

第一部分 Java 语言概述

1 选择题

1-1 在下列概念中，Java 语言只保留了_____。

- A. 运算符重载 B. 方法重载 C. 指针 D. 结构和联合

1-2 下列关于 Java 语言特征的描述中，错误的是_____。

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

1-3 下列关于 Java Application 程序在结构上特点的描述中，错误的是_____。

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

1-4Java 程序经过编译后生成的文件的后缀是_____。

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

1-5 下列关于运行字节码文件的命令行参数的描述中，正确的是_____。

- A. 第一个命令行参数（紧跟命令字的参数）被存放在 args[0] 中。
B. 第一个命令行参数被存放在 args[1] 中。
C. 命令行的命令字被存放在 args[0] 中。
D. 数组 args[] 的大小与命令行参数的个数无关。

2. 判断题

1-1Java 语言是在 C++ 语言之前问世的。()

1-2Java 语言具有较好的安全性和可移植性及与平台无关等特性。()

1-3Java 语言中取消了联合概念，保留了结构概念。()

1-4Java 语言中数据类型占内存字节数与平台无关。()

1-5Java 语言中可用下标和指针两种方式表示数组元素。()

1-6Java 语言的源程序不是编译型的，而是编译解释型的。()

1-7 操作系统中进程和线程两个概念是没有区别的。()

1-8Java 语言既是面向对象的又是面向网络的高级语言。()

1-9Java 程序分为两大类：一类是 Application 程序，另一类是 Applet 程序。前者又称 Java 应用程序，后者又称 Java 小应用程序。()

1-10JavaApplication 程序是由多个文件组成的，其中可以有也可以没有主文件。()

1-11 组成 JavaApplication 的若干类中，有且仅有一个主类，只有主类中含有主方法 main()_。()

1-12JavaApplication 程序中，必有一个方法 main()，该方法有没有参数都可以。()

1-13Java 程序中是不区分大小写字母的。()

1-14 下列两个命令：javac 和 java 是不同的，编译时用前者，运行时用后者。()

1-15Java 程序编译后生成的是字节码文件，其后缀是 .exe。()

1-16 字节码文件机器是不能直接识别的，它要经过解释器，边解释边执行。()

1-17 在 Java 程序中，使用 import 语句引入类是在解释器中实现的。()

1-18 在运行字节码文件时，使用 java 命令，一定要给出字节码文件的扩展名 .class。()

1-19 如果一个 Java 程序中有多个类，编译后只生成一个字节码文件，其名字同主类名。()

1-20Java 程序中出现的输出方法 println() 和 print () 是完全一致的。()

1. B 2. D 3. C 4. C 5. A

1. 错 2. 对 3. 错 4. 对 5. 错 6. 对 7. 错 8. 对 9. 对 10. 错

11. 对 12. 对 13. 错 14. 对 15. 错 16. 对 17. 对 18. 错 19. 错 20. 错

第二部分 Java 语言语法基础

1、选择题

2-1Java 语言所用的字符集中字符是__B__位。

A. 8 B. 16 C. 32 D. 64

2-2 下列关于标识符的描述中，正确的是__A__。

A. 标识符中可以使用下划线和美元符

B. 标识符中可以使用连接符和#号符

C. 标识符中大小写字母是无区别的

D. 标识符可选用关键字

2-3Java 语言的各种分隔符中，非法的是__D__。

A. 空白符 B. 分号 C. 逗号 D. 问号

2-4 下列是 Java 语言中可用的注释符，其中错误的是__C__。

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

2-5 Java 语言中字符型数据的长度是__B__位。

A. 8 B. 16 C. 32 D. 64

2-6 下列描述的两种数据类型的长度不相等的是__A__。

A. 字符型和布尔型 B 字节型和布尔型 C 短整型和字符型

D 整型和单精度浮点型

2-7 下列是 Java 语言中的复合数据类型，其中错误的是__C__。

A. 数组 B. 接口 C. 枚举 D. 类

2-8 下列是关于 Java 语言中数据类型的强制转换的描述，其中错误的是__CD__。

A. 当数据类型从高转换为低时需要强制转换

B. 强制转换时使用强制转换运算符，形如 (<类型>)

C. 浮点型数据强制转换为整型时，小数部分四舍五入

D. 赋值操作中都要采用强制类型转换

2-9 下面是关于常量的描述中，错误的是__BD__。

A. Java 语言的常量有 5 种 B. 浮点型数 12.456 是单精度的

C. 布尔型常量只有两个可选值：true 和 false D. 字符串常量含有结束符 ‘\0’

2-10 下列关于定义符号常量的描述中，错误的是__A__。

A. 定义符号常量是用关键字 const B. 定义符号常量时要赋初值

C. 符号常量既可定义为类成员，又可在方法中定义

D. 定义符号常量时必须给出数据类型

2-11 下列关于变量的默认值的描述，其中错误的是__BC__。

A. 定义变量而没有进行初始化时，该变量具有默认值

B. 字符型变量的默认值为换行符

C. 布尔型变量的默认值为 true

D. 变量的默认值是可以被改变的

2-12 下列关于变量定义的描述中，正确的是__AC__。

A. 定义变量时至少应指出变量名字和类型

B. 定义变量时没有给出初值，该变量可能是无意义值

C. 定义变量时，同一个类型多个变量间可用逗号分隔

D. 定义变量时必须给变量初始化

2-13 下列关于变量作用域的描述，错误的是__D__。

A. 在某个作用域定义的变量，仅在该作用域内可见，而在该作用域外不可见

B. 在类中定义的变量的作用域在该类中的方法内可以使用

C. 在方法中定义的变量的作用域仅在该方法体内

D. 在方法中作用域可嵌套，在嵌套的作用域中可以定义同名变量

2-14 下列关于增 1 (++) 减 1 (--) 运算符的描述中，正确的是__AB__。

- A. 增 1 减 1 运算符都是单目运算符
- B. 增 1 减 1 运算符都具有前缀运算和后缀运算
- C. 增 1 减 1 运算符不会改变变量的值
- D. 增 1 减 1 运算符前缀运算和后缀运算后表达式值是相同的

2-15 下列关于运算符优先级的描述中，错误的是 D。

- A. 在表达式中，优先级高的运算符先进行计算
- B. 赋值运算符优先级最低
- C. 单目运算符优先级高于双目和三目运算符
- D. 逻辑运算符优先级高于逻辑位运算符

2-16 下列关于表达式的描述中，正确的是 AC。

- A. 任何表达式都有确定的值和类型
- B. 算术表达式的类型由第一个操作数的类型决定
- C. 逻辑表达式的操作数是逻辑型的
- D. 赋值表达式的类型取决于右值表达式的类型

2-17 下列表达式中，非法的是 B。

```
int a=5,b=6; double c=1.1,d=2.2;
```

- A. a+c+++d
- B. (a+c)--
- C. c<<b
- D. a!=b?c:d

2-18 下列关于条件语句的描述中，错误的是 AC。

- A. if 语句可以有多个 else 子句和 else if 子句
- B. if 语句中可以没有 else 子句和 else if 子句
- C. if 语句中的<条件>可以使用任何表达式
- D. if 语句的 if 体、else 体内可以有循环语句

2-19 下列关于开关语句的描述中，错误的是 BCD。

- A. 开关语句中，default 子句可以省略
- B. 开关语句中，case 子句的（语句序列）中一定含有 break 语句
- C. 开关语句中，case 子句和 default 子句都可以有多个
- D. 退出开关语句的唯一条件是执行 break 语句

2-20 下列关于循环语句的描述中，错误的是 BD。

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

2-21 下列循环语句的循环次数是 D。

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

- A. 0
- B. 1
- C. 5
- D. 无限

2-22 下列循环语句中，循环体被执行的次数为 D。

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

- A. 3
- B. 4
- C. 不确定
- D. 无限

2-23 下列关于 Java 语言的数组描述中，错误的是 D。

- A. 数组的长度通常用 length
- B. 数组下标从 0 开始
- C. 数组元素是按顺序存放在内存的
- D. 数组在赋初值和赋值时都不判届

2-24 下列关于数组的定义形式，错误的是 ACD。

- A. int[]a;a=new int;
- B. char b[];b=new char[80]

C. `int[]c=new char[10];` D. `int[]d[3]=new int[2][]`

2-25 下列关于字符串的描述中，错误的是 B 。

A. Java 语言中，字符串分为字符串常量和字符串变量两种

B. 两种不同的字符串都是 String 类的对象

C. Java 语言中不在使用字符数组存放字符串

D. JavaApplication 程序的 `main()` 的参数 `args[]` 是一个 String 类的对象数组，用它可存放若干个命令行参数

2、判断题

2-1Java 语言标识符中可以使用美元符。(对)

2-2Java 语言标识符中大小写字母是没有区别的。(错)

2-3 分号 (;)、逗号 (,) 和冒号 (:) 都可作为 Java 语言中的分隔符。(对)

2-4 文档注释符 `/**...*/` 是 Java 语言特有的注释符。(对)

2-5Java 语言的基础数据类型有 4 种：整形、浮点型、字符型和布尔型。(对)

2-6Java 语言的复合数据类型有 3 种：数组、类和包。(错)

2-7Java 语言中，字节型与字符型是一样的。(错)

2-8Java 语言是一种强类型语言，数据类型转换有两种：隐含转换和强制转换。(对)

2-9 数据由高类型转换到低类型时，采用强制转换，数据精度要受到损失。(对)

2-10 布尔型常量可以自动转换为短整型变量。(错)

2-11Java 语言使用的是 Unicode 字符集，每个字符在内存中占 8 位。(错)

2-12Java 语言的字符中部隐含结束符。(对)

2-13Java 语言中定义符号常量使用 `final` 关键字。(对)

2-14Java 语言中不同数据类型的长度是固定的，不随机器硬件不同而改变。(对)

2-15Java 语言中只有无符号的字符型。(对)

2-16 字符型变量中只存放一个字符。(对)

2-17 定义变量时必须进行初始化，否则变量具有无意义的值。(错)

2-18 若定义变量时不进行初始化，则该变量一定具有默认值。(对)

2-19Java 语言中，变量出现的位置只有一种，即为类体内定义的成员变量。()

2-20Java 语言规定在嵌套的程序块中不允许定义同名变量。()

2-21 求余运算符 % 可用于整数和浮点数。()

2-22 关系运算符组成的表达式是逻辑型表达式。()

2-23 逻辑运算符 `&` 和 `&&` 是没有区别的，都具有逻辑与的功能。()

2-24 运算符 `&` 是逻辑与运算符还是按位与运算符取决于操作数的类型。()

2-25 运算符 `>>` 和 `>>>` 都是右移位运算符，其区别在于对向右移出的位置是舍弃还是保留。()

2-26 赋值运算符组成的赋值表达式中，通常将右边的表达式的类型转换为左边变量的类型后再赋值。()

2-27 三目运算符的 3 个操作数中，第一个操作数的类型必须是逻辑型的。()

2-28 强制类型转换运算符的功能是将一个表达式的类型转换为所指定的类型。()

2-29 对象运算符 `instanceof` 是一个单目运算符，其表达式的值是类的对象。()

2-30 内存分配运算符 `new` 可以为创建的数组分配内存空间，但不能为创建的变量分配空间。()

2-31 运算符的优先级中，最低的是三目运算符。()

2-32 运算符的结合性中，除了三目运算符是自右向左。()

2-33 关系表达式和逻辑表达式的值都是布尔型的。()

2-34 赋值表达式的值不能是逻辑型的。()

2-35 条件表达式的类型总是 “?” 和 “:” 之间的操作数的类型。()

2-36 块语句可以嵌套，外层块语句定义的变量在内层块语句中是可见的。()

2-37 条件语句的嵌套结构中，一个 `if` 子句最多有一个 `else` 子句与它配对，而且一定是距离它最近的。()

2-38 在条件语句的嵌套结构中，一个 `if` 子句最多有一个 `else` 子句与它配对，而且一定是距离它最近的。
()

- 2-39 开关语句中，case 子句后面的<语句序列>可以是块语句。()
- 2-40 循环语句中，<条件>表达式必须是逻辑型表达式。()
- 2-41 在 Java 语言中，break 语句不能作用在<标号名>的左边。()
- 2-42Java 语言中，数组在静态和动态赋值时都判越界。()
- 2-43 说明或声明数组时不分配内存大小，创建数组时分配内存大小。()
- 2-44 基本数据类型的数组在创建时系统将指定默认值。()
- 2-45 数组在定义时可以进行初始化，使用初始值表。()
- 2-46Java 语言中数组元素只有下标表示，没有指针表示。()
- 2-47 创建数组时，系统自动将数组元素个数存放 length 变量中，可供用户对数组操作时使用。()
- 2-48 字符串可分为字符串常量和字符串变量两种，它们都是对象。()
- 2-49Java 语言中不使用字符数组存放字符串。()
- 2-50 下列创建字符串 s 的方法是错误的。()

```
char ch[]={ 'm' ,m' , ' n' , ' n' };  
char s[]=new char(ch);
```

1. 对 2. 错 3. 对 4. 对 5. 对 6. 错 7. 错 8. 对 9. 对 10. 错 11. 错 12. 对 13. 对
14. 对 15. 对 16. 对 17. 错 18. 对 19. 错 20. 对 21. 错 22. 对 23. 错 24. 对 25. 错
26. 对 27. 对 28. 对 29. 错 30. 错 31. 错 32. 错 33. 对 34. 错 35. 错 36. 对 37. 错 38. 对
39. 对 40. 对 41. 错 42. 对 43. 对 44. 对 45. 对 46. 对 47. 对 48. 对 49. 对 50.
对

第三部分 Java 语言面向对象的特征

1、选择题

3-1 下列对封装性的描述中，错误的是_B ____。

- A. 封装体包含了属性和行为
- B. 封装体中的属性和行为的访问权限是相同的
- C. 被封装的某些信息在封装体外是不可见的
- D. 封装使得抽象的数据类型提高了可重用性

3-2 下列关于继承性的描述中，错误的是 _C_。

- A. 一个类可以同时生成多个子类
- B. 子类继承了父类的所有成员
- C. Java 语言支持单重继承和多重继承
- D. Java 语言通过接口可使子类使用多个父类的成员

3-3 下列对多态性的描述中，错误的是__A ____。

- A. Java 语言允许运算符重载
- B. Java 语言允许方法重载
- C. Java 语言允许变量覆盖
- D. 多态性提高了程序的抽象性和简洁性

3-4 在类的修饰符中，规定只能被同一包类所使用的修饰符是_B ____。

- A. public
- B. 默认
- C. final
- D. abstract

3-5 在成员变量的修饰符中，规定只允许该类自身访问的修饰符是_A ____。

- A. private
- B. public
- C. 默认
- D. protected

3-6 在成员方法的访问控制修饰符中，规定访问权限包含该类自身、同包的其他类和其他包的该类子类的修饰符是__D ____。

- A. public
- B. private
- C. 默认
- D. protected

3-7 下列关于构造方法的特点的描述中，错误的是__A ____。

- A. 不可重载
- B. 方法名同类名
- C. 无返回类型
- D. 系统自动调用

3-8 下列关于构造方法的特点的描述中，错误的是_ D ____。

- A. 在类体内说明静态方法使用关键字 static

- B. 静态方法只能处理静态变量或调用静态方法
- C. 静态方法不占用对象的内存空间，非静态方法占用对象的内存空间
- D. 静态方法只能用类名调用

3-9 下列对静态初始化起的描述中，错误的是_ D ____。

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

3-10 下列关于抽象类的描述中，错误的是_ C ____。

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

3-11 下列关于类的继承性的描述中，错误的是_ D ____。

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

3-12 下列关于子类继承父类的成员的描述中，错误的是__A ____。

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

3-13 下列关于接口的描述中，错误的是__ B ____。

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

3-14 下列关包的描述中，错误的是__A ____。

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

3-15 下类常用包中，存放用户图形界面类库的包是_ A____。

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

3-16 下列常用包中，存放用户图形界面类库的包是__ B____。

- A. Math
- B. Object
- C. System
- D. String

2、判断题

3-1 类是一种类型，也是对象的模板。(对)

3-2Java 语言只支持单重继承，不支持多重继承。(对)

3-3 类中说明的方法可以定义在类体外。(错)

3-4class 不是定义类的唯一关键字。(错)

3-5 某类的对象可以作为另一类的成员。(对)

3-6 在类体内说明成员变量是不允许赋初值。(错)

3-7 最终变量就是 Java 语言的符号常量。()

3-8 静态变量的引入只能使用对象。(错)

3-9 静态方法只能处理静态变量。(对)

3-10 抽象方法是一种只有说明而无具体实现的方法。(对)

3-11 最终方法是不能被当前子类重新定义的方法。(对)

3-12Java 语言中，方法调用一律都是传址的引用调用。(错)

3-13 非静态方法中，方法调用一律都是传址的引用调用。(错)

3-14 静态初始化其实在构造方法被自动调用之前运行的。(对)

- 3-15 抽象方法仅有方法头，而无方法体。(对)
- 3-16 抽象方法一定出现在抽象类中。(对)
- 3-17 最终类、最终方法和最终变量的修饰符都用 final。(对)
- 3-18 创建对象是系统将调用适当的构造方法给对象初始化。(对)
- 3-19 使用运算符 new 创建对象时，赋给对象的值实际上是一个地址值。(对)
- 3-20 使用构造方法只能给非静态成员变量赋初值。(对)
- 3-21 创建对象时，该对象占有的内存空间除了非静态的成员变量外，还有非静态的成员方法。(对)
- 3-22 Java 语言中，对象成员的标是指使用运算符“.”。(对)
- 3-23 Java 语言中，对象赋值实际上是同一个对象具有两个不同的名字，因为它们都有同一个地址值。(对)
- 3-24 对象可作方法参数，对象数组不能做方法参数。(错)
- 3-25 Java 语言中，所创建的子类都应有一个父类。(对)
- 3-26 Java 语言中，类的继承是可以传递的。(对)
- 3-27 Java 语言中，构造方法是可以继承的。(错)
- 3-28 子类中构造方法应包含自身类的构造方法和直接父类的构造方法。(对)
- 3-29 调用 this 或 super 的构造方法的语句必须放在第一条语句。(对)
- 3-30. 子类对象可以直接赋值给父类对象，而父类对象不可以赋值给子类对象。(错)
- 3-31 子类中所继承父类的成员都可以在子类中访问。(错)
- 3-32 成员方法的重载和父该是一回事。(错)
- 3-33 一个类可以实现多个接口。接口可以实现“多重继承”。(对)
- 3-34 实现接口的类不能是抽象类。(对)

1. B 2. C 3. A 4. B 5. A 6. D 7. A 8. D 9. D 10. C 11. D 12. A 13. B 14. A
15. A 16. B

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

第四部分 JavaApplet 及其应用

1、选择题

4-1 Applet 类是属于_____包的。

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

4-2 下列关于 Applet 程序的描述中，错误的是_____。

- A. Applet 程序的主类必须是 Applet 类的子类
B. Applet 程序的主类中应有一个 main() 方法
C. Applet 不是完整的独立程序
D. Applet 的字节码文件必须嵌套在一个 HTML 文件中

4-3 下列关于 Applet 程序的描述中，错误的是_____。

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

4-4 在 Applet 类的主要方法中，用来实现初始化操作的是_____。

- A. init() B. stop() C. start() D. paint()

4-5 下列关于 HTML 文件的描述中，错误的是_____。

- A. HTML 文件是使用一种超文本标记语言（HTML）书写的文件

- B. HTML 文件中使用的标记通常使用一对尖括号括起的，两个标记之间是正文内容
- C. HTML 文件中没有单独出现的标记
- D. HTML 中不区分大小写字母

4-6 下列关于向 Applet 程序传递参数的描述中，错误的是_____。

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

4-7 下列关于字体的描述中，错误的是_____。

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

4-8 下列用来获取当前颜色的方法是_____。

- A. `getColor()` B. `setColor()` C. `getRed()` D. `Color()`

4-9 下列各种会址矩形的方法中，绘制实心的矩形的方法是_____。

- A. `fillRect()` B. `drawRect()` C. `clearRect()` D. `drawRoundrRect()`

4-10 下列演示图像的描述中，错误的是_____。

- A. 使用图像类 Image 定义图像对象
- B. 使用方法 `getImage()` 获取图像信息
- C. 使用方法 `drawImage()` 显示图像
- D. 不可使用显示图像的方法进行缩放

2、判断题

4-1Applet 程序是通过浏览器中内置的 Java 解释器来解释执行的。（ ）

4-2Applet 程序是以 .java 为扩展名的 Java 语言源程序，该程序经过编译器后便成为可执行文件。（ ）

4-3 嵌入到 HTML 文件中的是 Applet 的源程序。（ ）

4-4 运行 Applet 程序是使用 AppletViewer 命令运行嵌入了字节码文件的 HTML 文件，从而获得运行结果。（ ）

4-5Applet 程序中的主类必须是 Applet 类的子类。（ ）

4-6Applet 类是 Java 语言类库中的一个重要的系统类，它被存放在 java.awt 包中。（ ）

4-7init() 方法是用来完成初始化操作的，在 Applet 程序运行期间只执行一次。（ ）

4-8 start() 方法被系统自动调用开启动住县城运行。通常在 Applet 程序被重新启动时，该方法被系统自动调用。（ ）

4-9 paint() 方法是在需要重画图形界面时被系统自动调用来显示输出结果的。（ ）

4-10 stop() 方法是用来自暂停执行操作的，它与 start() 方法不同，只被调用一次。（ ）

4-11init(), start(), stop() 和 destroy() 4 个方法构成 Applet 程序的生命周期。（ ）

4-12HTML 语言中 <HEAD> 和 </HEAD> 是用来表示 HTML 文件开始和结束的标记。（ ）

4-13 HTML 中多数标记是成对出现的，也有不成对出现的标记。（ ）

4-14 HTML 中是区分大小写字母的。（ ）

4-15 通过 HTML 文件中使用的 PARAM 标记可以向 Applet 程序传递参数的。（ ）

4-16parseInt() 方法的功能是将 int 型数据转换为 String 型数据。（ ）

4-17AWT 提供了许多标准的 GUI 组件和布局管理等类。（ ）

4-18Java 语言采用 16 位颜色标准。Java 的调色板保证 128 色。（ ）

4-19Graphics 类提供了 3 种绘制文本的方法，其方法名都是 drawString()。（ ）

4-20 绘制椭圆的方法是 drawOval(), 使用该方法也可以绘制圆。（ ）

1. B 2. B 3. D 4. A 5. C 6. A 7. C 8. B 9. A 10. D

1. 对 2. 错 3. 错 4. 对 5. 对 6. 错 7. 对 8. 对 9. 对 10. 错 11. 对 12. 错 13. 对
14. 错 15. 对 16. 错 17. 对 18. 错 19. 对 20. 对

第五章

选择题

5-1 下列关于容器的描述中，错误的是 D

- A. 容器是由若干个组建和容器组成的
- B. 容器是对图形界面中界面元素的一种管理
- C. 容器是一种对指定宽和高的矩形范围
- D. 容器都是可以独立的窗口

5-2 下列界面元素中，不是容器的是 A

- A. List B. Frame C. Dialog D. Panel

5-3 下列关于实现图形用户界面的描述中，错误的是 D

- A. 放在容器中的组件首先要定义，接着要初始化
- B. 放在容器中的多个组件是要进行布局的，默认的布局策略是 FlowLayout
- C. 容器中所有组件都是事件组件，都可以产生事件对象
- D. 事件处理是忧监听者定义的方法来实现的

5-4 下列关于组件类的描述中，错误的是 A

- A. 组件类中包含了文本组件类 (TextComponent) 和菜单组件类 (MenuComponent)
- B. 标签 (Label) 和按钮 (Button) 是组件类 (Component) 的子类
- C. 面板 (Panel) 和窗口 (Window) 是容器类 (Container) 的子类
- D. 文本框 (TextField) 和文本区 (TextArea) 是文本组件类 (TextComponent) 的子类

5-5 在对下列语句的解释中，错误的是 C

`but.addActionListener(this);`

- A. but 是某种事件对象，如按钮事件对象
- B. this 表示当前容器
- C. ActionListener 是动作事件的监听者
- D. 该语句的功能是将 but 对象注册为 this 对象的监听者

5-6 所有事件类的父类是 C

- A. ActionEvent B. AwtEvent C. KeyEvent D. MouseEvent

5-7 所有 GUI 标准组件类的父类是 C

- A. Buttom B. List C. Component D. Container

5-8 下列各种布局管理器中 Window 类、Dialog 类和 Frame 类的默认布局是 C

- A. FlowLayout B. CardLayout C. BorderLayout D. GridLayout

5-9 在下列各种容器中，最简单的无边框的又不能移动和缩放的只能包含在另一种容器中的容器是 D

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

5-10 下列关于菜单和对话框的描述中，错误的是 C

- A. Frame 容器是可以容纳菜单组件的容器
- B. 菜单条中可包含若干个菜单，菜单中又可包含若干菜单项，菜单项中还可包含菜单子项
- C. 对话框与 Frame 一样都可作为程序的最外层容器
- D. 对话框内不含有菜单条，它由 Frame 弹出

判断题

- 5-1 AWT 是抽象窗口工具包的英文缩写。(对)
- 5-2 容器中只能包含有组件，而不能再含有容器。(错)
- 5-3 容器分可以独立的窗口和不可以独立的窗口两种。(对)
- 5-4 单选按钮提供“多选一”的输入选择，下拉列表提供“多选多”的输入选择。(错)
- 5-5 所有容器的默认布局都是 **FlowLayout**。(错)
- 5-6 所有的组件都是事件源。(错)
- 5-7 Java 2 的事件处理方法是采用委托事件处理模型。(对)
- 5-8 事件组件都应注册一个事件监听者。(对)
- 5-9 事件监听者除了得知事件的发生外，还应调用相应方法处理事件。(对)
- 5-10 所有事件的父类是 **EventObject** 类。(对)
- 5-11 图形用户界面是由用户自定义成分、控制组件和容器组成的。(对)
- 5-12 所有 GUI 标准组件类的父类是 **Component** 类。(对)
- 5-13 所有容器类的父类是 **Frame** 类。(错)
- 5-14 标准和按钮都是事件源。(对)
- 5-15 文本去和文本框都可以引发两种事件：文本改变事件和动作事件。(错)
- 5-16 每个复选框有两种状态：“打开”和“关闭”状态。(对)
- 5-17 复选框组 (**CheckboxGroup**) 是由若干个按钮组成的。在一组按钮中可选多个。(错)
- 5-18 列表 (**List**) 是由多个选项组成的，只可在列表框的若干个选项选取一个。(错)
- 5-19 下拉列表 (**Choice**) 是一种“多选多”的输入界面。(错)
- 5-20 滚动条有两类：水平滚动条和垂直滚动条。(对)
- 5-21 画布 (**Canvas**) 是一种可容纳多个组件的容器。(错)
- 5-22 所有容器都采用顺序布局 (**FlowLayout**) 作为默认布局管理器。(错)
- 5-23 边界布局 (**BorderLayout**) 将容器分为东、西、南、北共 4 个区域。(错)
- 5-24 卡片布局 (**CardLayout**) 可使容器容纳多个组件，在同一时刻只显示若干个组件中的一个。(对)
- 5-25 容器可容纳组件和容器，同一个容器中的组件可被同时显示或者同时隐藏。(对)
- 5-26 所有容器都是有边框的。(错)
- 5-27 **Panel** 容器是可以嵌套使用的。(对)
- 5-28 **Applet** 实际是一种特殊的 **Panel** 容器。**Java Applet** 程序只负责它拥有的 **Applet** 容器内的无边框区域。(对)
- 5-29 窗口 (**Window**) 容器是一种独立窗口，但它不能作为程序的最外层容器。(对)
- 5-30 **Frame** 容器是有边框的容器，它也是一种独立窗口，只能作为最外层容器。(对)
- 5-31 **Frame** 容器可以容纳菜单组件，它实现了 **MenuContainer** 接口。(对)
- 5-32 **MenuComponent** 类是菜单组件的父类。(对)
- 5-33 菜单条 (**MunuBar**)、菜单 (**Munu**)、菜单项 (**MunuItem**) 是 3 中不同的菜单组件。(错)
- 5-34 对话框 (**Dialog**) 不是一种独立使用的容器。(错)
- 5-35 文件对话框 (**FileDialog**) 是对话框 (**Dialog**) 的子类，它是一种打开文件和保留文件的对话框窗口。(对)

分析程序输出结果

5-1 Exer5_1.java

```
import java.applet.*;
```

```
import java.awt.*;
```

```
public class Exer5_1 extends Applet
```

```
{
    private Label lab1,lab2;
    public void init()
    {
        lab1=new Label();
        lab1.setText("标签文本左对齐");
        lab1.setAlignment(Label.LEFT);
        lab2=new Label("标签文本右对齐",Label.RIGHT);
        setLayout(new GridLayout(3,3));
        add(lab1);add(lab2);
    }
}
```

运行改程序后，输出结果如图 5-5 所示。

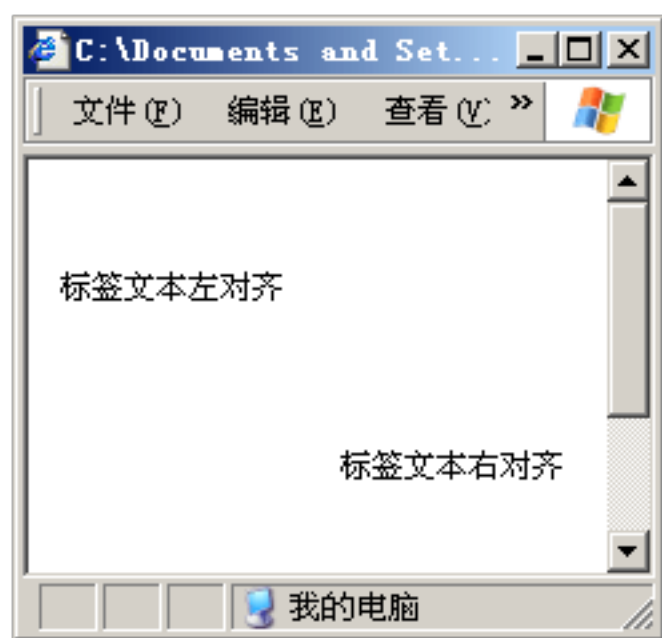


图 5-5 分析题 5-1 输出结果

5-2 Exer5_2.java

```
import java.applet.*;
```

```
import java.awt.*;
```

```
import java.awt.event.*;
```

```
public class Exer5_2 extends Applet implements ActionListener,ItemListener
```

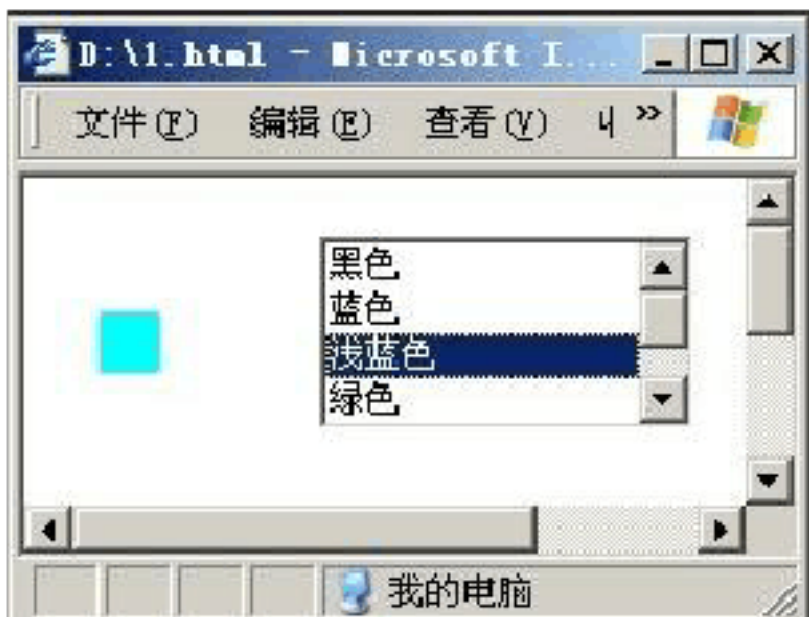
```
{
    List colList;
    String colName[]={"黑色","蓝色","浅蓝色","绿色","红色","黄色"};
    Color cols[]={Color.black,Color.blue,Color.cyan,Color.green,Color.red,Color.yellow};
    int t;
    public void init()
    {
        colList=new List(4,false);
        colList.addActionListener(this);
        colList.addItemListener(this);
        for(int i=0;i<colName.length;i++)
```

```

        colList.add(colName[i]);
        add(colList);
    }
    public void actionPerformed(ActionEvent e)
    {}
    public void itemStateChanged(ItemEvent e)
    {
        showStatus(colList.getSelectedItem()+";index"+colList.getSelectedIndex());
        t=colList.getSelectedIndex();
        repaint();
    }
    public void paint(Graphics g)
    {
        g.setColor(cols[t]);
        g.fillRect(15,30,20,20);
    }
}

```

运行改程序后，输出结果如图 5-6 所示。



5-3 Exer5_3.java

```

import java.awt.*;
import java.awt.event.*;
public class Exer5_3 extends Frame implements ActionListener
{
    Frame fr;
    Panel deck;
    CardLayout c1;
    Button but[];
    String name[]={"第一张","下一张","前一张","最末张"};
    public Exer5_3()
    {
        fr=new Frame("卡片窗口");
        deck=new Panel();
        c1=new CardLayout();
        deck.setLayout(c1);

        Panel c1=new Panel();
        Label lab1=new Label("卡片 1");
    }
}

```

```

        c1.add(lab1);
        deck.add(c1,lab1.getText());

        Panel c2=new Panel();
        Label lab2=new Label("卡片 2");
        c2.add(lab2);
        deck.add(c2,lab2.getText());

        Panel c3=new Panel();
        Label lab3=new Label("卡片 3");
        c3.add(lab3);
        deck.add(c3,lab3.getText());

        Panel c4=new Panel();
        Label lab4=new Label("卡片 4");
        c4.add(lab4);
        deck.add(c4,lab4.getText());

        Panel contr=new Panel();
        contr.setLayout(new GridLayout(2,2));
        but=new Button[name.length];
        for(int i=0;i<name.length;i++)
        {
            but[i]=new Button(name[i]);
            but[i].addActionListener(this);
            contr.add(but[i]);
        }
        fr.add(contr,BorderLayout.WEST);
        fr.add(deck,BorderLayout.EAST);
        fr.setSize(300,150);
        fr.show();
    }
    public void actionPerformed(ActionEvent e)
    {
        if(e.getSource()==but[0])
            c1.first(deck);
        if(e.getSource()==but[1])
            c1.next(deck);
        if(e.getSource()==but[2])
            c1.previous(deck);
        if(e.getSource()==but[3])
            c1.last(deck);
    }
    public static void main(String args[])
    {
        Exer5_3 m1=new Exer5_3();
    }

```

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/138005114115006046>