

一、选择题（去年的考试没有单选题，估计今年也不会有）

1. 实现结构化程序设计的主要途径是 [D]。
A. 自顶向下，逐步求精 B. 模块化 C. 语句结构化 D. 以上均是
2. 对象应该具有的基本特征是 [D]。
A. 自治性 B. 封闭性、通信性 C. 被动型、暂存性 D. 以上均是
3. Java 程序设计语言的基本特征是 [D]。
A. 简洁性、面向对象、分布式 B. 健壮性、结构中立、安全性、多线程
C. 可移植性、解释执行、高性能、动态性 D. 以上均是
4. 在定义抽象类时，class <类名> 前不可以使用的修饰符是 [A]。
A. final B. public C. 缺省 D. abstract
5. 下面是有关对象的论述，正确的说法是 [C]。
A. 对象是一种特殊类的描述 B. 声明引用型变量就是创建对象
C. 对象是类的实例化，具体化 D. 对象是类的抽象
6. 在方法的定义中，下面说法正确的是 [C]。
A. 必须有形式参数列表 B. 必须有 return 语句
C. 可通过使用 return 语句返回值 D. 不能将 private 作为方法的修饰符
7. 以下关于构造方法的描述，错误的是 [A]。
A. 构造方法的返回类型只能是 void 型
B. 构造方法是类的一种特殊方法，它的方法名必须与类名相同
C. 构造方法的主要作用是完成对类的初始化工作
D. 一般在创建新的对象时，系统会自动调用构造方法
8. Java 中，关于继承的论述中，错误的是 [B]。
A. 继承具有传递性 B. 继承关系也叫泛化，即（IS-A）关系
C. 子类继承父类中的所有成员 D. 子类继承父类，提高了系统的重用性
继承关系是泛化关系的反关系，也就是说子类是从父类继承的，而父类则是子类的泛化，从子到父是泛化，从父到子是继承
9. 定义一个名为 Key 的类，使之不能被继承，应是 [D]。
A. class Key { } B. protected class Key { }
C. abstract class Key { } D. final class Key { }
10. 下面是 Java 程序中有关接口的论述，错误的说法是 [C]。
A. 接口可以使得设计和实现分离 B. 声明接口可以达到规范某类对象操作的目的
C. 接口中定义的属性成员分为常量和变量两种 D. 接口不可以被实例化
11. 下面是有关 protected 访问属性的论述，正确的说法是 [D]。
A. 具有 protected 访问属性的成员变量和成员方法只能被本类访问
B. 具有 protected 访问属性的成员变量和成员方法只能被本包的类访问
C. 具有 protected 访问属性的成员变量和成员方法只能被子类访问
D. 具有 protected 访问属性的成员变量和成员方法只能被本包中的类和其他包中的子类访问
12. 下面是有关构造方法的论述，正确的说法是 [C]。
A. 构造方法是指方法名与类相同，且返回值为 int 类型的成员方法
B. 在同一个类中，构造方法之间不允许相互调用
C. 在创建对象时，系统自动调用构造方法，其主要任务是进行对象的初始化
D. 在任何情况下，子类中的构造方法都不需要调用父类的构造方法
13. 下面是有关监听器的论述，正确的说法是 [A]。
A. 监听器是一个实现了监听器接口的类对象 B. 监听器是一个处理事件的成员方法
C. 监听器的主要目的是处理用户界面的布局
D. 在 Java 程序中，所有事件都采用同一个监听器进行处理
14. 下列关于 JFrame 类的说法不正确的是 [D]。

- A. JFrame 类是顶层容器，其上面可添加各种通用容器和组件
B. JFrame 对象显示的效果是一个窗口
C. JFrame 被默认初始化不可见
D. JFrame 没有默认的布局管理器
15. 下面是有关 JFrame 类的论述，正确的说法是 [A]。
A. JFrame 组件是一种顶层容器
B. JFrame 组件是一种可以嵌套在另一个顶层容器中的容器
C. JFrame 组件是一种可以嵌套在另一个顶层容器中的普通组件
D. JFrame 组件是一种普通的组件
16. 一个好的类应该是 [B]
A. 尽量承担多个系统责任 B. 只承担一个系统责任
C. 包含很多成员方法 D. 包含很多成员变量
17. 在面向对象分析时，从大量的实体中选取系统所需要的实体，这项工作称为 [B]
A. 继承 B. 抽象 C. 多态 D. 封装
18. 继承是面向对象编程的一个重要特征，它可降低程序的复杂性并使程序 [B]
A. 可读性好 B. 代码可重用 C. 可以跨包访问 D. 运行更安全
19. 在下面的描述中，正确的是 [A]
A. 子类对象完全可以作为父类对象使用 B. 子类对象不可以作为父类对象使用
C. 父类对象完全可以作为子类对象使用 D. 以上说法均不对
20. 下面是有关 Java 异常处理的论述，正确的说法是 [C]。
A. 异常是指语法错误
B. 在 Java 中必须由产生异常的对象处理异常
C. 在 Java 中，所有异常类的父类是 Throwable
D. 在 Java 程序中，对任何类型的异常都可以不进行处理
21. JPanel 容器又称为面板容器，其主要特点不包括有 [A]
A. 可以作为顶层容器 B. 可以嵌套放置
C. 组件可直接放置在面板容器中 D. 默认的布局方式是 FlowLayout
22. 下面关于抽象类的描述中，错误的是 [BD]
A. 抽象类中可以定义构造方法 B. 抽象类中不可以定义一般方法
C. 抽象类中可以定义抽象方法 D. 抽象类中不可以定义 final 方法
23. 在面向对象的程序设计中，类可以作为 [C]
A. 类型定义机制 B. 封装机制 C. 类型定义机制和封装机制 D. 都不是
24. 在下面的描述中，哪个描述是对封装最准确的解释 [D]
A. 将方法集成为一个逻辑单元 B. 将变量集成为一个逻辑单元
C. 将方法和变量集成为一个逻辑单元
D. 将方法和变量集成为一个逻辑单元并隐藏某些细节
25. 在下面关于接口的描述中，错误的是 [D]
A. 接口与类的层次无关 B. 接口实现了设计和实现的分离
C. 接口定义了对应的交互界面 D. 接口与存储空间有关
26. 下列关于 JPanel 类的说法不正确的是 [C]
A. JPanel 类是非顶层容器 B. JPanel 的显示效果不是一个窗口
C. JPanel 中不可以再放置 JPanel D. JPanel 默认是流布局管理器
27. 下面是有关继承的论述，正确的说法是 [C]
A. 在 Java 程序中，子类只继承父类的成员变量
B. 在 Java 程序中，子类只继承父类的成员方法
C. 在 Java 程序中，子类可以覆盖父类中的成员方法
D. 在 Java 程序中，子类可以直接访问父类中的全部内容

28. 若已有下列类定义, 则选项中正确的是 [[C]]
- ```
class Animal { }
class Cat extends Animal { }
class PetCat extends Cat { }
```
- A. Cat cat = new Animal();                      B. PetCat cat = new Animal();  
C. Animal animal = new PetCat();                D. PetCat cat = new Cat();
29. 下列关于构造方法的描述, 错误的是。 [ [ B ] ]
- A. 构造方法名必须和类名相同                      B. 构造方法可以使用 void 来说明没有返回值  
C. 构造方法是对象做初始化工作                      D. 构造方法一般不能由编程人员显示调用
30. 下面描述正确的是。 [ [ A ] ]
- A. final 可以修饰类、方法和成员变量                      B. abstract 可以修饰类、方法和成员变量  
C. 定义抽象方法要定义返回值类型、方法名、形参表和方法体  
D. 用 final 修饰的变量, 在程序中可以对该变量的值进行修改
31. 下面是 Java 程序中有关接口的论述, 正确的说法是 [ [ D ] ]。
- A. 与类一样, Java 中的接口只能单继承                      B. 在 Java 程序中, 不能声明接口的引用变量  
C. 在接口中, 只可以包含抽象方法  
D. 接口是一种特殊的类, 其中仅包含抽象方法和静态常量
32. 下列关于容器的描述中, 正确的是 [ [ C ] ]
- A. 一个独立运行的 GUI 应用程序, 可以没有容器                      B. JFrame 中必须有 JPanel  
C. 组件可以加到容器中                      D. 按钮是一种容器
33. 使用哪个修饰符修饰的类不能被继承。 [ [ A ] ]
- A. final                      B. abstract                      C. synchronized                      D. static
34. 下面哪个不是面向对象技术的内容之一 [ [ D ] ]
- A. 多态                      B. 封装                      C. 继承                      D. 结构化
35. 构造方法何时被调用。 [ [ A ] ]
- A. 创建对象时                      B. 定义类时                      C. 调用对象的一般方法时                      D. 使用对象的属性时
36. 下列描述中, 哪个是正确的? [ [ A ] ]
- A. 类是变量和方法的集合体                      B. 数组是无序数据的集合  
C. 抽象类可以实例化                      D. 类成员数据必须是公有的
37. 当父类和子类在同一个包中时, 在子类中可以直接使用父类中的那些成员 [ [ B ] ]
- A. public、protected、private                      B. 友好、protected、private  
C. public、友好、protected                      D. public、友好、private
38. 如果类中的一个成员变量需要被该类的全部对象共同使用, 则该成员变量必须是。 [ [ C ] ]
- A. final 变量                      B. abstract 变量                      C. static 变量                      D. synchronized 变量
39. 下列描述中, 哪个是正确的? [ [ AC ] ]
- A. 抽象可以提高代码重用度                      B. 抽象可以隐藏细节  
C. 抽象可以降低问题的复杂性                      D. 抽象可以提高程序的安全性
40. 成员变量不能使用下面的哪个修饰符来定义 [ [ D ] ]
- A. public                      B. final                      C. protected                      D. abstract
41. 下列关于继承的描述, 哪个是正确的? [ [ A ] ]
- A. 子类能继承父类的所有方法和成员变量                      B. 子类只继承父类的非私有方法和成员变量  
C. 子类只能继承父类 public 方法和成员变量                      D. 子类只继承父类的方法, 而不继承成员变量
42. 封装就是将数据和对这组数据进行处理的操作组装在一起, 形成一个实体, 这个实体也就是 [ [ A ] ]
- A. 类                      B. 对象                      C. 方法                      D. 数据块
43. 下面是有关面向对象程序设计概念的论述, 正确的说法是 [ [ C ] ]。
- A. 面向对象的主要特性是正确性、健壮性、重用性、多态性  
B. 可以通过继承手段实现面向对象强调的信息隐藏

- C. 面向对象是一种以对象为中心，分析、设计、编写程序的方法  
D. 面向对象程序属于过程驱动
44. 下列有关适配器的论述中，正确的是 [ B ]  
A. 适配器是一种接口  
B. 适配器是一种实现了监听接口的类  
C. 适配器包含的成员方法都是抽象的  
D. 任何一个监听器接口都有对应的适配器
45. 下列类中，属于组合关系的是 [ CD ]  
A. Student 和 Person  
B. Car 和 SportCar  
C. Book 和 Person  
D. Car 和 Wheel
46. 下列说法中，正确的是 [ A ]  
A. 包含主类的 Java 源程序文件在命名时必须与主类名一致  
B. 类与类之间必须是平行的关系，即：一个类中不能嵌套定义另一个类  
C. 抽象类中定义的所有方法都是抽象的方法  
D. 以上均不正确
47. 下列选项中，属于 Java 接口的是 [ A ]  
A. ItemListener  
B. ActionEvent  
C. MouseAdapter  
D. actionPerformed
48. 抛出异常，应使用的语句是 [ B ]  
A. catch  
B. throw  
C. try  
D. finally
49. 下列关于接口的描述中，正确的是 [ B ]  
A. 接口中的成员变量既可以是常量，也可以是变量  
B. 接口中的成员方法都是抽象的  
C. 对于一个实现了接口的类，只需要实现接口中想要重新定义的成员方法即可  
D. 在接口中可以包含非抽象的成员方法

## 二、填空题

1. Java 的源程序文件的扩展名是 .java，经过编译后生成的代码叫做 字节码 文件，其扩展名是 class。Java 源文件中可以包含多个类，但只能有一个类是 主类 或 包含有 main() 方法的类。
2. 面向对象程序设计方法具有 4 个主要特征，它们是抽象性、封装性、继承性、多态性。
3. 接口是 常量 和 抽象方法 的集合，接口中没有方法的实现，接口中所有的方法需要在 子类 中实现。从本质上讲，接口是一种特殊的抽象类，通过接口可以实现 Java 类之间 多 继承结构。
4. 类和对象的关系：类是对象的 抽象，对象是类 具体化或实例化。
5. 类之间主要存在 3 种关系：依赖、泛化、聚合，飞机和发动机之间是 聚合 关系，动物和猫之间是 泛化 关系，人和自行车是 依赖 关系，公司与员工之间是 聚合 关系。
6. 在面向对象的程序设计中，封装是指将对象的属性和行为分别用 成员变量 和 成员方法 描述，并将它们绑定在一起形成一个可供访问的基本逻辑单元。
7. 创建一个名为 com 的包的语句是 package com;。
8. 定义抽象类的关键字是 abstract。
9. Java 中所有类的基类是 Object，一个类可以有 1 个直接父类，可以实现 多 个接口，接口中的方法默认都是 抽象的 或 abstract。
10. 接口中声明的方法默认都是 public abstract，接口中声明的变量都是 公共、静态、最终，接口之间 多继承 继承。
11. JPanel 的默认布局管理器是 FlowLayout，JFrame 的默认布局管理器是 BorderLayout。
12. 对象是用来描述现实世界中客体的部件，是面向对象软件系统在运行时 的基本单位。为了区分属于同一个类的不同对象，每个对象都有 一个唯一 的标识。



13. 异常的总父类是 Throwable，派生出 Error 异常和 Exception 异常。
14. 继承是类之间的一种常见关系。这种关系为共享数据和操作提供了一种良好的机制。通过继承，一个类的定义可以基于另外一个已经存在的类。利用继承机制可以大大提高程序的 代码重用 和 扩展性。
15. 事件处理机制的三要素包括 事件源、事件对象、事件监听器。
16. 一个类通过使用关键字 implements 声明自己实现的接口，如果一个类实现了某个接口，那么这个类必须重写该接口中的 全部抽象方法，否则该类必须是 抽象类。
17. 面向过程的程序设计采用的是 过程 抽象，而面向对象的程序设计采用的是 数据 抽象。
18. 在类的继承关系中，父类描述了实体的 一般 性，子类描述了实体的 特殊 性。
19. 不同的类对象收到同一个消息可以产生 不同的 响应效果，这种现象叫做多态。利用多态机制，用户可以发送一个通用的消息，而实现的细节由 子类决定，这样，同一个消息会调用 不同功能 的方法。
20. 在创建对象时，系统自动调用 构造方法，其主要任务是进行对象的初始化。
21. 一个类的核心内容是 数据，一般对外界 保密。
22. 抽象类中可以定义抽象方法，定义抽象方法的目的是 统一接口、实现多态。
23. 消息（事件）传递是对象之间相互联系的唯一途径。发送者发送消息，接收者通过调用 方法 来响应消息，这个过程被不断地重复，使得应用程序在人的有效控制下运转起来，最终得到相应的结果。消息 是驱动面向对象程序运转的源泉，对象之间相互联系（作用、影响）的唯一途径是 消息。
24. 在 Java 语言中，子类对象完全可以作为父类对象来使用，这种情况称为 动态多态。
25. 在静态成员方法中只能使用 static 成员变量，只能调用 static 成员方法。
26. 类的构造方法 不能 使用 static 来修饰，类的构造方法 不能 使用 abstract 来修饰。
27. 在子类中使用关键字 super 做前缀可调用被子类覆盖的父类中的方法或被子类隐藏的父类变量。
28. 在类的实例化时，调用构造方法的次序是先 父 后 子，在对象被撤消时，调用析构方法的次序是先 子 后 父。
29. 在 Java 语言中，类之间只能 单 继承，而 接口 是可以用来实现多重继承的数据类型。
30. 在 Java 中有两种实现多态的方法，一种是使用方法的 重载 实现静态多态（静态绑定），另一种是使用方法的 覆盖 实现动态多态（动态绑定）。
31. 容器可以分为 顶层 容器和 通用 容器，容器中的边框式布局管理器将整个容器分为 5 部分。
32. 在程序中捕获和处理异常，必须先捕获和处理 子类 异常，后捕获和处理 父类 异常。
33. 在 Java 语言中，传递对象时传递的是该对象的 地址。
34. 面向对象程序设计方法应该包含 类、对象、继承、消息、通信机制。
35. 用过程式语言编写的程序其主要特征是：程序= 过程 + 调用，而用面向对象程序设计出的程序，其主要特征是：程序= 对象 + 消息。
36. 类可以嵌套定义，在一个类中嵌套定义的类称为这个类的 内部类。
37. 对象是对现实世界的一个具体事物的 模拟（描述），而类是对一个对象集合的 统一描述。
38. Java 中 类的实例成员变量有 4 种访问控制符，按照访问权限从小到大的顺序，依次是 private、缺省、protected 和 public。
39. 事件监听器是 实现了事件处理接口的对象，适配器是实现了 事件处理接口的类。
40. 完成程序，下列程序的功能是打印 2 个点之间的距离。

```

class Point{
 private int x,y;
 public Point(int x,int y){
 ____ this.x=x; ____
 ____ this.y=y; ____
 }
 double distance(Point p){
 int dx = __ x-p.x; ____
 int dy = __ y-p.y; ____
 return Math.sqrt(dx * dx + dy * dy) ;
 }
}

public class Test {
 public static void main(String[] args) {
 Point p1 = new Point(2,5);
 Point p2 = new Point(6,8);
 System.out.println("2 点距离: " + __ p1.distance(p2) ____);
 }
}

```

## **P2.distance(p1)**

41. 面向对象是一种以\_\_**对象**\_\_为中心，分析、设计、编写程序的方法。
42. 实例成员变量属于\_\_**对象**\_\_，类成员变量属于\_\_**类**\_\_。
43. 重载方法调用时根据其参数的\_\_**类型**\_\_、个数和顺序来区分，Java 方法重载要求同名的方法必须有不同的\_\_**参数表**\_\_，靠返回类型不同不能实现两个重载的方法。
44. 异常处理的三个步骤是\_\_**异常**\_\_抛出、\_\_**捕获**\_\_异常、\_\_**处理**\_\_异常。
45. 当父类和子类中的方法\_\_**签名相同**\_\_，时，称子类覆盖(重写)了父类的方法。
46. 利用继承实现动态多态的条件是\_\_**子类覆盖父类方法**\_\_、\_\_**子类对象上转型**\_\_。
47. 父类和子类不在同一个包中，这时在子类中\_\_**可以**\_\_直接使用父类的 public 成员、\_\_**可以**\_\_直接使用父类的 protected 成员、\_\_**不能**\_\_直接使用父类的 private 成员、\_\_**不能**\_\_直接使用父类的 友好(默认) 成员。如果在同一个包中呢？
48. 在子类的构造方法中，首先要用\_\_**super()**\_\_来调用父类的\_\_**构造方法**\_\_。在类的构造方法中，还可以首先用 this 来调用\_\_**本类的其他构造方法**\_\_。
49. 每个对象中都有一个\_\_**super**\_\_成员和一个\_\_**this**\_\_成员，用来指定该对象本身和父类。
50. 对象应该具有的基本特性是\_\_**自治性**\_\_、\_\_**封闭性**\_\_、\_\_**通信性**\_\_、\_\_**被动性**\_\_、\_\_**暂时性**\_\_。
51. 一个对象要能够被克隆，必须实现\_\_**Cloneable**\_\_接口。
52. 在内部类中，\_\_**可以**\_\_直接使用外部类的成员。
53. JVM 的工作过程是\_\_**加载类**\_\_、\_\_**校验类**\_\_、\_\_**执行类**\_\_。

## 三、写出下面程序的运行结果

1.

```

class A {
 private int a;
 public A(int a) {this.a=a;}
 public String toString() {

```

```

 return "a="+a;
 }
}
public class B extends A{
 private int a;
 public B() {super(20) ; a=10;}
 public B(int a) {super(a*2); this.a=a;}
 public String getA() { return "a="+a;}
 public String toString() {
 return super.toString()+getA();
 }
 public static void main(String[] args) {
 System.out.println(new B(30));
 }
}
a=60a=30

```

2.

```

public class Test {
 public static void main(String[] args) {
 int[] a = {1,2,3};
 int s=0,i=1,j=2;
 try {
 switch(i) {
 case 1: {
 j=j/(j-1);
 System.out.println("i=" + i + "; ");
 i++; }
 case 2: {
 System.out.println("i=" + i + "; ");
 s = s + a[i];
 i++; }
 case 3: {
 System.out.println("i=" + i + "; ");
 s = s + a[i]; }
 }
 System.out.print("没有异常出现");
 }
 catch (ArithmeticException e) {
 System.out.println("捕获异常, "+"算术类异常:"); }
 catch (ArrayIndexOutOfBoundsException e1) {
 System.out.print("捕获异常, "+"数组下表越界异常:"); }
 }
}

```

**i=1;**

**i=2;**

**i=3;**

**捕获异常,数组下表越界异常:**

3、将程序补充完整。

接口定义如下:

```
Interface Figure {
 double showarea(); //计算面积
}
```

//定义矩形类（Rectangle 类），实现上述接口，仅写出必要的方法

```
class Rectangle _____ implements Figure _____ {
 double x, y;
 _____ public Rectangle() {x=y=1;} _____ //不带参数的构造方法
 _____ public Rectangle(double x,double y) {this.xx=this.y=y;}_ //带参数的构造方法
 _____ public double showarea() {return x*y;}_____ //必要的方法
 (其他语句忽略不写)
}
```

#### 4、将程序补充完整

有如下泛型类 Data 定义,请补充完整:

```
class Data<T>{
 private __T_____ first;
 public Data(){ }
 public Data(T first){ this.first=first; }
 public __T_____ getData() { return first; }
 public void setData(T first) { this.first=first; }
}
```

以下的泛型类使用，请将其补充完整。

```
public class DragDrawTest {
 public static void main(String args[]){
 Data < Integer> intpair= new Data _<Integer>_____ (new Integer(21));
 Data <String> strpair= new Data __<String>_____ ("Hello World");
 Integer val1,val2;
 String str1,str2;
 val1=intpair.getData();
 str1=strpair.getData();
 System.out.println("val1="+ val1);
 System.out.println("str1="+str1);
 }
}
```

#### 5、阅读下列程序，写出运行结果。

```
class Test {
 int value() {
 return 10;
 }
 int value(String str) {
 return 12;
 }
 int value(int v) {
 return v;
 }
 public static void main(String[] args) {
```



```

 Test t = new Test();
 System.out.print(t.value() + " ");
 System.out.print(t.value(8) + " ");
 System.out.print(t.value("Hello world"));
 }
}

```

**10 8 12**

6、

```

class Animal {
 public Animal() {
 this(2);
 System.out.print(" animal: ");
 }
 public Animal(int id) {
 System.out.print("id: " + id);
 }
}
class Cat extends Animal {
 public Cat() {
 System.out.println("cat ");
 }
}
public class Test {
 public static void main(String[] args) {
 new Cat();
 }
}

```

**id: 2 animal: cat**

7、

3.写出下面程序的运行结果

```

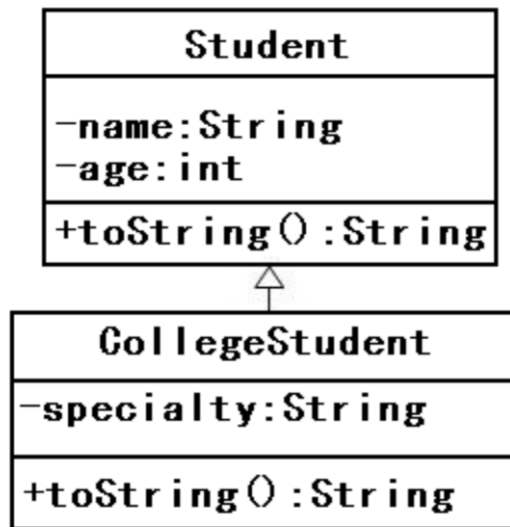
class A {
 private int a;
 public A(int a) {this.a=a;}
 public String toString() {
 return "a="+a;
 }
}
public class B extends A{
 private int a;
 public B() {super(20) ; a=10;}
 public B(int a) {super(a*2); this.a=a;}
 public String getA() { return "a="+a;}
 public String toString() {
 return super.toString()+getA();
 }
 public static void main(String[] args) {
 System.out.println(new B(30));
 }
}

```

```
}
}
```

#### 四、按照要求完成类定义

1. 请根据给定的 UML 类图，编写必要的公共类，UML 类图如下：



要求：

1). 每个类必须有 2 个公共构造方法，一个无参构造方法，一个有参构造方法，这些构造方法为类中的成员变量做初始化。

2). 父类的 toString() 方法返回学生的姓名和年龄，子类的 toString() 方法返回大学生的姓名、年龄和专业，例如：

```
Student s=new CollegeStudent("李平",23,"人文科学");
System.out.println(s);
```

这时，屏幕上将显示：**李平,23 岁,人文科学专业**

```
Public class Student {
 private String name;
 private int age;
 public Student() {name="aaa";age=15;}
 public Student(String name, int age) {this.name=name ; this.age=age;}
 //为2个变量配备 get/set 方法
 public String toString() {
 return name+", "+age+"岁";
 }
}

Public class CollegeStudent extends Student {
 private String specialty;
 public CollegeStudent() { specialty ="bbb";}
 public CollegeStudent(String name, int age, String specialty) {
 super(name,age) ;
 this.specialty=specialty;
 }
 //为1个变量配备 get/set 方法
```

```

 public String toString() {
 return super.toString()+" "+specialty;
 }
}

```

2. 设计一个课程与教材管理系统。其中应该包含课程类（CoureseClass）和教材类（ReferenceMaterialClass）。课程类（CoureseClass）包括课程编号、课程名称、学时、性质（选修、必修）、授课教师编号、使用教材书号等属性和相关行为的描述等；教材类（ReferenceMaterialClass）包括书号、书名、作者姓名、价格等属性和相关行为的描述。

要求：

- (1) 设计上述两个类应该包含的成员变量和成员方法（包括构造方法）。为了减少书写量，类中的所有成员方法只要求写出原型，不必写出方法体内部的代码。

```

class CoureseClass {
 private String no; //课程编号
 private String name; //课程名称
 private int time; //学时
 private boolean type; //性质（选修、必修）
 private String tno; //授课教师编号
 private ReferenceMaterialClass book; //教材
 public CoureseClass() {
 //省略
 }
 public CoureseClass(String no, String name, int time, boolean type,
 String tno, ReferenceMaterialClass book) {
 this.no=no;
 //省略
 }
 //为变量配备 get/set 方法
 public String toString() {
 return no;
 }
}

```

```

class ReferenceMaterialClass {
 private String no; //书号
 private String name; //书名
 private String bName; //作者姓名
 private float price; //价格
 public ReferenceMaterialClass() {
 //省略
 }
 public ReferenceMaterialClass(String no, String name, String bName, float price) {
 this.no=no;
 //省略
 }
 //为变量配备 get/set 方法
}

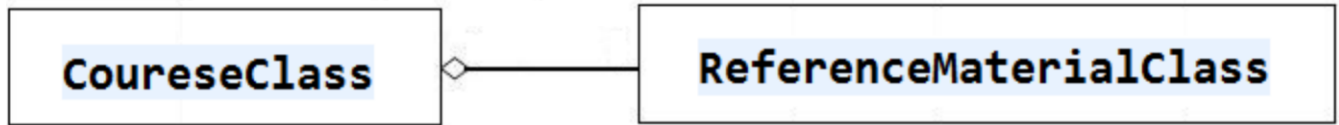
```

```

 public String toString() {
 return no;
 }
}

```

(2) 画出描述两个类之间关系的 UML 类图。



3、定义一个 Date 类，属性是私有的，包括 year,month,day，定义一个不带参数的构造方法，一个带参数的构造方法，一个 toString 方法。

【Date 类定义】

```

Public class Date {
 private int year,month,day;
 public Date() {year=2017;month=5;day=27;}
 public Date(int year, int month, int day) {
 this.year=year;
 this.month=month;
 this.day=day;
 }
 //为3个变量配备 get/set 方法
 public String toString() {
 return year+"年"+month+"月"+day+"日";
 }
}

```

定义一个 Book 类，属性是私有的，包括书名(name)、价格(price)和一个 Date 类型的出版日期，定义一个不带参数的构造方法，一个带参数的构造方法，一个公有方法 setName,一个公有方法 getPrice，一个 toString 方法。

【Book 类定义】

```

Public class Book {
 private String name;
 private float price;
 private Date date;
 public Book() {
 name="";
 price=0;
 date=new Date(2017,5,20);
 }
 public Book(String name, float price, Date date) {
 this.name=name;
 this.price=price;
 this.date=date;
 }
 public void setName(String name) {this.name=name;}
 public float getPrice() {return price;}
 public String toString() {
 return name+", 价格"+price+"元, 出版日期"+date.toString();
 }
}

```

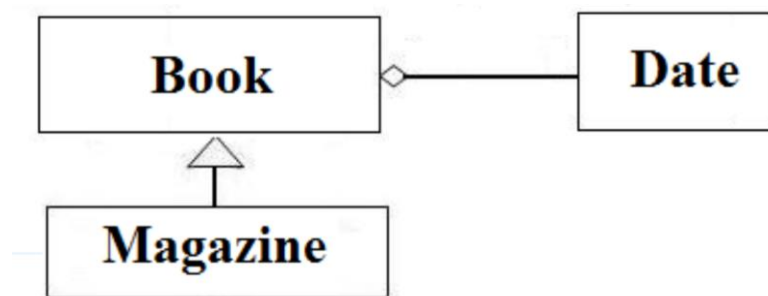
```
}
```

定义一个 Magazine 类，从 Book 类继承，增加一个 color 属性，定义一个不带参数的构造方法，一个带参数的构造方法，一个公有方法 seColor, 一个 toString 方法。

【Magazine 类定义】

```
Public class Magazine extends Book{
 private String color;
 public Magazine() {
 color="";
 }
 public Magazine(String name, float price, String color, Date date) {
 super(name, price, date);
 this.color = color;;
 }
 public void setColor(String Color) {this.color=Color;}
 public String toString() {
 return super.toString()+" "+color+"色";
 }
}
```

画出描述上述三个类 Date、Book、Magazine 之间关系的 UML 类图。



4、定义一个类 Vehicle，属性有字符串类型的 rego，该属性只能被本类的方法访问，定义一个不带参的构造方法，一个带参的构造方法，一个 toString 方法。

```
Vehicle v1 = new Vehicle("High"); System.out.println(v1); // 显示 REGO: High
```

```
Vehicle v2 = new Vehicle("High"); System.out.println(v2); // 显示 REGO: Low
```

【Vehicle 类定义】

```
Public class Vehicle {
 private String rego;
 public Vehicle() {rego="Low";}
 public Vehicle(String rego) {this.rego=rego;}
 //为1个变量配备 get/set 方法
 public String toString() {
 return "REGO:"+rego;
 }
}
```

定义 Vehicle 的子类 Truck，增加一个 boolean 型的属性 fourWheelDrive，该属性只能被本类的方法访问，定义一个带参的构造方法，一个 toString 方法。若 fourWheelDrive 的值为真显示 rego 的值和“4WD”，否则显示 rego 的值和“2WD”

```
Truck t1 = new Truck("Low", true); System.out.println(t1); // 显示 REGO: Low 4WD
```

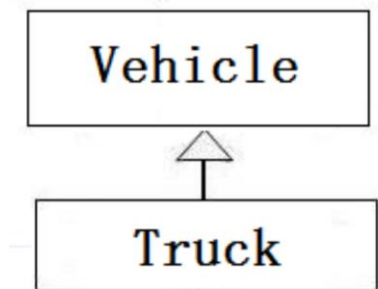
```
Truck t2 = new Truck("Low", false); System.out.println(t2); // 显示 REGO: Low 2WD
```

【Truck 类定义】

```

Public class Truck extends Vehicle {
 private boolean fourWheelDrive;
 public Truck() {fourWheelDrive=true;}
 public Truck(String rego, boolean fourWheelDrive) {
 super(rego);
 this.fourWheelDrive=fourWheelDrive;
 }
 //为1个变量配备 get/set 方法
 public String toString() {
 return super.toString()+((fourWheelDrive)?"4WD":"2WD");
 }
}

```



## 五、按要求将程序补充完整

### 1、下列程序实现如下两个功能：

- 1) 程序运行后，显示如图 1 所示的窗口外观。
- 2) 当在两个文本框中输入 int 型数值，且点击任意一个按钮时后，将会在两个文本框之间显示相应的运算符，并在右面显示运行结果，如图 2 所示。



图 1 初始显示状态

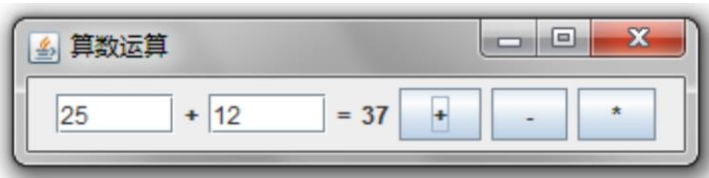


图 2 点击“+”按钮之后的显示状态

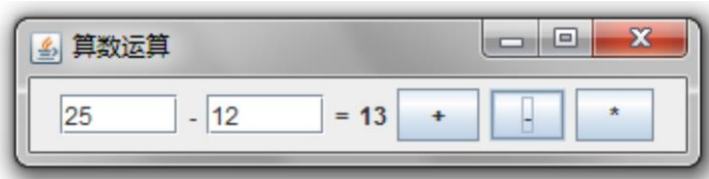


图 3 点击“-”按钮之后的显示状态



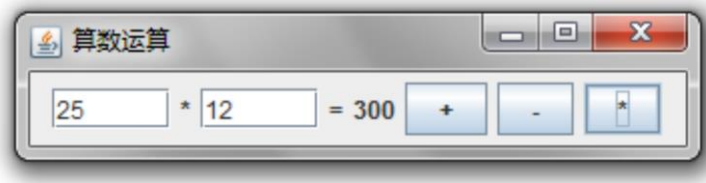


图4 点击“\*”按钮之后的显示状态

```
import javax.swing.*;
public class Test {
 public static void main(String[] args) {
 JFrame frame = new JFrame("算数运算");
 frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
 frame.getContentPane().add(new Operator());
 frame.pack();
 frame.setVisible(true);
 }
}
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
public class Operator extends JPanel {
 private JTextField a1=new JTextField(5);
 private JTextField a2=new JTextField(5);
 private JLabel a3=new JLabel("=");
 private JLabel a4=new JLabel(" ");
 private JButton b1=new JButton("+");
 private JButton b2=new JButton("-");
 private JButton b3=new JButton("*");

 public Operator() {
 add(a1);add(a2);add(a3);add(a4);
 add(b1);add(b2);add(b3);
 M1 m1=new M1();
 b1.addActionListener(m1);
 b2.addActionListener(m1);
 b3.addActionListener(m1);
 }
 class M1 implements ActionListener {
 @Override
 public void actionPerformed(ActionEvent e) {
 double a=Double.parseDouble(a1.getText());
 double b=Double.parseDouble(a2.getText());
 double c=0;
 Object obj=e.getSource();
 if (obj==b1) {c=a+b; }
 if (obj==b2) {c=a-b; }
 if (obj==b3) {c=a*b; }
 a4.setText(""+c);
 }
 }
}
```

```

 }
 }
}

```

## 2、将下列程序补充完整。

1) 程序运行后，显示如图 1 所示的窗口外观。

2、当在第 1 个文本框中输入一个正整型数值后，点击任意一个按钮，将会在第 2 个文本框中显示这个数值对应的相应进制数值，如图 2 所示。



图 1 初始显示状态



图 2 点击“二进制”按钮之后的显示状态

提示：进制转换不用自己写，调用下面的方法。布局管理器用默认的就可以。

在 `Integer` 类中含有下面三个有关数值进制转换的静态方法：

`public static String toHexString(int k)` //将 `k` 转换为十六进制，返回类型为 `String`

`public static String toOctalString(int k)` //将 `k` 转换为八进制，返回类型为 `String`

`public static String toBinaryString(int k)` //将 `k` 转换为二进制，返回类型为 `String`

`import javax.swing.*;`

`public class Test {`

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

`JFrame frame = new JFrame("进制转换");`

`frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);`

`frame.getContentPane().add(new Operator());`

`frame.pack();`

`frame.setVisible(true);`

`}`

`}`

`import javax.swing.*;`

`import java.awt.*;`

```

import java.awt.event.*;

public class Change extends JPanel {
private JTextField a1=new JTextField(5);
private JTextField a2=new JTextField(15);
private JButton b1=new JButton("二进制");
private JButton b2=new JButton("八进制");
private JButton b3=new JButton("十六进制");
public Change() {
 add(a1);add(a2);
 add(b1);add(b2);add(b3);
 M1 m1=new M1();
 b1.addActionListener(m1);
 b2.addActionListener(m1);
 b3.addActionListener(m1);
}
class M1 implements ActionListener {
 @Override
 public void actionPerformed(ActionEvent e) {
 int a=Integer.parseInt(a1.getText());
 Object obj=e.getSource();
 if (obj==b1) { a2.setText(Integer.toBinaryString(a)); }
 if (obj==b2) { a2.setText(Integer.toOctalString(a)); }
 if (obj==b3) { a2.setText(Integer.toHexString(a)); }
 }
}
}

```

### 3、根据如下题目要求，编写一个 Java 应用程序。

在一个窗口中实现下列操作：当用户点击“画直线”按钮后，在中央区域拖动鼠标实现画直线的功能，当点击“显示当前坐标”按钮后，在中间区域任意位置按下鼠标，则显示当前的坐标位置，当用户点击“红色”按钮后，显示区域中的内容以红色显示，当用户点击“绿色”按钮后，显示区域中的内容以绿色显示，如下图所示。

#### 【提示】

默认颜色为红色

Color k = Color.red;

MouseEvent 事件类中的 2 个方法：

public int getX(); // 返回当前鼠标点的 x 坐标，

public int getY(); // 返回当前鼠标点的 y 坐标。

**MouseAdapter**    **MouseListener**

public void mousePressed(MouseEvent e);

**MouseMotionAdapter**    **MouseMotionListener**

public void mouseDragged(MouseEvent e);

Graphics 中的 2 个方法：

drawLine(int x1,int y1,int x2,int y2);

drawString(String str, int x,int y)



图 1 选画直线后，拖动鼠标画线



图 2 选显示当前坐标后，按下鼠标显示坐标

```
public class I extends JFrame {
 public static final int POS=1;
 public static final int LINE=2;
 private JButton b1=new JButton("红色");
 private JButton b2=new JButton("绿色");
 private JButton b3=new JButton("画直线");
 private JButton b4=new JButton("显示当前位置");
 private Color color=Color.black;
 private int flag=0;
 private int x,y,x1,y1;

 public T() {
```

```

 add(b1, BorderLayout.WEST); add(b2, BorderLayout.EAST);
 add(b3, BorderLayout.SOUTH); add(b4, BorderLayout.NORTH);
 B1 b=new B1();
 b1.addActionListener(b); b2.addActionListener(b);
 b3.addActionListener(b); b4.addActionListener(b);
 addMouseListener(new M1()); addMouseMotionListener(new M2());
 setSize(500,400);
 setVisible(true);
 }
 class B1 implements ActionListener {
 @Override
 public void actionPerformed(ActionEvent e) {
 Object obj=e.getSource();
 if (obj==b1) { color=Color.red; }
 if (obj==b2) { color=Color.green; }
 if (obj==b3) { flag=LINE; }
 if (obj==b4) { flag=POS; }
 }
 }
 class M1 extends MouseAdapter {
 public void mousePressed(MouseEvent e) {
 x=x1=e.getX(); y=y1=e.getY();
 if (flag==POS) {
 Graphics g=getGraphics();
 g.setColor(color);
 g.drawString("(" +x+", "+y+")", x,y);
 g.dispose();
 }
 }
 public void mouseReleased(MouseEvent e) {
 if (flag==LINE && (x!=x1 || y!=y1)) {
 Graphics g=getGraphics();
 g.setColor(color);
 g.drawLine(x,y,x1,y1);
 g.dispose();
 }
 }
 }
 class M2 extends MouseMotionAdapter {
 public void mouseDragged(MouseEvent e) {
 x1=e.getX(); y1=e.getY();
 }
 }
 public static void main(String[] args) {
 new T();
 }
}

```

#### 4、根据要求，编写一个 Java 应用程序。

JFrame 窗口采用采用流式布局管理器，在窗口中放置一个 JButton 按钮，当鼠标光标移入按钮时，该按钮的背景颜色变为红色，当鼠标光标移出按钮时，该按钮的背景颜色变为黄色。

##### 【编程提示】

1. Color.red 为红色，Color.yellow 为黄色。

2. JButton 按钮的 setBackground(Color) 方法可以改变按钮的背景颜色。

3. **MouseAdapter** 中的 2 个事件处理方法：

public void mouseEntered(MouseEvent e) 方法处理鼠标移入

public void mouseExited(MouseEvent e); 方法处理鼠标移出

程序的部分代码如下，请将程序构造方法中的代码补充完整(不允许修改给定的代码)：

```
public class Test extends JFrame{
 private JButton btn=new JButton("按钮"); //按钮
 public static void main(String[] args) { //主方法
 new Test();
 }
 public Test() { //构造方法(请补充代码)
 super("鼠标事件处理") ;
 add(btn,"North");
 MouseAdapter m1=new MouseAdapter() {
 public void mouseEntered(MouseEvent e) {
 btn.setBackground(Color.red);
 }
 public void mouseExited(MouseEvent e) {
 btn.setBackground(Color.yellow);
 }
 };
 btn.addMouseListener(m1);
 setSize(500,400);
 setVisible(true);
 }
}
```



## 一、填空题

1. 创建类的对象时，使用运算符 new 给对象分配内存空间。
2. 定义类的构造方法不能有返回值类型，其名称与 类 名相同，一个文件中有 最多 1 个公共类，要用 公共类 的类名做文件名。
3. Java 中，接口 是可以用来实现类间多重继承功能的结构。
4. 若有类定义：class B extends A{ } 则类 B 是类 A 的 子类。
5. 接口中的成员只有公共静态常量和 公共抽象方法。
6. 在子类中使用关键字 super 做前缀可调用被子类覆盖的父类中的方法。
7. Java 语言中，通常把可能发生异常的方法调用语句放到 try 块中，并用紧跟其后的 catch 块来捕获和处理异常。
8. 在一个 Java 源文件中定义了 3 个类，编译该 Java 源文件时会产生 3 个字节码文件。
9. 如果子类中的某个方法的名字、返回值类型和参数列表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（重写） 了父类的同名方法。
10. 调用一个类的静态（static）方法，可以不创建该类的对象，直接通过 类名 做前缀来调用。
11. 如果定义一个方法时，其方法体中的语句可能抛出异常，但又没有通过 try...catch 处理异常，则应在其方法头中使用关键字 throws 声明该方法 抛出 异常。

## 二、判断

- (1). 匿名类中可以定义构造函数。  
【 错 】
- (2). 用 final 类实现一个接口时，必须要重写接口中定义的全部抽象方法。 【 对 】
- (3). 在抽象类中必须要有抽象方法。  
【 错 】
- (4). 在内部类中可以直接使用外部类中的任何成员。 【 错 】
- (5). 用 protected 修饰的方法可以被任何包中的类使用。 【 错 】
- (6). 类一定有构造函数。  
【 对 】
- (7). 在静态内部类中可以直接使用外部类中的所有成员。 【 错 】
- (8). 接口中定义的方法的访问描述符都是 private。 【 错 】
- (9). 运行时异常一定要由程序自身来捕获和处理。  
【 错 】
- (10). 一个 Java 源程序文件中可以有多个主类（公共类）。 【 错 】
- (11). 方法的重载是指类中的两个方法的方法名和形参表完全一样。 【 错 】

- (12). 通过父类句柄来使用子类对象，这种做法不会产生任何问题。 【 对 】
- (13). Java 中类之间只能单继承，接口之间也只能单继承。 【 错 】
- (14). 在静态方法中不能使用 this 和 super。 【 对 】
- (15). 在 Java 图形界面中，所有组件的父类是 Component 类。 【 对 】
- (16). 事件适配器是 Java 类库提供的已经实现了某个事件接口的类。 【 对 】
- (17). 布局管理器用来管理容器中组件的位置和大小的对象。 【 对 】
- (18). ActionListener 接口中定义了 actionPerformed 方法。 【 对 】
- (19). Java 程序的文件名必须和主类名一样。 【 对 】
- (20). final 和 abstract 修饰符可以同时用来修饰同一个方法。 【 错 】
- (21). private 方法同时也是 final 方法。 【 对 】
- (22). 在捕获异常时，要先捕获子类描述的异常，后捕获父类描述的异常。 【 对 】
- (23). BorderLayout 布局将容器分成 4 个区域。 【 错 】
- (24). 在子类的构造函数中，使用 this( ) 来调用父类的构造函数。 【 错 】

### 三、阅读程序，回答问题

1.

```
1: import java.util.Arrays;
2:
3: public class SortArray {
4: public static void main(String args[]) {
5: String[] str = {"size", "abs", "length", "class"};
6: Arrays.sort(str);
7: for (int i=0; i<str.length; i++)
8: System.out.print(str[i]+" ");
9: }
10: }
```

(1) 写出程序运行后的结果。(2 分)

**abs      class      length      size**

(2) 方法调用 str[0].length() 的返回结果是多少？(3 分)

**排序后是 3，排序前是 4**

2.

```
1: //TestKeyListener.java
2: import java.applet.Applet;
3: import java.awt.*;
4: import java.awt.event.*;
```

```

5:
6: public class TestKeyListener extends JPanel {
7:
8: Color clr;
9:
10: public void TestKeyListener () {
11:
12: addKeyListener(
13: new KeyAdapter() {
14: public void keyTyped(KeyEvent e) {
15: if(e.getKeyChar()=='r' || e.getKeyChar()=='R')
16: clr=Color.red;
17: else if(e.getKeyChar()=='g' || e.getKeyChar()=='G')
18: clr=Color.green;
19: else if(e.getKeyChar()=='b' || e.getKeyChar()=='B')
20: clr=Color.blue;
21: else
22: clr=Color.black;
23: setBackground(clr);
24: }
25: });
26: }
27: }

```

(1) 程序第 15、16 行实现的功能是什么？(2 分)

输入字母 r 时选择红色

(2) 如果敲击键盘的'b'键，有什么结果？(3 分)

面板背景色为蓝色

3.

```

1: class A
2: {
3: int a=1;
4: double d=2.0;
5: void show() {
6: System.out.println("Class A: a="+a+"\td="+d);
7: }
8: }
9: class B extends A
10: {

```

```

11: float a=3.0f;
12: String d="Java program.";
13: void show() {
14: super.show();
15: System.out.println("Class B: a="+a+"\td="+d);
16: }
17: }

```

(1) 若在应用程序的 main 方法中有以下语句：

```

A a=new A();
a.show();

```

则输出的结果如何？(2 分)

**Class A: a=1 d=2.0**

(2) 若在应用程序的 main 方法中定义类 B 的对象 b：

```

A b=new B();
b.show();

```

则输出的结果如何？(3 分)

**Class A: a=1 d=2.0**

**Class B: a=3.0 d=Java program.**

4.

```

1: public class A {
2: public static void main(String args[]) {
3: MyInterface obj2=new B("B");
4: obj2.show();
5: }
6: }
7:
8: interface MyInterface {
9: double G=88;
10: void show();
11: }
12:
13: class B implements MyInterface {
14:
15: String str;
16: public B(String s) {

```

```

17: str=s;
18: }
19: public void show() {
20: System.out.println(str+": G="+G);
21: }
22: }

```

(1) 程序第 8~11 行定义的 MyInterface 是类还是接口？(2 分)

**接口**

(2) 程序第 4 行输出的结果是什么？(3 分)

**B: G=88.0**

## 四、程序编制

1. 设计一个 Java 的 GUI 应用程序，实现下列功能：

- (1) 程序运行后，在窗口中显示如图 1 所示的字符串。
- (2) 当点击鼠标时，在窗口中显示鼠标当前的坐标位置，显示如图 2 所示。
- (3) 当拖动鼠标时，在窗口中随着鼠标的移动绘制曲线，并显示鼠标当前的坐标位置，如图 3 所示。

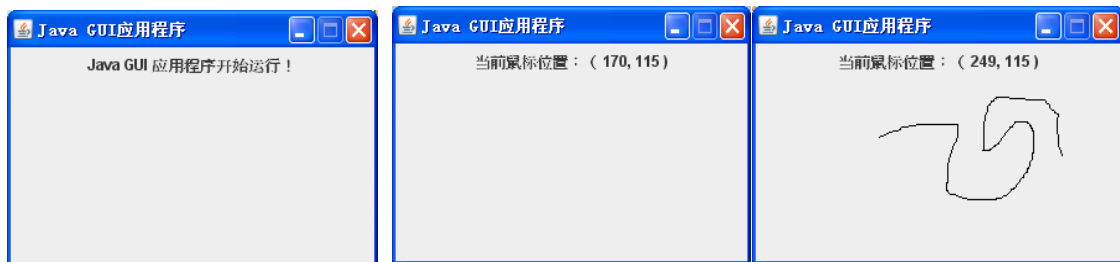


图 1

图 2

图 3

要求：根据题目要求写出三个鼠标事件响应成员方法的内容。

```

public class I extends JFrame{

 private int x,y; //鼠标按下时的位置

 private int x1,y1; //鼠标拖动时的位置

 private JLabel pos=new JLabel("Java GUI 应用程序开始运行",JLabel.CENTER);

 public T() {

 super("Java GUI 应用程序");

 add(pos,"North");

 addMouseListener(new M1());

 addMouseMotionListener(new M2());

 setSize(500,400);
 }
}

```

```

 setVisible(true);
 }

 //处理鼠标按下事件的内部类
 class M1 extends MouseAdapter {
 public void mousePressed(MouseEvent e) {
 x=e.getX();y=e.getY();

 pos.setText("当前鼠标位置("+x+", "+y+"");
 }
 }

 //处理鼠标拖动事件的内部类
 class M2 extends MouseMotionAdapter {
 public void mouseDragged(MouseEvent e) {
 x1=e.getX();y1=e.getY();
 Graphics2D g2d=(Graphics2D) getGraphics();
 g2d.draw(new Line2D.Double(x,y,x1,y1));
 g2d.dispose();
 x=x1;y=y1;

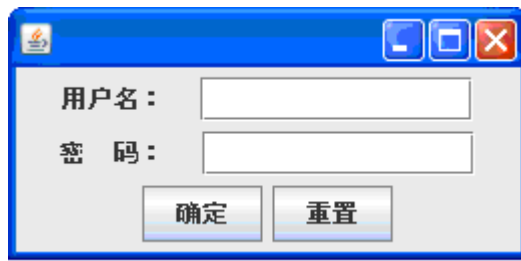
 pos.setText("当前鼠标位置("+x+", "+y+"");
 }
 }

 public static void main(String[] args) {
 new T();
 }
}

```

2. 根据题目要求，编写一个用户登录的 Java 应用程序。





操作过程为：

(1) 用户输入用户名和密码。

(2) 点击“确定”按钮，验证用户名和密码；如果正确，弹出一个“欢迎进入”的提示框；否则，弹出一个“用户名/密码输入错误！”的提示框；点击“重置”按钮，清空所有文本框的内容。

【提示】

(1) 用户名、密码均自己规定；

(2) 显示提示框的方法为：

`JOptionPane.showMessageDialog(null, String message);` //message 是提示框中显示的信息内容。

```
public class I extends JFrame{

 private JTextField t1=new JTextField();

 private JPasswordField p1=new JPasswordField();

 private JButton b1=new JButton("确定");

 private JButton b2=new JButton("重置");

 public T() {

 setLayout(new GridLayout(3,2));

 add(new JLabel("用户名:"));add(t1);

 add(new JLabel("密 码:"));add(p1);

 add(b1);add(b2);

 M1 m1=new M1();

 b1.addActionListener(m1);

 b2.addActionListener(m1);

 pack();

 setVisible(true);

 }

}
```

//处理动作事件的内部类

```
class M1 implements ActionListener {

 public void actionPerformed(ActionEvent e) {

 Object obj=e.getSource();

 if (obj==b1) {

 String tt1=t1.getText();

 String pp1=new String(p1.getPassword());

 if (tt1.equals("123") && pp1.equals("123"))

 JOptionPane.showMessageDialog(null , "正确");

 else

 JOptionPane.showMessageDialog(null , "错误");

 }else if (obj==b2) {

 t1.setText("");

 p1.setText("");

 }

 }

}

public static void main(String[] args) {

 new T();

}

}
```

