Java 语言面向对象高级程序设计　　习　题　四

一、填空题

1．在面向对象系统中，消息分为（　）和（　）两类。

2．类的访问控制符有（　）和（　）两种，（　）类具有跨包访问性而（　）类不能被跨包访问。

3．类成员的访问控制符有（　）、（　）、（　）和默认四种。

4．public 类型的类成员可被（　）、同一包中的（　）和不同包中的（　）的代码访问引用。

5．protected 类型的类成员可被（　）、同一包中的（　）和不同包中的（　）的代码访问引用。

6．default 类型的类成员只能被（　）、同一包中的（　）的代码访问引用。

7．private 类型的类成员只能被其所在类中的代码访问引用，它只具有（　）域访问性。

8．系统规定用（　）表示当前类的构造方法，用（　）表示直接父类的构造方法，在构造方法中两者只能选其一，且须放在第一条语句。

9．若子类和父类在同一个包中，则子类继承父类中的（　）、（prote　cted）和（　）

成员，将其作为子类的成员，但不能继承父类的（　）成员。

10．若子类和父类不在同一个包中，则子类继承了父类中的（　）和（　）成员，将其作为子类的成员，但不能继承父类的（　）和（　　）成员。

11．(　）直接赋值给（　　）时，子类对象可自动转换为父类对象，(　）赋值给（　）时，必须将父类对象强制转换为子类对象。

12．Java 的多态性主要表现在（　）、（　）和（　）三个方面。

13．重写后的方法不能比被重写的方法有（　）的访问权限，重写后的方法不能比被重写的方法产生更多的异常。

14．Java 语言中，定义子类时，使用关键字（　）来给出父类名。如果没有指出父类，则该类的默认父类为（　　）。

15．Java 语言中，重载方法的选择是在编译时进行的，系统根据（　）、（　）和参数顺序寻找匹配方法。

16．实现接口中的抽象方法时，必须使用（完全相同）的方法头，并且还要用（public）修饰符。

17．接口中定义的数据成员均是（　），所有成员方法均为（　）方法，且没有（　）方法。

18．this 代表（　）的引用，super 表示的是当前对象的直接父类对象。

19．如果一个类包含一个或多个 abstract 方法，则它是一个（　）类。

20．Java 不直接支持多继承，但可以通过（　）实现多继承。类的继承具有（　）性。

21．没有子类的类称为（　），不能被子类重载的方法称为（　），不能改变值的量称为常量，又称为（　）。

22．一个接口可以通过关键字 extends 来继承（　）其他接口。

23．接口中只能包含（public static final）类型的成员变量和（public abstract）类型的成员方法。

24．一般地，内部类又分为定义在方法体外的（　）和定义在方法体内的（　）两种。

25．静态内部类可直接通过外部类名引用，其一般格式是（　）。

26．匿名类一般分为（　）和（　）类两种。

27．面向对象的软件设计中，根据目的不同模式可分为（　）、（　）和（　）三种。

二、选择题

1．下面关于类的继承性的描述中，错误的是（　）。

A．继承是在已有的基础上生成新类的一种方法

B．Java 语言要求一个子类只有一个父类

C．父类中成员的访问权限在子类中将被改变

D．子类继承父类的所有成员，但不包括私有的成员方法

2．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（　　）。

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

3．在类的修饰符中，规定只能被同一包类所使用的修饰符是（　　）。

A．public　　B．默认　　C．final　　D．abstract

4．下列关于子类继承父类的成员描述中，错误的是（　　）。

A．当子类中出现成员方法头与父类方法头相同的方法时，子类成员方法覆盖父类中的成员方法。

B．方法重载是编译时处理的，而方法覆盖是在运行时处理的。

C．子类中继承父类中的所有成员都可以访问。

D．子类中定义有与父类同名变量时，在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。

5．定义一个类名为“MyClass.java”的类，并且该类可被一个工程中的所有类访问，则下面哪些声明是正确的？（　）

A．public class MyClass extends Object　　B．public class MyClass

C．private class MyClass extends Object　　D .class MyClass extends Object

6．下列关于继承性的描述中，错误的是（　）。

A．一个类可以同时生成多个子类B ．子类继承了父类中除私有的成员以外的其他成员

C．Java 支持单重继承和多重继承D ．Java 通过接口可使子类使用多个父类的成员

7．下列关于抽象类的描述中，错误的是（　）。

A．抽象类是用修饰符 abstract 说明的　　B．抽象类是不可以定义对象的

C．抽象类是不可以有构造方法的　　D．抽象类通常要有它的子类

8．设有如下类的定义：

```
public class parent {

int change() {}
```

}

class Child extends Parent { }

则，下面哪些方法可加入 Child 类中？（ ）

A．public int change(){ } B．int chang(int i){ }

C．private int change(){ } D．abstract int chang(){ }

9．下列关于构造方法的叙述中，错误的是（ ）。

A．构造方法名与类名必须相同 B．构造方法没有返回值，且不用 void 声明

C．构造方法只能通过 new 自动调用 D．构造方法不可以重载，但可以继承

10．下面叙述中，错误的是（ ）。

A．子类继承父类 B．子类能替代父类 C．父类包含子类 D．父类不能替代子类

11．下列对多态性的描述中，错误的是（ ）。

A．Java 语言允许方法重载与方法覆盖 B．Java 语言允许运算符重载

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

12．下面关于接口的描述中，错误的是（　　）。

A．一个类只允许继承一个接口　　B．定义接口使用的关键字是 interface

C．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

D．接口实际上是由常量和抽象方法构成的特殊类

13．欲构造 ArrayList 类的一个实例，此类继承了 List 接口，下列哪个方法是正确的？（　　）

A．ArrayList myList=new Object();　　B．ArrayList myList=new List();

C．List myList=new ArrayList();　　D．List myList=new List();

14．下列哪些方法与方法 public void add(int a){}为合理的重载方法？（　　）

A．public void add(char a)　　B．public int add(int a)

C．public void add(int a,int b)　　D．public void add(float a)

15．MAX_LENGTH 是 int 型 public 成员变量，变量值保持为常量 100，其定义是（　　）。

A．public int MAX_LENGTH=100;　　B．final public int MAX_LENGTH=100;

C．public final int MAX_LENGTH=100;　　D．final int MAX_LENGTH=100;

三、判断题

1．Java 语言中，构造方法是不可以继承的。(　　)

2．子类的成员变量和成员方法的数目一定大于等于父类的成员变量和成员方法的数目。(　　)

3．抽象方法是一种只有说明而无具体实现的方法。(　　)

4．Java 语言中，所创建的子类都应有一个父类。(　　)

5．调用 this 或 super 构造方法的语句必须放在第一条语句。(　　)

6．一个类可以实现多个接口，接口可以实现“多重继承”。 (　　)

7．实现接口的类不能是抽象类。(　　)

8．使用构造方法只能给实例成员变量赋初值。(　　)

9．Java 语言不允许同时继承一个类并实现一个接口。(　　)

四、分析题

1．分析下面的程序，写出运行结果。

```
public class Exercises6_1 extends TT{

public void main(String args[]){

Exercises6_1 t = new Exercises6_1("Tom");

}

public Exercises6_1(String s){

super(s);

System.out.println("How do you do?");

}

public Exercises6_1(){

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);
}
}
```

运行结果是：(　　)

2．分析下面的程序，写出运行结果。

```
public class Exercises6_2 {
private static int count;
private String name;
```

```
public class Student {

private int count;

private String name;

public void Output(int count) {

count++;

this.count++;

Exercises6_2.count++;

Exercises6_2.this.count++;

System.out.println(count + " " + this.count + " "

+ Exercises6_2.count + " " + Exercises6_2.this.count++);

}

}

public Student aStu() {
```

```
return new Student();

}

public static void main(String args[]) {

Exercises6_2 g3 = new Exercises6_2();

g3.count = 10;

Exercises6_2.Student s1 = g3.aStu();

s1.Output(5);

}

}
```

运行结果是：(　　)

3．分析下面的程序，写出运行结果。

```
class Exercises6_3 {

class Dog {
```

```
private String name;

private int age;

public int step;

Dog(String s, int a) {

name = s;

age = a;

step = 0;

}

public void run(Dog fast) {

fast.step++;

}

}

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