l=new FullTime (2010,3,10,9,30,45);
        System.out.println(l.toString());
    }
}

```

}

34、定义一个体育活动类 (Sports)作为基类，它有一个进行活动的方法 play(方法名命名规范、 J2EE 方向、小写字母开头 . NET 方向 :大写字母开头)。

足球 (Football) 和篮球 (Basketball) 都是体育活动类的衍生类。

请在测试类 (Test)中编写一个方法 howToPlay(Sports sp) ,该方法要求传递一个 Sports 类型的参数。该方法的作用是、

1) 当传入的实例类型为 Football 时，控制台上应打印、足球是用脚踢的！

2) 当传入的实例类型为 Basketball 时，控制台上应打印、篮球是用手打的！

在 main 方法中调用 howToPlay() 验证你的代码是对的。

```

public class Sports{
    public void play(){
        System.out.println (" 运动 ");
    }
}
class Football extends Sports{
    public void play(){
        System.out.println (" 足球是用脚踢的 ");
    }
}

```

```

}
class Basketball extends Sports{
public void play(){
System.out.println (" 篮球是用手打的 ");
}
}
class Test{
public void howToPlay(Sports sp){
sp.play();
}
public static void main(String[]args){
Test t=new Test();
t.howToPlay(new Basketball());
}
}

```

35、乐器（ Instrument ）分为、钢琴（Piano）、小提琴（Violin）

- * 各种乐器的弹奏（ play ）方法各不相同。
- * 编写一个测试类 InstrumentTest，要求、
- * 编写方法 testPlay，对各种乐器进行弹奏测试。 要依据乐器的不同，进行相应的弹奏。
- * 在 main 方法中进行测试

```

public interface Instrument {
    void play();
}
public class Piano implements Instrument {
    public Piano() {
    }
    public void play() {
        System.out.println("Piano Playing");
    }
}
public class Violin implements Instrument{
    public Violin() {
    }
    public void play() {
        System.out.println("Piano Playing");
    }
}
public class InstrumentTest {
    public InstrumentTest() {

```

```

    }
    public static void main(String[] args) {
        Instrument p=new Piano();
        Instrument v=new Violin();
        p.play();
        v.play();
    }
}

```

36、声明一个包括各种车辆的抽象类，通过继承机制定义公共汽车和自行车两个类，构造对应的方法区分公共汽车和自行车，并统计各自的数量，完成公共汽车和自行车的实例化。

```

abstract class Jizhi{
    int count=0;
}
class Cart extends Jizhi{
public Cart(){
System.out.println(" 这是一辆公共汽车  ");
}
}
class Bike{
public Bike(){
System.out.println(" 这是一辆自行车  ");
}
}
public class A{
public static void main(String[] args){
    Cart cart=new Cart();
    Bike bike=new Bike();
}
}

```

37、编写学生类 (class student)继承人类 (class preson)的程序 ,要求父类的构造函数为 (preson(string name),子类的构造函数要调用父类的构造函数.p()方法用于输出该对象的信息 (如 The person's name is Zhangheng),学生类有 study()方法 ,并分别创建一个对象并调用各个方法

```

public class luanxie {
    String name;
    luanxie(String name) {
        this.name = name;
    }
    public void p() {

```

```

        System.out.println(" 我的名字叫犀利，人称    ，犀利哥 ?是也！ ");
    }
}
public class luanxie2 extends luanxie {
    luanxie2() {
        super("");
    }
    public void study() {
        System.out.println(" 我啊，想都没想过要出名。一次意外让我在
网络上这么流行！ ");
    }
    public static void main(String[] args) {
        luanxie per = new luanxie("");
        per.p();
        luanxie2 stu = new luanxie2();
        stu.study();
    }
}

```

38、设计一个学生类 Student，其数据成员有 name(姓名)、age(年龄)和 degree(学位)。由 Student 类派生出本科生类 Undergraduate 和研究生类 Graduate，本科生类 Undergraduate 增加成员 specialty(专业)，研究生类增加成员 direction(研究方向)。每个类都有 show()方法，用于输出数据成员信息

```

package bianti;
public class show {
    public String name;
    public int age;
    public String degree;
    public show(String n1, int a1, String d1) // 构造方法
    {
        name = n1;
        age = a1;
        degree = d1;
    }
    public void show() {
        System.out.print(" 学生、" + name + "    " + " 年龄、" + age + "    "
+ " 学位、 "
        + degree + '\n');
    }
}

```

```

public class Undergraduate extends show {
    public String specialty;
    public Undergraduate(String name, int age, String degree, String spe) {
        super(name, age, degree);
        specialty = spe;
    }
    public void show() {
        super.show();// 调用父类的方法
        System.out.println(" 专业 " + specialty);
    }
}

class Graduate extends show{
    public String direction;
    public Graduate(String name, int age, String degree, String dir) {
        super(name, age, degree);
        direction = dir;
    }
    public void show() {
        super.show();// 调用父类的方法
        System.out.println(" 研究方向 " + direction);
    }
}

public class Student1 {
    public static void main(String[] args) {
        show stu1 = new show(" 张 1", 20," 普通 ");
        stu1.show();
        Undergraduate stu2 = new Undergraduate(" 陈定 ", 21, " 本科生 ", "
计算机 ");
        stu2.show();
        Graduate stu3 = new Graduate(" 李生 ", 26, " 研究生 ", " 信息管理 ");
        stu3.show();
    }
}

```

39、编写学生类 (class student)继承人类 (class preson)的程序 ,要求父类的构造函数为 (preson(string name),子类的构造函数要调用父类的构造函数.p()方法用于输出该对象的信息 (如 The person's name is Zhangheng),学生类有 study()方法 ,并分别创建一个对象并调用各个方法

```

package bianti;
public class Person {
    String name;

```

```

public Person(String name){
    this.name=name;
}
public void p(){
    System.out.println(" 这人的名字是、人妖！  ");
}
public String getName() {
    return name;
}
public void setName(String name) {
    this.name = name;
}
}
package bianti;

```

```

public class student extends Person {
    student(){
        super("猪妖！ ");
    }
    public void read(){
        System.out.println(" 他还是一个学生！  ");
    }
    public static void main(String[] args) {
        Person p=new Person("张");
        p.p();
        student s=new student();
        s.read();
    }
}

```

40、在父类 Base 和子类 Sub 中都定义了成员变量 var 及成员方法 method()。在 Sub 类中，可通过 super.var 和 super.method() 来访问 Base 类的成员变量 var 及成员方法 method()。

```

public class d {
    String var = " 基本值！ ";
    public void method() {
        System.out.println(" 基本方法！ ");
    }
}
public class d1 extends d {
    String var = " 基本值 "; // 隐藏父类的 var 变量
}

```

```

public void method() {
    System.out.println(" 前面的方法！  ");
} // 覆盖父类的 method() 方法
void test() {
    String var = " 值"; // 局部变量
    System.out.println(" 值是  " + var); // 打印 method()方法的局部变
量
    System.out.println("this  的值是、  " + this.var); // 打印 Sub 实例的
实例变量
    System.out.println("super  的值是、  " + super.var); // 打印在 Base
类中定义的实例变量
    method(); // 调用 Sub 实例的 method() 方法
    this.method(); // 调用 Sub 实例的 method() 方法
    super.method(); // 调用在 Base 类中定义的 method() 方法
}
public static void main(String args[]) {
    d1 sub = new d1();
    sub.test();
}
}

```

41、 创建一个类 Queue 代表队列（其特点为、先进先出），添加方法 add(Object obj) 以及 get()，并添加 main() 方法进行效果验证。（要求、使用 LinkedList 实现队列、在向 LinkedList 中添加时，使用 addLast() 方法，在从 LinkedList 中取出时，使用 removeFirst() 方法）

```

import java.util.LinkedList;
public class Queue {
    LinkedList queue;
    public Queue() {
        queue = new LinkedList();
    }
    void add(Object o) {
        queue.addLast(o);
    }
    Object get() {
        Object o = null;
        if (!queue.isEmpty()) {
            o = queue.getFirst();
            queue.removeFirst();
        }
    }
}

```

```

    return o;
}
public static void main(String[] args){
    Queue q=new Queue();
    System.out.println(q.get());
    q.add("1");
    System.out.println(q.get());
    q.add("2");
    q.add("3");
    System.out.println(q.get());
    q.add("4");
    q.add("5");
    System.out.println(q.get());
    System.out.println(q.get());
    System.out.println(q.get());
    System.out.println(q.get());
}
}

```

42、定义了一个学生的类 Student,包括成员变量 “学号”、“班号”、“姓名”、“性别”、“年龄”,成员方法:“获得学号”、“获得班号”、“获得性别”、“获得姓名”、“获得年龄”。

请为 Student 类定义构造函数初始化所有的成员变量。增加一个方法 public String toString() 把 Student 类对象的所有信息组合成一个字符串。

答案：

源程序： StudentDemo.java

```

public class StudentDemo {
    public static void main(String args[]) {
        Student s1= new Student("20050101",92,'m'," 李军 ",20);
        System.out.println(s1.toString()); } }
class Student {
    private String studentId;
    private int classId;
    private String studentName;
    private char studentGender;
    private int studentAge;
    String getStudentId() { return studentId; }
    int getClassId() { return classId; }
    char getStudentGender() { return studentGender; }
    String getStudentName() { return studentName; }
}

```



```

int getStudentAge() { return studentAge; }
boolean setStudentAge(int newAge) {
if( newAge>0 && newAge<130) {
studentAge = newAge;
return true; }
else {
System.out.println(" 学生年龄输入错误  !");
return false; } }
Student(String id, int clsId, char gender, String name, int age) {
studentId = id;
classId = clsId;
studentName = new String(name);
studentAge = age;
studentGender = gender; }
public String toString() {
return "学生信息： \n" + "学号：" + this.getStudentId() + "\t 班号：" +
this.getClassId() + "\t 姓名：" + this.getStudentName() + "\t 性别：" +
((this.getStudentGender()) == 'm' ? "男" : "女") + "\t 年龄：" +
this.getStudentAge();
}
}

```

43、构造函数是否可以被重载？试举例。

答案：构造函数可以被重载

源程序： ConstructorTest.java

```

public class ConstructorTest {
public static void main(String args[]) {
System.out.println(" 创建父类对象： ");
SuperClass sc = new SuperClass();
System.out.println("\n 创建第一个子类对象： ");
SubClass sc1 = new SubClass();
System.out.println("\n 创建第二个子类对象： ");
SubClass sc2 = new SubClass(1); }
}
class SuperClass {
SuperClass() {
System.out.println(" 父类的构造函数 "); }
}
class SubClass extends SuperClass {
SubClass() {
System.out.println(" 子类的第一个构造函数 "); }
}

```

```
SubClass(int i) {
System.out.println(" 子类的第二个构造函数  "); }
}
```

44、构造一个类来描述屏幕上的一个点， 该类的构成包括点的 x 和 y 两个坐标，以及一些对点 进行的操作，包括：取得点的坐标值，对点的坐标进行赋值，编写应用程序生成该类的对象并 对其进行操作。

答案：

源程序： Point.java

```
public class Point {
private int x;
private int y;
public Point(int x,int y) {
this.x=x;
this.y=y; }
public void setPoint(int x1,int y1) {
x=x1;
y=y1; }
public void getPoint() {
System.out.println("point x: "+ x+",y: "+ y); }
public static void main(String[] args) {
Point origin=new Point(10,10);
origin.getPoint();
origin.setPoint(20,20);
origin.getPoint(); } }
```

45、编程创建一个 Box 类，在其中定义三个变量表示一个立方体的长、宽、高，再定义一个方法 setDemo 来对这 3 个变量进行初始化，然后定义一个方法来求立方体的体积。 创建一个对象，求给定尺寸的立方体的体积。

答案：

源程序： Box.java

```
public class Box {
private double myLength;
private double myWidth;
private double myHeight;
public void setDemo(double l,double w,double h) {
myLength=l;
myWidth=w;
myHeight=h;
}
public double Volume(){
```

```
return myLength*myWidth*myHeight;
}
}
```

源程序： BoxDemo.java

```
public class BoxDemo {
    public static void main(String args[])
    {
        Box box1=new Box();
        box1.setDemo(10.00,20.00,30.00);
        System.out.println("the box's volume:"+box1.V olume());
    }
}
```

46、定义一个类实现银行帐户的概念，包括的变量有“帐号”和“存款余额”，包括的方法有“存款”、“取款”和“查询余额”。定义主类，创建帐户类的对象，并完成相应操作。

．答案：

源程序： BankAccount.java

```
public class BankAccount {
    private double leftmoney;
    private long account;
    public BankAccount(long acco,double lmoney ) {
        account=acco;
        leftmoney=lmoney;
    }
    public double getleftmoney()
    { return leftmoney; }
    public void savemoney(double money){
        leftmoney+=money;
    }
    public void getmoney(double money){
        if(money<=leftmoney)
            leftmoney-=money;
        else
            System.out.println(" 您的帐户余额不足，只能取： "+leftmoney);
    }
}
```

源程序： TestAccount.java

```
public class TestAccount {
    public static void main(String args[]) {
        BankAccount myAccount=new BankAccount(12345678,1000);
        myAccount.savemoney(5000);
    }
}
```

```

System.out.println("    存    入        5000    元    后    余    额    为    :
"+myAccount.getleftmoney());
myAccount.getmoney(2000);
System.out.println("    取    出        2000    元    后    余    额    为    :
"+myAccount.getleftmoney());
}
}
int age;
public Person(String n,char s,int a) {
name=n;
sex=s;
age=a; }
public String toString(){
return " 姓名 : " +name+"\n  性别 : " +sex+"\n  年龄 :"+age;
}
}
public class Student extends Person {
long number;
int phi,eng,comp;
public Student(long k,String s,char c,int i,int i1,int i2,int i3) {
super(s,c,i);
number=k;
phi=i1;
eng=i2;
comp=i3; }
public double aver() { return(phi+eng+comp)/3.0; }
public int max() {
int temp,max;
temp=phi>eng?phi:eng;
max=temp>comp?temp:comp;
return max; }
public int min() {
int temp,min;
temp=phi<eng?phi:eng;
min=temp<comp?temp:comp;
return min; }
public String toString(){
return " 学号 : " +number+" "+super.toString()+"\n    平均分 : " +aver()+" 最
高分 :"+max()+" 最低分 :    "+min(); }
}

```

47、编写 Student 类的测试程序 程序要求实现：学生信息（学号、姓名性别、年龄、平均分、最高分、最低分）在程序中设置，要求通过运行程序，在屏幕中输出如下的结果：例如：学号：200501 姓名：李丽 性别：女 年龄：20 平均分：90.0 最高分：95 分 最低分：87。

答案：

TestStudent.java

```
public class TestStudent {
    public static void main(String args[]) {
        Student stud=new Student(200501,"李丽",'女',20,95,87,88);
        System.out.println(stud.toString()); } }
```

48、定义 Point、Circle、Cylinder 三个类，三个类都拥有 x,y 属性，而 Circle 又增加了属性 radius，Cylinder 类又增加了属性 height。都有获得 x,y 坐标，设置 x,y 坐标，求面积、体积的方法。

答案：

Test.java

```
class Point {
    protected int x,y;
    public Point(){ setPoint(0,0); }
    public Point(int a,int b){ setPoint(a,b); }
    public void setPoint(int a,int b) {
        x=a; y=b; }
    public int getX() { return x; }
    public int getY() { return y; }
    public String toString()
    { return "("+x+","+y+""; }
    public double area() { return 0.0; }
    public double volume() { return 0.0; } }
class Circle extends Point //继承 Point 类
{
    protected double radius;
    public Circle() { //隐含调用父类的构造函数
        setRadius(0); }
    public Circle (double r,int a,int b) {
        super(a,b); //调用父类的构造函数
        setRadius(r); }
    public void setRadius(double r) {
        radius=(r>=0?r:0); }
    public double getRadius(double r) { return radius; }
    public double area() { return Math.PI*radius*radius; }
    public String toString()
```

```

{ return    "Center="+super.toString()+"Radius="+radius; } }
class Cylinder extends Circle          //继承 Point 类
{
protected double height;
public Cylinder () {          //隐含调用父类的构造函数
setHeight(0); }
public Cylinder (double h,double r,int a,int b) {
super(r,a,b);          //调用父类的构造函数
setHeight(h); }
public void setHeight (double h) {
height =(h>=0?h:0); }
public double getHeight () {    return height; }
public double area() { return 2*super.area()+2* Math.PI*radius*radius; }
public double volume() { return super.area()*height;    }
public String toString()
{ return    super.toString()+"Height="+ height; } }
public class Test {
public static void main(String args[]) {
Point point=new Point(10,12);
Circle circle=new Circle(3.5,22,9);
Cylinder cylinder=new Cylinder(10,3.3,10,10);
System.out.println(point.toString()+":"+ "\narea="+
point.area()+"\nvolume="+point.volume());
System.out.println(circle.toString()+":"+ "\narea="+
circle.area()+"\nvolume="+ circle.volume());
System.out.println(cylinder.toString()+":"+ "\narea="+
cylinder.area()+"\nvolume="+ cylinder.volume());
}
}

```

49 . 请编写程序由三个类定义组成 : 类 A 中只定义了一个类型为 String 的对象 x 和带有一个 参数的构造函数 (用来初始化 x); 类 B 是类 A 的子类 , 其中定义了一个 int 类型的变量 x 和 29 一个带两个参数的构造函数 (用来为父类中的对象 x 和类 B 中的变量 x 做初始化) , 还定义了一个方法 myPrint() , 功能为输出父类 A 中的 x 和其子类 B 中的 x 的值 ; 测试类中定义了 B 类的对象 b , 请通过创建 b 为上述两个 x 赋值 (类 A 中的 x 的值为 “ 这是父类的 x. ” , 类 B 中的 x 的值为 68 , 显示两个 x 的值。

B.java

```

class A {
String x;

```

```

A(String x1){ x=x1; } }
class B extends A {
int x;
B(String x1,int x2){super(x1); x=x2; }
void myPrint()
{ System.out.println(" 父类 A 中 x 的值="+super.x+" 子类 B 中 x 的
值="+x);
}}

```

源程序： Test.java

```

public class Test {
public static void main(String args[]) {
B b=new B(" 这是父类的 x.",68 );
b.myPrint(); } }

```

50 .请编写程序，它由三个类定义组成，类名分别为 A、B 和 Applet1：类 A 中定义了一个类型为 float 的属性 f 和一个类型为 byte 的私有属性 k；类 A 还定义了一个带有一个参数的构造函数（用于初始化 f），和两个公共方法 setk() 和 getk()，其中 setk() 的功能为给私有属性 k 赋值，getk() 的功能为返回私有属性 k 的值。类 B 是类 A 的子类，其中仅定义了一个类型为 boolean 的属性 b 和一个带两个参数的构造函数（用来为父类中的变量 f 和本类的变量 b 做初始化）。Test 类中定义了一个初值为 123 的 int 型变量 n 和一个 B 类的对象 x，请通过创建 x 为上述 f 和 b 赋值（f 的值为 68.963,b 的值为 true）；在 paint() 方法中，给 A 类的私有属性 k 赋值 66，然后显示 f，k，b，n 的值。

答案：

源程序： B.java

```

class A {
float f;
private byte k;
A(float f1){ f=f1; }
public byte getk( ){ return k; }
public void setk(byte m){ k=m; } }
class B extends A {
boolean b;
B(float ff,boolean bb){ super(ff); b=bb; } }

```

源程序： Test.java

```

public class Test {
public static void main(String args[]) {
int n=123;
B x=new B(68.963f,true);

```

```
x.setk((byte)66);
System.out.println("f="+x.f+"      k="+x.getk( )+"      b="+x.b+"      n="+n);
}}
```

51 . 设有 1 个 Professor 类，继承自 Human 类，该类是教师的蓝图，它除继承相应的属性和方法外，还具有下面的成员属性：

讲授课程属性： private String course

教学效果属性： private int result 该属性的取值为 0—3，1 表示效果良好，2 表示可以

接受，3 表示效果不佳，0 表示未予评价 成员方法： public String getDetails() 将姓名、性别、年龄和讲授课程属性以字符串方式返回。

public boolean set_reslut(int x) 该方法用于设置教学效果，必须考虑设置是否合理（若不是 0-3 之间，则不允许设置）构造方法：用于设置姓名，性别和年龄和讲授课程。

public Professor(String name, boolean sex, int age, String course)

答案：

源程序： Human.java

```
class Human {
String name;
boolean sex;
int age;
public Human(String n,boolean s,int a) {
name=n;
sex=s;
age=a; }
public String getDetails(){
return "姓名：" +name+"\n 性别：" +sex+"\n 年龄：" +age; } }
class Professor extends Human{
private String course;
private int result;
public String getDetails(){
return super.getDetails()+"\t"+course; }
public boolean set_reslut(int x){
if(x>=0&&x<=3){
result=x;
return true; }
else return false; }
public Professor(String name,boolean sex,int age,String course){
super(name,sex,age);
this.course=course; }
public Professor(String name,int age,String course){
```



```
this(name,false,age,course); } }
```

52、定义一个接口或多个接口，并用一个类实现。

答案：源程序： student.java

```
interface Student_info      //定义学生情况接口
{
int year=2006;
int age();
void output();
}
interface Student_score     //定义学生成绩接口
{
float total();
void output();
}
public class student implements Student_info,Student_score
{      //实现学生情况接口、学生成绩接口
String name;
int birth_year;
float math,english,computer;
public student(String n1,int y,float a,float b,float c) {
name=n1;
birth_year=y;
math=a;
english=b;
computer=c; }
public int age() {
return year- birth_year; }
public float total() {
return math+english+computer; }
public void output() {
System.out.print(this.name+" "+this.age()+" 岁 ");
System.out.println(math+" "+english+" "+computer+" "+total());
}
public static void main(String args[]) {
student s1=new student("张军 ",1982,90,84,78);
s1.output(); }}
```

53、定义一个抽象类，并定义它的两个非抽象子类。

答案：

```
abstract class A {
public int r;
```

```

abstract public int area(); }
class B extends A {
public int r ;
B(int r) {this.r=r;}
public int area() {return 3*r*r;} }
class C extends A {
public int width,height;
C (int w, int h) {
width=w;
height=h; }
public int area() {
return width*height; } }

```

54、定义接口 A，接口内有两个方法： method1(int x) 和 method2(int x,int y)，它们的返回值类型均为 int。请编写一个 Applet，并使其实现接口 A：令方法 method1(int x)的功能是求 5 的 x 次方、方法 method2(int x,int y) 的功能是求两参数的最大值。在 paint() 方法中调用这两个方法，显示 method1(2)和 method2(2,8)的结果。

答案：

```

public class MyClass implements A {
public void paint() {
System.out.println (" 调用 method1(2)的结果： "+method1(2));
System.out.println (" 调用 method2(2,8)的结果： "+method2(2,8));
}
public int method1(int x) {
int result=1;
for(int i=1;i<=x;i++)
result*=5;
return result; }
public int method2(int x,int y) {
return x>y?x:y; } }
interface A {
public abstract int method1(int x);
public abstract int method2(int x,int y);
}

```

55、定义名为 VolumeArea 的抽象类，在其中定义圆周率 的值为 3.14159，并定义两个抽象方法 volume(double r)和 area(double r)，它们的返回值类型均为 float。再定义以类 VolumeArea 为父类的子类 VolumeAndArea，在该子类中实现父类中的抽象方法：方法 volume(double r) 的功能是求半径为 r 的球的体积（公式为： $\frac{4}{3}\pi r^3$ ）；方法 area(double r)的功能是求半径为 r 的圆的面积（公式为：

2)。请编写一个 Application , 在其主类中定义一个 VolumeAndArea 类的对象 x , 通过对象 x 求半径为 r(r 的值由命令行给定)的球的体积及该球最大切面的圆的面积, 并输出计算结果。

答案 :

```
public class Class1 {
static VolumeAndArea x=new VolumeAndArea( );
public static void main(String[] args) {
float r=Float.valueOf(args[0]).floatValue( );
System.out.println(" 半径为 "+r+" 的球的体积 =" +x.volume(r));
System.out.println(" 半径为 "+r+" 的圆的面积 =" + x.area(r));
}}
abstract class VolumeArea {
static final float myPI=3.14159f;
public abstract float volume(double r);
public abstract float area(double r); }
class VolumeAndArea extends VolumeArea {
public float volume(double r) {
return (float)(4*myPI*r*r*r/3); }
public float area(double r){
return (float)(myPI*r*r); } }
```

56 . 定义一个图形的抽象类 , 具有求面积和画图形的方法 , 再定义点、线、圆的类继承这个抽象类实现它的方法。

答案 :

```
abstract class Graphic{
public static final double PI = 3.1415926;
abstract double area();
abstract void draw(); }
class Point extends Graphic {
protected double x, y;
public Point(double x, double y) {
this.x = x;
this.y = y; }
double area() { return 0; }
void draw() { // 在此模拟实现画一个点
System.out.println("Draw a point at (" +x+", "+y+""); }
public String toString(){
return "(" +x+", "+y+""); } }
class Line extends Graphic{
protected Point p1, p2;
public Line(Point p1, Point p2){
```

```

this.p1 = p1;
this.p2 = p2; }
double area(){ return 0; }
void draw(){
// 在此模拟实现画一条线
System.out.println("Draw a line from "+p1+" to "+p2);
}}
class Circle extends Graphic{
protected Point o;
protected double r;
public Circle(Point o, double r) {
this.o = o;
this.r = r; }
double area() {
return PI * r * r; }
void draw() {
// 在此模拟实现画一个圆
System.out.println("Draw a circle at "+o+" and r="+r);
}}
public class GraphicDemo{
public static void main(String []args){
Graphic []g=new Graphic[3];
g[0]=new Point(10,10);
g[1]=new Line(new Point(10,10),new Point(20,30));
g[2]=new Circle(new Point(10,10),4);
for(int i=0;i<g.length;i++){
g[i].draw();
System.out.println("Area="+g[i].area());
}}}

```

57、请定义一个名为 Card 的扑克牌类，该类有两个 private 访问权限的字符串变量 face 和 suit ,分别描述一张牌的牌面值和花色。 定义 Card 类中的 public 访问权限的构造方法， 为类中的变量赋值； 定义 protected 访问权限的方法 getFace()，得到扑克牌的牌面值；定义 protected 访问权限的方法 getSuit()，得到扑克牌的花色；定义方法 toString()，返回表示扑克牌的花色和牌面值字符串。

若应用程序的 main 方法中，定义字符串数组 f 和 s；分别表示扑克牌的牌面值和花色；定义 52 个元素的 Card 类型数组 deck，用来存放 4 个花色的 52 张牌，如下所示、 string f[]={ “ A ”, ” 2 ”,? string } s[]={ 黑桃 ” 红桃 ” 梅花 ” 方块 ” } Card[] deck=new Card[52]; 使用 Card 类的构造方法给 deck 数组的 52 张牌赋值， 要求数组中先存

放黑桃花色的 A、2、3.....K ; 然后是红桃花色的 A、2、3.....K ; 梅花花色的 A、2、3.....K ; 方块花色的 A、2、3.....K ; 把所有的牌信息打印出来。

再编写模拟洗牌的程序段 , 即把数组 deck 中的扑克牌随机打乱存放程序 , 分发给 4 人。

```
public class Card {
    private String face, suit;
    public Card() {
        face = "";
        suit = "";
    }
    public Card(String face, String suit) {
        this.face = face;
        this.suit = suit;
    }
    public String getFace() { return face;}
    public void setFace(String face) { this.face = face;}
    public String getSuit() {return suit;}
    public void setSuit(String suit) { this.suit = suit;}
    public String toString() { return suit + face;}
    public static void main(String[] sd) {
        Card[] desk = new Card[52];
        String[] f = { "A", "2", "3", "4", "5", "6", "7", "8", "9", "10", "J",
            "Q", "K" };
        String[] s = { " 黑桃 ", " 红桃 ", " 梅花 ", " 方块 " };
        for (int i = 0; i < desk.length; i++) {
            desk[i] = new Card(f[i % 13], s[i / 13]);
            // System.out.println(desk[i].toString());
        }
        for (int first = 0; first < desk.length; first++) {
            for (int k = 0; k < desk.length - first; k++) {
                int second = (int) (Math.random() * (desk.length - first));
                Card temp = desk[first];
                desk[first] = desk[second];
                desk[second] = temp;
            }
        }
        for (int i = 0; i < desk.length; i++) {
            System.out.print(desk[i].toString() + " ");
            if (i == 12 || i == 25 || i == 38) {
                System.out.println();
            }
        }
    }
}
```

58、在 CS 枪战游戏中 , 有 Ak47 和 USP 两种枪 , 每种枪弹夹的子弹数和发射子弹数都不一样。设计一个枪战游戏程序 , 定义一个 Gun 类 ,

该类具有枪名称的属性 name，子弹数量的名称 num，枪手的编号 id 和记录当前枪手数量的属性 count(静态)，具有 num 和 name 属性的 getter 和 setter 方法和发射的方法 fire(); 一个构造方法为属性赋初始值，Gun.count++;// 计数器累加 this.id = Gun.count; name = "Gun"; num = 12; 设计一个 Ak47 类，由 Gun 类继承而来，每发射一次消耗 3 发子弹，每装一个弹夹，上 30 发子弹；设计一个 USP 类，由 Gun 类继承而来，每发射一次消耗 1 发子弹，每装一个弹夹，上 12 发子弹；设计一个主类，分别创建不同的 Ak47 和 USP 对象，对象调用类里面的方法，装子弹，联系发射 10 次，显示枪里面还有多少子弹？

```
package xsc;
public class Gun {
    protected String name;// 名字
    protected int num;// 子弹数量
    public int id;// 编号
    public static int count;// 计数器
    public Gun() {
        Gun.count++;// 计数器累加
        this.id = Gun.count;
        name = "Gun";
        num = 12;}
    public String getName() { return name; }
    public void setName(String name) { this.name = name;}
    public int getNum() { return num;}
    public void setNum(int num) {this.num=num;}
    public void Fire() {
    }
}
package xsc;
public class Ak47 extends Gun {
    @Override
    public void Fire() {
        this.num-=3;
        System.out.println(this.name+" 正在使用 Ak47 开枪 ");
    }
    @Override
    public void setNum(int num) {
        this.num=30*num;
    }
}
package xsc;
```

```

public class USP extends Gun {
    @Override
    public void Fire() {
        this.num--;
        System.out.println(this.name+" 正在使用  USP 开第枪 ");
    }
    @Override
    public void setNum(int num) {
        this.num=12*num;
    }
}

package xsc;
public class Test {
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        USP usp1 = new USP();
        Ak47 ak1 = new Ak47();
        USP usp2 = new USP();

        //Gun gun=new Gun();
        System.out.println("usp1.id="+usp1.id);
        System.out.println("usp2.id="+usp2.id);
        System.out.println("ak1.id="+ak1.id);
        System.out.println(" 总枪数 =" +Gun.count);
        usp1.setNum(1);
        ak1.setNum(1);
        Gun nowGun;
        nowGun=usp1;
        for(int i=0;i<10;i++){
            nowGun.Fire();
        }
        nowGun=ak1;
        for(int i=0;i<10;i++){
            nowGun.Fire();
        }
        System.out.println("usp1 此枪还有 "+usp1.getNum()+" 个子弹 ");
        System.out.println("ak1 此枪还有 "+ak1.getNum()+" 个子弹 ");
    }
}

```

- 59、编程实现矩形类，其中应包括计算矩形周长和面积的方法。
- (1) 使用矩形类，编程统计若干块土地的相关信息。由用户输入每块土地的长与宽（自己确定），程序将计算其面积并显示出来。
 - (2) 从矩形类派生一个子类、长方体类。对长方体类的操作是求表面积和体积。编程检查、运行所编写的长方体类。
- 60、编写一个类 Book，代表教材、
- (1) 具有属性、名称（ title ） 页数（ pageNum ），其中页数不能少于 200 页，否则输出错误信息，并赋予默认值 200 ；
 - (2) 具有方法、 detail，用来在控制台输出每本教材的名称和页数；
 - (3) 具有带参数的构造方法、用来完成对象的初始化工作，并在构造方法中完成对页数的最小值限制。

第七章 数据库的连接

一、选择题

- | | | | |
|-------------|--------------|-------------|--------------|
| 1---5 DCBCA | 6--10 CACAA | 11-15 CBDBA | 16--20 BDCBD |
| 21-25 CABCD | 26--30 CDDCD | 31--32 BC | |

二、填空题

- 1、 msbase.jar , mssqlserver.jar , msutil.jar
- 2、 next()
- 3、 Statement
- 4、 jdbc:microsoft:sqlserver
- 5、 java.sql
- 6、 127.0.0.1 , localhost
- 7、 absolute(int row)
- 8、 first()
- 9、 getRow()
- 10、 isFirst() , isLast()

三、判断题

- 1--10 T F F F F T F T T T

四、编程题

```
1、 import java. sql.*;
public class fieldlist {
public static void main(String args[]) {
try {
Class.forName("sun.jdbC odbC JdbcOdbcDriver");
Connection con=DriverManager.getConnection("jdbc:odbc:dsntest");
```



```

Statement stmt=con.createStatement();
ResultSet rs=stmt.executeQuery("select * from test");
for(int i=1;i<=rs.getMetaData().getColumnCount();i++) {
System.out.print(rs.getMetaData().getColumnName(i)+"\t");
}// 跟踪显示各个列的名称
rs.close();
stmt.close();
con.close();
}
catch(Exception e){
System.out.println(e);
}
}
}

```

```

2、 import java.sql.*;
public class ResultSetTest {
public static void main(String args[]) {
try {
Class.forName("sun.jdbc_odbC JdbcOdbcDriver");
Connection con=DriverManager.getConnection("jdbc:odbc:dsntest");
Statement stmt=con.createStatement();
ResultSet rs=stmt.executeQuery("select * from test");
for(int i=1;i<=rs.getMetaData().getColumnCount();i++) {
System.out.print(rs.getMetaData().getColumnName(i)+"\t");
}// 跟踪显示各个列的名称
while(rs.next()) { //跟踪显示各个列的值
for(int j=1; j<=rs.getMetaData().getColumnCount(); j++) {
System.out.print(rs.getObject(j)+"\t"); } }
rs.close();
stmt.close();
con.close(); }
catch(Exception e) {
System.out.println(e); }} }

```

```

3、 import java. sql.*;
public class dbinsert {
public static void main(String args[]) {
try {
Class.forName("sun.jdbc_odbC JdbcOdbcDriver");

```

```

Connection con=DriverManager.getConnection("jdbc:odbc:testdsn");
Statement stmt=con.createStatement();
ResultSet rs=stmt.executeQuery("select * from test");
stmt.executeUpdate("Insert into student(id,name,age,sex)
values(2,'huang',30,true)");
rs=stmt.executeQuery("select * from test");
System.out.println(" 更新后的结果 :  ");
while(rs.next()) {
System.out.print(rs.getString("id")+"\t");
System.out.print(rs.getString("name")+"\t");
System.out.print(rs.getString("age")+"\t");
System.out.println(rs.getString("sex")); }
rs.close();
stmt.close();
con.close(); }
catch(Exception e)
{System.out.println(e);} } }
4、 import java sql.*;
public class dbinsert {
public static void main(String args[]) {
try {
Class.forName("sun.jdbc odbc JdbcOdbcDriver");
Connection con=DriverManager.getConnection("jdbc:odbc:testdsn");
Statement stmt=con.createStatement();
ResultSet rs=stmt.executeQuery("select * from test");
stmt.executeUpdate("Delete from student where id=100");
rs=stmt.executeQuery("select * from test");
System.out.println(" 更新后的结果 :  ");
while(rs.next()) {
System.out.print(rs.getString("id")+"\t");
System.out.print(rs.getString("name")+"\t");
System.out.print(rs.getString("age")+"\t");
System.out.println(rs.getString("sex")); }
rs.close();
stmt.close();
con.close(); }
catch(Exception e)
{System.out.println(e);} } }
5、 import java sql.*;
public class dbinsert

```

```

{      public static void main(String args[])
{try {
Class.forName("sun.jdbc odbc JdbcOdbcDriver");
Connection con=DriverManager.getConnection("jdbc:odbc:testdsn");
Statement stmt=con.createStatement();
ResultSet rs=stmt.executeQuery("select * from test");
stmt.executeUpdate("Update student set name='张三 ' where id=101");
rs=stmt.executeQuery("select * from test");
System.out.println(" 更新后的结果：  ");
while(rs.next()) {
System.out.print(rs.getString("id")+"\t");
System.out.print(rs.getString("name")+"\t");
System.out.print(rs.getString("age")+"\t");
System.out.println(rs.getString("sex"));
}
rs.close();
stmt.close();
con.close(); }
catch(Exception e)
{System.out.println(e);}
}
}

```

6、 略

7、 略

第八章 多线程

一、填空题

1、 健壮的 多线程

2、 run()

3、 Thread Runnable

4、 10 1

5、 synchronized 、 wait() 、 notify()

6、 init() 、 start() 、 paint() 、 stop() destroy()

7、 并发

8、 线程

9、 线程不安全性

10、 死锁

11、 finally

12、 死锁 、 优先级倒置 、 饥饿

13、 数据竞争

- 14、 sleep ()

15、 synchronized

16、 抢先式调度

17、 false

18、 线程 代码

19、 对象流

20、 对象串行化

21、 事件源

22、 Thread

23、 main

24、 loadpicture

25、 sleep ()

26、 java.lang.Tnread

27、 可运行

28、 运行状态

29、 yield()

30、 run()

31、 Thread

32、 wait () notify ()

33、 数据

34、 run ()

35、 新建状态、 可运行状态、 运行状态、 终止状态。

二、 选择题

1--5 BBCCD 6—10 CBDAD 11--15 BDAAB 16--20 CBCCA

21--25 CBADD 26--30CCABB 31--35 DBDDB 36--40 CBCBD

41--45 ACDBB 46--50 BDACB 51--55 DCDCA 56--60 CCCBD

61--65 CBBA A 66--70 BABBA 71--75 ADBAA 76--80 BCBDD

81--85 CCCDB 86--90 AADAB 91--95 BBACA

96--100ABAAC

101--105 DCCDB 106-110 DABAB 111-115 CDBDC

三、 多项选择

1、 ACD 2、 ABCD 3、 ABC 4、 AB

5、 ABC 6、 AD 7、 ABCD

四、 判断题

1--10 T T T T T T F F F T 11--21 T F T T F F F T F F T

五、 程序分析题
- 396

1、输出结果： ACBBBD

2、答案、 run() 方法作为接口中的方法缺省是 public 的，题中缩小了范围。改正如下、

```
public class Test implements Runnable
{
    Thread t = new Thread(this);
    t.start();
    public void run()
    {
        System.out.println(" thread is running now ");
    }
}
```

3、该程序的运行结果为：

Current thread

After name change

3

2

1

六、程序设计题

1、有三个线程 ID 分别是 A、B、C,请有多线程编程实现，在屏幕上循环打印 10 次 ABCABC

```
public static void main(String[] args) {
    XunleiLock lock = new XunleiLock();
    new Thread(new XunleiPrinter("A", lock)).start();
    new Thread(new XunleiPrinter("B", lock)).start();
    new Thread(new XunleiPrinter("C", lock)).start();
}

class XunleiPrinter implements Runnable {
    private String name = "";
    private XunleiLock lock = null;
    private int count = 10;
    public XunleiPrinter(String name, XunleiLock lock) {
        this.name = name;
        this.lock = lock;
    }
    @Override
    public void run() {
        while(count > 0) {
            synchronized (lock) {
                if
```



```

        System.out.println(name + ":" + new
            Date(System.currentTimeMillis()));
        Thread.sleep(pauseTime);
    } catch (Exception e) {
        System.out.println(e);
    }
}
}
static public void main(String args[]) {
    TimePrinter tp1 = new TimePrinter(1000, "Fast Guy");
    tp1.start();
    TimePrinter tp2 = new TimePrinter(3000, "Slow Guy");
    tp2.start();
}
}

```

3、试着编写一个程序，实现 Runnable 接口，创建一个线程。

//通过 Runnable 接口创建的另外一个线程；

```
class thread2 implements Runnable
```

```
{ //自定义线程的 run() 方法；
```

```
    public void run()
```

```
{
```

```
        System.out.println("Thread2 is running , ");
```

```
}
```

```
}
```

//程序的主类

```
class Multi_Thread //声明主类；
```

```
{
```

```
    public static void main(String args[]) //声明主方法；
```

```
{
```

```
        thread1 threadone=new thread1(); //用 Thread 类的子
```

类创建线程；

```
        Thread threadtwo=new Thread(new thread2()); //用
```

Runnable 接口类的对象创建线程；

```
        threadone.start(); threadtwo.start(); //start() 方法启动线
```

程；

```
}
```

```
}
```

4、实现通过四个售票点发售某日某次列车的 100 张车票，一个售票点用一个线程表示。

```
public class ThreadDome1{
```

```

    public static void main(String[] args){
        ThreadTest t = new ThreadTest();
        t.start();
        t.start();
        t.start();
        t.start();
    }
}
class ThreadTest extends Thread{
    private int ticket = 100;
    public void run(){
        while(true){
            if(ticket > 0){
                System.out.println(Thread.currentThread().getName() +
                    "is saling ticket" + ticket--);
            }else{
                break;
            }
        }
    }
}

public class ThreadDemo1{
    public static void main(String[] args){
        new ThreadTest().start();
        new ThreadTest().start();
        new ThreadTest().start();
        new ThreadTest().start();
    }
}
class ThreadTest extends Thread{
    private int ticket = 100;
    public void run(){
        while(true){
            if(ticket > 0){
                System.out.println(Thread.currentThread().getName() +
                    " is saling ticket" + ticket--);
            }else{
                break;
            }
        }
    }
}

```



```
}  
}
```

5、CrazyCoder 工作室为例，现在 6 个工作室成员因工作需要，工作室成员需要做一个项目，因为是多个人做，设计用多线程来做。首先定义成员类 UserInfo(表示成员信息),还有一个方法 Work() 表示工作。

```
public class UserInfo {  
    private string _name;  
        /// 用户名称  
    public string Name  
    {  
        get { return _name; }  
        set { _name = value; }  
    }  
}
```

```
///
```

法
/// 线程执行具体方法 注意:必须是 Void 类型，不能为静态方

```
/// by crzycoder http://www.crazycoder.cn
```

```
///
```

```
public void Work()  
{  
    //这里写要做的工作代码  
}
```

```
}
```

现在开始写线程代码

```
public class WorkThread  
{
```

```
    ///
```

```
    /// 线程执行代码
```

```
    /// by crzycoder http://www.crazycoder.cn
```

```
    ///
```

```
public void CrazyCoderThread()  
{
```

```
    //假设工作室 6 名成员开始工作
```

```
    for (int i = 1; i <= 6; i++)
```

```
    {
```

```
        UserInfo u = new UserInfo();
```

```
        u.Name = " 疯狂工作室成员 " + i.ToString();
```

```
        Thread t = new Thread(u.Work);
```

```
        t.Start();
```

```
    }
```

```
}
```

```
}
```

6、用 JAVA SOCKET 编程，实现简单的 Echo 功能。 如、 客户端从键盘输入 hi (当用户输入 exit 退出程序)，服务端响应为 hi(服务器要求为多线程)

解、服务器程序、

```
import java.io.*;
```

```
import java.net.*;
```

```
public class MyServer extends Thread{
```

```
private Socket cq ;
```

```
public MyServer(Socket cq){
```

```
this.cq = cq;
```

```
}
```

```
public void run(){
```

```
try{
```

```
        BufferedReader
```

```
        in=new
```

```
        BufferedReader(new
```

```
InputStreamReader(cq.getInputStream()));
```

```
    PrintWriter out=new PrintWriter(cq.getOutputStream());
```

```
    while(true){
```

```
        String str=in.readLine();
```

```
        System.out.println(str);
```

```
        out.println("message: "+str);
```

```
        out.flush();
```

```
        if(str.equals("exit"))
```

```
            break;
```

```
    }
```

```
}
```

```
catch(IOException e)
```

```
{
```

```
    System.out.println(e.getMessage());
```

```
}
```

```
}
```

```
public static void main(String[] args) throws IOException{
```

```
    ServerSocket server=new ServerSocket(8009);
```

```
    while(true)
```

```
    {
```

```
        Socket s=server.accept();
```

```
        new MyServer(s).start();
```

```
    }
```

```

    }
}
客户端程序、
import java.net.*;
import java.io.*;
public class MyClient{
    public static void main(String[] args)throws Exception
    {
        Socket server=new Socket("localhost",8009);
        BufferedReader in=new      BufferedReader(new
InputStreamReader(server.getInputStream()));
        PrintWriter out=new PrintWriter(server.getOutputStream());
        BufferedReader br=new      BufferedReader(new
InputStreamReader(System.in));
        while(true)
        {
            String str=br.readLine();
            out.println(str);
            out.flush();
            if(str.equals("exit")){
                break;
            }
            System.out.println(in.readLine());
        }
        server.close();
    }
}

```

7、提货单真实数据一个调用者在调用耗时操作 ,不能立即返回数据时 ,先返回一个提货单 .然后在过一断时间后凭提货单来获取真正的数据 .去蛋糕店买蛋糕 , 不需要等蛋糕做出来 (假设现做要很长时间) , 只需要领个提货单就可以了 (去干别的事情) , 等到蛋糕做好了 , 再拿提货单取蛋糕就可以了。

```

public class Host {
    public Data request(final int count, final char c) {
        System.out.println("request(" + count + ", " + c + ") BEGIN");
        // (1) 建立 FutureData 的实体
        final FutureData future = new FutureData();
        // (2) 为了建立 RealData 的实体 , 启动新的线程
        new Thread() {

```

```

public void run() {
//在匿名内部类中使用    count、future、c。
RealData realdata = new RealData(count, c);
future.setRealData(realdata);
}
}.start();
System.out.println("request(" + count + ", " + c + ") END");
public class RealData implements Data { private final String content;
public RealData(int count, char c) {
System.out.println("making    RealData(" + count + ", " + c + ")
BEGIN");
char[] buffer = new char[count];
for (int i = 0; i < count; i++) {
buffer[i] = c;
try {
Thread.sleep(1000);
} catch (InterruptedException e) {
}
}
System.out.println("making    RealData(" + count + ", " + c + ")
END");
this.content = new String(buffer);
}
public String getContent() {
return content;
}
}
public class FutureData implements Data {
private RealData realdata = null;
private boolean ready = false;
public synchronized void setRealData(RealData realdata) {
if (ready) {
return; // 防止 setRealData 被调用两次以上。
}
this.realdata = realdata;
this.ready = true;
notifyAll();
}
public synchronized String getContent() {
while (!ready) {

```

```

try {
    wait();
} catch (InterruptedException e) {
}
}
return realdata.getContent();
}
}

```

顾客做完自己的事情后，会拿着自己的“提货单”来取蛋糕、

```
System.out.println("data1 = " + data1.getContent());
```

这时候如果蛋糕没做好，就只好等了、

```

while (!ready) { try {
    wait();
} catch (InterruptedException e) {
}
}

```

//等做好后才能取到

```
return realdata.getContent();
```

8、一个从 1 加到 10,第 2 个从 11 加到 20...线程同步执行 ,最后再把十个线程结果相加

```

public class Test extends Thread {
    static int n = 0;
    private int startNum = 0 ;
    public Test (int sn)
    {
        this.startNum = sn ;
    }
    public static synchronized void addSum (int num)
    {
        n += num ;
    }
    public static void main(String[] args) {
        Thread [] thList = new Thread [10] ;
        for (int i = 0; i < 10; i ++) {
            thList [i] = new Test(i * 10 + 1) ;
            thList [i].start();
        }
        for (int i = 0 ; i < 10 ; i ++)
        {
            thList [i].join () ;
        }
    }
}

```

```
System.out.println ("Sum is : " + n) ;  
}
```

```
public void run() {  
int sum = 0 ;  
for (int i = 0; i < 10; ++i) {  
sum += sn + i ;  
}  
addSum (sum) ;  
}  
}
```

9、一个多线程的摇奖器的例子

在第一个类中定义三个 void 方法用于产生随机数

```
import java.util.Random;  
public class YaoJiangQi {  
private int baseint=9;  
public void winningone()  
{  
Random random=new Random();  
random.setSeed(11l);//这里可以用不同的种子 (long)  
int[] numberline=new int[7];  
System.out.print(" 一等奖号码为、 ");  
for(int i=0;i<=6;i++)  
{  
if(i==0)  
numberline[i]=random.nextInt(baseint)+1;  
else  
{  
for(int j=0;j<i;j++)  
{  
if(numberline[i]==numberline[j])  
numberline[i]=random.nextInt(baseint)+1;  
}  
}  
System.out.print(numberline[i]);  
}  
System.out.print("\n");  
}  
public void winningtwo()  
{
```

```

Random random=new Random();
random.setSeed(221);
int[] numberline=new int[7];
System.out.print(" 二等奖号码为、    ");
for(int i=0;i<=6;i++)
{
    {
        if(i==0)
            numberline[i]=random.nextInt(baseint)+1;
        else
        {
            for(int j=0;j<i;j++)
            {
                if(numberline[i]==numberline[j])
                    numberline[i]=random.nextInt(baseint)+1;
            }
        }
    }
    System.out.print(numberline[i]);
}
System.out.print("\n");
}

public void winningthree()
{
    Random random=new Random();
    random.setSeed(331);
    int[] numberline=new int[7];
    System.out.print(" 三等奖号码为、    ");
    for(int i=0;i<=6;i++)
    {
        {
            if(i==0)
                numberline[i]=random.nextInt(baseint)+1;
            else
            {
                for(int j=0;j<i;j++)
                {
                    if(numberline[i]==numberline[j])
                        numberline[i]=random.nextInt(baseint)+1;
                }
            }
        }
    }
}

```

```

    }
    }
    System.out.print(numberline[i]);
}
}
}
}

```

定义三个线程

```

public class Yaoone extends Thread{
public void run()
{
    YaoJiangQi yao1=new YaoJiangQi();
    yao1.winningone();
}
}
public class Yaotwo extends Thread{
public void run()
{
    YaoJiangQi yao2=new YaoJiangQi();
    yao2.winningtwo();
}
}
public class Yaothree extends Thread{
public void run()
{
    YaoJiangQi yao3=new YaoJiangQi();
    yao3.winningthree();
}
}
}

```

测试

```

public class TryYaoJiangQi {
    public static void main(String[] args) {
        try{
            Thread thread1=new Yaoone();
            Thread thread2=new Yaotwo();
            Thread thread3=new Yaothree();
            thread1.setPriority(3);
            thread2.setPriority(2);
            thread3.setPriority(1);
            thread1.start();
            thread2.start();

```



```

        thread3.start();
    }
    catch(Exception e)
    {
        e.printStackTrace();
    }
}
}

```

10、编写一个多线程程序实现如下功能、线程 A 和线程 B 分别在屏幕上显示信息 “ A start ” “ B start ” 后 调用 wait 等待；线程 C 开始后调用 sleep 休眠一段时间，然后调用 notifyall，使线程 A 和线程 B 继续运行。线程 A 和线程 B 恢复运行后输出信息 “ A END ” “ B END ” 后结束，线程 C 在判断线程 A 线程 B 结束后，自己也结束运行。

、 ThreadDemo.java

```

-----
public class ThreadDemo {
    public static void main(String[] args) {
        Object obj = new Object();
        ThreadA ta = new ThreadA(obj);
        ThreadB tb = new ThreadB(obj);
        ThreadC tc = new ThreadC(obj, ta, tb);
        ta.start();
        tb.start();
        tc.start();
    }
}

```

// 线程 A

```

class ThreadA extends Thread {
    private Object obj;
    public ThreadA(Object obj) {
        this.obj = obj;
    }
    public void run() {
        System.out.println("A Start");
        synchronized(obj) {
            try {
                obj.wait();
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    }
}

```

```

        }
    }
    System.out.println("A End");
}

// 线程 B
class ThreadB extends Thread {
    private Object obj;
    public ThreadB(Object obj) {
        this.obj = obj;
    }
    public void run() {
        System.out.println("B Start");
        synchronized(obj) {
            try {
                obj.wait();
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
        System.out.println("B End");
    }
}

```

```

// 线程 C
class ThreadC extends Thread {
    private Object obj;
    private ThreadA ta;
    private ThreadB tb;
    public ThreadC(Object obj, ThreadA ta, ThreadB tb) {
        this.obj = obj;
        this.ta = ta;
        this.tb = tb;
    }
    public void run() {
        System.out.println("C Start");
        synchronized(obj) {
            try {
                sleep(1000);
            }
        }
    }
}

```

```

        obj.notifyAll();
    } catch (Exception e) {
        e.printStackTrace();
    }
}
while(ta.isAlive() && tb.isAlive()); // 判断 A 线程和 B 线程是否已经结束
System.out.println("C End");
}

```

11、设计 4 个线程，其中两个线程每次对 j 增加 1，另外两个线程对 j 每次减少 1。写出程序。

```

public class ThreadTest1{
    private int j;
    public static void main(String args[]){
        ThreadTest1 tt=new ThreadTest1();
        Inc inc=tt.new Inc();
        Dec dec=tt.new Dec();
        for(int i=0;i<2;i++){
            Thread t=new Thread(inc);
            t.start();
            t=new Thread(dec);
            t.start();
        }
        private synchronized void inc(){
            j++;
            System.out.println(Thread.currentThread().getName()+"-inc:"+j);
        }
        private synchronized void dec(){
            j--;
            System.out.println(Thread.currentThread().getName()+"-dec:"+j);
        }
        class Inc implements Runnable{
            public void run(){
                for(int i=0;i<100;i++){
                    inc();
                }
            }
        }
    }
}

```



```

    t4.print();
}
}

```

13、要实现一个 socket 服务器

接受客户端 1 发来的消息，客户端 1 每隔一段时间就会发送消息给服务器，服务器将消息存储数据库，同时发送到客户端 2 用于实时显示，是通过两个端口分别与两个客户端进行通信，一收一发还是仅监听一个端口，收消息，然后再调用方法进行发送。

```

import java.awt.*;
import java.awt.event.*;
import java.applet.*;
import java.net.*;
import java.io.*;
import java.util.*;

```

```

public class ChatClient extends Applet {

```

```

    TextField tfName = new TextField(15); // 姓名输入文本域
    Button btConnect = new Button(" 连接 "); // 连接按钮
    Button btDisconnect = new Button(" 断开连接 ");
    TextArea tfChat = new TextArea(8,27); // 显示聊天信息文本框
    Button btSend = new Button(" 发送 ");
    TextField tfMessage = new TextField(30); // 聊天输入
    java.awt.List list1 = new java.awt.List(9); // 显示在线用户信息

```

```

    Socket socket=null; // 连接端口
    PrintStream ps=null; // 输出流
    Listen listen=null; // 监听线程

```

```

    public void init() { //Applet 初始化
        tfChat.setEditable(false); // 设置信息显示框为不可编辑
        Panel panel1 = new Panel(); // 实例化面板
        Panel panel2 = new Panel();
        Panel panel3 = new Panel();
        tfChat.setBackground(Color.white); // 设置组件背景颜色
        panel1.setBackground(Color.orange);
        panel2.setBackground(Color.pink);
        panel3.setBackground(Color.orange);
        panel1.add(new Label(" 姓名、 ")); // 增加组件到面板上
        panel1.add(tfName);
    }

```

```

panel1.add(btConnect);
panel1.add(btDisconnect);
panel2.add(tfChat);
panel2.add(list1);
panel3.add(new Label(" 聊天信息 "));
panel3.add(tfMessage);
panel3.add(btSend);
setLayout(new BorderLayout()); // 设置 Applet 布局管理器
add(panel1, BorderLayout.NORTH); // 增加面板到 Applet 上
add(panel2, BorderLayout.CENTER);
add(panel3, BorderLayout.SOUTH);
}

public boolean action(Event evt, Object obj){ // 事件处理
try{
if(evt.target==btConnect){ // 点击连接按钮
if (socket==null){
socket=new Socket(InetAddress.getLocalHost(),5656); // 实例化一个套接字
ps=new PrintStream(socket.getOutputStream()); // 获取输出流
StringBuffer info=new StringBuffer("INFO: ");
String
userinfo=tfName.getText()+":"+InetAddress.getLocalHost().toString();
ps.println(info.append(userinfo)); // 输出信息
ps.flush();
listen=new Listen(this,tfName.getText(),socket); // 实例化监听线程
listen.start(); // 启动线程
}
}
else if(evt.target==btDisconnect){ // 点击断开连接按钮
disconnect(); // 调用断开连接方法
}
else if(evt.target==btSend){ // 点击发送按钮
if(socket!=null){
StringBuffer msg=new StringBuffer("MSG: ");
String msgtxt=new String(tfMessage.getText());
ps.println(msg.append(tfMessage.getText())); // 发送信息
ps.flush();
}
}
}
}

```

```

}
catch (Exception ex){
ex.printStackTrace(); // 输出错误信息
}
return true;
}
public void disconnect() { // 断开连接方法
if(socket!=null){
ps.println("QUIT"); // 发送信息
ps.flush();
}
}

class Listen extends Thread{ // 监听服务器传送的信息
String name=null; // 用户名
BufferedReader reader ; //输入流
PrintStream ps=null; // 输出流
Socket socket=null; // 本地套接字
ChatClient client=null; // 客户端 ChatClient 实例

public Listen(ChatClient p,String n,Socket s) {
client=p;
name=n;
socket=s;

try{
reader = new BufferedReader(new
InputStreamReader(s.getInputStream())); // 获取输入流
ps=new PrintStream(s.getOutputStream()); // 获取输出流

}
catch(IOException ex){
client.disconnect(); // 出错则断开连接
ex.printStackTrace(); // 输出错误信息
}
}

public void run(){
String msg=null;
while(socket!=null){

```

```

try{
msg=reader.readLine(); // 读取服务器端传来信息
}
catch(IOException ex){
client.disconnect(); // 出错则断开连接
ex.printStackTrace(); // 输出错误信息
}
if (msg==null) { // 从服务器传来的信息为空则断开此次连接
client.listen=null;
client.socket=null;
client.list1.removeAll();
return;
}
StringTokenizer st=new StringTokenizer(msg,":"); // 分解字符串
String keyword=st.nextToken();

if(keyword.equals("newUser")) { // 新用户连接信息
client.list1.removeAll(); // 移除原有用户名
while(st.hasMoreTokens()) { // 取得目前所有聊天室用户名
String str=st.nextToken();
client.list1.add(str); // 增加到用户列表中
}
}
else if(keyword.equals("MSG")) { // 聊天信息
String usr=st.nextToken();
client.tfChat.append(usr); // 增加聊天信息到信息显示框
client.tfChat.append(st.nextToken("\0"));
client.tfChat.append("\n");
}
else if(keyword.equals("QUIT")) { // 断天连接信息
System.out.println("Quit");
try{
client.listen=null;
client.socket.close(); // 关闭端口
client.socket=null;
}
catch(IOException e){}
client.list1.removeAll(); // 移除用户列表
return;
}
}

```


14、假定一个缓冲区只能保存一条信息。即缓冲区只有两个状态、有信息或没有信息。

```
void reader_function ( void );
void writer_function ( void );
char buffer;
int buffer_has_item=0;
pthread_mutex_t mutex;
struct timespec delay;
void main ( void ){
pthread_t reader;
/* 定义延迟时间 */
delay.tv_sec = 2;
delay.tv_nsec = 0;
/* 用默认属性初始化一个互斥锁对象 */
pthread_mutex_init (&mutex,NULL);
pthread_create(&reader, pthread_attr_default, (void *)&reader_function,
NULL);
writer_function( );
}
void writer_function (void){
while(1){
/* 锁定互斥锁 */
pthread_mutex_lock (&mutex);
if (buffer_has_item==0){
buffer=make_new_item( );
buffer_has_item=1;
}
/* 打开互斥锁 */
pthread_mutex_unlock(&mutex);
pthread_delay_np(&delay);
}
}
void reader_function(void){
while(1){
pthread_mutex_lock(&mutex);
if(buffer_has_item==1){
consume_item(buffer);
buffer_has_item=0;
}
pthread_mutex_unlock(&mutex);
```

```
pthread_delay_np(&delay);
}
}
```

15、模拟火车票联网售票系统、多个线程同时出票，保证每张出票的编号连续且不重复。

```
public class Demo
{
    public static void main(String args )
    {
        SellTicket st1=new SellTicket();
        st1.setName("线程 1");
        SellTicket st2=new SellTicket();
        st2.setName("线程 2");
        SellTicket st3=new SellTicket();
        st3.setName("线程 3");
        st1.start();
        st2.start();
        st3.start();
    }
}

public class SellTicket extends Thread
{
    private static int id=0;
    public void run() {
        for(int i=0;i<20;i++) {
            sell();
        }
    }
    public synchronized void sell() {
        id++;
        System.out.println(currentThread().getName()+" 买票 "+id);
    }
}
```

16、JAVA 多线程编程。 用 JAVA 写一个多线程程序，如写四个线程，二个加 1，二个对一个变量减一，输出。

```
package First;

import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;
```

```

import java.io.PrintStream;

import com.sun.media.sound.Toolkit;

public class test {
    static int num = 0;
    public static void main(String[] args) {
        System.out.println("now:" + num );
        new test().start();
    }

    class Thd implements Runnable{// 创建线程

        public void run() {
            num ++;
            System.out.println( num + "\n");
        }
    }

    class Thd1 implements Runnable{// 创建线程
        public void run() {
            num --;
            System.out.println( num + "\n");
        }
    }

    public void start() {
        for ( int i = 0;i < 2; i ++){
            Thd th1 = new Thd();
            new Thread(th1).start();// 线程启动
        }
        for ( int i = 0;i < 2; i ++){
            Thd1 th = new Thd1();
            new Thread(th).start();// 线程启动
        }
    }
}

```

17、编写一个多线程程序实现如下功能、线程 A 和线程 B 分别在屏幕上显示“。。。start”后，调用 wait 等待；线程 C 开始后调用 sleep 休眠一段时间，然后调用 notifyAll ，使线程 A 和线程 B 继续运行。线程

A 和线程 B 恢复运行后输出信息 “。。 end ” 后结束，线程 C 判断线程 A 和 B 结束后，C 自己也结束。

```
public class ABC {

    boolean available=false;
    String name;
    int count=0;
    public ABC(String name){
        this.name=name;
    }
    public synchronized void st(){
        System.out.println("... start.");
        count++;
        if(count>1){
            available=true;
        }
        try{
            wait();
        }catch(Exception e){}
        System.out.println("... end.");
    }
    public synchronized void end(){
        System.out.println("... end.");
    }
    public synchronized void wake() {
        notifyAll();
    }

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        ABC x=new ABC("haha");
        TestThread a=new TestThread(x);
        TestThread b=new TestThread(x);
        C c=new C(x);
        (new Thread(a)).start();
        (new Thread(b)).start();
        (new Thread(c)).start();
    }
}
class TestThread implements Runnable{
```

```

ABC abc;
public TestThread(ABC abc){
    this.abc=abc;
}
public void run(){
    if(!abc.available){
        abc.st();
    }
    else abc.end();

}
}
class C implements Runnable{
    ABC abc;
    public C(ABC abc){
        this.abc=abc;
    }
    public void run(){
        try{
            Thread.sleep(1000);
// System.out.println("sjlfsj jfsdl j");      // 验证是否执行到这里
        }catch(InterruptedException e){};
        abc.wake();
    }
}

```

18、有一个南北向的桥，只能容纳一个人，现桥的两边分别有 10 人和 12 人，编制一个多线程程序让这些人到达对岸，每个人用一个线程表示，桥为共享资源。在过桥的过程中显示谁在过桥其其走向。

```
package cn.thread;
```

```

public class Through_out_bridge {

    public static void main(String[] args) {
        Bridge b = Bridge.getInstance();// 实例化桥
        //实例化左端 9 个人，此时所有人都不能过桥，桥的可以状态标志为
        不可以用
        for (int i = 1; i <= 9; i++) {
            Thread t = new Thread(new Person(false, i, b));
            t.start();
        }
    }
}

```

//实例化右端 12 个人，此时所有人都不能过桥， 桥的可以状态标志为不可以用

```
for( int i=1 ;i<=12;i++)
{
    Thread t = new Thread(new Person(true,i,b));
    t.start();
}
//桥的可用状态给左端第 10 个人，可以自定义给谁
b.state = true;
Thread t = new Thread(new Person(false, 10, b));
}
}
```

class Person implements Runnable {

```
    public Bridge bridge;// 桥
    private String hand;// 在桥哪一端
    int No;// 序号
```

```
    public Person(boolean side, int i, Bridge bridge) {
```

```
        this.No = i;
        this.bridge = bridge;
        if(bridge.state)
        {
            System.out.println(i+" 已经过桥。 ");
        }
        if (side) {
            this.hand = new String(" 右");
        } else {
            this.hand = new String(" 左");
        }
    }
}
```

//过桥方法

```
    public synchronized void through() throws InterruptedException {
```

```
        if (bridge.state) {
            System.out.println(hand+" 边第 "+No + " 在过桥 ");
        }
    }
}
```

```

        bridge.open( No);
    } else {
        bridge.lock(No);

    }

}

    public void run() {
try {
    Thread.sleep(1000);
//    System.out.println(No+hand+"  边已经过桥 !");
    through();
} catch (InterruptedException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}
    }
}

class Bridge {
    //可用状态判断  true、可用
    public boolean state = false;
    //自行实例化
    public static Bridge getInstance() {
return new Bridge();
    }

    public synchronized void open(int i) throws InterruptedException {
if (state) {
    Thread.sleep(1000);
    this.state=true;
    notify();
}
    }

    public synchronized void lock(int i) throws InterruptedException {
if (!state) {
    this.state=false;
    System.out.println(i + "  在等待 .");
    wait();
}
    }
}

```

```
}
```

19、利用多线程技术编写 Applet 程序，其中包含一个滚动的字符串。字符串从左向右运动，当所有的字符都从屏幕的右边消失后，字符串重新从左边出现并继续向右移动。

答案：源程序： Xianch .java

```
import java.awt.* ;
import java.applet.* ;
public class Xianch extends Applet implements Runnable
{
    final String rollingMessage = "SAILING , SAILING, I AM SAILING..." ;
    Thread m_Draw = null ;
    int beginX ;
    public void init()
    {
        m_Draw = new Thread(this) ;
    }
    public void paint(Graphics g)
    {
        g.drawString(rollingMessage, beginX, 40) ;
    }
    public void start()
    {
        m_Draw.start() ;
        try{
            Thread.sleep(50) ;
        }catch(InterruptedException e){}
    }
    public void stop()
    {
        //if(m_Draw.isAlive())
        //m_Draw.stop() ;
    }
    public void run()
    {
        try
        {
            while(true)
            {
                beginX = ++beginX % getWidth() ;
                repaint() ;
            }
        }
    }
}
```



```

ThreadD sleep(100) ;
}
}
catch(InterruptedException e)
{
}
}
}
}

```

20、编写一个应用程序，在线程同步的情况下来实现“生产者 消费者”问题。

答案：源程序：

```

public class ConsumeInteger extends Thread {
private HoldIntegerSynchronized cHold ;
public ConsumeInteger( HoldIntegerSynchronized h )
{
super( "ConsumeInteger" ) ;
cHold = h ;
}
public void run()
{
int val, sum = 0 ;
do {
// sleep for a random interval
try {
ThreadD sleep( (int) ( Math.random() * 3000 ) ) ;
}
catch( InterruptedException e ) {
System.err.println( e.toString() ) ;
}
val = cHold getSharedInt() ;
sum += val ;
}
while ( val != 10 ) ;
System.err.println( getName() + " retrieved values totaling: " + sum +
"\nTerminating " + getName() ) ;
}
}
public class HoldIntegerSynchronized {
private int sharedInt = -1 ;

```

```

private boolean writeable = true ; // condition variable
public synchronized void setSharedInt( int val )
{
while ( !writeable ) { // not the producer's turn
try {
wait() ;
}
catch ( InterruptedException e ) {
e.printStackTrace() ;
}
}
System.err.println(   ThreadD   currentThread().getName()   +   "   setting
sharedInt to " +
val ) ;
sharedInt = val ;
writeable = false ;
notify() ; // tell a waiting thread to become ready
}
public synchronized int getSharedInt()
{
while ( writeable ) { // not the consumer's turn
try {
wait() ;
}
catch ( InterruptedException e ) {
e.printStackTrace() ;
}
}
writeable = true ;
notify() ; // tell a waiting thread to become ready
System.err.println(   ThreadD   currentThread().getName()   +   "   retrieving
sharedInt value
" + sharedInt ) ;
return sharedInt ;
}
}
public class ProduceInteger extends Thread {
private HoldIntegerSynchronized pHold ;
public ProduceInteger( HoldIntegerSynchronized h )
{

```

```

super( "ProduceInteger" ) ;
pHold = h ;
}
public void run()
{
for ( int count = 1 ;
count <= 10 ;
count++ ) {
// sleep for a random interval
try {
Thread.sleep( (int) ( Math.random() * 3000 ) ) ;
}
catch( InterruptedException e ) {
System.err.println( e.toString() ) ;
}
pHold.setSharedInt( count ) ;
}
System.err.println( getName() + " finished producing values" +
"\nTerminating " +
getName() ) ;
}
}
public class SharedCell {
public static void main( String args[] )
{
HoldIntegerSynchronized h = new HoldIntegerSynchronized() ;
ProduceInteger p = new ProduceInteger( h ) ;
ConsumeInteger c = new ConsumeInteger( h ) ;
p.start() ;
c.start() ;
}
}

```

21、利用多线程设计一个程序，同时输出 50 以内的奇数和偶数，以及当前运行的线程数。

答案：

```

Public class threadprint extends thread
{
Int k=0;
Public threadprint(string name,int k)
{

```

```

super(name);
this.k=k;
}
Public void run()
{
Int i=k;
System.out.print(getname()+" : ");
While(i<50)
{
System.out.print(i+" ");
i+=2;
}
System.out.println(getname()+" end! ");
}
Public static void main (string[] args)
{
Threadprint t1=new threadprint(" thread1 ",1);
Threadprint t2=new threadprint(" thread2 ",2);
T1.start();
T2.start();
System.out.println(" activecount=" +t2.activecount());
}
}

```

22、 编写采用原子量类的多线程程序模拟银行用户存取钱的并发程序

```

import java.util.concurrent.atomic.AtomicLong;
class AtomicAccount {
    AtomicLong balance;
    public AtomicAccount(long money) {
        balance = new AtomicLong(money);
        System.out.println("Totle Money: " + balance);
    }
    publicvoid deposit(long money) {
        balance.addAndGet(money);
    }
    publicvoid withdraw(long money, int delay) {
        long oldvalue = balance.get();
        if (oldvalue >= money) {
            try {
                Thread.sleep(delay);
            }
        }
    }
}

```

```

        } catch (InterruptedException e) {
            e.printStackTrace();
        }
        if (balance.compareAndSet(oldvalue, oldvalue - money)) {
            System.out.println(Thread.currentThread().getName()
                + " withdraw " + money + " successful!" +
balance);
        } else {
            System.out.println(Thread.currentThread().getName()
                + "thread concurrent, withdraw failed!" +
balance);
        }
    } else {
        System.out.println(Thread.currentThread().getName()
            + " balance is not enough,withdraw failed!" +
balance);
    }
}
public long get() {
    return balance.get();
}
}

```

```

class AtomicAccountDeposit extends Thread {
    AtomicAccount account;
    int delay;

    public AtomicAccountDeposit(AtomicAccount account, int
delay) {
        this.account = account;
        this.delay = delay;
    }
    public void run() {
        account.deposit(100);
    }
}

```

```

public class AtomicAccountTest extends Thread {
    AtomicAccount account;
    int delay;
}

```

```

public AtomicAccountTest(AtomicAccount account, int delay) {
    this.account = account;
    this.delay = delay;
}
public void run() {
    account.withdraw(100, delay);
}
public static void main(String[] args) {
    AtomicAccount account = new AtomicAccount(100);
    AtomicAccountTest accountThread1 =
new AtomicAccountTest(account, 1000);
    AtomicAccountDeposit accountThread2 =
new AtomicAccountDeposit(account, 0);
    AtomicAccountTest accountThread3 = new
AtomicAccountTest(account, 500);
    accountThread1.start();
    accountThread2.start();
    accountThread3.start();
}
}

```

23、用线程实现动画

```

import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;
public class AnimatorCanvas extends Canvas implements Runnable {
    Image[] frames;
    int left, top;
    boolean alive;
    int interval;
    int currentFrame;
    int numFrames;
    MIDlet midlet;
    Thread thread;

    public AnimatorCanvas(MIDlet midlet, Image[] frames, int left, int top, int interval) {
        this.frames = frames;
        this.left = left;
        this.top = top;
        this.interval = interval;
    }
}

```

```

this.alive=true;
this.currentFrame=0;
this.numFrames=frames.length;
this.midlet=midlet;
thread=new Thread(this);
thread.start();
}
public AnimatorCanvas(MIDlet midlet, Image[] frames, int interval){
this(midlet, frames, 0, 0, interval);
this.left=(this.getWidth()-frames[0].getWidth())/2;
this.top=(this.getHeight()-frames[0].getHeight())/2;
}
public void paint(Graphics g){
g.setColor(0x00000000);
g.fillRect(0,0,getWidth(),getHeight());

g.drawImage(frames[currentFrame++],left,top,Graphics.LEFT|Graphics.TOP);
if(currentFrame>=numFrames){
currentFrame=0;
}
}
public void keyPressed(int keyCode){
if(getGameAction(keyCode)==FIRE){
alive=!alive;
}else{
thread=null;
((Animator)midlet).exit();
}
}
public void run(){
while(true){
if(alive){
repaint();
try{
Thread.sleep(interval);
}catch(InterruptedException e){
}
}
}
}
}

```

}
}

第九章 异常处理

一、选择题

- | | | | | | | |
|--------|-------|--------|-------|--------|-------|--------|
| 1--5 | CDAAC | 6--10 | DAAAB | 11--15 | AADBB | 16--20 |
| | CDAAD | | | | | |
| 21--25 | DBBCD | 26--30 | ADAAC | 31--35 | BCACD | 36--40 |
| | ACBCC | | | | | |
| 41--45 | AACDD | 46--49 | BAAD | | | |

二、填空题

- 1、 例外 程序运行 正常执行
- 2、 throws
- 3、 try-catch-finally throw/throws
- 4、 finally
- 5、 异常处理
- 6、 Exception
- 7、 抛出异常方法
- 8、 catch
- 9、 抛弃
- 10、 捕获
- 11、 public String toString () public String getMessage () public void printStackTrace ()
- 12、 NullPointerException
- 13、 catch catch finally
- 14、 Throwable
- 15、 throw
- 16、 try 多个
- 17、 catch
- 18、 多个 catch
- 19、 finally
- 20、 finally
- 21、 Exception
- 22、 ClassNotFoundException 运行时异常 NullPointerException
ArithmeticException ClassCastException
ArrayIndexOutOfBoundsException
- 23、 Exception 两
- 24、 调用栈
- 25、 抛出
- 26、 参数内容 空值 .
- 27、 匹配

- 28、 异常
- 29、 try-catch-finally
- 30、 finally
- 31、 Error Excetion
- 32、 对象
- 33、 捕获异常
- 34、 抛出
- 35、 异常抛出 异常捕获
- 36、 数组下标越界
- 37、 编译器
- 38、 ChecKedException
- 39、 Exception
- 40、 throws
- 41、 try、 catch、 finally 、 throw 、 throws
- 42、 覆盖
- 43、 Exception
- 44、 捕获
- 45、 catch
- 46、 异常 错误
- 47、 catch
- 48、 try
- 49、 不同方式
- 50、 Error Exception
- 51、 throws
- 52、 throws throws
- 53、 throws
- 54、 java.io

三、 判断题

- | | | | |
|-------|---------------------|-------|---------------------|
| 1—10 | T T T F T T F F F F | 11—20 | T F F F T T F T F F |
| 21—30 | T T T T F T T T F F | | |
| 31—35 | T F T F T | | |

四、 程序阅读题

1、 答案、

i=2

i=1

testEx2, catch exception

testEx2, finally; return value=false

testEx1, finally; return value=false

testEx, finally; return value=false

2、 答案、

in procedure, catch ArithmeticException: java.lang.ArithmeticException: /
by zero

3、数组下标越界

4、算术异常

5、catch(FileNotFoundException e)

6、ArithmeticException

7、Exception

8、答、无论是出于何种原因，只要执行离开 try/catch 代码块，就会执行 finally 代码块。即无论 try 是否正常结束，都会执行 finally 定义的最后代码。如果 try 代码块中的任何代码或它的任何 catch 语句从方法返回，也会执行 finally 代码块。但本题中在 try 代码块中执行了“System.exit(0);”语句，执行了这一语句后，Java 虚拟机（JVM）将被终止。那么 finally 语句块一定不会被执行。

五、程序题

1、创建一个 Exception 类的子类 DivideByZeroEx，代表除数为 0 异常；编写一个 TestDivideByZero 类，该类包括一个方法 div(double a, double b)，实现两个参数的相除操作，如果 b 为 0，则生成异常对象，并抛出异常，否则得到 a/b 的结果。然后在 main 函数中调用 div 方法，然后处理异常。

```
class DivideByZeroEx extends Exception {  
    private String s;  
    public DivideByZeroEx(String s){  
        this.s=s;  
    }  
    public String getS() {  
        return s;  
    }  
    public void setS(String s) {  
        this.s = s;  
    }  
}  
  
public class TestDivideByZero {  
    public TestDivideByZero() {  
        // TODO Auto-generated constructor stub  
    }  
  
    public static double div(double a, double b) throws DivideByZeroEx {  
        if(b==0){  
            throw new DivideByZeroEx(" 除数不能为零！ ");  
        }  
        else  
            return a/b;  
    }  
}
```

```

}
public static void main(String[] args) {
try {
double result=div(5,1);
System.out.println(result);
} catch (DivideByZeroEx e) {
System.out.println(" 除数不能为零 !");
e.printStackTrace();
}
}
}

```

2、编写一个程序，以说明 catch (Exception e) 如何捕获各种异常。

```

public class ExceptionExam{
    public static void main(String [] args){
        try{
            int i=0;
            i=3/i;// 将产生算术异常。
        }
        catch(ArithmeticException e)// 找到了匹配的异常， 执行该 catch 块。
        {
            System.out.println(" 发生了 ArithmeticException 异常 ");
        }
        catch(AarryIndexOutOfBoundsException e)// 不是匹配的异常， 且不会
        再捕获
        //异常，如果发生数组索引超出范围所产生的异常， 将执行该 catch
        块。
        {
            System.out.println(" 发生了 AarryIndexOutOfBoundsException 异常
        ");
        }
        catch(Exception e)
        //前两个 catch 块均无法捕获 try 块中发生的异常时，才会执行该
        catch 块。
        {
            System.out.println(" 发生了异常 ");
        }
        finally
        { System.out.println(" 执行 d Finally"); }
        }
    }
}

```

3、运行 try-catch 语句的实例，熟悉 try-catch 语句的语法形式和应用、答、

```
Public class TryCatchDemo{
    Public static void main(String argsp[]){
        String s1= " 0356 " ;
        Double d1=0;
        Try{
            //程序执行转换操作，如果 S 变量为字符串将产生一个异常
            d1=Double.parseDouble(s1);
            System.out.println( 转换成功， s 变量是数字，等于 "+d1) ;
        } catch ( Exception e ) {
            System.out.println( 转换失败， s 变量是文字，等于 "+d1) ;
        } } }
```

4、异常类型匹配的处理方式。程序功能、首先输出“这是一个异常处理的例子”，然后在你程序中主动地产生一个 ArithmeticException 类型被 0 除而产生的异常，并用 catch 语句捕获这个异常。最后通过 ArithmeticException 类的对象 e 的方法 getMessage 给出异常的具体类型并显示出来。

```
public class ExceptionExam1{
    public static void main(String [] args){
        try
        {int i=0;
        i=3/i;
        }
        catch(ArithmeticException e){
            System.out.println(" 异常是、 "+e.getMessage());
        }
        finally{
            System.out.println("finally 语句被执行 ");
        }
    }
}
```

5、根据所给创建自定义异常类的使用格式，编写一个自定义异常的小程序。

```
class Excp1 extends Exception {}
class Excp2 extends Excp1 {}
public class ExceptionExam7{
    public static void main(String [] args) throws Exception{
        try {
            throw new Excp2();
        }
    }
}
```

```

    }
    catch(Excp2 e)    {
        System.out.println("catch1");
        throw new Excp1();
    }
    catch(Excp1 e)    {
        System.out.println("catch2");
        throw new Exception();
    }
    catch(Exception e)    {
        System.out.println("catch3");
    }
    finally    {
        System.out.println("finally");
    }    }    }

```

6、编写一个程序，捕获两种不同类型的异常。

```

public class ExceptionExam3{
    public static void main(String args[]){
        try{
            int i = args.length;
            System.out.println("i =" + i);
            int j=5/i;
            int k[]={ 1,2,3 };
            k[5]=0;
        }
        catch(ArithmeticException e){
            System.out.println(" 被零除 : " + e);}
        catch(ArrayIndexOutOfBoundsException e){
            System.out.println("Array index out of bound exception: " + e);}
        System.out.println(" 执行 catch 块后的语句块 ");
    }
}

```

7、编写一个程序，嵌套 try 语句。

```

public class ExceptionExam2{
    public static void main(String args[]){
        int data1[]={2,4,6,8,10,12};
        int data2[]={1,0,2,4,3};
        try{
            for(int i=0;i<data1.length;i++) {
                try{
                    System.out.println(data1[i]+ "/" + data2[i]+ "is" + data1[i]/data2[i]);
                }
            }
        }
    }
}

```

```
}
catch(ArithmeticException e){
System.out.println(" 不能被零除！  ");
}}}
catch(ArrayIndexOutOfBoundsException e){
System.out.println(" 程序被终止！  ");}}}
```

- 8、略
- 9、略
- 10、略

第十章 文件操作

一、选择题

1—5	CBAAC	6—10	CDDAA	11—15	BCCBA	16—20
	CDDAB					
21—25	DCCAA	26—30	BDDDB	31—35	AADAA	36—40
	DDADB					
41—45	BACBB	46—50	ABCCA	51—55	BBBAC	56—60
	AABDA					
61—65	ADBCA	66—70	AADCD	71—73	AAA	

二、填空题

- 10、File , java.io
- 11、文本文件操作 流文件
- 12、FileReader FileWriter
- 13、BufferedReader java.io
- 14、FileInputStream java.io
- 15、read()
- 7、字符串 Boolean
- 8、文件 boolean, 不为空
- 9、FileReader FileWriter java.io
- 10、目录
- 11、目录名
- 12、文件对象 String name 父路径字符串
- 13、FileInputStream FileOutputStream
- 14、write()
- 15、顺序 RandomAccessFile
- 16、DataInput 接口和 DataOutPut 接口 .
- 17、顺序存取 随机存取
- 18、输入流
- 19、DataOutputStream FileInputStream
- 20、java.lang.Object RandomAccessFile 数据输出
- 21、对象 parent child 路径

22、 InputStream Reader BufferedReader
 23、 FileInputStream FileOutputStream
 24、 DataInput DataOutput
 25、 InputStream System.in.read()
 26、 字节流 字符流
 27、 System.in
 28、 read()
 29、 文件文件操作 流文件
 30、 new FileReader(File)
 31、 byte[]array byte[]array int length 输入流 int 字节数
 off off
 32、 数组
 33、 FileInputStream FileOutputStream
 34、 相对 绝对 绝对 相对
 35、 InputStream Reader BufferedReader
 36、 OutputStream、 Writers 字节流 字符
 37、 print 读写 字符流
 38、 数据交流
 39、 Object 类 输入 输出流
 40、 子类 8 位字节 字节
 41、 FileInputStream FileOutputStream
 42、 InputStream OutputStream
 43、 DataInput DataOutput
 44、 System.in System.out
 45、 FilterOutputStream
 46、 抽象
 47、 输出流
 48、 File java.io
 49、 FileReader FileWriter
 50、 BufferedReader java.io
 51、 FileInputStream java.io
 52、 read()
 53、 InputStream OutputStream
 54、 顺序存取 随机存取
 55、 in out err
 56、 字节 字符
 57、 ObjectInput ObjectOutput
 58、 String InputStream
 59、 InputStream System.in.read()
 60、 PrintStream FilterOutputStream

- 61、 8 位 字节 16 字符
- 62、 输入字符流
- 63、 File、 RandomAccessFile、 FileInputStream、 FileOutputStream
- 64、 输入流
- 65、 DataOutputStream FileInputStream
- 66、 java.lang.Object RandomAccessFile 数据输出
- 67、 字节 int
- 68、 int 写入 字符串
- 69、 序列化 序列化
- 70、 String write (int)
- 71、 字符 字节流到字符流的

三、判断题

- | | | | |
|-------|---------------------|-------|---------------------|
| 1—10 | T F T F T T F F F T | 11—20 | F F F T T F F F F T |
| 21—30 | F T T T T T T T F F | 31—40 | F T T T T F F T T T |
| 41—50 | F T T T F F T T T F | 51—60 | T F T T T F T T T F |
| 61—70 | T T T F F T F T F T | 71—80 | T F T T T T T F T T |
| 81—87 | F T T F T T T | | |

四、读程序题

- 1、 throws IOException c = (char)br.read();
- 2、 FileInputStream IOException
- 3、 System.in.read() IOException
- 4、 throws IOException BufferedReader (char)br.read()
- 5、 InputStreamReader br.readLine() 2
- 6、 FileInputStream(file1) FileOutputStream fis.available()
- 7、 BufferedReader in=br.readLine())!=null
- 8、 InputStreamReader String str=br.readLine();
- 9、 throws IOException InputStreamReader
- 10、 java.io.File mkdir() txt
- 11、 InputStreamReader String str=br.readLine();
- 12、 BufferedReader in=br.readLine())!=null
- 13、 FileInputStream(file1) FileOutputStream fis.available()

五、问答题

- 1、 Java 程序通过流来完成输入和输出，流是输入或输出信息的抽象。流再通过 Java 的输入 /输出系统与外设连接进行数据通信。流是抽象的对象，具体实现代码在 java.io 包中。
- 2、 类 FileInputStream 和类 FileOutputStream 分别直接继承于 InputStream 和 OutputStream，它们重写或实现了父类中的所有方法，通过这两个类可以打开蹦迪机器的文件，进行顺序的读写。
- 3、 分别是 int read(),int read(byte b[]) 和 int read(byte b[],int off,int len)
int read()方法的用途是从输入流中读取下一个字节流数据，其返回

值是 0~255 之间的一个整数。该方法是 abstract 的，因此子类必须实现它才能生成对象。而 `int read(byte b[])` 是从输入流中读取长度为 `b.length` 的数据，写入字节数组 `b`，返回本次读取的字节数。`int read(byte b[],int off ,int len)` 是从输入流中读取长度为 `len` 的数据，写入字节数组 `b` 中从下标 `off` 开始的位置，返回本次读取的字节数。

4、FileReader 类的创建语法是、`FileReader(File file)` 和 `FileReader(String fileName)`，

`FileWriter` 类的创建语法是、`FileWriter(File file)` 和 `FileWriter(String fileName)`。

5、BufferedReader 类的创建语法是 `BufferedReader(Reader in)`，

`BufferedWriter` 类的创建语法是、`BufferedWriter(Writer out)`。

6、FileInputStream 类的创建语法是、`FileInputStream(File file)` 和 `FileInputStream(String name)`，

`FileOutputStream` 类的创建语法是、`FileOutputStream(File file)` 和 `FileOutputStream(String name)`。

7、System 类中的 `out` 是一个 `PrintStream` 类型的变量，它是标准输出流，默认情况下，输出到控制台。

System 类中的成员变量 `in` 是 `InputStream` 类型的变量，它是标准输出流，默认情况下，指的是键盘输入。

System 类中的成员变量 `err` 是一个 `PrintStream` 类型的变量。它指的是标准错误流，它与 `out` 变量类似，默认情况下输出到控制台。

8、它一般有 `void write(int b)`、`flush()` 和 `close()` 这三个方法，其中 `void write(int b)` 方法是用来将指定的字节 `b` 作为数据写入输出流。

`flush()` 的作用是清空输出流，并输出所有被缓冲的字节。

而 `close()` 类的作用是关闭流的使用方法及注意，与 `InputStream` 中的 `close()` 方法一样。

9、可以，使用的类是 `FileReader` 和 `FileWriter`。

10、可以，使用的类是 `FileInputStream` 和 `FileOutputStream`。

11、Serialization 便是应用文件的读入和写出功能将程序的变量保存在硬盘上和从硬盘上读入数据。

12、可以，使用的方法是 `read(byte[],int off ,int len)`。

13、可以，使用的类是 `SequenceInputStream`。

14、FileReader 是一个文件读入类，它是 `InputStreamReader` 的子类。而 `BufferedReader` 类的含义是缓冲输入字符流。

15、所谓流是指同一台计算机或网络中不同计算机之间有序运动着的数据序列，Java 把这些不同来源和目标的数据都统一抽象为数据流。数据流可分为输入流和输出流，输入流代表从其他设备流入计算机的数据序列，输出流代表从计算机流向外部设备的数据序列。

流式输入输出的特点是数据的获取和发送沿数据序列的顺序进行，即每一个数据都必须等待排在它前面的数据，等前面的数据读入

或送出之后才能被读写。所以流和队列一样，只能以“先进先出”的方式对其中的数据进行读写，而不能随意选择读写的位置。

16、两者都作为 `Object` 类的直接子类，基本输入流（无效 `Stream`）和基本输出流（`OutputStream`）是处理以 8 位字节为基本单位的字节流类；`Reader` 和 `Writer` 类是专门处理 16 位字符流的类。

17、（1）`public File(String pathname)`、创建一个对应于参数 `pathname` 的 `File` 类对象。参数 `pathname` 是包含目录和文件名的字符串。如果没有文件名，则代表目录。

（2）`public File(String parent, String child)`、该构造函数将 `pathname` 分成两部分 `parent` 和 `child`，参数 `parent` 表示目录或文件所在路径，参数 `child` 表示目录或文件名称。

（3）`public File(File parent, String child)`、该构造函数与上面一个不同在于将 `parent` 的参数类型由 `String` 变为 `File`，代表 `parent` 是一个已经创建的 `File` 类文件对象（指向目录）。

常用方法有、

（1）`public boolean canWrite()`、返回文件是否可写。

（2）`public boolean canRead()`、返回文件是否可读。

（3）`public boolean createNewFile()`、当文件不存在时创建文件。

18、`RandomAccessFile` 类提供了两个构造函数、

`public RandomAccessFile(File file, String mode);`

`public RandomAccessFile(String name, String mode);`

`RandomAccessFile` 类包括的一些常用方法，有、

`readLine()`、用于文本文件的读取，可以一次读取文件的一行，文件指针自动移到下一行开始位置，行的结束标记使用回车符（`\r`）、换行符（`\n`）或回车换行符。

`public void close()`、关闭输入输出流；

`public long length()`、获取文件的长度。

19、`FileOutputStream fout=new FileOutputStream("myfile.dat");`

`DataOutputStream out=new DataOutputStream(fout);`

20、`DataInputStream` 类和 `DataOutputStream` 类是从过滤流类继承过来，这两个流的对象均不能独立地实现数据的输入和输出处理，必须与其他输入流与输出流对象一起使用，才能实现不同类型数据的读写。这两个类与 `FileInputStream` 类和 `FileOutputStream` 类相配合可以完成对格式数据的读写。

21、Java 的 I/O 功能主要是通过输入输出类库 `java.io` 包来实现的，`java.io` 包中包括了可以处理字节流、字符流、文件流和对象流等的类。

22、`RandomAccessFile` 类用来实现对文件的随机读写操作。它继承自 `Object` 类，实现了 `DataInput` 和 `DataOutput` 接口。构造方法如下、

`RandomAccessFile raf=new RandomAccessFile(< 文件名 >, < 访问模`

式>)、

说明如下、

(1) 文件名参数，可以是 File 类对象或文件名字符串的形式提供。

(2) 访问模式，有 2 种取值、“r”只读，或者“rw”读写。

23、在 Java.io 包中，能够直接访问内存的类有 3 个。

(1) ByteArrayInputStream，用于从字节数组中读取数据，主要方法有、read()，skip()，reset()，close()等。

(2) ByteArrayOutputStream，用于向字节数组写入数据，主要方法有、write()，writeTo()，reset()，close()以及 toByteArray()，size()，toString()。

(3) StringBufferInputStream 类。

这几个类都继承自 InputStream 和 OutputStream 类。

24、在 Java 中，不同的 Java 程序、线程或代码段相互之间的数据交换通过管道流实现。java.io 包中的 2 个类、PipedInputStream 和 PipedOutputStream 类实现了管道机制。

六、编程题

1、编写一个应用程序，完成文件的拷贝功能，文件名从命令行得到。

```
import java.io.*;

class FileCopy{
    public static void main(String[] args){
        FileInputStream in;
        FileOutputStream out;
        if (args.length<2) {
            System.out.println("Usage: java copy srcfile destfile");
            System.exit(-1);
        }
        try {
            in = new FileInputStream(args[0]);
            out = new FileOutputStream(args[1]);
            copyFile(in,out);
        }
        catch (Exception e){
            System.out.println(e);
        }
    }

    private static void copyFile(FileInputStream in, FileOutputStream out){
        int length;
        byte buf[] = new byte[1024];
        try{
```

```

        while ((length=in.read(buf,0,1024))!=-1)
        {
            out.write(buf, 0, length);
        }
    }
    catch (Exception e){
        System.out.println("Error:"+e);
        System.exit(-1);
    }
}
}

```

2、请编写用 RandomAccessFile oureFile; String s= “ information to Append \n mon ! ”;

oureFile = new RandomAccessFile oureFile(“phone.nember.numbers”, “ rw ”);模式打开访问 oureFile 文件 ,并在文件尾加 s 的 addFile 程序。

Import java.io.*

Class addFile {

Public static void main(String args[])throws IOException {

RandomAccessFile ourFile;

String= “ information to Append\n mon! \n ”;

oureFile=new RandomAccessFile oureFile(“ phone.nember.numbers “ , ” rw ”);

oureFile.seek(oureFile.length());

oureFile.writebytes(s);

oureFile.close();

}

}

3、利用文件输入输出流编写一个实现文件拷贝的程序 , 源文件名和目标文件名通过命令行参数传入。

import java.io.*;

class CopyFile{

public static void main(String[] args) {

String file1,file2 ;

int ch = 0 ;

try {

file1=args[0];

file2=args[1];

FileInputStream fis = new FileInputStream(file1);

FileOutputStream fos=new FileOutputStream(file2);

```

while ((ch=fis.read())!=-1)
fos.write(ch);
fis.close();
fos.close();
}
catch(FileNotFoundException e){
System.out.println(" 源文件 :未找到 !");
}catch(ArrayIndexOutOfBoundsException e){
System.out.println(" 缺少运行参数 !  ");
System.exit(-1);
}
catch (IOException e){
System.out.println(e.toString());
}
}
}

```

4、编写一个程序，在当前目录下创建一个子目录 test，在这个新创建
的子目录下创建一个文件，并把这个文件设置成只读。

```

import java.io.*;
public class NewFile{
public static void main(String args[]){
File f=new File("test");
if(f.exists()&&f.isDirectory())
System.err.println(" 目录、 "+f.toString()+" 已经存在 !");
else{
if(f.mkdir()){
System.out.println(" 目录 "+f.getAbsolutePath()+" 创建结束 !");
File f2=new File(f,"my.txt");
try{
f2.createNewFile();
f2.setReadOnly();
}catch(IOException e){
System.out.println(e.toString());
}
System.out.println(" 文件、 "+f2.getAbsolutePath()+" 创建结束 !");
}
else
System.out.println(" 目录 "+f.getAbsolutePath()+" 创建失败 !");
}
}
}

```

```
}
```

5、编写一个程序，从键盘输入一串字符，从屏幕输出并将其存入 a.txt 文件中。

```
import java.io.*;
public class InFile{
    public static void main(String[] args)
    {
        int ch;
        try{
            FileOutputStream out=new FileOutputStream("a.txt");
            while((ch=System.in.read())!='\r'){
                System.out.write(ch);
                out.write(ch);
            }
            out.close();
            System.out.write('\n');
        }catch(IOException e){
            System.out.println(e.toString());
        }
        System.out.println(" 输出至文件完毕！  ");
    }
}
```

6、随机读取文件中的字符信息。

```
import java.io.*;
public class Readtext
{
    public static void main(String[] args)
    {String str1;
    char ch1;
    int n;
    try{
        File mytxt=new File("read.txt");
        RandomAccessFile ra=new RandomAccessFile(mytxt,"rw");
        ra.write("This is the first sentence".getBytes());
        ra.writeChar('\n');
        ra.write("Java Program Design".getBytes());
        ra.writeChar('\n');
        ra.seek(0);
        str1=ra.readLine();
        System.out.println(str1);
    }
}
```

```

ra.seek(18);
System.out.println(ra.readLine());
ra.close()
}
catch(IOException e)
{
System.out.println(e.toString());
}
}
}

```

程序运行结果、

This is the first sentence
sentence

7、从键盘敲入一系列字母，将其存储到文件中，

```

import java.io.*;
public class Exer6_7 {
    //从键盘读入一组字符并返回
    public static String Keyboard()
    {
BufferedReader InputStream = new BufferedReader(new InputStreamReader(System.in));
try
{
        return(InputStream.readLine());
}
        catch (IOException e){return "Error occur when input from Keyboard!";}
    }
    //将一组字符存入文件
    public static void PuttoFile(String s,String fileName)
    {
        //String fileName="d:\\testInput.txt";
try
    {
        FileWriter writer = new FileWriter(fileName);
        writer.write(s); // 将随机数写入流
        writer.close(); // 关闭流
    }
    catch (IOException iox)
    {

```

```

        System.out.println("Error occur when PuttoFile!");
    }
}

```

8、从键盘敲入一系列字母， 将其存储到文件中， 对其进行升序排序后， 存到另一个文件中， 并显示在屏幕上

```

import java.io.*;
public class Exer6_7 {
    //从键盘读入一组字符并返回
    public static String Keyboard()
    {
        BufferedReader InputStream = new BufferedReader(new InputStreamReader(System.in));
        try
        {
            return(InputStream.readLine());
        }
        catch (IOException e){return "Error occur when input from Keyboard!";}
    }
    //将一组字符存入文件
    public static void PuttoFile(String s,String fileName)
    {
        //String fileName="d:\\testInput.txt";
        try
        {
            FileWriter writer = new FileWriter(fileName);
            writer.write(s); // 将随机数写入流
            writer.close(); // 关闭流
        }
        catch (IOException iox)
        {
            System.out.println("Error occur when PuttoFile!");
        }
    }
    //将数据从文件中读取出来
    public static String GetFromFile(String fileName)
    {
        String s="";
        //String fileName="d:/testInput.txt";
    }
}

```



```

try
{
    BufferedReader in = new BufferedReader(new FileReader(fileName
e));
    s=in.readLine(); // 读取一串字符保存到    s
}
catch(IOException io)
{
    System.out.println("Error occur when GetFromFile!");
}
return s;
}
//将一组字符排序 (冒泡排序 升序排序 ) 并将排序后的字符串返回
public static String sort(String s)
{
    char[] c=s.toCharArray(); // 将字符串转换成字符数组
    char temp; //排序用到的临时变量
    for(int i=0;i<c.length-1;i++)
        for(int j=i+1;j<c.length;j++)
            if(c[i]>c[j])
            {
                temp=c[i];
                c[i]=c[j];
                c[j]=temp;
            }
    s=new String(c); // 注意 不能用 c.toString
    return s;
}

public static void main(String args[]){

    String s=Keyboard(); // 从键盘输入一组字符
    PuttoFile(s,"d:\\testInput.txt"); // 将字符保存到 d:\\testInput.txt
    s=GetFromFile("d:/testInput.txt"); // 从文件 d:\\testInput.txt 读出数据
    s=sort(s); //将 s 进行排序
    PuttoFile(s,"d:\\testOutput.txt"); // 将 排 序 后 的 内 容 保 存
到 d:\\testOutput.txt
    System.out.println(s); // 将排序后的内容输出到屏幕上
}

```

```
}  
/*
```

练习 \Java 练习 \java\10>java Exer6_7

asdfkjadslgkjlkdslaglk

aaadddfggjkkkkllllsss

9、编写一段程序，将 c:\jdk1.3\example\newFile.txt 中保存的信息输出到屏幕上。

```
import java.io.*;  
public class File3{  
    public static void main(String asd[])throws IOException{  
        int ch;  
        File file 1 = new File( "c:\\jdk1.3\\example\\newFile.txt ");  
        try{  
            FileInputStream fin = new FileInputStream(file1);  
            System.out.println( "文件中的信息为、 ");  
            ch = fin.read();  
            while(ch!=-1){  
                System.out.print((char)ch);  
                ch = fin.read();}  
            fin.close();  
        }catch(FileNotFoundException e){System.out.println(e);}  
        Catch(IOException e){  
            System.out.println(e);  
        }  
    }  
}
```

10、编写一段程序，实现从键盘输入以“？”结束的一串字符，然后将该串中“？”前面的所有字符顺序写入磁盘文件 c:\jdk1.3\example\newFile.txt 中。

```
import java.io.*;  
public class File2{  
    public static void main(String asd[]){  
        char ch;  
        File file1 = new File( "c:\\jdk1.3\\example\\newFile.txt ");  
        Try{  
            FileOutputStream fout = new FileOutputStream(file1);  
            System.out.println( "输入任一字符串，以？结束 ");  
            ch = (char)System.in.read();  
            while(ch!= ??){  
                fout.write(ch);
```

```

ch=(char)System.in.read();
}
Fout.close();
}catch(FileNotFoundException e){
System.out.println(e);}catch(IOException e){
System.out.println(e);}
}
}

```

11、编写程序，接收用户从键盘输入的数据，回车后保存到文件 test.txt 中。若用户输入符号 #，则退出程序。

```

import java.io.*;
public class WriteFile{
public static void main(String args[])
{byte buffer[]=new byte[128];
System.out.println(" 请输入数据，回车后保存到文件    test.txt");
System.out.println(" 输入 # 则退出！ ");
try{
FileOutputStream f=new FileOutputStream("test.txt");
while(true){
int n=System.in.read(buffer);
if(buffer[0]=='#' ) break;
f.write(buffer,0,n);
f.write('\n');
}
f.close();
}catch(IOException e)
{
System.out.println(e.toString());
}
}
}

```

12、某人在玩游戏的时候输入密码 123456 后成功进入游戏（输错 3 次则被强行退出），要求用程序实现密码验证的过程。

解析、（不考虑类名的情况下）

```

public static void main(String []args)throws IOException{
BufferedReader br = new BufferedReader(new
InputStreamReader(System.in))
String password = " ";
for(int i=0;i<3;i++){
Ststem.our.println( "请输入密码、 ")

```

```

Password = br.readLine();
if(password.equals( "123456 "))
    break;}else
    {if(i==2){
        Stsrem.out.println( '累计密码错误达到    3 次 , 退出游戏 !    ');
        Ststem.exit(0);
    }
    }
}

```

13、创建一个输入 “ 你好！ ” 的 JAVA 语言程序，在屏幕上显示出来。

```

Import java.io.* ;

```

```

    class BRRead{
    public static void main(String[]args)throws IOException{
        char c;
        BufferedReader      br      =      new      BufferedReader(new
InputStreamReader(System.in));
        c = (char)br.read();
        System.out.println(c);
    }
}

```

14、编程、编写一个截取字符串的函数， 输入为一个字符串和字节数，输出为按字节截取的字符串。 但是要保证汉字不被截半个，如 “ 我 ABC ” 4，应该截为 “ 我 AB ”，输入 “ 我 ABC 汉 DEF ”，6，应该输出为 “ 我 ABC ” 而不是 “ 我 ABC+ 汉的半个 ”。

```

package test;

```

```

class SplitString {
String SplitStr;
int SplitByte;
public SplitString(String str,int bytes) {
    SplitStr=str;
    SplitByte=bytes;
    System.out.println("The String is:"+SplitStr+";SplitBytes="+SplitByte);
}

```

```

public void SplitIt() {
    int loopCount;
    loopCount=(SplitStr.length()%SplitByte==0)?(SplitStr.length()/SplitByte

```

```

):(SplitStr.length()/SplitByte+1);
    System.out.println("Will Split into "+loopCount);
    for (int i=1;i<=loopCount ;i++ ) {
        if (i==loopCount){
            System.out.println(SplitStr.substring((i-1)*SplitByte,SplitStr.length()));
        } else {
            System.out.println(SplitStr.substring((i-1)*SplitByte,(i*SplitByte)));
        }
    }
}

public static void main(String[] args) {
    SplitString ss = new SplitString("test 中 dd 文 dsaf 中男大 3443n 中国
43 中国人 0ewldfls=103",4);
    ss.SplitIt();
}
}

```

15、编写一段程序，要求从键盘输入 123bc 并显示出来。

```

    Import java.io.* ;

    class BRRead{
    public static void main(String[]args)throws IOException{
        char c;
        BufferedReader br = new BufferedReader(new
InputStreamReader(System.in));
        c = (char)br.read();
        System.out.println(c);
    }
}

```

16、编写一段使用 FileInputStream 类与 FileOutputStream 类复制文件的代码。

```

import java.io.*;
class ReadWriteFile{
public static void main(String[] args) {
String file1,file2 ;
int ch = 0 ;
file1 = "readme.txt" ;
file2="readme.bak";
try{

```

```

FileInputStream fis = new FileInputStream(file1);
FileOutputStream fos=new FileOutputStream(file2);
int size=fis.available();
System.out.println(" 字节有效数、 "+size);
while ((ch=fis.read())!=-1){
System.out.write(ch);
fos.write(ch);
}
fis.close();
fos.close();
}
catch (IOException e){
System.out.println(e.toString());
}
}
}

```

17、编写一段代码，使其能够将读入的信息用字节数组存放，转换成字符串后再输出。

```

import java.io.*;
public class StandardIO2 {
public static void main(String []args)
{
byte ch[]=new byte[128];
System.out.println(" 请输入一行字符 ");
try{
System.in.read(ch,0,128);
System.out.println(" 你输入的内容如下、 ");
String str=new String(ch); // 转换成字符串
System.out.println(str);
}catch(IOException e){
System.out.println(e.toString());
}
}
}

```

18、编写一段程序，将 c:\jdk1.3\example\newFile.txt 中保存的信息输出到屏幕上。

```

import java.io.*;
public class File3{
public static void main(String asd[])throws IOException{
int ch;

```

```

File file 1 = new File( "c:\\jdk1.3\\example\\newFile.txt ");
try{
FileInputStream fin = new FileInputStream(file1);
System.out.println( "文件中的信息为、 ");
ch = fin.read();
while(ch!=-1){
System.out.print((char)ch);
ch = fin.read();}
fin.close();
}catch(FileNotFoundException e){System.out.println(e);}
Catch(IOException e){
System.out.println(e);
}
}
}

```

19、编写代码实现从键盘输入整型、单精度型、布尔型数据、然后将这些数据写入磁盘。

```

import java.io.*;
public class File4{
public static void main(String asd[]){
int ch ;
InputStreamReader iin = new InputStreamReader(System.in);
BufferedReader bin = new BufferedReader(iin);
File file1= new File( "c:\\jdk1.3\\example\\dataFile.txt ");
try{
FileOutputStream fout = new FileOutputStream(file1);
DataOutputStream dout = new DataOutputStream(fout);
System.out.println( "输入整数、 ");
int l = Integer.parseInt(bin.readLine());
System.out.println( "输入浮点数、 ");
float f=Float.parseFloat(bin.readLine());
System.out.println( "输入布尔量 ");
Boolean b = new Boolean(bin.readLine()).booleanValue();
dout.write(i);
dout.writeFloat(f);
dout.writeBoolean(b);
dout.close();
}
catch(FileNotFoundException e){
System.out.println(e);
}
}
}

```

```

    }catch(IOException e){System.out.println(e);}
    }
}

```

20、编写程序将磁盘文件 c:\jdk1.3\example\dataFile.txt 中保存的各类数据复制到磁盘文件 c:\jdk1.3\example\outFile.txt 中。

```

import java.io.*;
public class File4{
    public static void main(String asd[])throws IOException{
        int ch;
        File file1 = new File( "c:\\jdk1.3\\example\\example\\dataFile.txt ");
        File file2 = new File( "c:\\jdk1.3\\example\\example\\outFile.txt ");
        try {
            FileInputStream fin = new FileInputStream(file1);
            DataInputStream din=new DataInputStream(fin);
            int i = din.readInt();
            float f = din.readFloat();
            boolean b = din.readBoolean();
            din.close();
            FileOutputStream fout = new FileOutputStream(file2);
            DataOutputStream dout = new DataOutputStream(fout);
            dout.writeInt(i);
            dout.writeFloat(f);
            dout.writeBoolean(b);
            dout.close();
            System.out.println( "整数、 " +i);
            System.out.println( "浮点数、 " +f);
            System.out.println( "布尔量、 " +b);
        }catch(FileNotFoundException e){
            System.out.println(e);}
        catch(IOException e){System.out.println(e);}
    }
}

```

21、利用文件输入输出流编写一个实现文件拷贝的程序，源文件名和目标文件名通过命令行参数传入

```

import java.io.*;
class CopyFile{
    public static void main(String[] args) {
        String file1,file2 ;
        int ch = 0 ;
        try {

```



```

file1=args[0];
file2=args[1];
File 无效 Stream fis = new File 无效 Stream(file1);
FileOutputStream fos=new FileOutputStream(file2);
while ((ch=fis.read())!=-1)
fos.write(ch);
fis.close();
fos.close();
}
catch(FileNotFoundException e){
System.out.println(" 源文件 :未找到 !");
}catch(ArrayIndexOutOfBoundsException e){
System.out.println(" 缺少运行参数！ ");
System.exit(-1);
}
catch (IOException e){
System.out.println(e.toString());
}
}
}

```

22、编写一个程序， 从键盘输入 10 个整数， 并将这些数据排序后在标准输出上输出

```

import java.io.*;
public class Sort{
public static void main(String args[])
{
int a[]=new int[10];
byte b[]=new byte[10];
int t;
String str;
System.out.println(" 请输入 10 个整数、 ");
try{
for(int i=0;i<10;i++)
{System.out.print("No. "+(i+1)+" : ");
System.in.read(b);
str=new String(b);
str=str.trim();
a[i]=Integer.parseInt(str);
}
}catch(IOException e){

```

```

System.out.println(e.toString());
}
catch(NumberFormatException e){
System.out.println(e.toString());
}
for(int i=0;i<9;i++)
for(int j=i+1;j<10;j++)
{
if(a[i]>a[j]){
t=a[i];
a[i]=a[j];
a[j]=t;
}
}
for(int i=0;i<10;i++)
System.out.println(a[i]+"\\t");
}
}

```

23、编写一个程序，从键盘输入一串字符，从屏幕输出并将其存入 a.txt 文件中

```

import java.io.*;
public class InFile{
public static void main(String[] args)
{
int ch;
try{
FileOutputStream out=new FileOutputStream("a.txt");
while((ch=System.in.read())!='\\r'){
System.out.write(ch);
out.write(ch);
}
out.close();
System.out.write('\\n');
}catch(IOException e){
System.out.println(e.toString());
}
System.out.println(" 输出至文件完毕！ ");
}
}

```

24、编写一个程序，从键盘输入一串字符，统计这串字符中英文字母、

数字、其他符号的字符数。

```
import java.io.*;
public class Count{
    public static void main(String[] args)
    {
        int x=0,y=0,z=0;
        int ch;
        try{
            while((ch=System.in.read())!='\r'){
                if(ch>='A'&&ch<='Z' || ch>='a'&&ch<='z')
                    x++;
                else if(ch>='0'&&ch<='9')
                    y++;
                else
                    z++;
            }
        }catch(IOException e){
            System.out.println(e.toString());
        }
        System.out.println(" 英文字母、    "+x);
        System.out.println(" 数字字符、    "+y);
        System.out.println(" 其它字符、    "+z);
    }
}
```

25、编写一段代码，使其实现从键盘输入 8，能得到一个值为 2 的作用。

```
import java.io.*;
    public class Test2 {
        public static void main(String args[])throws Exception{
            int a=4;
            BufferedReader br=new BufferedReader(new
InputStreamReader(System.in));
            System.out.println(" 请输入一个数字 ");
            String input=br.readLine();
            int b=Integer.parseInt(input);
            if(b>a){
                int sum=b/a;
                System.out.println(sum);
            }else{
                System.out.println(" 输入错误 ");
            }
        }
    }
```

}
}

第十一章 GUI 界面设计

一、选择题

1-5 ADDDC	6-10 DBACB	11-15 BCBCD	16-20 BDBCD
21-25 CBBDB	26-30 ABDDDB	31-35 ABBBA	36-40
BCAAB			
41-45 DCCBD	46-50 CCBCD	51-55 ABDCC	56-60
ABBCA			
61-65 CCBCB	66-70 BCCAD	71-75 ABDDDB	76-80 BDACA
81-85 DDCCD	86-90 CDBCB	91-95 CCDAC	96-100
BABDC			
101-105 AACCA	106-110 ACCDB	111-115 CDCAA	116-120
CCCCA			
121-125 ADCDB	126-130 ACDAB	131-135 ADBBA	136-140
CDCAB			
141-145 CBCBC	146-150 ACCAB	151-154 BBBB	

二、填空题

- 1、FlowLayout、BorderLayout、CardLayout、CridLayout、GridBagLayout
- 2、BoxLayout、ScrollPaneLayout、ViewportLayoutr、Overlayout
- 3、FlowLayout
- 4、Swing、水平、垂直
- 5、KeyEvent
- 6、MouseListener、MouseMotionListener、MouseApdapter
- 7、MouseListener、MouseMotionListener
- 8、JMenuBar、JMenu、JMenuItem
- 9、创建一个 JMenuBer 对象并将其加入到 JFrame 中
创建 JMenu 对象
创建 JMenuulten 对象并将 33 其添加到 JMenu 对象中
将 JMenu 对象添加到 JMenuBar 中
- 10、Panel、Swing
- 11、JPanel
- 12、Swing
- 13、FlowLayout
- 14、复选框、当选按钮、列表、下拉列表、标签、文本编辑区、按钮、菜单
- 15、JComponent
- 16、GUI
- 17、java.awt
- 18、Swing

- 19、 AWT 、 Swing、 SWT/JFace
- 20、 容器类组件、布局类组件、普通组件类
- 21、 JOptionPane
- 22、 JOptionPane
- 23、 Java
- 24、 JOptionPane
- 25、 JDialog
- 26、 JFileChooser
- 27、 ProgressMonitor
- 28、 下拉式菜单、弹出式菜单
- 29、 复选框、单选按钮、列表框、组合框
- 30、 非模式 (non-modal)、模式 (modal)
- 31、 低级事件
- 32、 低级事件、高级事件
- 33、 AWT
- 34、 AWT 组件
- 35、 Swing
- 36、 AWT
- 37、 Swing
- 38、 Application 、或 Frame、主窗口、或 Applet、主窗口
- 39、 FlowLayout 、 BorderLayout 、 CardLayout、 GridLayout
- 40、 框架
- 41、 AWT 、 Swing、 JFC
- 42、 按钮 (Button)、单选按钮 (CheckboxGroup)、复选框 (Checkbox)、下拉列表框 (Choice)、画板 (Canvas)、列表框 (List)、标签 (Label)、文本组件 (Textfield 和 TextArea)、滚动条 (Scrollbar)、菜单 (Menu)
- 43、 事件
- 44、 Javax.swing
- 45、 Window 类、 panel 类、 scrollpane 类
- 46、 add()
- 47、 getIconImage()
- 48、 setTitle()
- 49、 setVisible()
- 50、 getText()
- 51、 getEchoChar()
- 52、 Label()
- 53、 3
- 54、 LEFT 、 CENTER 、 RIGHT
- 55、 显示标题
- 56、 修饰符、返回值类型、方法名、方法体

57、 JpasswordField
58、 list
59、 标签、 状态
60、 Checkbox()
61、 组件
62、 选中、 未选中
63、 Checkbox
64、 高度
65、 是否允许复选
66、 boolean
67、 Menubar
68、 awt
69、 快
70、 setSize()
71、 AbstractButton
72、 按钮的图标
73、 String getText()
74、 setEnabled()、 是否禁用
75、 isSelected()
76、 按钮

三、 判断题

1-5 TTTFT 6-10 TFFTT 11-15 TTTTT 16-20 TTTTT
21-25 FTTTT 26-30 TTFTT 31-35 TTTTT 36-40 TTTTT
41-42 TT

四、 程序填空

```
1、 import java.awt. *;  
import javax.swing.*;  
2、 extends  
stdno=new JTextField();  
name=new JTextField();  
3、 c.add(button1);c.add(button2);c.add(button3);  
jframe.show();  
4、 cp.add("a",jbt1);  
cp.add("d",jbt4);  
card.next(cp);  
5、 JFrame  
    show();  
implements  
6、 jpanel.add(jbutton2);  
jframe.show();
```

```

7、 implements KeyListener
    addKeyListener(this);
keyPressed
keyReleased
keyTyped
8、 cp.addMouseListener(new MouseEventHandler());
show();
text.setText(s);
9、 Frame f = new Frame();
f.add(button2);
f.add(button3);
10、 frame = new JFrame("JSplitPane 组件 ");
    frame.setVisible(true);
11、 plist.addActionListener(new ActionListener()
frame.setVisible(true);
12、 JPanel valuePanel = new JPanel();
setVisible(true);
13、 ActionListener
frame.setVisible(true);
14、 List list=new List(4,true)
15、 Choice choice=new Choice()
16、 Label label=new Label( '我的标签 ');
Container c=getContentPane();
17、 JPanel
frame.setVisible(true);
18、 import javax.swing.*;
setVisible(true);
19、 JApplet ItemListener
italic.addItemListener(this);
import java.awt.*;
getContentPane().add(t, BorderLayout.NORTH);

```

五、简答题

1、 答、 1) 创建组成界面的各个部件，确定它们的属性和位置关系，调整外观尽量完美

2) 定义组成界面的各个部件对不同事件的响应，实现图形用户界面的交互功能。

即为各个成分和元素编写代码，以达到预期的功能。

2、 答、 使用更少的系统资源
增加大量更高级的组件
允许控制程序的外观

- 3、答；(1) 容器类组件
(2) 布局类组件
(3) 普通组件类
- 4、答、引入合适的包和类
使用缺省的观感或设置自己的观感
设置一个顶层的容器
使用缺省的布局管理器或设置另外的布局管理器
定义组件并将它们添加到容器
对组件或事件编码
- 5、第一步、程序开始装载事件包；
第二步、实现事件监听者所对应的接口，有三种方法可以实现接口类、
 - (1) 组件所在类本身作为事件监听类
 - (2) 用外部类作为专门的事件监听类
 - (3) 用匿名内部类作为事件监听类第三步、对事件源，即组件添加事件监听者，事件源实例化以后，必须进行授权，注册该类事件的监听器，通常用 `addXXXListener(XXXListener)` 方法来注册监听器；
第四步，重写相应事件接口类中的方法。
- 6、事件，就是发生在用户界面上的用户交互行为所产生的一种效果。
Event Source (事件源) 产生事件的对象。
Event handler/Listener (事件处理器) 接收事件对象并对其进行处理的方法。
- 7、
 - 1). 引入包
 - 2). 选择“外观和感觉”
 - 3). 设置顶层容器
 - 4). 设置布局管理
 - 5). 向容器中添加组件
 - 6). 对组件进行事件处理
- 8、
 - 1) 用 `getContentPane()` 方法获得 `JFrame` 的内容面板，再对其加入组件、`frame.getContentPane().add(childComponent)`
 - 2) 建立一个 `Jpanel` 或 `JDesktopPane` 之类的中间容器，把组件添加到容器中，用 `setContentPane()` 方法把该容器置为 `JFrame` 的内容面板、
- 9、
 - (1) 创建一个 `JMenuBar` 对象并将其加入到 `JFrame` 中。
 - (2) 创建 `JMenu` 对象。
 - (3) 创建 `JMenuItem` 对象并将其添加到 `JMenu` 对象中。
 - (4) 将 `JMenu` 对象添加到 `JMenuBar` 中。
- 10、所谓模式对话框就是指必须经过确认才能继续程序的运行的对话框，创建的模式对话框有、口令对话框、删除文件对话框、选择参数对话框、错误提示对话框等，这些对话框都是要求输入数据或获

得确认按键后才能关闭对话框的

所谓非模式对话框就是指显示对话框后，用户可以不用关闭此对话框继续程序的运行，如查找 / 替换对话框、插入符号对话框就是非模式对话框。实际上大部分对话框都是模式对话框。

11、 1.设置窗口大小

2.获得当前的窗口大小

3.隐藏和显示一个窗口

4.设置窗口标题

5.关闭 Frame 窗口

12、 1.创建组件

2.指定布局

3.响应事件

六、编程题

1、编写程序，包括一个标签、一个文本框和一个按钮，当单击按钮时把文本框中的内容复制到标签中显示。

```
package test;
import java.awt.Toolkit;
import javax.swing.SwingUtilities;
import javax.swing.UIManager;
import java.awt.Dimension;
public class Application {
    boolean packFrame = false;
    public Application() {
        Frame1 frame = new Frame1();
        if (packFrame) {
            frame.pack();
        } else {
            frame.validate();
        }
        Dimension screenSize =
Toolkit.getDefaultToolkit().getScreenSize();
        Dimension frameSize = frame.getSize();
        if (frameSize.height > screenSize.height) {
            frameSize.height = screenSize.height;
        }
        if (frameSize.width > screenSize.width) {
            frameSize.width = screenSize.width;
        }
        frame.setLocation((screenSize.width - frameSize.width) / 2,
                           (screenSize.height - frameSize.height) /
```

```

2);
    frame.setVisible(true);
}
public static void main(String[] args) {
    SwingUtilities.invokeLater(new Runnable() {
        public void run() {
            try {
                UIManager.setLookAndFeel(UIManager.
getSystemLookAndFeelClassName());
            } catch (Exception exception) {
                exception.printStackTrace();
            }

            new Application();
        }
    });
}
}

```

2、练习使用对话框。包括一个 JLabel 和两个 JButton，按任何一个 JButton 都会产生一个输入对话框，按确定后将输入内容在 JLabel 中显示出来

```

import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
public class dTest{
public static void main(String args[]){
Form form=new Form();
form.setSize(300,200);
form.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
form.setVisible(true);
}
}
class Form extends JFrame implements ActionListener{
private Dial dia;
JLabel lab=new JLabel(" 请输入 ");
public Form(){
JButton btn1=new JButton("Yes");
JButton btn2=new JButton("No");
getContentPane().setLayout(new GridLayout(3,1));

```

```

add(lab);
add(btn1);
add(btn2);
setTitle("inputprac");
btn1.addActionListener(this);
btn2.addActionListener(this);
}
public void actionPerformed(ActionEvent event){
if(event.getActionCommand()==" 输入"){
dia=new Dial(this," 请输入内容 ",true);
dia.setVisible(true);
lab.setText(dia.getSt());
}
else{
System.exit(0);
}
}
}
class Dial extends JDialog implements ActionListener{
private JTextField text=new JTextField("change");
private JButton btn1=new JButton(" 确定 ");
private JPanel panel=new JPanel();

public Dial(JFrame frame,String title,boolean b){
super(frame,title,b);
getContentPane().setLayout(new FlowLayout());
panel.add(text);
panel.add(btn1);
btn1.addActionListener(this);
getContentPane().add(panel);
pack();
}
public void actionPerformed(ActionEvent event){
this.dispose();
}
public String getSt(){
return text.getText();
}
}

```

3、.练习使用 JTable。包括姓名、学号、语文成绩、数学成绩、总分共

5 项

```
import java.awt.*;
import javax.swing.*;

public class table{
public static void main(String args[]){
JFrame form=new JFrame("Table 试验 ");
final Object[] row={" 姓名 "," 学号 "," 语文成绩 "," 数学成绩 "," 总分 "};
Object[][] rowData = {
{" 张三 ", "01", "81", "80", "160"},
{" 李四 ", "02", "93", "72", "165"},
{" 王麻子 ", "03", "99", "97", "196"},
};
JTable tb=new JTable(rowData,row);
JScrollPane pane= new JScrollPane (tb);
form.getContentPane().add(tb);
form.setVisible(true);
form.pack();
}
}
```

4、一个简单的 Swing GUI 应用程序。 在一个框架窗口中显示两个标签和一个按钮、上面的标签显示一串固定的文字信息，选择按钮后在下方的标签上显示系统现在的时间。

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.util.*;

public class SwingDemo extends JFrame
                                implements ActionListener{

    JButton b1;
    JLabel l1,l2;
    SwingDemo(){
        super("Swing 应用程序的例 ");
        l1=new JLabel(" 一个 GUI 应用程序的例子 ",JLabel.CENTER);
l2 =new    JLabel(" ");
        b1=new JButton(" 现在时间 [T]");
        b1.setMnemonic(KeyEvent.VK_T);
        b1.setActionCommand("time");
        b1.addActionListener(this);
        getContentPane().add(l1,BorderLayout.NORTH);
```

```

        getContentPane().add(l2, BorderLayout.CENTER);
        getContentPane().add(b1, BorderLayout.SOUTH);
    }
    public void actionPerformed(ActionEvent e)
    {
        Calendar c1 = Calendar.getInstance();
        if(e.getActionCommand().equals("time")){
            l2.setText(" 现在是 "
                +c1.get(Calendar.HOUR_OF_DAY)
                +" 时 "+c1.get(Calendar.MINUTE)+" 分");
            l2.setHorizontalAlignment(JLabel.CENTER);
        }else System.exit(0);
    }

    public static void main(String args[]){
        JFrame.setDefaultLookAndFeelDecorated(true);
        JFrame frame = new SwingDemo();
        frame.setDefaultCloseOperation(
            JFrame.EXIT_ON_CLOSE);
        frame.pack();
        frame.setVisible(true);
    }
}

```

5、创建一个界面质量管理器，并显示课程建设、一般课程、精品课程，并在精品课程张显示子菜单。

```

import java.awt.*;
import javax.swing.*;
import javax.swing.tree.*;
import java.awt.event.*;
public class TestJtree extends JFrame {
    JPanel cp=new JPanel();
    JTree jtree;
    DefaultMutableTreeNode root;
    public TestJtree() {
        this.setSize(300,300);
        this.setTitle("JTree 组件 ");
        cp=(JPanel)this.getContentPane();
        cp.setLayout(new BorderLayout());
        root=new DefaultMutableTreeNode(" 质量管理 ");
        createTree(root);
    }
}

```

```

        jtree=new JTree(root);
        cp.add(jtree, BorderLayout.CENTER);
    }
    public static void main(String[] args) {
        TestJtree TestJtree = new TestJtree();
        TestJtree.setVisible(true);
    }
    private void createTree(DefaultMutableTreeNode root)    {
        DefaultMutableTreeNode classroom=null;
        DefaultMutableTreeNode number=null;
        classroom=new DefaultMutableTreeNode(" 课程建设 ");
        root.add(classroom);
        number=new DefaultMutableTreeNode(" 一般课程 ");
        classroom.add(number);
        number=new DefaultMutableTreeNode(" 重点课程 ");
        classroom.add(number);
        number=new DefaultMutableTreeNode(" 精品课程 ");
        classroom.add(number);
        number.add(new DefaultMutableTreeNode(" 国家精品课程 "));
        number.add(new DefaultMutableTreeNode(" 市级精品课程 "));
        number.add(new DefaultMutableTreeNode(" 校级精品课程 "));
        number=new DefaultMutableTreeNode(" 其他课程 ");
        classroom.add(number);    }
    protected void processWindowEvent(WindowEvent e)    {
        if(e.getID()==WindowEvent.WINDOW_CLOSING)
            System.exit(0);
    }
}

```

6、使用文件打开、关闭对话框（ JFileChooser ）。将选 择的文件名显示到文本区域中。

```

import java.io.*;
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import javax.swing.filechooser.*;
public class JFileChooserDemo extends JFrame {
    public JFileChooserDemo() {
        super("使用 JFileChooser");
        final JTextArea ta = new JTextArea(5,20);
        ta.setMargin(new Insets(5,5,5,5));
    }
}

```

```

ta.setEditable(false);
JScrollPane sp = new JScrollPane(ta);
final JFileChooser fc = new JFileChooser();
JButton openBtn = new JButton(" 打开文件 ...");
openBtn.addActionListener(new ActionListener(){
    public void actionPerformed(ActionEvent e) {
        int returnVal = fc.showOpenDialog(
            JFileChooserDemo.this);
        if (returnVal == JFileChooser.APPROVE_OPTION) {
            File file = fc.getSelectedFile();
            ta.append("打开 : " + file.getName() + ".\n");
        } else ta.append("取消打开命令  .\n");
    }
});
JButton saveBtn = new JButton(" 保存文件 ...");
saveBtn.addActionListener(new ActionListener(){
    public void actionPerformed(ActionEvent e){
        int returnVal = fc.showSaveDialog(
            JFileChooserDemo.this);
        if (returnVal == JFileChooser.APPROVE_OPTION)
        {
            File file = fc.getSelectedFile();
            ta.append("Saving: " + file.getName() + ".\n");
        } else ta.append("取消保存命令。  \n");
    }
});
JPanel buttonPanel = new JPanel();
buttonPanel.add(openBtn);
buttonPanel.add(saveBtn);
Container c = getContentPane();
c.add(buttonPanel, BorderLayout.NORTH);
c.add(sp, BorderLayout.CENTER);
}

public static void main(String[] args) {
    JFrame frame = new JFileChooserDemo();
    frame.setDefaultCloseOperation(EXIT_ON_CLOSE);
    frame.pack();
    frame.setVisible(true);
}
}

```

7、编写一个对话框程序。

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
public class JOptionPaneDemo{
    public JFrame aFrame;
    public JButton aButton;
    public JOptionPane anOptionPane;
    static JLabel aLabel;
    public JOptionPaneDemo(){
        aFrame = new JFrame("使用 JOptionPane");
        aButton = new JButton(" 显示对话框 ");
        aLabel = new JLabel(" 您作出的选择是、    ",
            JLabel.CENTER);
        aButton.addActionListener(new OptionListener());
        anOptionPane = new JOptionPane();
        aFrame.getContentPane().add(aButton,
            BorderLayout.NORTH);
        aFrame.getContentPane().add(aLabel,
            BorderLayout.SOUTH);
        aFrame.setVisible(true);
        aFrame.setDefaultCloseOperation(
            JFrame.EXIT_ON_CLOSE);
    }
    public class OptionListener implements ActionListener{
        public void actionPerformed(ActionEvent e){
            String[] choices = {" 喜欢 ", " 不喜欢 "};
            int n = anOptionPane.showOptionDialog(aFrame,
                "您喜欢音乐吗？ ", " 请选择 ",
                JOptionPane.YES_NO_OPTION,
                JOptionPane.QUESTION_MESSAGE, null,
                choices,choices[0]);
            if(n == 0)
                JOptionPaneDemo.aLabel.setText(" 喜欢音乐 ");
            else JOptionPaneDemo.aLabel.setText("不喜欢音乐 ");
        }
    }
    public static void main( String[] args){
        new JOptionPaneDemo();
    }
}
```



```
}
```

8、编写一个功能选择程序，方法自定。

```
import javax.swing.*;
```

```
import java.awt.*;
```

```
import java.awt.event.*;
```

```
public class JMenuDemo extends JFrame implements ActionListener{
```

```
    JLabel j1 = new JLabel(" 请选择菜单、  ",JLabel.CENTER);
```

```
    JMenuItem aaMenuItem,baMenuItem;
```

```
    JMenuDemo(){
```

```
        super("使用 JMenu");
```

```
        JMenuBar aMenuBar = new JMenuBar();
```

```
        Icon i1 =    new ImageIcon("images/girl.gif");
```

```
        Icon i2 =    new ImageIcon("images/boy.gif");
```

```
        JMenu aMenu = new JMenu(" 菜单  A");
```

```
        JMenu bMenu = new JMenu(" 菜单  B");
```

```
        JMenuItem aaMenuItem = new JMenuItem(" 菜单项  AA",i1);
```

```
        JMenuItem abMenuItem = new JMenuItem(" 菜单项  AB",i2);
```

```
        JMenuItem baMenuItem = new JMenuItem(" 菜单项  BA");
```

```
        aMenuBar.add(aMenu);
```

```
        aMenuBar.add(bMenu);
```

```
        aMenu.add(aaMenuItem);
```

```
        aMenu.addSeparator();
```

```
        aMenu.add(abMenuItem);
```

```
        bMenu.add(baMenuItem);
```

```
        aaMenuItem.addActionListener(this);
```

```
        abMenuItem.addActionListener(this);
```

```
        baMenuItem.addActionListener(this);
```

```
        setJMenuBar(aMenuBar);
```

```
        getContentPane().add(j1,BorderLayout.CENTER);
```

```
    }
```

```
    public void actionPerformed(ActionEvent e)
```

```
    {
```

```
        JMenuItem source = (JMenuItem)(e.getSource());
```

```
        j1.setText(" 选择了菜单  :"+source.getText());
```

```
        j1.setHorizontalAlignment(JLabel.CENTER);
```

```
    }
```

```
    public static void main(String args[])
```

```
    {
```

```
        JFrame frame = new JMenuDemo();
```

```
        frame.setDefaultCloseOperation(
```

```

JFrame.EXIT_ON_CLOSE);
    frame.setVisible(true);
    frame.pack();
}
}

```

9、编辑一个登录界面，并用密码进行验证。

```

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
public class JPasswordFieldDemo extends JApplet
    implements ActionListener{

    JLabel jl1,jl2;
    JPasswordField jp1,jp2;
    JButton jb1,jb2;
    public void init(){
        Container c = getContentPane();
        jl1 = new JLabel(
            "<html><h3> 请输入您的密码 :",JLabel.CENTER);
        jl2 = new JLabel(
            "<html><h3> 请再次输入密码 :",JLabel.CENTER);
        jp1 = new JPasswordField(8);
        jp2 = new JPasswordField(8);
        jb1 = new JButton("<html><h3> 提交 ");
        jb2 = new JButton("<html><h3> 取消 ");
        c.setLayout(new GridLayout(3,2));
        c.add(jl1);
        c.add(jp1);
        c.add(jl2);
        c.add(jp2);
        c.add(jb1);
        c.add(jb2);
        jb1.addActionListener(this);
        jb2.addActionListener(this);
    }
    public void actionPerformed(ActionEvent e){
        if(e.getSource()==jb1)
            if(jp1.getPassword().length>0)
                if(String.valueOf(jp1.getPassword()).equals
                    (String.valueOf(jp2.getPassword())))
                    showStatus("密码输入成功！ ");
    }
}

```

```

        else{
            showStatus("两次输入的密码不同，请重新输入！");
            jp1.setText("");
            jp2.setText("");
        }
    else showStatus("密码不能为空！");
    if(e.getSource()==jb2){
        jp1.setText("");
        jp2.setText("");
        showStatus("");
    }
}
}
}

```

10、工资额计算器应用程序

```

package jiemian;
import java.awt.Color;
import java.awt.Font;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.*;
import org.dyno.visual.swing.layouts.Constraints;
import org.dyno.visual.swing.layouts.GroupLayout;
import org.dyno.visual.swing.layouts.Leading;

public class Frame1 extends JFrame {
    private static final long serialVersionUID = 1L;
    private JLabel jLabel0;
    private JLabel jLabel1;
    private JLabel jLabel2;
    private JLabel jLabel3;
    private JTextField jTextField0;
    private JTextField jTextField1;
    private JLabel jLabel4;
    private JButton jButton0;
    private JToggleButton jToggleButton0;
    private static final String PREFERRED_LOOK_AND_FEEL =
"javax.swing.plaf.metal.MetalLookAndFeel";

    public Frame1() {
        initComponents();
    }
}

```

```

private void initComponents() {
    setTitle(" 工资额计算器  ");
    setBackground(new Color(64, 128, 128));
    setLayout(new GroupLayout());
    add(getJLabel1(), new Constraints(new Leading(69, 10, 10), new
Leading(
    78, 10, 10)));
    add(getJLabel0(), new Constraints(new Leading(99, 169, 10, 10),
    new Leading(12, 35, 12, 12)));
    add(getJLabel2(), new Constraints(new Leading(57, 10, 10), new
Leading(
    72, 10, 10)));
    add(getJTextField0(), new Constraints(new Leading(134, 157, 10,
10),
    new Leading(69, 28, 12, 12)));
    add(getJLabel3(), new Constraints(new Leading(68, 12, 12), new
Leading(
    122, 10, 10)));
    add(getJTextField1(), new Constraints(new Leading(132, 160, 12,
12),
    new Leading(119, 28, 12, 12)));
    add(getJButton0(), new Constraints(new Leading(80, 77, 10, 10),
    new Leading(251, 12, 12)));
    add(getJToggleButton0(), new Constraints(new Leading(216, 82, 10,
10),
    new Leading(251, 10, 10)));
    add(getJLabel4(), new Constraints(new Leading(102, 162, 12, 12),
    new Leading(182, 34, 10, 10)));
    setSize(343, 300);
}

private JToggleButton getJToggleButton0() {
    if (jToggleButton0 == null) {
        jToggleButton0 = new JToggleButton();
        jToggleButton0.setFont(new Font("Dialog", Font.BOLD, 14));
        jToggleButton0.setText(" 取消 ");
    }
    jToggleButton0.addActionListener(new c(this));
    return jToggleButton0;
}

```

```
}
```

```
private JButton getJButton0() {  
    if (jButton0 == null) {  
        jButton0 = new JButton();  
        jButton0.setFont(new Font("Dialog", Font.BOLD, 14));  
        jButton0.setText(" 确定 ");  
    }  
    jButton0.addActionListener(new a(this));  
    return jButton0;  
}
```

```
private JLabel getJLabel4() {  
    if (jLabel4 == null) {  
        jLabel4 = new JLabel();  
        jLabel4.setFont(new Font("Dialog", Font.BOLD, 14));  
        jLabel4.setForeground(Color.red);  
    }  
    return jLabel4;  
}
```

```
private JTextField getJTextField1() {  
    if (jTextField1 == null) {  
        jTextField1 = new JTextField();  
        jTextField1.setText("");  
    }  
    return jTextField1;  
}
```

```
private JTextField getJTextField0() {  
    if (jTextField0 == null) {  
        jTextField0 = new JTextField();  
        jTextField0.setText("");  
    }  
    return jTextField0;  
}
```

```
private JLabel getJLabel3() {  
    if (jLabel3 == null) {  
        jLabel3 = new JLabel();  
    }
```

```

        jLabel3.setFont(new Font("Dialog", Font.BOLD, 14));
        jLabel3.setText(" 工资额、 ");
    }
    return jLabel3;
}

private JLabel getJLabel2() {
    if (jLabel2 == null) {
        jLabel2 = new JLabel();
        jLabel2.setFont(new Font("Dialog", Font.BOLD, 14));
        jLabel2.setText(" 工作时间、 ");
    }
    return jLabel2;
}

private JLabel getJLabel1() {
    if (jLabel1 == null) {
        jLabel1 = new JLabel();
    }
    return jLabel1;
}

private JLabel getJLabel0() {
    if (jLabel0 == null) {
        jLabel0 = new JLabel();
        jLabel0.setFont(new Font("Dialog", Font.BOLD, 26));
        jLabel0.setForeground(new Color(128, 0, 0));
        jLabel0.setText(" 工资额计算器 ");
    }
    return jLabel0;
}

class a implements ActionListener {
    private Frame1 adaptee;

    a(Frame1 adaptee) {
        this.adaptee = adaptee;
    }

    public void actionPerformed(ActionEvent e) {

```

```

        adaptee.b(e);
    }
}

public void b(ActionEvent l) {
    try {
        String a = jTextField0.getText();
        String b = jTextField1.getText();
        double m = Double.parseDouble(a);
        double n = Double.parseDouble(b);
        if (m < 0) {
            JOptionPane.showMessageDialog(null, " 工作时间不能小于
0 ! ", " 消息信息 ",
                JOptionPane.INFORMATION_MESSAGE);
        } else {
            if (m == 0 || m <= 40) {
                jLabel4.setText(" 总工资为、 " + String.valueOf(m * n));
            }
            if (m > 40) {
                jLabel4.setText(" 总工资为、 "
                    + String.valueOf(40 * n + 1.5 * (m - 40) * n));
            }
        }
    } catch (Exception ee) {
        JOptionPane.showMessageDialog(null, " 请输入正确的数字信
息 ! ", " 消息信息 ",
            JOptionPane.INFORMATION_MESSAGE);
    }
}

class c implements ActionListener {
    private Frame1 adaptee;

    c(Frame1 adaptee) {
        this.adaptee = adaptee;
    }

    public void actionPerformed(ActionEvent e) {
        adaptee.d(e);
    }
}

```

```

    }

    public void d(ActionEvent l) {
        jTextField1.setText("");
        jTextField0.setText("");
        jLabel4.setText("");
    }

    private static void installLnF() {
        try {
            String lnFClassname = PREFERRED_LOOK_AND_FEEL;
            if (lnFClassname == null)
                lnFClassname =
                UIManager.getCrossPlatformLookAndFeelClassName();
            UIManager.setLookAndFeel(lnFClassname);
        } catch (Exception e) {
            System.err.println("Cannot install " +
            PREFERRED_LOOK_AND_FEEL
            + " on this platform:" + e.getMessage());
        }
    }

    public static void main(String[] args) {
        installLnF();
        SwingUtilities.invokeLater(new Runnable() {
            @Override
            public void run() {
                Frame1 frame = new Frame1();

                frame.setDefaultCloseOperation(Frame1.EXIT_ON_CLOSE);
                frame.setTitle(" 工资额计算器 ");
                frame.getContentPane().setPreferredSize(frame.getSize());
                frame.pack();
                frame.setLocationRelativeTo(null);
                frame.setVisible(true);
            }
        });
    }
}

```

11、略

12、略