

JAVA 第三、四章测验

一、 单选题（20 题，每题 2 分，共 40 分）

- 1、 下列对封装性的描述中，错误的是（ ）
 - A. 封装体包含了属性和方法
 - B. 封装体中的属性和方法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
- 2、 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. 子类不能继承父类所有的方法和变量
 - C. JAVA 语言支持单重继承和多重继承
 - D. JAVA 语言通过接口克服 JAVA 单继承的缺点
- 3、 下列对多态性的描述中，错误的是（ ）
 - A. JAVA 语言允许运算符重载
 - B. JAVA 语言允许方法重载
 - C. JAVA 语言允许变量覆盖
 - D. 多态性提高了程序的抽象性和简洁性
- 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（ ）
 - A. public
 - B. 默认
 - C. final
 - D. abstract
- 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. private
 - B. public
 - C. 默认
 - D. protected
- 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（ ）
 - A. public
 - B. private
 - C. static
 - D. protected
- 7、 下列关于构造方法的特点的描述中，错误的是（ ）
 - A. 不可重载
 - B. 方法名同类名
 - C. 无返回类型
 - D. 系统自动调用
- 8、 下列关于静态方法的描述中，错误的是（ ）
 - A. 在类体内说明静态方法使用关键字 static
 - B. 静态方法只能处理静态成员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
- 9、 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
- 10、 关于 super，下面说法中错误的是（ ）
 - A. 用来访问父类被隐藏的成员变量
 - B. 用来调用父类中被重写的方法
 - C. 用于调用父类的构造方法
 - D. 用于定义父类
- 11、 下列关于子类继承父类的成员的描述中，错误的是（ ）

- A. 子类可继承父类的所有成员，并都可以直接访问
 - B. 当子类中定义有与父类同名的变量时：子类继承的父类方法中，使用继承的父类的变量；子类执行自己的方法中，使用的是自己定义的变量
 - C. 当子类中出现成员方法与父类相同的成员方法时，子类成员方法覆盖父类中的成员方法
 - D. 方法重载是编译时处理的，而方法覆盖是在运行时处理的
- 12、 下列关于接口的描述中，错误的是（ ）
- A. 接口实际上是由常量和抽象方法构成的特殊类
 - B. 一个类只允许继承一个接口
 - C. 定义接口使用的关键字是 `interface`
 - D. 在继承接口的类中通常要给出接口中定义的抽象方法的具体实现
- 13、 下列关于包的描述中，错误的是（ ）
- A. 包是一种特殊的类
 - B. 包是若干个类的集合
 - C. 包是使用 `package` 语句创建的
 - D. 包有有名包和无名包两种
- 14、 下列常用包中，存放用户图形界面类库的包是（ ）
- A. `java.awt`
 - B. `java.lang`
 - C. `java.util`
 - D. `java.io`
- 15、 下列是系统提供的常用的类，所有类的父类的类是（ ）
- A. `Math`
 - B. `Object`
 - C. `System`
 - D. `String`
- 16、 分析下列程序：（ ）

```
public class Test{
    public void changeInt(int x){
        x+=12;
        System.out.print(x+" ");
    }
    public static void main(String[] args){
        int intVar=10;
        System.out.print(intVar+" ");
        Test Obj=new Test();
        Obj.changeInt(intVar);
        System.out.println(intVar);
    }
}
```

- A. 10 22 22
 - B. 10 22 10
 - C. 22 22 22
 - D. 以上结果都不正确
- 17、关于下面的程序，结论是正确的是（ ）

```
public class Test{
    Test (){
        System.out.println("2");
    }
    public static void main(String[] args){
        Test obj=new Test();
        obj.Test ();
        System.out.println("1");
    }
}
```

- ```

 }
}

```
- A. 程序可以通过编译并正常运行，结果输出“21”
  - B. 程序可以通过编译并正常运行，结果输出“221”
  - C. 程序无法通过编译
  - D. 程序可以通过编译，但无法正常运行

18、关于下面的程序，结论是正确的（ ）

```

public class Test extends A{
 int data;
 Test(int sd,int d){
 data=d;
 super(sd);
 }
 public static void main(String args[]){
 Test obj=new Test(2,3);
 System.out.println(obj.superData+obj.data);
 }
}
class A{
 int superData;
 A(int d){
 superData=d;
 }
}

```

- A. 程序可以通过编译并正常运行，结果输出“5”
- B. 程序可以通过编译并正常运行，结果输出“23”
- C. 程序无法通过编译
- D. 程序可以通过编译，但无法正常运行

19、关于下面的程序，结论是正确的（ ）

```

public class Test extends B{
 Test(){
 super();
 System.out.print("3");
 }
 public static void main(String args[]){
 Test obj=new Test();
 }
}
class A{
 A(){
 System.out.print("1");
 }
}
class B extends A{

```

```

B(){
 System.out.print("2");
}
}

```

- A. 程序可以通过编译并正常运行，结果输出“123”
  - B. 程序可以通过编译并正常运行，结果输出“23”
  - C. 程序可以通过编译并正常运行，结果输出“213”
  - D. 程序无法通过编译
- 20、下面方法的定义置于画线处将会导致编译错误的是（ ）

```

class A{
 float getNum(){
 return 3.5f;
 }
}
class B extends A{

}

```

- A. float getNum(){return 4.5f;}
- B. double getNum(){return 4.5;}
- C. void getNum(double d){}
- D. double getNum(float f){return 4.5;}

## 二、 判断题（每小题 1 分，共 10 分）

- 1、在类体内说明成员变量时不允许赋初值。（ ）
- 2、抽象方法是一种只有说明而无具体实现的方法。（ ）
- 3、最终方法是不能被当前子类重新定义的方法。（ ）
- 4、JAVA 语言中，方法调用一律都是传址的引用调用。（ ）
- 5、使用构造方法只能给非静态成员变量赋初值。（ ）
- 6、JAVA 语言中，所创建的子类都应有一个父类。（ ）
- 7、JAVA 语言中，构造方法是可以继承的。（ ）
- 8、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。（ ）
- 9、子类对象可以赋值给父类对象；而父类对象不可以直接赋值给子类对象。（ ）
- 10、实现接口的类不能是抽象类。（ ）

## 三、 写出下列程序的运行结果（每小题 5 分，共 20 分）

- 1、class A{
 static int a=0;
 int b;
 public A(int a){
 this.a=a;
 }
 public A(int a,int b){

```

 this(a);
 this.b=b;
 }
 public int add(){
 return a+b;
 }
 public void display(){
 System.out.println("a="+a+",b="+b);
 System.out.println("a+b="+this.add());
 }
}
public class Test{
 public static void main(String args[]){
 A a=new A(10,5);
 A b=new A(20,8);
 a.display();
 b.display();
 }
}

```

2、

```

class A{
 int a=10;
 static int b=20;
 void f1(){
 System.out.println("A.a="+a++);
 }
 static void f2(){
 System.out.println("A.b="+b++);
 }
}
public class Test extends A{
 int a=100;
 static int b=200;
 void f1(){
 System.out.println("Test.a="+a++);
 }
 static void f2(){
 System.out.println("Test.b="+b++);
 }
}

```

```

 public static void main(String args[]){
 A t=new Test();
 t.f1();
 t.f2();
 System.out.println("a="+t.a);
 System.out.println("b="+t.b);
 Test s=new Test();
 System.out.println("a="+s.a);
 System.out.println("b="+s.b);
 }
}

```

```

3、 class A{
 int a;
 A(int i){
 a=i;
 }
}
class B extends A{
 int a,b;
 B(int i,int j){
 super(i);
 b=j;
 a=i+j;
 }
}
public class AB{
 public static void main(String args[]){
 A a1=new A(9),a2;
 B b=new B(3,5);
 System.out.println(b.a);
 a2=a1;
 System.out.println(a2.a);
 a2=b;
 System.out.println(a2.a);
 b=(B)a2;
 System.out.println(b.a);
 }
}

```

```
 }
}
```

```
4、class A{
 A(){
 System.out.println("class A");
 }
}
class B extends A{
 int b;
 B(int i){
 b=i;
 }
}
class C extends B{
 private int c;
 C(int i,int j){
 super(j);
 c=i;
 }
 int returnc(){
 return c;
 }
}
class Test{
 public static void main(String args[]){
 C c=new C(5,18);
 System.out.println("c:"+c.b*c.returnc());
 }
}
```

#### 四、程序设计题（3 题，每题 10 分，共 30 分）

1、定义一个 Employee（雇员）类，其中包含：

属性：String 类型的 name(姓名)、int 类型的 salary（薪水）和 String 类型的 sex（性别）。

方法：Employee(String,int,String)(构造方法)、show()(显示雇员信息的方法)。

并编写一个测试类 **Demo**，测试类中的所有方法。（测试数据自拟）

2、定义一个 **Manager**(经理)类，该类继承 **Employee** 类，其中包含：

属性：**String** 类型的 **department**（部门）。

方法：**Manager(String,int,String,String)**(构造方法，要求调用父类的构造方法)、**show()**(显示经理信息的方法，要求调用父类的显示方法)。

并编写一个测试类 **Demo**，测试类中的所有方法。（测试数据自拟）

3、编写一个 **abstract** 类，类名为 **Geometry**，该类有一个 **abstract** 方法：**public abstract double**



`getArea()`。编写一个矩形类 `Rect`，自行定义其需要的属性，并重写 `getArea()` 方法（计算矩形的面积）。编写一个圆类 `Circle`，同样自行定义其需要的属性，并重写 `getArea()` 方法（计算圆的面积）。编写 `TotalArea` 类，用来计算各种图形的面积之和，该类有一个名为 `tuxing` 的对象数组，数组元素类型为 `Geometry`，该类还有一个 `public double computerTotalArea()` 方法，该方法返回 `tuxing` 的元素调用 `getArea()` 方法返回的面积之和。最后写一个测试类 `Test`，计算一个圆和一个矩形的面积之和。（测试数据自拟）