

---

## 第一部分

### 基础知识练习

#### 目标

本章对应于《学生指南》各章的内容分别提供了练习题集，包括：

- 第一章 Java 入门
- 第二章数据类型和运算符
- 第三章流程控制与数组
- 第四章封装
- 第五章继承
- 第六章抽象类与接口
- 第七章多态
- 第八章异常
- 第九章多线程机制
- 第十章输入输出流
- 第十一章使用泛型和集合框架
- 第十二章基于 Swing 的图形用户界面(GUI)设计
- 第十三章 Java 事件驱动编程

---

## 第一章练习题（Java 入门）

1. 下列哪项不是 JDK 所包含的内容？（选一项）
  - 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
  
2. 下列关于 JDK、JRE 和 JVM 的描述。哪项正确？
  - A. JDK 中包含了 JRE，JVM 中包含了 JRE
  - B. JRE 中包含了 JDK，JDK 中包含了 JVM
  - C. JRE 中包含了 JDK，JVM 中包含了 JRE
  - D. JDK 中包含了 JRE，JRE 中包含了 JVM
  
3. 下列哪个工具可以编译 java 源文件？
  - A. javac
  - B. jdb
  - C. javadoc
  - D. junit
  
4. JDK 工具 javadoc 的作用是哪项？
  - A. 生成 Java 文档
  - B. 编译 Java 源文件
  - C. 执行 Java 类文件
  - D. 测试 Java 代码
  
5. 以下哪些包是 Java 标准库中常用的包？（选三项）
  - A. java. lang
  - B. javax. servlet .http
  - C. java. io
  - D. java. sql

- 
6. 使用 JDK 工具生成的 **Java** 文档的文件格式是？
- A. XML 格式
  - B. 自定义格式
  - C. 二进制格式
  - D. HTML 格式
7. 以下关于 JVM 的叙述，哪项正确？（选两项）
- A. JVM 运行于操作系统之上，它依赖于操作系统
  - B. JVM 运行于操作系统之上，它与操作系统无关
  - C. JVM 支持 **Java** 程序运行，它能够直接运行 **Java** 字节码文件
  - D. JVM 支持 **Java** 程序运行，它能够直接运行 **Java** 源代码文件
8. 以下关于支持 **Java** 运行平台的叙述，哪项错误？
- A. **Java** 可在 **Solaris** 平台上运行
  - B. **Java** 可在 **Windows** 平台上运行
  - C. **Java** 语言与平台无关。**Java** 程序的运行结果依赖于操作系统
  - D. **Java** 语言与平台无关。**Java** 程序的运行结果与操作系统无关
9. 以下关于 **Applet** 和 **Java** 程序之间关系的叙述，哪项错误？
- 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
10. 以下关于 **Java HotSpot** 的描述，哪两项错误？（选两项）
- A. **JavaHotSpot** 是一种热编译技术，在编译 **Java** 源程序时会被使用
  - B. **Java HotSpot** 是一种热编译技术，在运行 **Java** 代码时会被使用
  - C. **Java HotSpot** 是一种热编译技术，它只对程序的部分字节码进行优化
  - D. **Java HotSpot** 是一种热编译技术，它会对程序的全部字节码进行优化

---

11. 环境变量 **PATH** 中含有多个路径时，路径和路径之间可以用哪项来隔开？

- A.:
- B. ,
- C.\*
- D:|

12. **CLASSPATH** 中的 “.” 的含义是哪项？

- A. 省略号
- B. 当前目录
- C. 所有目录
- D. 上级目录

13. **JVM** 在执行一个 **Java** 类时，大致采用以下过程？

- A. 执行类中的代码→装载类→校验类
- B. 校验类→装载类→执行类中的代码
- C. 装载类→执行类中的代码→校验类
- D. 装载类→校验类→执行类中的代码

14. 当运行一个 **Java** 程序时，传递参数的格式是哪项？

- A. **java** 类名参数 1，参数 2
- B. **javac** 类名参数 1 参数 2
- C. **java** 类名参数 1 参数 2
- D. **java** 类名参数 1+参数 2

15. 以下关于 **Java** 文件名的叙述，正确的有？（选两项）

- A. **Java** 源文件的扩展名应为 **.java**
- B. **Java** 源文件的文件名应与文件中的类名一致
- C. **Java** 字节码文件的扩展名应为 **.java**
- D. 一个 **Java** 源文件中只能包含一个 **Java** 类

---

IC  
2D  
3A  
4A  
SACD  
6D  
7AC  
8C  
9B  
10 A D  
11 A  
12 B  
13 D  
14 C  
15 A B

---

## 第二章练习题（数据类型和运算符）

1. 下列哪项不属于 Java 语言的基本数据类型？
  - A. int
  - B. String
  - C. double
  - D. boolean
  
2. 下列哪项不是 int 类型的字面量？
  - A. \u03A6
  - B. 077
  - C. 0xABBC
  - D. 20
  
3. 下列哪项不是有效的标识符？
  - A. userName
  - B. 2test
  - C. \$change
  - D. \_password
  
4. 下列哪项是 Java 语言中所规定的注释样式？（选三项）
  - A. // 单行注释
  - B. --单行注释
  - C. /\*  
    \*单行或多行注释  
    \*/
  - D. /\*\*  
    \*文档注释  
    \*/
  
5. 下列哪项不是 Java 语言的关键字？
  - A. goto
  - B. sizeof
  - C. instanceof
  - D. volatile

---

6. 现有如下五个声明:

Line1: int a\_really\_really\_really\_long\_variable\_name=5 ;

Line2: int \_hi=6;

Line3: int big=Integer.getInteger("7");

Line4: int \$dollars=8;

line5: int %opercent=9;

哪行无法通过编译?

A. Line1

B. Line3

C. Line4

D. Line5

7. 现有:

1. class Top {

2. static int x=1;

3. public Top (inty) { x\*=3; }

4. }

5. class Middle extends Top {

6. public Middle() {x+=1; }

7. public static void main (String [] args) {

8. Middle m = new Middle();

9. System.out.println (x);

10. }

11. }

结果为:

A. 1

B. 2

C. 3

D. 编译失败

8. 现有:

1. class Passer f

2. static final int X=5;

3. public static void main (String [] args) {

4. new Passer(). go (x);

5. System.out.print (x);

6. }

7. void go (int x) {

8. System.out.print(x++);

9. }

10. }

---

结果是什么？

- A. 55
- B. 56
- C. 65
- D. 66

9. 现有：

```
1. class Wrench f
2. public static void main(String [] args) {
3. Wrench w=new Wrench();    Wrench w2=new Wrench();
4.     w2=go (w, w2);
5. System.out.print (w2==w);
6.     }
7. static Wrench go (Wrench wr1,  Wrench wr2) {
8. Wrench wr3=wr1; wr1=wr2; wr2=wr3;
9.     return wr3;
10.    }
11. }
```

结果是什么？

- A. false
- B. true
- C. 编译失败
- D. 运行的时候有异常抛出

10. 现有：

```
5. class Wrench2 {
6.     int size;
7. public static void main(String [] args) {
8. Wrench2 w=new Wrench2();
9.     w.size=11;
10.    Wrench2 w2=go(w, w.size);
11.    System. out .print (w2. size);
12.    )
13.    static Wrench2 go(Wrench2 wr. int s) {
14.    S=12;
15.    return wr;
16.    }
17. }
```



---

结果为:

- A. 11
- B. 12
- c. 编译失败。
- D. 运行时异常被抛出

11. 现有:

```
class Test2 {  
    public static void main (String [] args) {  
        short a, b, C;  
        a=1;  
        b=2;  
        C=a+b;  
        a+=2;  
    }  
}
```

以上代码中, 哪一句是错误的?

- A. a=1;
- B. C=a+b;
- C. a+=2;
- D. short a, b,C;

12. 表达式:  $1-2/5+2 \times 5$  的结果是哪项?

- A. 10.6
- B. 9.8
- C. 9

13. 现有代码片段:

```
String s="123";  
String sl=S+456;  
请问 sl 的结果是哪项?
```

- A. 123456
- B. 579
- C. 编译错误
- D. 运行时抛出异常

---

14. 基本数据类型 `float` 的包裹类是哪项?

- A. Integer
- B. Double
- C. Float
- D. Character

15. 现有:

```
1. class Test4 {  
2.     public static void main (String [] args) {  
3.         boolean X=true;  
4.         boolean y=false;  
5.         short Z=42;  
6.  
7.         if((z++==42) && (y=true))z++;  
8. if((x=false) || (++z==45)) z++;  
9.  
10.    System. out.println("z="+z);  
11.    }  
12. }
```

结果为:

- A. Z=42
- B. z=44
- C. Z= 45
- D. z= 46

---

## 第二章练习题

参考答案

1B

2A

3B

4ACD

5B

6D

7D

8A

9B

10 A

11 B

12 D

13 A

14 C

15 D

---

## 第三章练习题

### 第三章练习题（流程控制与数组）

1. 现有：

```
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哪项？

- A. 0123
- B. 012456789
- C. 0123456789
- D. 012

2. 程序：

```
class TestApp{
public static void main (String[] args){
int x=6;
if (x>1)
System.out.println("x>1");
else if (x>5)
System.out.println("x>5");
else if (x<10)
System.out.println("x<10");
else if (x<29)
System.out.println("x<29");
else
System.out.println ("以上都不是");
}
}
```

上述程序运行后的结果是哪项？

- A. x>5
- B. x>1
- C. x<10
- D. x<29

---

3. 现有:

```
class TestApp{
public static void main (String[] args){
int [5] myarray={ 10, 11, 12, 13, 14};
int sum=0;
for (int x : myarray)
sum+=x;
System. out. println( "sum= "+sum);
}
}
```

上述程序运行后的结果是哪项?

- A. sum=10
- B. sum=70
- C. sum=60
- D. 运行时抛出异常

4. 下列有关数组的声明中, 正确的是哪项? (选两项)

- A. int s[10];
- B. int[10] s;
- C. int[5] s={1,2,3,4,5};
- D. int s[];

5. 已知数组 array, 其最后一个元素的下标是?

- A. array. size
- B. array. length-1
- C. array. size-1
- D. array. length

6. 程序:

```
class TestApp{
public static void main (String[] args){
int X=5:
switch (x){
case 1:
case 2:
case 3:
System. out. println (“一季度”);
break;
case 4:
case 5:
```

---

```
case 6:
System.out. println (“二季度”);
break;
default:
System.out. println (“三季度以上”);
break;
)
}
)
```

上述程序运行后的结果是哪项？

- A. 一季度
- B. 二季度
- c. 三季度以上
- D. 无输出

7. 为将数组 myArray 的长度由 3 改为 6，现采取以下编码：

```
int[]myArray=new int [3];
myArray=new int[6];
```

代码执行后，以下叙述哪项是正确的？

- A. 数组 myArray 的长度已由 3 改为 6，其中前 3 个元素的值不变，后 3 个元素的值为空。
- B. 数组 myArray 的长度已由 3 改为 6，其中前 3 个元素的值不变，后 3 个元素需再经过初始化后才能使用。
- C. 数组 myArray 的长度没有变化。
- D. 数组 myArray 的长度已由 3 改为 6，原来 3 个元素的值全部丢失。

8. 现有：

```
1. class lfs {
2. public static void main (String [] args) {
3. boolean state=false;
4. int i=2;
5. if( ++i>2) && (state=true))
6. i++;
7. if( ++i>4) || (state=false))
8. i++;
9. System.out .println (i);
10. }
11. }
```

---

结果为:

- A. 6
- B. 5
- C. 4
- D. 编译失败

9. 现有:

```
3. public class Tester {  
4. public static void main (String[] args) {  
5.     int x=5;  
6.     Integer x1=x; Integer x2=x;  
7. int x3=new Integer(5);  
8.     system.out.print(x1.equals(x));  
9.     system.out.print(x1==x);  
10.    system.out.print(x2.equals(x1));  
11.    system.out.print(x2==x1);  
12-   system.out.print(x2==x3);  
13-   system.out.print(x2.equals(x3));  
    " 14. }  
15. }
```

结果为:

- A. 编译失败
- B. falsefalse true true true true
- C. true true true true true true
- D. false false true true true false
- E. true false true false false true
- F. 运行时异常被抛出

10. 现有:

```
1. class Rectangle {  
2' public static void main(string [] args) {  
3.     int[] x={1, 2, 3};  
4.     x[1]= (x[1]>1) ? x[2]: 0;  
5.     System.out. println(x[1]);  
6.     }  
7.     }
```

---

结果为:

- A. 3
- B. 2
- C. 1
- D. 0

11. 现有:

```
1. class Output {  
2.     public static void main (String[] args) {  
3.         int i=5;  
4.         System.out.print( "4"+i+"");  
5.         System.out.print (i+5+"7");  
6.         System.out.println (i+"8");  
7.     }  
8. }
```

结果为:

- A. 9 9722
- B. 9 55758
- C. 45 10758
- D. 45 9722

12. 以下哪种初始化数组的方式是错误的?

- A. `String[] names={"zhang", "wang", "li"};`
- B. `String names[] =new String [3];`  
`names [0] ="zhang"; names [1] ="wang"; names [2] ="li";`
- C. `String[3] names={"zhang", "wang", "li"};`
- D. 以上皆正确

13. 现有:

```
1. class WhileTests {  
2.     public static void main (String [] args) {  
3.         int X=5;  
4.         while (++x<4) {  
5.             --x;  
6.         }  
7.         System.out.println( "x="+x);  
8.     }  
9. }
```



---

结果是什么？

- A.X=6
- B. X=5
- C.X=2
- D. 编译失败

14. 现有：

```
1. class Test2 f
2. public static void main (String [] args) {
3.     boolean X= true;
4.     boolean y=false;
5.     short Z=20;
6.
7. if((x==true) && (y=true)) z++;
8.     if((y==true) || (++z==22)) z++;
9.
10.    System.out.println("z="+z);
11.    }
12. }
```

结果是什么？

- A.Z=21
- B. z=22
- C. z=23
- D. Z= 24

15.现有：

```
1. class Foo {
2. public static void main (String [] args) {
3.     int x=0;
4.     int y=4;
5. for (int z=0; z<3; Z++; X++) {
6.     if(x>1& ++y<10)
7.         y++;
8.     }
9. System.out.println (y);
10.    }
11. }
```

结果是什么？

- A. 7
- B. 8
- C. 10
- D. 12

---

I D  
2B  
3C  
4CD  
5B  
6B  
7D  
8A  
9C  
11 C  
12 C  
13 A  
14 B  
15 B

---

## 第四章练习题（封装）

别

1. 下列有关类、对象和实例的叙述，正确的是哪一项？
  - A. 类就是对象，对象就是类，实例是对象的另一个名称，三者没有差别
  - B. 对象是类的抽象，类是对象的具体化，实例是对象的另一个名称
  - C. 类是对象的抽象，对象是类的具体化，实例是类的另一个名称
  - D. 类是对象的抽象，对象是类的具体化，实例是对象的另一个名称
2. 下列类 Account 的构造方法中，声明正确的是？
  - A. Account (String name){}
  - B. Account (String name)
  - C. Account (name)()
  - D. NewAccount (String name){}
3. 类 Account 中字段声明正确的是哪项？
  - A. class Account f  
name;  
amount;  
)
  - B. class Account{  
String name=l.0;  
double amount="Mike";  
)
  - C. class Accountf  
String name;  
double amount;  
)
  - D. class Account{  
String name= "Mike,,, double amount=1000.0;  
)
4. 类 Account 中方法声明正确的是哪一项？
  - A. class Account f  
deposit();  
)
  - B. class Accountf  
void deposit();  
}

---

C.    class   Accountf  
void deposit(){  
  )  
  )

D. class Accountf  
void   deposit{()  
  )  
  )

5. 下列有关类声明的代码片段，哪一项是正确的？

A. package school;  
import java. sql.\*;  
class Student {  
  }  
  }

B. import java. sql.\*;  
package school;  
class   Student f  
  )  
  )

C.    package school;  
class Student{  
  }  
  }

import java. sql.\*j  
D. package school;  
import java. sql.\*;  
private String name;  
class Student {  
  }  
  }

6. 有关 **new** 关键字的描述正确的是哪项？

A.创建对象实例的时候可以不使用 **new** 关键字

B. **new** 所创建的对象不占用内存空间

C. **new** 会调用类的构造器来创建对象

D. **new** 所创建的对象一定存在引用变量

7. 下列哪些是方法 `public int add (int a)` 的重载方法？（选三项）

A. `public int add (long a);`

B. `public void add (int a);`

C. `public void add (long a);`

D. `public int add (float a);`

---

8. 我们定义一个 **Account** 类来描述银行账户，银行账户有账户名、金额等属性

特征，同时有存款、取款等行为特征，下述代码适合描述的是哪项？

A. class Accountf

String name; // 账户

String amount; // 金额

Account (String name){ }

void deposit (double mount){ // 存款

)

void withdraw (double mount){ // 取款

}

)

B. class Accountf

String name; // 账户

doujole amount; // 金额

Account (double amount){ }

void deposit (double mount){ // 存款

)

void withdraw (double mount){ // 取款

)

)

C. class Accountf

String name; // 账户

double amount; // 金额

Account (String name){ }

void deposit (double mount){ // 存款

)

void withdraw (double mount){ // 取款

)

)

D. class Accountf

String name; // 账户

double amount; // 金额

Account (String name){ }

void deposit(){ // 存款

)

void withdraw(){ // 取款

)

)

---

9. 现有:

```
1. class Banana2 f
2.     static int X=2;
3. public static void main (String [] args) {
4.     int X=2;
5. Banana2 b=new Banana2();
6.     b. go(x);
7. }
8.     static {x+=x; }
9.     void go (int x) {
10.        ++x;
11.        System. out.println (x);
12.    }
13. }
```

结果为:

- A. 7
- B. 5
- C. 3
- D. 2

10. 现有:

```
1. class TestFoo f
2.     int x;
3.     String y;
4. int  getX() { return x; }
5. String getY() {return y; }
6.     void setX (int x) {
7.         int Z=7;
8.         this.x=x;
9.     }
10. }
```

可以添加多少个修饰符来封装此类?

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

---

11. 定义枚举如下:

```
public enum Direction{  
    EAST,SOUTH,WEST,NORTH  
}
```

F 列正确使用该枚举类型的语句是哪项?

- A. Direction Direction=EAST;
- B. Direction direction=Direction.WEST;
- C. int a= Direction.NORTH;
- D. Direction direction=2;

12. 定义类:

```
package utils;  
public class Rep{  
    public static String twice (String s){return s+s ; }  
}
```

再定义另一个类 Demo:

```
1.//insert code here  
2.public class Demo{  
3.    public static void main (String[] args){  
4.System.out.println( twice( "Hello"));  
5.    }  
6. }
```

在第一行插入哪项代码, 可以使程序正常编译和执行?

- A. import utils.\*;
- B. import utils.Rep.\*;
- C. import static utils.Rep.twice;
- D. static import utils.Rep.twice;

13. 现有:

```
public class TestDemo{  
    private int X=2;  
    static int y=3;  
    public void method(){  
        final int i=100;  
        int j =10;  
        class Cinner {  
            public void mymethod(){  
                //Here  
            }  
        }  
    }  
}
```

---

在 Here 处可以访问的变量是哪些？（选三项）

- A. X
- B. y
- C. j
- D. i

14. 现有如 F 包结构：

```
com
|--- X
|   |--- Alpha.class
|   |
|   |--- y
|       |--- Beta.class
|
|--- Gamma.class
```

和类：

```
class Test { Alpha a; Beta b; Gamma c; }
```

哪三个必须加入到类 Test 中，以使其通过编译？（选三项）

- A. package y;
- B. package com;
- C. import com.x.y.\*;
- D. import com.x.\*;

15. 现有 2 个文件：

```
1. package x;
2. public class X{
3. public static void doX() {System.out.print("doX");}
4. }
```

和：

```
1. class Find{
2. public static void main(String [] args) {
3. //insert code here
4. }
5. }
```

哪两行分别插入到类 Find 的第 3 行将编译并产生输出 “doX”？（选两

项）

- A. doX();
- B. X. doX();
- C. x. X. doX();
- D. x.X myX=new x.X();myX.doX();



---

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

---

## 第五章练习题（继承）

1. 下列关于继承优点的叙述正确的是哪几项？（选三项）

- A. 可以创建更为特殊的类型
- B. 消除重复代码
- C. 执行效率高
- D. 便于维护

2. 在子类中调用父类中被覆盖的方法时需要使用哪项关键字？

- A. this
- B. super
- C. new
- D. 以上都不是

3. 现有

```
public class Parentt  
public void change (int x){  
}  
public class Child extends Parent{  
    // 覆盖父类 change 方法  
}
```

下列哪个声明是正确的覆盖了父类的 `change` 方法？

- A. `protected void change (int x){}`
- B. `public void change(int x, int y){}`
- C. `public void change (int x){}`
- D. `public void change (String s){}`

4. 如果想要一个类不能被任何类继承的话，需要使用哪个关键字来修饰该类？

- A. `abstract`
- B. `new`
- C. `static`
- D. `final`

---

5. 为了使得 `System.out.println()` 输出对象引用的时候得到有意义的信息我们应该覆盖 `Object` 的哪个方法?

- A. `equals`
- B. `toString`
- C. `hashCode`
- D. `notify`

6. 现有:

```
public class Pet()
public class Cat extends Pet()
执行代码
```

```
Cat c= new Cat();
```

```
Pet p= (Pet)c;
```

后下列哪项是正确的?

- A. `Pet p=(Pet)c` 运行错误
- B. `Pet p=(Pet)c` 编译错误
- C. `Pet p= (Pet)c` 正常执行
- D. 以上都不对

7. 程序:

```
public class Pet{
public void speak(){
System.out.print ("pet");
}
}
```

```
public class Cat extends Pet{
public void speak(){
System.out.print("Cat");
}
}
```

```
public class Dog extends Pet(
public void speak(){
System.out.print("Dog");
}
}
```

执行代码

```
Pet[] p= {new Cat (),new Dog(),new Pet()};
for (int i=0;i<p.length;i++)
p[i].speak();
```

---

后输出的内容是哪项？

- B. Cat Cat Cat
- C. Cat Dog Dog
- D. Cat Dog Pet

8. 现有：

- 1. class Dog{ }
- 2. class Harrier extends Dog { }
- 3.
- 4. class DogTest{
- 5. public static void main (String [] args) {
- 6. Dog d1=new Dog();
- 7. Harrier h1=new Harrier();
- 8. Dog d2=h1;
- 9. Harrier h2= (Harrier) d2;
- 10. Harrier h3=d2;
- 11. }
- 12. }

下面哪一项是正确的？

- A. 2 个 Dog 对象被创建
- B. 2 个 Harrier 对象被创建
- C. 3 个 Harrier 对象被创建
- D. 编译失败

9. 现有：

- 2. class Cat {
- 3. Cat (int c) {System.out.print {"cat"+c+" "}; }
- 4. }
- 5. class SubCat extends Cat {
- 6. SubCat (int c){super (5); System.out.print ("cable");}
- 7. SubCat() { this (4); }
- 8. public static void main (String [] args) {
- 9. SubCat s= new SubCat();
- 10. }
- 11. }

结果为：

- A. cat5
- B. cable
- C. cat5 cable
- D. cable cat5

---

10. 现有：

```

1. class Guy {String greet()    {return "hi"; } }
2.   class Cowboy extends Guy  (   String greet()    (   return
"howdy ";   ) )
3.   class Surfer extends Guy  (String greet()    (return "dude! ";} }
4.
5.   class Greetings  {
6. public static void main (String [] args)    {
7. Guy [] guys =  ( new Guy(), new Cowboy(), new Surfer() );
8.   for (Guy g:   guys)
9. System.out.print (g.greet());
10.  }
11. }
```

结果为：

- A. hi howdy dude!
- B.运行时异常被抛出。
- c. 第 7 行出现一个错误，编译失败。
- D. 第 8 行出现一个错误，编译失败。

11. 现有：

```

1.   class Over  {
2. int dolt (long x) { return 3; }
3. }
4.
5. class Under extends Over  {
6.      / / insert code here
7. }
```

和四个方法：

```

short dolt (int y) { return 4; }
int dolt(long Xr long y) { return 4; }
private int dolt(short y) { return 4; }
protected int dolt (long x) { return 4; }
```

分别插入到第 6 行，有几个可以通过编译？

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

---

12 现有

```
1. class Beverage {  
2.     Beverage () { System.out.print ("beverage "); }  
3. }  
4. class Beer extends Beverage {  
5.     public static void main(string [] args) {  
6.         Beer b = new Beer (14) ;  
7.     }  
8.     public int Beer(int x) {  
9.         this () ;  
10.        System.out.print ("beerl") ;  
11.    }  
12.    public Beer() { System.out.print("beer2 "); }  
13. }
```

结果是什么？

- A. beerl beverage
- B. beer2 beverage
- C. beverage beer2 beerl
- D.编译失败

13.现有:

```
1. class Bird {  
2.     void talk() { System.out.print("chirp "); }  
3. }  
4. class Parrot2 extends Bird {  
5.     protected void talk() { System.out.print("hello "); }  
6.     public static void main(String [] args) {  
7.         Bird [] birds = {new Bird(), new Parrot2 () };  
8.         for( Bird b : birds)  
9.             b.talk () ;  
10.    }  
11. }
```

结果是什么 ？

- A. chirp chirp
- B. hello hello
- C. chirp hello
- D:编译错误

---

14. 现有:

```
1. class SuperFoo(  
2. SuperFoo doStuff (int x) {  
3. return new SuperFoo();  
4. }  
5. }  
6.  
7. class Foo extends SuperFoo {  
8. // insert code here  
9. }
```

和四个声明:

```
Foo doStuff (int x) { return new Foo(); }  
Foo doStuff (int x) { return new SuperFoo(); }  
SuperFoo doStuff(int x) { return new Foo(); }  
SuperFoo doStuff(int y) { return new SuperFoo(); }
```

分别插入到第 8 行, 有几个可以通过编译?

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

15. 现有:

```
1. class HorseRadish {  
2. // insert code here  
3. protected HorseRadish (int x) {  
4. System.out.println ("bok choy");  
5. }  
6. }  
7. class Wasabi extends HorseRadish {  
8. public static void main (String [] args) {  
9. Wasabi w= new Wasabi();  
10. }  
11. }
```

分别插入到第 2 行, 哪两项允许代码编译并产生 "bok choy" 输出结果?

(选

两项)

- A. protected HorseRadish() {this (42); }
- B. protected HorseRadish() {}
- C. // just a comment
- D. protected HorseRadish() { new HorseRadish (42);}

---

## 参考答案

- 1ABD
- 2B
- 3C
- 4D
- SB
- 6A
- 7D
- 8D
- 9C
- 10 A
- 11 D
- 12 D
- 13 C
- 14 D
- 15 AD



---

## 第六章练习题（抽象类与接口）

1. 下列有关抽象类的叙述正确的是哪项？
  - A. 抽象类中一定含有抽象方法
  - B. 抽象类的声明必须包含 **abstract** 关键字
  - C. 抽象类既能被实例化也能被继承
  - D. 抽象类中不能有构造方法
  
2. 下列有关抽象方法叙述正确的是哪项？（选两项）
  - A. 抽象方法和普通方法一样，只是前面多加一个修饰符 **abstract**
  - B. 抽象方法没有方法体
  - C. 抽象方法可以包含存任何类中
  - D. 包含抽象方法的类的具体子类必须提供具体的覆盖方法
  
3. 下列有关接口的叙述错误的是哪项？
  - A. 接口中只能包含抽象方法和常量
  - B. 一个类可以实现多个接口
  - C. 类实现接口时必须实现其中的方法
  - D. 接口不能被继承
  
4. 下列关于接口的定义哪项是正确的？
  - A. `interface C{int a; }`
  - B. `public interface A implements B {}`
  - C. `public interface A {int a(); }`
  - D. `abstract interface D {}`

---

5. 现有:

```
1. interface Animal {
2.     void eat();
3. }
4.
5. // insert code here
6.
7. public class HouseCat implements Feline {
8.     public void eat() { }
9. }
```

和以下三个接口声明:

```
interface Feline extends Animal ( )
interface Feline extends Animal {void eat(); }
interface Feline extends Animal {void eat() { } }
```

分别插入到第 5 行, 有多少行可以编译?

- A. 0
- B. 1
- C. 2
- D. 3

6. 现有:

```
1. interface Color { }
2. interface Weight { }
3. //insert code here
```

和以下六个声明:

```
class Boat extends Color, extends Weight { }
class Boat extends Color and Weight { }
class Boat extends Color, Weight { }
class Boat implements Color, implements Weight { }
class Boat implements Color and Weight { }
class Boat implements Color, Weight { }
```

分别插入到第 3 行, 有多少行可以编译?

- A. 0
- B. 1
- C. 2
- D. 3

---

7. 现有:

```
1. abstract class Color {  
2. protected abstract String getRGB();  
3. }  
4.  
5. public class Blue extends Color {  
6.     // insert code here  
7. }
```

和四个声明:

```
public String getRGB() { return "blue"; }  
String getRGB() { return "blue"; }  
private String getRGB() { return "blue"; }  
protected String getRGB() { return "blue"; }
```

分别插入到第 6 行, 有几个可以通过编译?

- A. 0
- B. 1
- C. 2
- D. 3

8. 现有:

```
1. abstract class Color2 {  
2.     //insert code here  
3. }  
4.  
5. public class Blue2 extends Color2 {  
6. public String getRGB() { return "blue"; }  
7. }
```

和 4 个声明:

```
public abstract String getRGB();  
abstract String getRGB();  
private abstract String getRGB();  
protected abstract String getRGB();
```

分别插入到第 2 行, 有多少行可以编译?

- A. 0
- B. 1
- C. 2
- D. 3

---

9. 现有:

```
1. class Top {  
2.     static int X=1;  
3.     public Top() { x*=3; }  
4. }  
5. class Middle extends Top {  
6. public Middle() {x+=1; }  
7. public static void main(String [] args) {  
8. Middle m=new Middle();  
9. System.out.println (x)  
10. }  
11. }
```

结果是什么?

- A. 2
- B. 3
- C. 4
- D. 编译失败

10. 现有两个文件:

```
1. package X;  
2. public class X {  
3. public static void doX() {System.out.print ("doX"); }  
4. }
```

和:

```
1. import x.X;  
2. class Find {  
3.     public static void main (String [] args) {  
4.         X myX=new X(); myX.doX();  
5.         X.doX();  
6.         x.X.doX():  
7.         x.X myX2=new x.X(); myX2.doX();  
8.     }  
9. }
```

结果为:

- A. Find 类中第 4 行出现一个错误, 编译失败。
- B. Find 类第 5 行出现一个错误, 编译失败。
- C. Find 类第 6 行出现一个错误, 编译失败。
- D. doX doX doX doX

---

11.现有:

```
1. class Tree {  
2.     private static String tree = "tree";  
3.     String getTree () { return tree; }  
4. }  
5. class Elm extends Tree {  
6.     private static String tree = "elm";  
7.     public static void main (String [] args) {  
8.         new Elm().go (new Tree()) ;  
9.     }  
10.    void go (Tree t) {  
11.        String s = t.getTree () +Elm.tree + tree +
```

(new

```
        Elm().getTree ()) ;  
12.    System.out.println (s) ;
```

结果为:

- A. elm elm elm elm
- B. tree elm elm elm
- C. tree elm elm tree
- D. tree elm tree elm

12.现有:

```
1. interface Animal {  
2.     void eat () ;  
3. }  
4.  
5. //insert code here  
6.  
7. public class HouseCat extends Feline {  
8.     public void eat() {}  
9. }
```

和五个申明

```
abstract class Feline implements Animal { }  
abstract class Feline implements Animal { void eat () ; }  
abstract class Feline implements Animal { public void eat();}  
abstract class Feline implements Animal { public void eat() {} }  
abstract class Feline implements Animal { abstract public void eat();}
```

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

- 13. 现有：
  1. interface I { void go(); }
  - 2.
  3. abstract class A implements I { }
  - 4.
  5. class C extends A {
  6.     void go(){ }
  7. }

结果是什么？

- A. 代码通过编译
- B. 由于第 1 行的错误导致编译失败
- C. 由于第 3 行的错误导致编译失败
- D. 由于第 6 行的错误导致编译失败

14. 现有：

1. interface Data {public void load();}
  2. abstract class Info {public abstract void load();}
- 下列类定义中正确使用 Data 和 Info 的是哪项？
- A. public class Employee implements Info extends Data {  
   public void load() { /\* do something\* \*/ }  
   }  
 B. public class Employee extends Info implements Data {  
   public void load() { /\* do something\* \*/ }  
   }  
 C. public class Employee implements Info extends Data {  
   public void Data.load() { /\* do something\* \*/ }  
   public void load() { /\* do something\* \*/ }  
   }  
 D. public class Employee extends Info implements Data {  
   public void Data.load() { /\* do something\* \*/ }  
   public void info.load() { /\* do something\* \*/ }  
   }

---

15. 下列代码正确的是哪项?

- A. `public class Session implements Runnable, Cloneable{  
 public void run ();  
 public Object clone () ;  
}`
- B. `public class Session extends Runnable, Cloneable {  
 public void run() { /*do something*/  
 public Object clone() { /*make a copy*/  
}`
- C. `public abstract class Session  
 implements Runnable, Cloneable {  
 public void run() { /*do something*/  
 public Object clone() { /*make a copy*/  
}`
- D. `public class Session  
 implements Runnable, implements Cloneable {  
 public void run() { /*do something*/  
 public Object clone() { /*make a copy*/  
}`

---

参考答案

1B

2BD

3D

4C

5C

6B

7C

8D

9C

10 D

11 C

12 C

13 D

14 B

15 C



---

## 第七章练习题（多态）

1. 下列自‘关多态的叙述正确的是哪项？（选两项）
  - A. 父类的引用指向子类的实例是一种多态
  - B. 子类的引用指向子类的实例是一种多态
  - c. 接口的引用指向实现该接口类的实例是一种多态
  - D. 抽象类的引用指向抽象类的实例是一种多态
2. Java 中方法绑定有哪些形式？（选两项）
  - A.编译时刻绑定
  - B.运行时刻绑定
  - c. 静态绑定
  - D. 私有绑定
3. 表达式"hello" instance of String 返回的值是哪项？
  - A. true
  - B. false
  - C. 1
  - D. 0
4. 求平方根方法 `public static double sqrt (double a)`可以传递的参数类型有哪些？（选三项）
  - A. byte
  - B. float
  - C. String
  - D. long
5. 涉及类 `MyClass` 的方法签名是 `public void find(MyClass a)`，那么该  
方法可接收的实际参数的类型可以是哪些？（选两项）
  - A. `MyClass` 类的类型
  - B. `MyClass` 子类的类型
  - C. `Object` 类型
  - D. 所有接口

---

6. 使用下列哪些关键字可以判定实参的具体类型?

- A. as
- B. is
- C. instanceof
- D. extends

7. 现有:

```
class Pencil {  
    public void write (String content){  
        System.out.println ("Write"+content);  
    }  
}  
class RubberPencil extends Pencil{  
    public void erase (String content){  
        System.out.println ("Erase"+content);  
    }  
}
```

执行下列代码的结果是哪项?

```
Pencil pen=new RubberPencil();  
pen.write ("Hello");  
pen.erase ("Hello");
```

- A. Write Hello  
    Erase Hello
- B. Erase Hello  
    Write Hello
- C. 编译错误
- D. 运行时抛出异常

8. 现有:

```
class Pencil {  
    public void write (String content){  
        System.out.println ("Write"+content);  
    }  
}  
class RubberPencil extends Pencil{  
    public void write (String content){  
        System.out.println ("Rubber Write"+content);  
    }  
    public void erase (String content){  
        System.out.println ("Erase "+content);  
    }  
}
```

---

执行下列代码的结果是哪项？

```
Pencil pen=new RubberPencil();  
pen.write("Hello");
```

- A. Write Hello
- B. Rubber Write Hello
- C.编译错误
- D.运行时抛出异常

9 下列哪些方法是在编译时刻绑定的？（选三项）

- A. 静态方法
- B. **private** 方法
- C. **final** 方法
- D.非 **private** 方法

10. 现有：

```
class Pencil {  
    public void write (String content){  
        System.out.println( "Write",+content){  
    }  
}  
  
class RubberPencil extends Pencil{  
    public void write (String content){  
        System.out.println("Rubber Write"+content);  
    }  
    public void erase (String content){  
        System.out.println( "Erase "+content);  
    }  
}
```

执行下列代码的结果是哪项？

```
Pencil pen=new Pencil();  
(( RubberPencil) pen).write( "Hello");
```

- A. Write Hello
- B. Rubber Write Hello
- c. 编译失败
- D. 运行时抛出异常

---

11. 现有:

```
class TestA {  
    public void start() { System.out.println("TestA"); }  
}  
public class TestB extends TestA {  
    public void start() { System.out.println("TestB"); }  
    public static void main(String[] args) {  
        ((TestA)new TestB()).start();  
    }  
}
```

运行结果是哪项?

- A. TestA
- B. TestB
- C. 编译失败
- D. 运行时抛出异常

12. 现有:

```
class A {public String name="a"}  
class B extends A {public String name="b"}  
执行如下代码后的结果是哪项?
```

```
A a=new B();  
System.out.println(a.name);
```

- A. a
- B. b
- C. 编译失败
- D. 运行时抛出异常

13. 现有:

```
1 interface F{}  
2 class A implements F{}  
3 class B extends A{}  
4 class C extends B{  
5     public static void main(String[] args){  
6         B b=new B();  
7         // insert code here  
8     }  
9 }
```

---

下列哪行代码插入到第 7 行，将抛出 `java.lang.ClassCastException` 异常 7

- A. `A a=b;`
- B. `F f= (C)b;`
- C. `F f= (A)b;`
- D. `B bb= (B)(A)b;`

14. 现有：

```
1. class Guy { String greet() { return "hi"; } }
2. class Cowboy extends Guy { String greet() { return.
"howdy";}}
3. class Wrangler extends Cowboy { String greet()
{ return "orch!"; }}
4.
5. class Greetings2 {
6. public static void main (String [] args) {
7.     Guy g=new Wrangler();
8.     Guy g2=new Cowboy();
9.     Wrangler w2=new Wrangler();
10.    System.out.print (g.greet()+g2.greet()+w2.greet());
11. }
12. }
```

结果是什么？

- A. hi hi ouch!
- B. ouch! howdy ouch!
- C. hi howdy ouch!
- D. 编译失败
- E.运行的时候有异常抛出

15. 现有：

```
class ClassA {}
class ClassB extends ClassA {}
class ClassC extends ClassA {}
```

以及：

```
ClassA p0=new ClassA();
ClassB p1=new ClassB();
ClassC p2=new ClassC();
ClassA p3=new ClassB();
ClassA p4=new ClassC();
```

- 
- 下列哪些是正确的？（选三项]

- A. `p0=pl;`
- B. `p1 =p2;`
- C. `p2=p4;`
- D. `p2 = (ClassC)pl;`
- E. `p1 = (ClassB)p3;`
- F. `p2 = (Classc)p4;`

---

参考答案

1AC

2AB

3A

4ABD

5AB

6C

7C

8B

9ABC

10 D

11 B

12 A

13 B

14 B

15AEF

---

## 第八章练习题（异常）

1. Java 语言中异常的分类是哪项？
  - A. 运行时异常和异常
  - B. 受检异常和非受检异常
  - C. 错误和异常
  - D. 错误和运行时异常
  
2. 所有异常的父类是哪项？
  - A. Throwable
  - B. Error
  - C. RuntimeException
  - D. Exception
  
3. 关于 try...catch...finally 结构，描述正确的是哪项？（选两项）
  - A. 可以有多个 catch
  - B. 只能有一个 catch
  - C. 可以没有 catch
  - D. finally 必须有
  
4. 下列属于非受检异常的是哪项？
  - A. SQLException
  - B. IOException
  - C. NullPointerException
  - D. OutOfMemoryError
  
5. 假设有自定义异常类 ServiceException，那么抛出该异常的语句正确的是  
哪项？
  - A. raise ServiceException
  - B. throw new ServiceException()
  - C. throw ServiceException
  - D. throws ServiceException



---

6. 在方法的声明中，要求该方法必须抛出异常时使用哪个关键字？

- A. throw
- B. catch
- C. finally
- D. throws

7. 现有：

```
1. class Number{
2.     public static void main(String [] args) {
3.         try {
4.             System.out.print (Integer.parseInt ("forty"));
5.         } catch (RuntimeException r) {
6.             System.out.print ("runtime");
7.         } catch (NumberFormatException e) {
8.             system..ut.print("number");
9.         }
10.    }
11. }
```

结果是什么？

- A. number
- B. runtime
- C. forty number
- D. 编译失败

8. 现有：

```
9. void topGo() {
10.     try {
11.         middleGo();
12.     } catch (Exception e) {
13.         System.out.print("catch");
14.     }
15. }
16. void middleGo() throws Exception {
17.     go();
18.     system.out.print("late middle");
19. }
20. void go() throws ExceptiOn {
21.     throw new Exception();
22. }
```

---

如果调用 `topGo ()` ,则结果为:

- A. late middle
- B. catch
- C. late middle catch
- D. catch late middle

9.现有

```
1. class Calc {  
2.     public static void main(String [] args) {  
3.         try {  
4.             int x = Integer.parseInt ("42a") ;  
5.             //insert code here  
6.             System.out.print ("oops");  
7.         }  
8.     }  
9. }
```

下面哪两行分别插入到第五行, 会导致输 "oops"? (选两项)

- A. } catch (IllegalArgumentException e) {
- B. } catch (IllegalStateException c) {
- C. } catch (NumberFormatException n) {
- D. } catch (ClassCastException c) {

10.现有:

```
1. class Flow {  
2.     public static void main(String [] args) {  
3.         try {  
4.             System.out.print ("before") ;  
5.             doRiskyThing ( ) ;  
6.             System.out.print ("after ") ;  
7.         } catch (Exception fe) {  
8.             System.out.print ("catch") ;  
9.         }  
10.        System.out.println ( " done") ;  
11.    }  
12.    public static void doRiskyThing() throws Exception{  
13.        // this code returns unless it throws an Exception  
        }  
    }
```

可能会产生哪两项结果 ? (选两项)

- A. before catch
- B. before after done
- C. before catch done
- D. before after catch

---

11.现有:

```
1. class Birds {
2.     public static void main (String [] args) {
3.         try {
4.             throw new Exception ();
5.         } catch (Exception e) {
6.             try {
7.                 throw new Exception ();
8.             } catch (Exception e2) { System.out.print ("inner
           "); }
9.             System.out.print ( "middle" );
10.        }
11.        System.out.print ("outer");
12.    }
13. }
```

A. inner outer  
B. middle outer  
C. inner middle outer  
D. 编译失败

12.现有:

```
1. class Parser extends Utils {
2.     public static void main (String [] args) {
3.         try { System.out.print (new Parser ( ).getInt ("42"));
4.     } catch (NumberFormatException n) {
5.         System.out.println ( "NFExc" ); }
6.     }
7.     int getInt (String arg) throws NumberFormatException{
8.         return Integer.parseInt (arg) ;
9.     }
10. }
11. class Utils {
12.     int getInt (String arg) { return 42; }
13. }
```

给果为 :

- A. NFExc  
B. 42  
C. 42NFExc  
D. 编译失败

---

13. 现有:

```
1. class Parser extends Utils {  
2. public static void main (String[] args) {  
3. try{System.out.print (new Parser().getInt("42"));  
4. } catch (Exception e) {  
5. System.out.println("Exc"); }  
6. }  
7. int getInt (String arg) throws Exception {  
8. return Integer.parseInt (arg);  
9. }  
10. }  
11. class Utils {  
12. int getInt (String arg) {return 42; }  
13. }
```

结果为:

- A. 42
- B. Exc
- C. 42Exc
- D. 编译失败

14. 现有:

```
1. class Propeller2 {  
2. public static void main (String[] args) // add code here?  
3. { new Propeller2().topGo(); }  
4.  
5. void topGo() // add code here?  
6. { middleGo(); }  
7.  
8. void middleGo() // add code here?  
9. { go(); System.out.println ("late middle"); }  
void go() // add code here?  
12. {throw new Exception(); }  
13. }
```

为使代码通过编译, 需要在哪一行加入声明 throws Exception?

- A. 只在第 11 行
- B. 在第 8 行和第 11 行
- C. 在第 5 行、第 8 行和第 11 行
- D. 在第 2 行、第 5 行、第 8 行和第 11 行

---

15.现有

```
1. class Parser extends Utils {  
2.     public static void main (String [] args) {  
3.         try { System.out.print (new Parser () .getInt ("42"))  
4.     } catch (Exception e) {  
5.         System.out.println ("Exc"); }  
6.     }  
7.     int getInt (String arg) throws Exception {  
8.         return Integer.parseInt (arg) ;  
9.     }  
10. }  
11. class Utils {  
12.     int getInt () { return 42; }  
13. }
```

结果是什么？

- A. 42Exc
- B. Exc
- C. 42
- D.编译失败

---

参考答案

1B

2A

3AC

4D

5B

6D

7D

8B

9 A C

10 B C

11 C

12 B

13 D

14 D

15 C

---

## 第九章练习题（多线程机制）

1. 线程由以下哪些部分组成？（选三项）
  - A. 虚拟的 CPU
  - B. CPU 执行的代码
  - C. 支持代码工作的数据
  - D. 进程
2. 创建线程的时候必须实现哪项接口？
  - A. Runnable
  - B. Thread
  - C. Run
  - D. Start
3. 启动一个线程的方法是哪项？
  - A. join()
  - B. run f)
  - C. start()
  - D. sleep()
4. 当一个处于阻塞状态的线程解除阻塞后，它将回到哪个状态？
  - A. 运行中状态
  - B. 结束状态
  - C. 新建状态
  - D. 可运行状态
5. 可以通过继承哪个类来创建线程？
  - A. Thread
  - B. Runnable
  - C. tart
  - D. run

---

、 6. 为了保证方法的线程安全，声明方法的时候必须用哪个修饰符？

- A. new
- B. transient
- C. void
- D. synchronized

7. 线程的默认优先级是哪项？

- A. 0
- B. 1
- c. 5
- D. 10

8. 现有：

- 1. class Thread2 implements Runnable (
- 2. void run() {
- 3. System.out.print ("go");
- 4. }
- 5.
- 6. public static void main(String [] args) {
- 7. Thread2 t2=new Thread2();
- 8. Thread t=new Thread(t2);
- 9. t.start();
- 10. }
- 11. }

结果为：

- A. go
- B.运行时异常被抛出
- c. 代码运行，无输出结果
- D. 编译失败

9. 程序员想要创建一个名为 **MyThread** 的类以便在 **main** 方法中用 **Thread** 实例

化。对于下面三行：

**MyThread** 必须继承 **Thread**。

**MyThread** 必须实现 **Thread**。

**MyThread** 必须覆盖 **public void run()**。



---

有几行是正确的 7

- A. 0
- B. 1
- C. 2
- D. 3

10. 现有:

```
1. class ThreadBoth extends Threaa implements Runnable {  
2.  
3. public void run() (System.out.print("hi"); }  
4.  
5. public static void main (String [] args) {  
6. Thread t1=new ThreadBoth():  
7. Thread t2 = new Thread (t1):  
8.     t1.run():  
9.     t2.run():  
10. }  
11. }
```

结果为:

- A. hi hi
- B. hi
- C. 编译失败
- D. 运行时异常被抛出

11. 现有: t 是一个合法的 Thread 对象的引用, 并且 t 的合法 run()方法如下:

```
9. public void run() {  
10.     System.out.print ("go");  
11. }
```

及:

```
18. t.start();  
19. t.start();  
1 n    t.run();
```

哪一个结果是?

- A. go go
- B. go go go
- C. go 之后跟着一个异常
- D. go go 之后跟着一个异常

---

12. 现有:

```

1. class ThreadExcept implements Runnable {
2.     public void run() { throw new RuntimeException("exception
"); }
3.
4.     public static void main(String [] args) {
5.         new Thread (new ThreadExcept()).start();
6.         try {
7.             int x=Integer.parseInt (args [0]);
8.             Thread. sleep (x);
9.             System.out.print("main");
10.        } catch (Exception e) { }
11.    }
12. }
```

和命令行: `java ThreadExcept 1000`

哪一个结果是?

- A. main
- B. 编译失败
- C. main java.lang.RuntimeException: exception
- D. 代码运行, 但没有输出

13. 现有:

```

5. class Order2 implements Runnable {
6.     public void run() {
7. for (int x= 0; x<4; x++) {
8. try{Thread.sleep(100); }catch (Exception e) { }
9.     System.out.print("r");
10.    } }
11.     public static void main(string [] args) {
12.         Thread t=new Thread(new order2());
13.         t. start();
14.         for(int x=0; x<4; x++) {
15.             // insert code here
16.             System.out.print("m");
17.         } } }
```

哪一个插入到第 15 行, 最有可能产生输出 `rmrmrmrm`?

- A. `Thread. sleep(1);`
- B. `Thread. sleep(100);`
- C. `Thread.sleep(1000);`
- D. `try{ Thread.sleep(1); } catch (Exception e) { }`
- E. `try{Thread.sleep(100); } catch (Exception e) { }`
- F. `try{Thread.sleep(1000); }catch (Exception e) { }`

---

14. 现有:

```
1. class Waiting implements Runnable {  
2.     boolean flag=false;  
3. public synchronized void run() {  
4.     if (flag) {  
5.         flag=false;  
6. System.out.print ("1");  
7.     try { this.wait(); } catch (Exception e) { }  
8. System.out.print ("2");  
9.     }  
10.    else {  
11.        flag=true;  
12.        System.out.print ("3");  
13.        try{Thread.sleep (2000); } catch(Exception e) {}  
14.        System.out.print ("4");  
15.        notify();  
16.    }  
17. }  
18. public static void main (String [] args) {  
19.     Waiting w=new Waiting();  
20.     new Thread (w) .start();  
21.     new Thread (w) .start();  
22. }  
23. }
```

以下哪两项是正确的? (选两项)

- A.代码输出 1 3 4
- B.代码输出 3 4 1
- C.代码输出 1 2 3 4
- D.代码输出 1 3 4 2
- E.代码运行完毕
- F. 代码不会完成

---

```
5. class Order implements Runnable {  
6. public void run () {  
7. try { Thread.sleep (2000) ; } catch (Exception e)  
8. System.out.print("in") ;  
10. public static void main (String [] args) {  
11. Thread t = new Thread (new Order ()) ;  
12. t.start () ;  
13. System.out.print ("pre ") ;  
14. try { t.join () ; } catch (Exception e) {}  
15. System.out.print ("post") ;
```

可产生哪两项结果? (选两项)

- A. pre in post
- B. pre in
- C. in post pre
- D. in pre post
- E. pre post in

---

参考答案

1ABD

2A

3C

4D

SA

6D

7C

8D

9A

10 A

11 C

12 C

13 E

14 B F

15 A D

---

## 第十章练习题（输入输出流）

1. 下列哪项是 Java 语言中所定义的字节流？
  - A. Output
  - B. Reader
  - C. Writer
  - D. InputStream
  
2. 在输入流的 `read` 方法返回哪个值的时候表示读取结束？
  - A. 0
  - B. 1
  - C. -1
  - D. null
  
- 3 为了从文本文件中逐行读取内容，应该使用哪个处理流对象？
  - A. `BufferedReader`
  - B. `BufferedWriter`
  - C. `BufferedInputStream`
  - D. `BufferedOutputStream`
  
4. 为了实现自定义对象的序列化，该自定义对象必须实现哪个接口？
  - A. `Volatile`
  - B. `Serializable`
  - C. `Runnable`
  - D. `Transient`
  
5. 以下关于 `File` 类的叙述，哪两项正确？（选两项）
  - A. `File` 类位于 `java.io` 包中
  - B. 创建一个 `File` 类的实例将导致打开指定文件进行读写
  - c. `File` 实例不能用于封装目录
  - D. `File` 实例封装了指定文件的信息

---

6. 删除 File 实例所对应文件的方法是？

- A. mkdir
- B. exists
- C. delete
- D. isHidden

7. 现有：

```
1. import java.util.*;
2. class ScanStuff {
3. public static void main (String [] args) {
4. String S= "x,yy,123";
5. Scanner sc = new Scanner (s);
6. while (sc.hasNext())
7. System.out.print (sc.next() +" ");
8. }
9. }
```

结果是什么？

- A. x yy
- B. x,yy,123
- C. x yy 123
- D. x, yy
- E. 编译失败
- F. 运行的时候有异常抛出

8. 现有：

- f 对一个 java .io. File 型实例的合法引用
- fr 对 ‘个 java.io.FileReader 型实例的合法引用
- br 对一个 java.io.BufferedReader 型实例的合法引用

和：

```
34. String Line=null;
35.
36. //insert code here
37. System.out.println (line);
38. }
```

---

哪一行代码插入到 36 行将循环通过一个文本文件并存文本域中每次输出一行？

- A. while( (line=f.read()) !=null) {
- B. while( (line=fr.read()) !=null) {
- C. while( (line=br.read()) !=null) {
- D. while( (line=br.readLine()) !=null) {
- E. while( (line=fr.readLine()) !=null) {
- F. while( (line=f.readLine()) !=null) {

9. 现有：

```
int x= reader.read();
```

下面哪一项正确？

- A. 阅读器不是 `FileReader` 或者 `BufferedReader` 类型
- B. 阅读器可以是 `FileReader` 或者 `BufferedReader`
- C. 阅读器可以是 `FileReader` 类型，但不能是 `BufferedReader` 类型
- D. 阅读器可以是 `BufferedReader` 类型，但不能是 `FileReader` 类型

10. 现有：

```
31. String s="write a line to a file";
```

```
32. w. print(s+"\n");
```

哪一个是对的？

- A. w 即可以是 `PrintWriter` 类型，也可以是 `BufferedWriter` 类型。
- B. w 即不可以是 `PrintWriter` 类型，也不可以是 `BufferedWriter` 类型。
- C. w 可以是 `BufferedWriter` 类型，但不可以是 `PrintWriter` 类型。
- D. w 以是 `PrintWriter` 类型，但不可以是 `BufferedWriter` 类型。

11. 现有：

-f 是一个合法的 `java.io.File` 实例的引用

- fr 是一个合法的 `java.io.FileReader` 实例的引用

- br 是一个合法的 `java.io.BufferedReader` 实例的引用



---

哪两个是合法的？（选两项）

- A. File f2=new File (f);
- B. FileReader fr2=new FileReader (fr);
- C. FileReader fr2=new FileReader(f);
- D. FileReader fr2=new FileReader(br);
- E. BufferedReader br2 =new BufferedReader (f);
- F. BufferedReader br2=new BufferedReader (fr);

12. 现有:

- f 是一个对 java.io. File 实例的合法引用
  - fw 是 ‘个对 java.io.FileWriter 实例的合法引用
  - bw 是一个对 java.io.BufferedWriter 实例的合法引用
- 以下哪一项无效？

- A. PrintWriter prtWtr = new PrintWriter (f);
- B. PrintWriter prtWtr = new PrintWriter( fw);
- C. PrintWriter prtWtr = new PrintWriter(bw):
- D. BufferWriter bufWtr = new BufferedWitr(f);
- E. BufferWriter bufWtr = new BufferedWriter (fw);
- F. BufferWriter bufWtr = new BufferedWriter (bw):

13.现有

- 1. import java.io.PrintWriter;
- 2.
- 3. class DoFormat{
- 4.public static void main (String [] args) {
- 5. int x=42;
- 6. int y=12345;
- 7. float Z=7 ;
- 8.System.out.format("—%4d-", X)j
- 9. System.out.format ("-%4d-", y);
- 10. System.out.format ("-%4.1d-", z);
- 11. }
- 12. }

结果为:

- A.编译失败
- B. -42- -1234- -7.0-
- C. - 42- -1234- -7.0-
- D. - 42- -12345- -7.0-
- E. 运行时异常被抛出

---

’ 14. 现有:

10. class Car implements Serializable ()

12. class Ford extends Car {}

如果试图序列化一个 Ford 实例, 结果为:

- A. 编译失败
- B. 两个对象被序列化
- c. 一个对象被序列化
- D. 运行时异常被抛出

15. 现有:

1. class Pet implements Serializable {

2.Collar c= new Collar();

3. }

4. class Collar implements Serializable {

5.collarPart cpl=new CollarPart ("handle");

6.CollarPart cp2=new CollarPart ("clip");

7. }

8.

9. class CollarPart implements Serializable()

如果 Pet 实例被序列化, 则多少对象将被序列化?

- a. 0
- b. 1
- c. 2
- D. 3
- E. 4
- F. 5

---

参考答案

1D

2C

3A

4B

5AC

6C

7B

8D

9B

10D

11C F

12 D

13 E

14 C

15 E

---

## 第十一章练习题（使用泛型和集合框架）

1. 集合 API 中 Set 接口的特点是哪项？
  - A. 不允许重复元素，元素有顺序
  - B. 允许重复元素，元素无顺序
  - C. 允许重复元素，元素有顺序
  - D. 不允许重复元素，元素无顺序
2. 实现了 Set 接口的类是哪项？
  - A. ArrayList
  - B. HashTable
  - C. HashSet
  - D. Collection
3. 表示键值对概念的接口是哪项？
  - A. Set
  - B. List
  - C. Collection
  - D. Map
4. List 接口的特点是哪项？
  - A. 不允许重复元素，元素无顺序
  - B. 不允许重复元素，元素有顺序
  - C. 允许重复元素，元素有顺序
  - D. 允许重复元素，元素无顺序
5. 下列哪些项是泛型的优点？（选两项）
  - A. 不用向下强制类型转换
  - B. 代码容易编写
  - C. 类型安全
  - D. 运行速度快

---

6. 创建一个只能存放 **String** 的泛型 **ArrayList** 的语句是哪项?

- A. `ArrayList<int> al=new ArrayList<int>();`
- B. `ArrayList<String> al=new ArrayList<String>();`
- C. `ArrayList al=new ArrayList<String>();`
- D. `ArrayList<String> al =new List<String>();`

7. 现有

```
public class Demo{
public static void main (String[] args){
List al=new ArrayList();
a1. add("1");
a1. add("2");
a1. add("2");
a1. add("3");
System.out.println (al);
}
}
```

上述程序执行后的输出是哪项?

- A. `[1, 2, 3]`
- B. `[1,2, 3, 3]`
- C. `[1,2, 2, 3]`
- D. `[2,1,3,2]`

8. 现有:

```
3. import java.util.*;
4. class ForInTest {
5.static List list=new ArrayList();
6.
7.public static void main (String [] args) {
8.list. add("a"); list. add("b");list. add( "c");
9. //insert code here
10. System.out.print (o);
11. }
12. }
```

哪一行插入到第 9 行将导致输出 “abc”?

- A. `for (Iterator o : list.iterator(); o.hasNext (); )`
- B. `for (Iterator O : list)`
- C. `for (Object o : list.iterator())`
- D. `for (Object o : list)`

---

9. 现有:

```
5. import java.util.*;
6. class AddStuff2 {
7. public static void main(String [] args) {
8. TreeSet<String>t=new TreeSet<String>();
9.     if (t.add("one"))
if (t.add("two"))
if (t.add ("three"))
12.     add("four");
13.     for (String s : t)
14.         System.out.print (s);
15. }
16. }
```

结果为:

- A. one
- B. one three two
- C. one two three
- D. one two three four
- E. four one three two
- F.编译失败

10. 哪两个是同步的? (选两项)

- A. java.util.TreeSet
- B. java.util.Hashtable
- C. java.util.Vector
- D. java.util.LinkedList

11. 现有:

- list 是一个合法的集合引用
- getCollection()返回一个合法集合的引用

哪两个是合法的? (选两项)

- A. for(Object o ; list)
- B. for(Object o : getCollection())
- C. for(Object o : list.iterator())
- D. for(Iterator i; list.iterator() ; i.hasNext () )
- E. for(Iterator i=list.iterator(); i.hasNext (); )

---

12. 现有如下类型:

a - java.util.Hashtable

b - java.util.List

c - java.util.ArrayList

d - java.util.SortedSet

和定义:

1-使用本接口, 允许用户控制集合中每个元素的插入位置。

2-使用本集合, 确保用户可以按照递增或元素的自然顺序遍历集合。

3-本具体类型允许空元素及基于索引的访问。

4-本集合是同步的。

哪一组匹配是对的?

A. 2 描述 d; 3 描述 b。

B.1 描述 b; 3 描述 C。

C.3 描述 a; 4 描述 b。

D.4 描述 a; 2 描述 C。

13. 现有:

```
1. import java.util.*;
```

```
2. class AddStuff {
```

```
3. public static void main (String [] args) {
```

```
4. TreeSet<String> s= new TreeSet<String>();
```

```
5.     s.add( "one");
```

```
6.     s.add( "two");
```

```
7.     // insert code here
```

```
8. for (String s2: sorted)
```

```
9. System.out.print (s2+" ");
```

```
10. }
```

```
11. }
```

和四个代码片段:

```
s1.sortedSet sorted=s.tailSet(s.first());
```

```
s2.sortedSet<String> sorted=s.tailSet(s.first());
```

```
s3.sortedSet sorted= (SortedSet)s .tailSet(s.first());
```

```
s4.SortedSet sorted= (SortSet<String>)s .tailSet(s.first());
```

---

分别插入到第 7 行，哪项可以编译？

- A. s2
- B. s1
- C. s2 和 s3
- D. s2 和 s4
- E. s2, s3 和 s4
- F. s1, s2, s3 和 s4

14. 现有：

```
1. import java.util.*;
2. class ForInTest {
3. static List list = new ArrayList();
4.
5. static List getList() { return list; }
6.
7. public static void main (String[] args) {
8. list.add("a"); list.add("b"); list.add("c");
9.      // insert code here
10.    System.out.print (o);
11.  }
12. }
```

第 9 行插入哪一项将输出 abc？

- A. for(char o: list)
- B. for (Object o: o.getList())
- C. for(Object o: getList();)
- D. for(Object o: getList())
- E. for(Object o: o.getList();)

15. 现有：

```
1. import java.util.*;
2.
3. Class FindStuff {
4. public static void main (String[] args) {
5.      // insert code here
6.    c.put ("X", 123);
7.  }
8. }
```



---

分别插入到第 5 行，哪三行允许代码编译？(选三项)

- A. `Map c= new SortedMap();`
- B. `HashMap c= new HashMap();`
- C. `HashMap c= new Hashtalole();`
- D. `SortedMap c= new TreeMap();`
- E. `ArrayList c= new ArrayList();`
- F. `MaD c = new LinkedHashMap();`

---

参考答案

1D

2C

3D

4C

5AC

6B

7C

8D

9E

10 B C

11 B E

12 B

13 A

14 D

15 B D F

---

## 第十二章练习题（基于 **Swing** 的图形用户界面设计）

1. 下列有关 **Swing** 的叙述，哪项错误？
  - A. **Swing** 是 Java 基础类(JFC)的组成部分
  - B. **Swing** 是可用来构建 GUI 的程序包
  - C. **Swing** 是 AWT 图形工具包的替代技术
  - D. Java 基础类(JFC)是 **Swing** 的组成部分
  
2. **Swing** GUI 通常由哪几类元素组成？（选三项）
  - A. GUI 容器
  - B. GUI 组件
  - C. 布局管理器
  - D. GUI 事件侦听器
  
3. 以下关于 **Swing** 容器叙述，哪项错误？
  - A. 容器是一种特殊的组件，它可用来放置其它组件
  - B. 容器是组成 GUI 所必需的元素
  - C. 容器是一种特殊的组件，它可被放置在其它容器中
  - D. 容器是一种特殊的组件，它可被放置在任何组件中
  
4. 以下关于 **BorderLayout** 类功能的描述，哪项错误？
  - A. 它可以与其它布局管理器协同工作
  - B. 它可以对 GUI 容器中的组件完成边框式的布局
  - C. 它位于 **java.awt** 包中
  - D. 它是一种特殊的组件
  
5. **JTextField** 类提供的 GUI 功能是？
  - A. 文本区域
  - B. 按钮
  - C. 文本字段
  - D. 菜单

- 
6. 将 GUI 窗口划分为东、西、南、北、中五个部分的布局管理器是：
- A. FlowLayout
  - B. GridLayout
  - C. CardLayout
  - D. BorderLayout
- 7 在 Swing GUI 编程中， `setDefaultCloseOperation(JFrame . EXIT_ON_CLOSE)`语句的作用是：
- A. 当执行关闭窗口操作时，不做任何操作。
  - B. 当执行关闭窗口操作时，调用 `WindowsListener` 对象并将隐藏 `JFrame`
  - C. 当执行关闭窗口操作时，退出应用程序
  - D. 当执行关闭窗口操作时，调用 `WincowsListener` 对象并隐藏和销毁 `Jframe`
8. 下列哪些是 Swing 容器的顶层容器？（选两项）
- A. `JPanel`
  - B. `JScrollPane`
  - C. `JWindow`
  - D. `JFrame`
9. 界面设计中我们希望实现一个对话框，该对话框上有 5 个垂直方向排列的大小一致的按钮，这些按钮要充满对话框的空余空间，并且当对话框改变大小的时候按钮也要跟着改变大小，同时保证大小一致。那么对话框应该选择哪种布局方式？
- A. `FlowLayout`
  - B. `BorderLayout`
  - C. `GridLayout`
  - D. `null (no LayoutManager)`
10. 组件的 `setSize()`方法签名正确的是哪项？（选两项）
- A. `setSize(int width,int height)`
  - B. `setSize(int x,int y,int width,int height)`
  - C. `setSize(Dimension dim)`
  - D. 以上皆不是

---

11. Applet 的布局设置默认是 `FlowLayout`, 下列哪项代码可以改变 Applet 的布局方式?

- A. `setLayoutManager(new GridLayout());`
- B. `setLayout (new GridLayout(2,2));`
- C. `setGridLayout (2,2);`
- D. `setBorderLayout();`

12. 组件的 `getBounds()` 方法返回值的类型是哪项?

- A. `Dimension`
- B. `Rectangle`
- C. `Point`
- D. `int[]`

13. 假设有一个面板 `PanelOne`, 它的布局管理方式是 `BorderLayout`。它的背景颜色被显式地设为黑色而前景颜色被显式地设为白色。有另一个面板 `PanelTwo`, 它的布局管理方式是 `FlowLayout`, 被添加到 `PanelOne` 的 `center` 区域, `PanelTwo` 的前景颜色是红色, 没有背景颜色被指定。一个带有文字的 `Label` 被添加到 `PanelTwo` 上, `Label` 没有设置颜色, 下列叙述正确的是哪项? (选两项)

- A. `Label` 的文字是白色
- B. `Label` 的文字是红色
- C. `PanelTwo` 的布局管理器覆盖 `PanelOne` 的布局管理器
- D. `PanelTwo` 是黑色背景

14. 现有:

```
package mypack;
import javax.swing.JButton;
import javax.swing.JFrame;
public class Butt extends JFrame {
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        Butt butt=new Butt();
    }
    Butt(){
        JButton jbl=new JButton("Hello");
        JButton jb2=new JButton("World");
        add(jbl);
        add(jb2);
        setSize(300, 300);
        setVisible (true);
    }
}
```

---

下列关于代码运行效果的叙述正确的是哪项？

- A. 2 个按钮紧靠着排列，Hello 在 World 的左侧
- B. Hello 占据 Frame 的整个区域
- C. World 占据 Frame 的整个区域
- D. 2 个按钮都在 Frame 的顶部

15. 现有：

```
package mypack;
import java.awt.Color;
import java.awt.FlowLayout;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JPanel;
public class ButtList extends JFrame {
    public static void main(String[] args) {
        ButtList bl = new ButtList();
    }
    ButtList(){
        JPanel pl=new JPanel();
        pl.setBackground (Color.pink);
        p1.add(new JButton("one"));
        p1.add(new JButton("Two"));
        p1.add(new JButton("Three"));
        add("South", p1);
        setLayout(new Fl.wLayout());
        setSize(200,200);
        setVisible(true);
    }
}
```

上述代码运行后的显示效果是哪项？

- A. 3 个按钮从左到右排列在 Frame 的底部
- B. 3 个按钮从左到右排列在 Frame 的顶部
- C. 按钮无法显示
- D. 只有按钮 Three 显示出来并占据 Frame 整个区域

---

参考答案

1D

2A B C

3D

4D

5C

6D

7C

8CD

9C

10A C

11 B

12 B

13 B D

14 C

15 B

---

## 第十三章练习题（Java 事件驱动编程）

1. 以下关于 GUI 事件处理模型的叙述，哪两项是错误的（选两项）？
  - A. GUI 事件处理模型是委托式模型，其委托对象是事件处理器。
  - B. 用户与 GUI 的交互需要通过事件机制来完成。
  - C. GUI 事件处理模型是层次模型，因此一个事件可被多个组件处理。
  - D. 一个事件源只能注册一个事件侦听器。
  
2. GUI 事件模型的组成元素包括（选三项）：
  - A. 事件
  - B. 事件处理器
  - C. GUI 容器
  - D. 事件源
  
3. 以下各项哪些不能成为 GUI 事件源？
  - A. GUI 按钮
  - B. GUI 窗口，例如 JFrame
  - C. 鼠标
  - D. 文本字段
  
4. 编写 JButton 组件的事件处理器类时，需实现哪个接口？
  - A. ItemListener
  - B. ActionListener
  - C. ButtonListener
  - D. WindowListener
  
5. 以下哪些接口是事件侦听器接口？(选二项)
  - A. ActionListener
  - B. ItemListener
  - C. WindowListener
  - D. ButtonListener



- 
6. 事件适配器类的作用是：（选三项）
- A. 为编写事件侦听器程序提供简便手段
  - B. 创建一种全新的事件侦听机制
  - C. 是由相应的事件侦听器接口继承而来
  - D. 定义在 `java.awt.event` 中
7. 以下哪个方法不是鼠标事件侦听器接口（`MouseListener`）定义的？
- A. `mousePressed`
  - B. `mouseEntered`
  - C. `mouseDragged`
  - D. `mouseClicked`
8. 下列叙述正确的是哪项？（选两项）
- A. `MouseListener` 接口定义了处理鼠标点击事件的方法
  - B. `MouseMotionListener` 接口定义了处理鼠标点击事件的方法
  - C. `MouseClickedListener` 接口定义了处理鼠标点击事件的方法
  - D. `ActionListener` 接口定义了处理按钮点击事件的方法
9. 下列哪个组件会产生 `Action` 事件？
- A. Buttons
  - B. Labels
  - C. Check Boxes
  - D. Windows
10. 下列叙述正确的是哪项？（选三项）
- A. `TextField` 能产生 `ActionEvent` 事件
  - B. `TextArea` 能产生 `ActionEvent` 事件
  - C. `Button` 能产生 `ActionEvent` 事件
  - D. `MenuItem` 能产生 `ActionEvent` 事件

- 
11. 假如我们想要对象 `eh` 来处理 `TextArea` 对象 `t` 的 `TextEvent` 事件, 那么, 我们应如何把 `eh` 添加为 `t` 的事件处理程序?
- A. `t.addTextListener (eh)`
  - B. `eh.addTextListener (t)`
  - C. `addTextListener (eh,t)`
  - D. `addTextListener (t, eh)`
12. 处理一个对象事件的首选方式是哪项?
- A. 覆盖对象的 `handleEvent()` 方法
  - B. 添加一个或多个事件监听来处理事件
  - C. 覆盖对象的 `processEvent()` 方法
  - D. 覆盖对象的 `dispatchEvent()` 方法
13. 当 2 个或多个对象被添加作为同一个事件的监听器, 那么当事件触发的時候  
哪个监听器对象被首先调用?
- A. 第一个被添加的监听器对象
  - B. 最后一个被添加的监听器对象
  - C. 无法确定哪个监听器对象被首先调用
  - D. 为同一个事件添加多个监听器是无法做到的
14. 下列叙述正确的是哪项?
- A. 事件继承模型取代事件委托模型
  - B. 事件继承模型比事件委托模型更加高效
  - C. 事件委托模型使用事件监听器来定义事件处理类的方法
  - D. 事件委托模型使用 `handleEvent()` 方法来支持事件处理
- 15 在事件委托类的继承体系中, 最高层次的类是哪项?
- A. `java.util.EventListener`
  - B. `java.util.EventObject`
  - C. `java.awt.AWTEvent`
  - D. `java.awt.event.AWTEvent`

---

参考答案

1CD

2ABD

3D

4B

SABC

6ACD

7C

8AD

9A

10 A C D

11 A

12 B

13 C

14 C

15 B