《Java 程序设计》实验报告

年级、专业、班级	2018 级计科卓越班	姓名	秦凡
实验题目	基于 GUI 和套接字的网络绘图板程序设计与实现		
实验时间	2020.5.5～2020.5.26	实验地点	四川省南充市蓬安县河舒工业园区桂花路 12 号
实验成绩		实验性质	□验证性 □设计性 ■综合性

教师评价：

□算法/实验过程正确；□源程序/实验内容提交 □程序结构/实验步骤合理；

□实验结果正确； □语法、语义正确； □报告规范；

其他：

评价教师签名：

一、实验目的

运用面向对象程序设计思想，基于 java 的 GUI、套接字和多线程机制，实现基绘图板程序。

二、实验项目内容

1.借鉴 实验一的类的架构构建绘图类集合，shape、shape 类的派生类和 graphics 类；

2.将实验一 的 OpenGLApp 改为 Jframe 类的派生类，包含 Graphics 类实例作为数据成员，利用特定的布局管理器（layout）构建主窗口，在窗口中创建相应的控件允许用户选择当前绘制图形的形状、线条颜色、填充颜色等；

3.添加相应的事件监听和相应方法（处理用户的输入），例如绘制图形的形状、线条颜色、填充颜色的选择变化；

4.重载主窗口的 paintComponent（Graphics g）方法，用于在主窗口中绘制 Graphics 类实例的图形数据，绘制过程中调用这些图形数据自己的 draw()方法，可以把主窗口的 g 传给每个图形元素类的 draw 方法 ；

5. 在主窗口中注册的 MouseListener 接口中相关处理方法，根据当前的用户选择实现类似 window 附件中 painter 工具的功能，例如：绘制指定图形和移动图形。

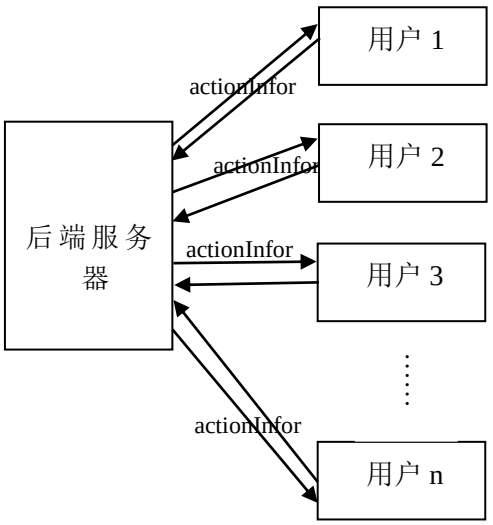
6. 利用 sock、多线程机制，修改程序结构实现一个基于 java sock 的网络白板程序(多个用户协同白板绘图程序)的设计与实现。

报告创建时间：2020 年 5 月 19 日 星期二 18 时 46 分

三、实验过程或算法（参照 MVC 模式进行类的设计，程序实现流程设计和算法设计）

1） 总设计思想：

用户面板与服务器实现分离。当用户 1 号在自己的面板进行了操作后，将其操作的信息通过类 actionInfor 传递到服务器，然后服务器将这个操作信息类 actionInfor 的实例分别传递给各个用户，当各个用户接收到了这个操作实例后，产生相应动作，如绘制，移动图形，给图形填充颜色，改变图形形状。



actionInfor 类成员变量：

```
// 0 代表新绘制图形， 1 代表 move 图形， 2 代表 reshape 图形， 3 代表 fill 图形
public int type=0;  // 默认取值为 0
// 当 type 等于 0 时有效，此时 addedShape 表示某一个用户新绘制的图形
public shape addedShape=null;
// pointS 表示用户按下鼠标时的鼠标位置
public point pointS=null;
// pointS 表示用户松开鼠标时的鼠标位置
public point pointE=null;
```

2） 监听用户的各个操作：

drawListener 类用来监听用户的各个操作。在 drawListener 类中添加鼠标监听事件，具体包括鼠标按下事件，鼠标释放事件，鼠标拖动事件以及鼠标点击事件。对于用户选择功能的实现，还添加了按钮的点击监听事件，以确定用户操作的类型。

drawListener 成员变量：

```
//x1,y1 分别是鼠标按下时的横坐标和纵坐标
//x2,y2 分别是鼠标释放时的横坐标和纵坐标
```

```
    private int x1, y1, x2, y2;
    // g2 是用于绘制图形的画笔
    private Graphics2D g2;
    //str 为功能按钮的字符，取值有"Move","Reshape","Rect", "Triangle","Cube"
    private String str = "Triangle";
    //绘制的图形的颜色
    private Color color;
    //所有要绘制的图形的集合
    private QFGraphics graphics;    // 自己写的图形集合类
    //绘制的面板
    private JFrame j;
    //用户新绘制的图形
    private shape addedShape;
    //用户找面板上的操作，要传递给服务器
    private actionInfor ai=null;
```

3）绘图用户类 PaintClient。

这个类用来将用户与服务器后端联系在一起，两者进行数据交互。PaintClient 通过 socket = **new** Socket(**host**, port)开启一个 socket 通信，等待服务器处理请求，然后通过 socket.getInputStream()和 socket.getOutputStream()建立与服务器的输入输出流联系。通过监听用户是否产生了新的操作，一旦检测到用户有新的操作后，通过操作生成新的 actionInfor 实例，将其传递到后端。同时用户类 PaintClient 也将一直等待后端传递来的 actionInfor 实例，接受到后产生相应的操作。

PaintClient 类的成员变量：

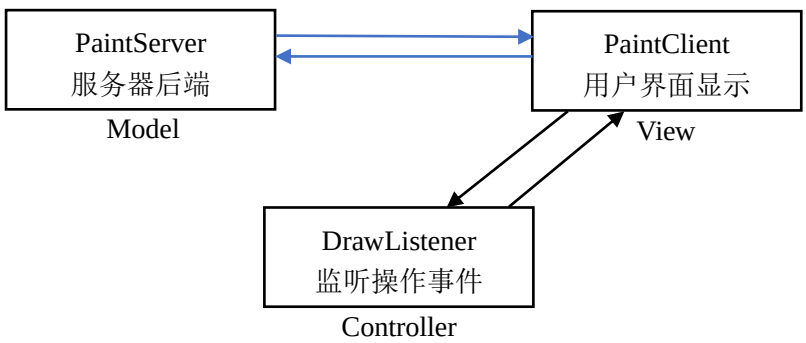
```
  //用户产生的操作
  private actionInfor ai=null;
  // Create and initialize a title label
  private JLabel jlblTitle = new JLabel();
  // Create and initialize a status label
  private JLabel jlblStatus = new JLabel();
  // Input and output streams from/to server
  private ObjectInputStream fromServer=null;
  private ObjectOutputStream toServer=null;
```

4）后端服务器类：

后端服务器类共有两个类，PaintServer 类用于建立与用户的联系，为每一个用户开启一个线程类：HandleASession 类。HandleASession 类用来处理请求及接受和发送数据。在 HandleASession 类中，为了使接受数据和发送数据不相互制约，在 HandleASession 类中分别为数据发送和数据接受写了线程内部类，这样发送数据就不用收到接受数据的制约了，接受数据也不会受到发送数据的制约。

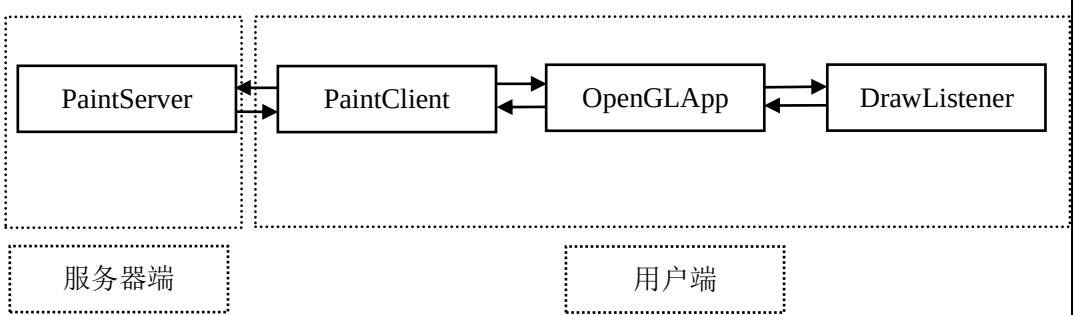
5）MVC 设计模式及思想：

服务器后端开可以看做 Model 层，服务器后端接收数据后，对已经被数据序列化后的数据进行反序列化，然后再将反序列化后的数据传送到每个用户。PaintClient 可以看做是 View 层，其负责处理用户与后端的交互，将检测到的用户的新操作发送给后端，同时也接收用户的新操作传递给 DrawListener 类，它不负责数据的处理，只是数据的传递。DrawListner 类可以看做是 Controller 层，其负责数据的处理，对接收到的 actionListener 实例进行处理，同时又将监听到的操作通过产生 actionListener 实例发送给 PainClient，再通过 PainLient 发送到服务器后端。



6）数据流向：

将所有数据都封装在 actionInfor 类里面，一是简化了代码，二也可以方便程序的扩展，比如想要实现其他功能，传递的信息只需改变 actionInfor 类即可。下面的图反应了 actionInfor 类数据的流向。



7）双缓冲技术

在编写程序的时候，我们绘制图形时存在严重的闪烁。这种闪烁虽然不会给程序的效果造成太大的影响，但着实有违设计初衷，也给程序的使用者造成了些许不便。有时候我们通过 repaint()来调用 paint()，其实也是产生了一个 AWT 线程，然后通过 AWT 线程来调用 update()，再由 update()来调用

paint()。在这里，我直接对 paint（）方法进行了重载。以下是实现双环醇，解决闪烁问题的核心代码。

核心代码，重载了原本了 paint（）方法：

```
// 重载 paint（）方法，实现双缓冲技术，去掉闪烁
public void paint(Graphics g){
  //创建 Image 空对象，保存绘制内容
  Image iBuffer = null;
  //创建 Graphics 空对象，作为参数传递给原方法函数的输入
  Graphics gBuffer=null;
  if(iBuffer==null) {
     //创建 Image 空对象，并设置大小
     iBuffer=this.createImage(this.getSize().width,this.getSize().height);
   //获得 Graphics 实例
     gBuffer=iBuffer.getGraphics();
  }
//设置背景
gBuffer.setColor(this.getBackground());
  //调用原本的方法，只是传入的是我们自己的 Graphics
  super.paint(gBuffer);
  //画出，呈现
  g.drawImage(iBuffer,0,0,this);
}
```

8）图形的 reshape 操作。

图形的 reshape 操作，及更改图形特征点，使得图形变为符合用户想得到的形状。当用户选择了 reshape 按钮后，我们首先检测用户按下鼠标的位置，然后根据这个位置去判断哪个图形包含这个位置（对于图形重叠情况，至于顶层的会优先选中），如果图形 S 包含了这个鼠标坐标，再检测这个鼠标坐标为中的且边长为 15px 的正方体时候包含这个图形的特征点，如果包含某个特征点，则这个特征点就是即将被操作的点，当用户释放了鼠标后，将选中的这个点变为鼠标释放时的坐标点。

9）图形的 move 操作。

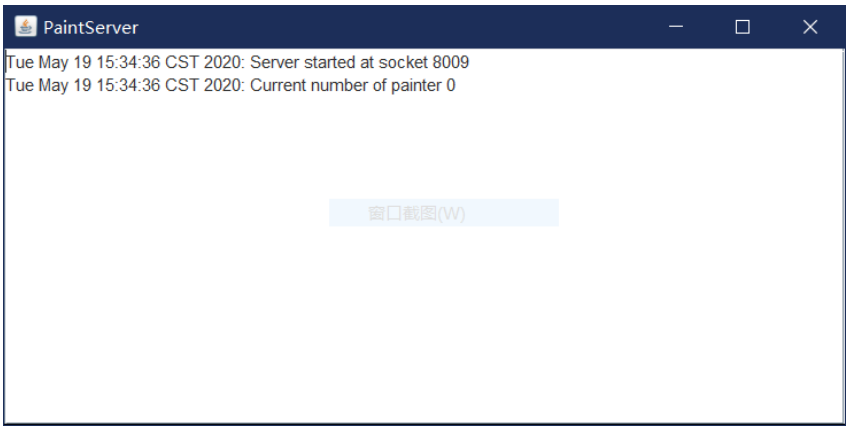
前面步骤和 reshape 操作类似，首先检测鼠标按下的点在哪个图形中，比如在图形 s 中，则将 s 的所有特征点进行偏移处理。偏移的信息来说鼠标释放和鼠标按下的坐标，及将两个坐标相减（横坐标减横坐标，纵坐标减纵坐标），得到的偏移量用来移位图形。

四、　实验结果展示及分析和（或）源程序调试过程

程序运行方法：

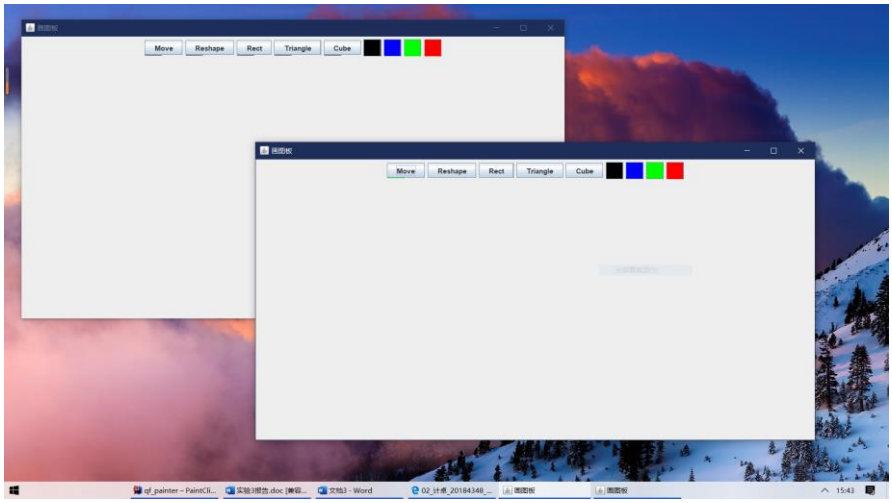
先运行 PainServe 类，再运行 PaintClient 类，其中 PaintClient 类多次运行即可产生多个用户。

4.1 启动后端服务器

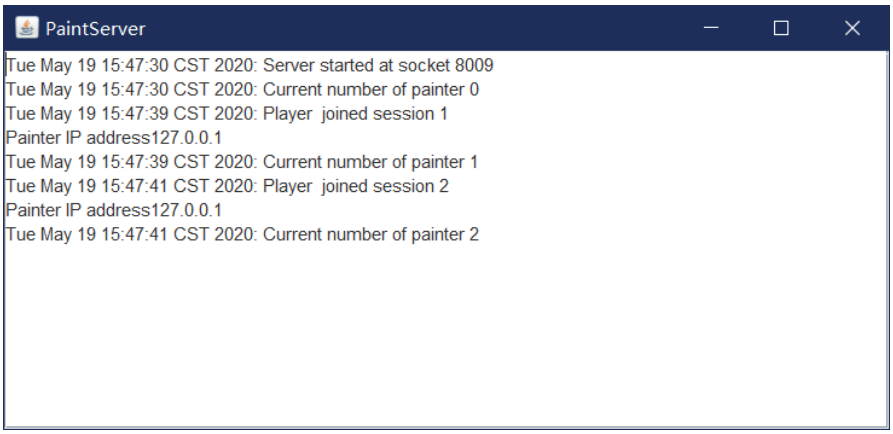


4.2 启动前端服务器。

4.2.1 总共启动了两个绘图面板，如下图

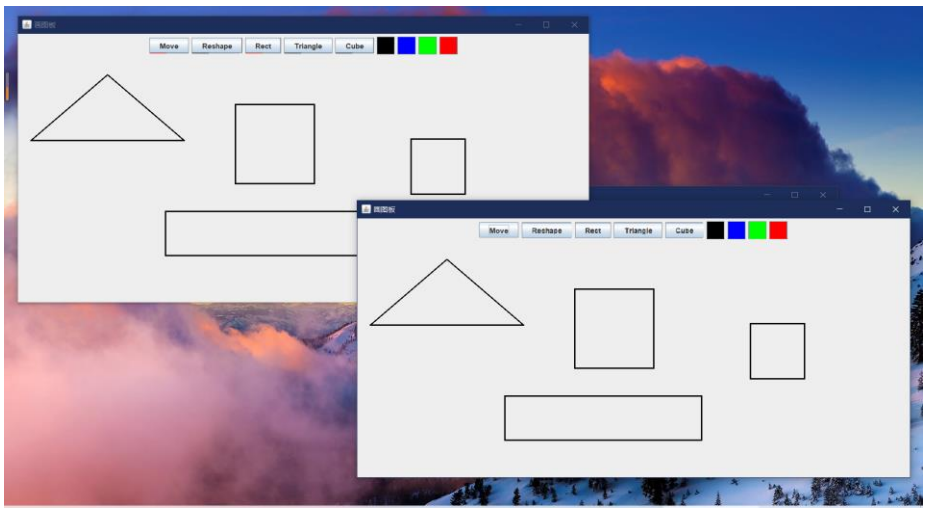


4.2.2 下图是启动两个用户面板后后端的信息，可以看见，其显示已经有两个绘图者加入



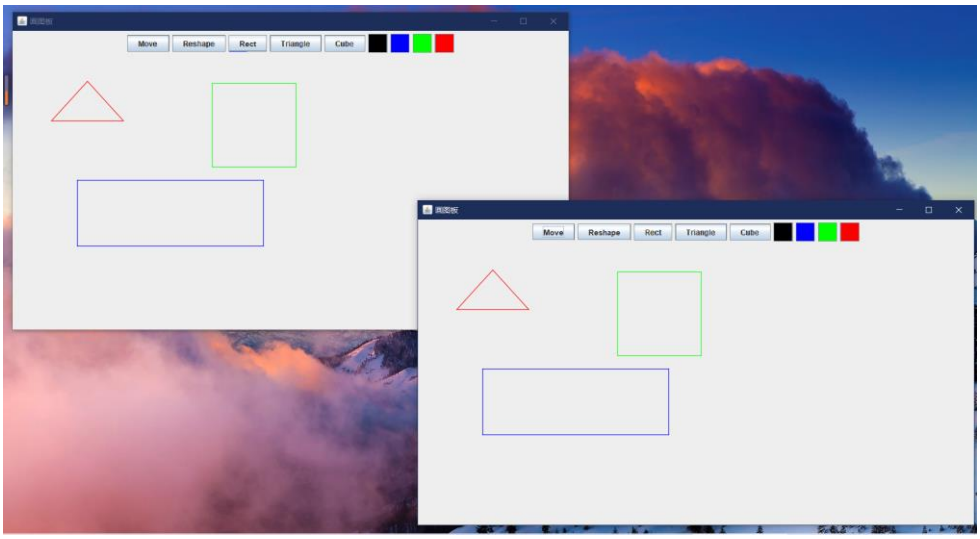
4.3 绘制图形

4.3.1 首先使用默认颜色（黑色）绘制了一个三角形，两个正方形以及一个矩形



可以发现，两个面板的图形是一模一样的

4.3.2 改变颜色，绘制图形

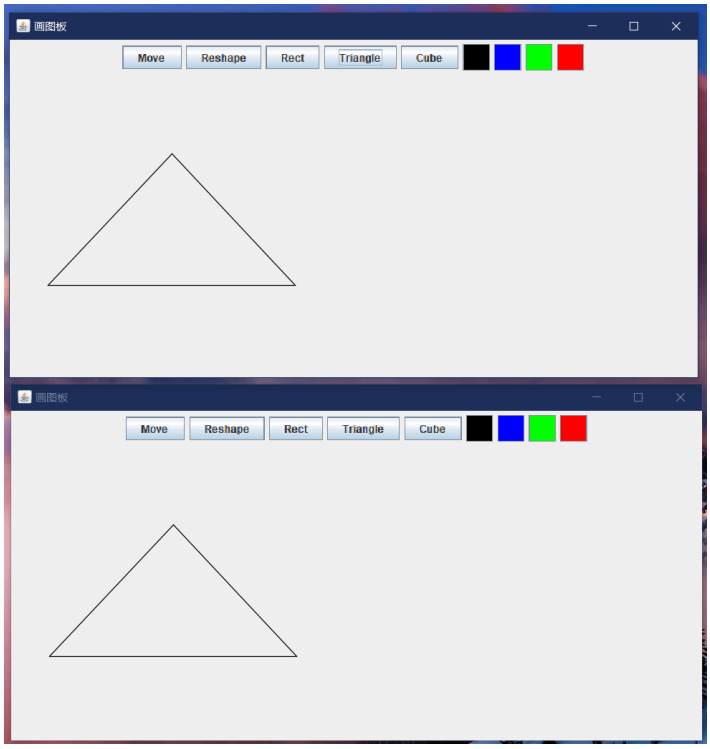


此时在面板一（左上角）绘制了一个红色的三角形，一个绿色的正方形，一个蓝紫色的矩形。

4.4 改变图形形状

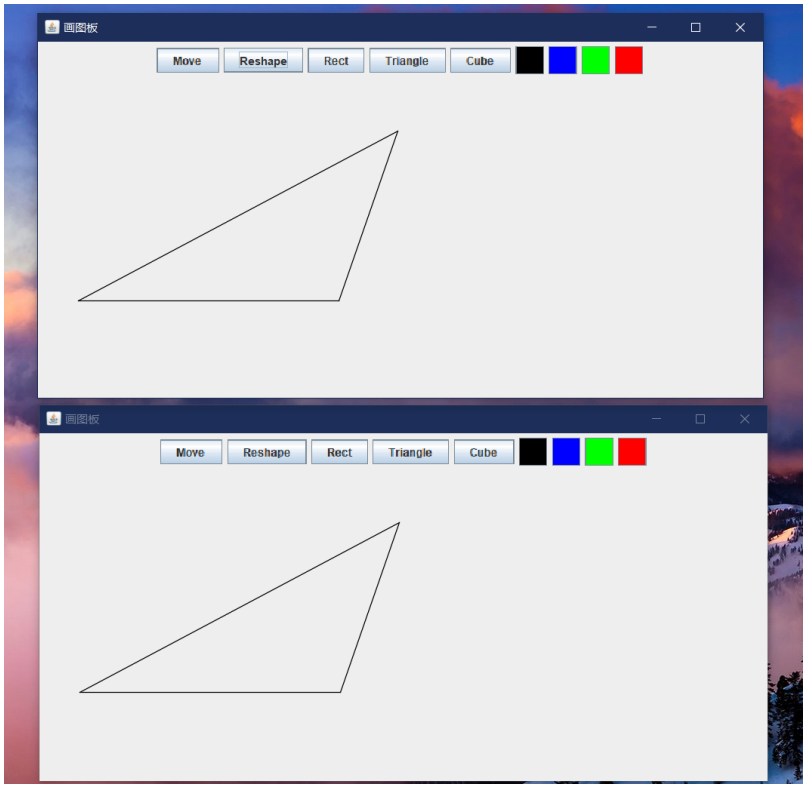
4.4.1 三角形：

4.4.1.1 改变前的图形的形状

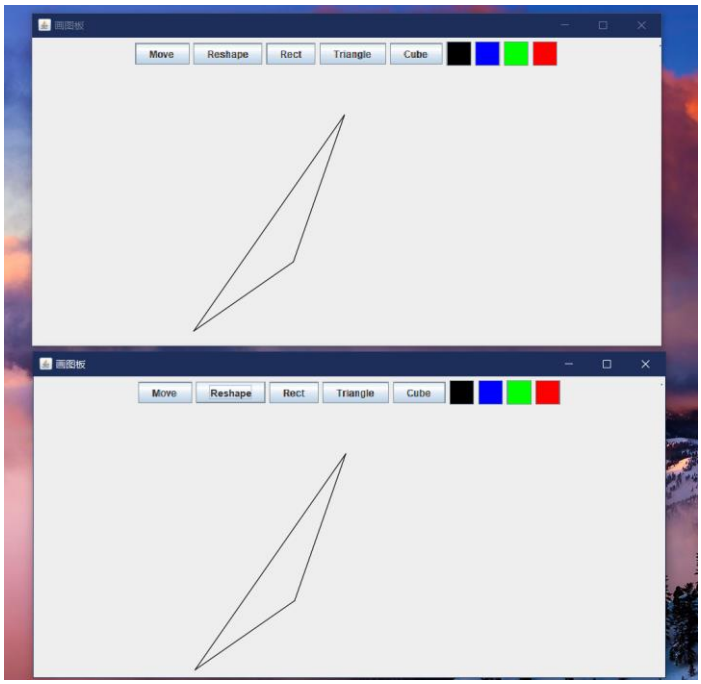


4.4.1.2 改变后的图形的形状

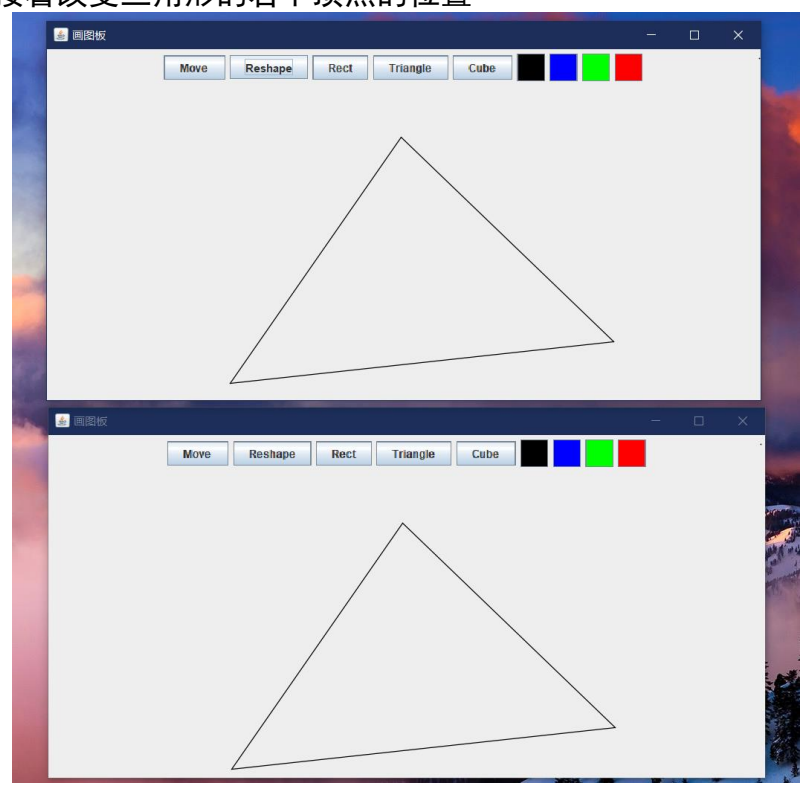
4.4.1.2.1 改变三角形的上顶点的位置



4.4.1.2.2 接着改变三角形的左下顶点的位置

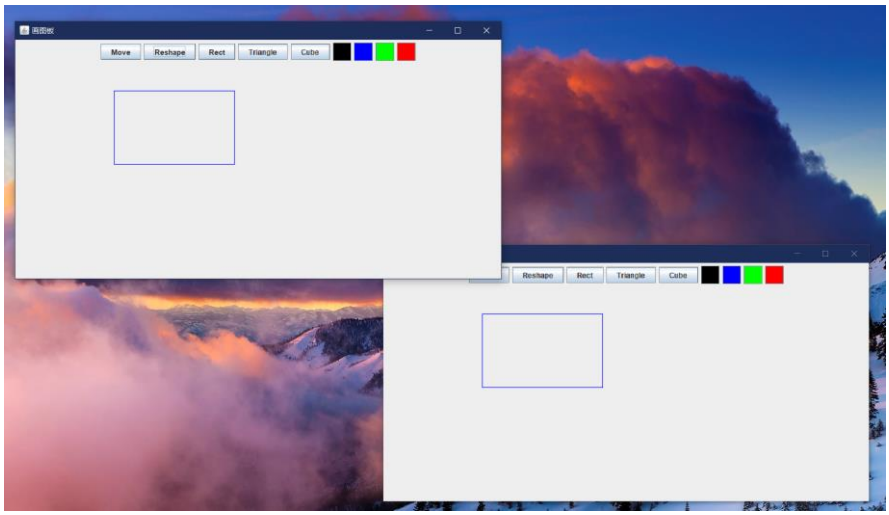


4.4.1.2.3 接着改变三角形的右下顶点的位置



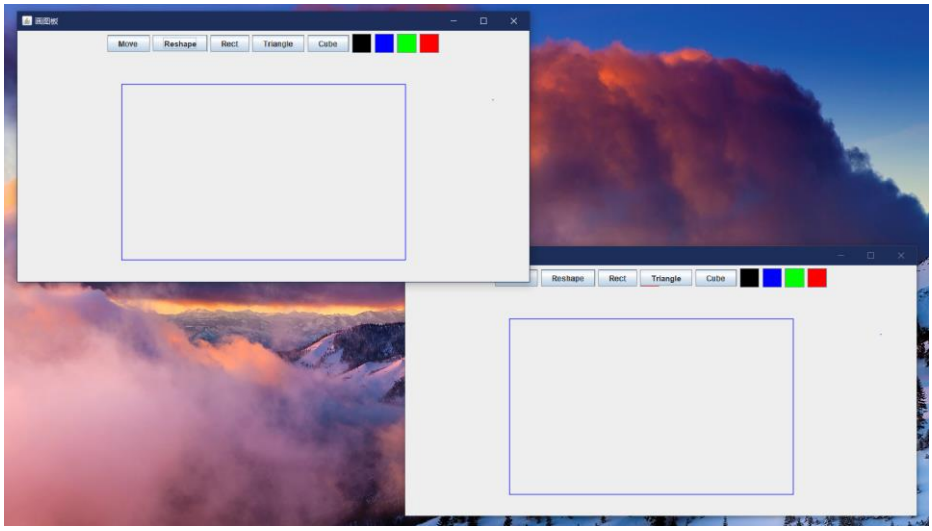
4.4.2 矩形：

4.2.2.1 改变前的图形的形状



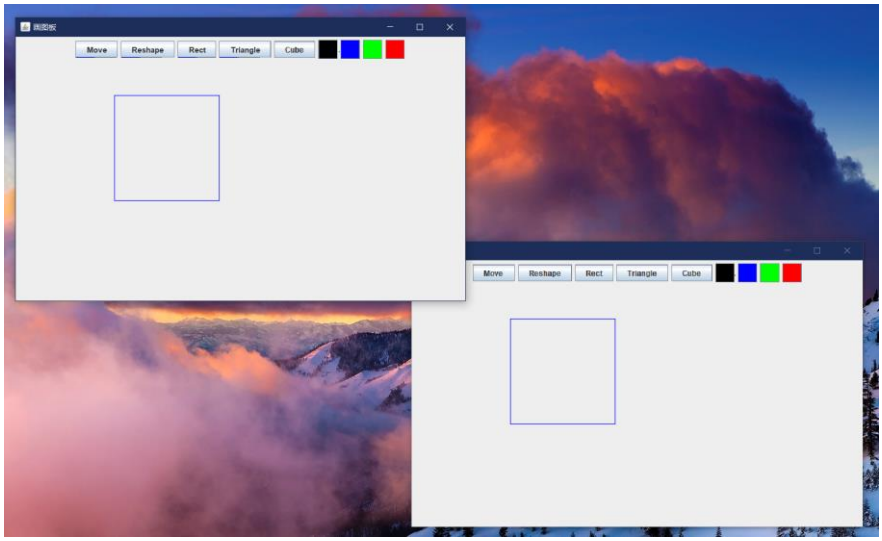
4.2.2.2 改变后的图形的形状

我们将这个矩形的右下角的顶点向右下角拉动，然后矩形被放大



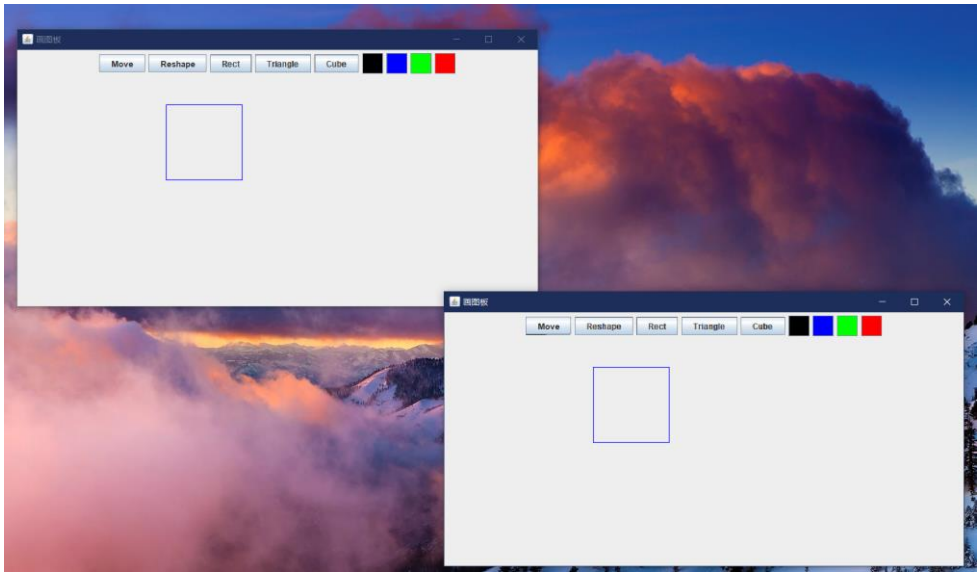
4.4.3 正方形

4.4.3.1 改变前的图形的形状



4.4.3.2 改变后的图形的形状

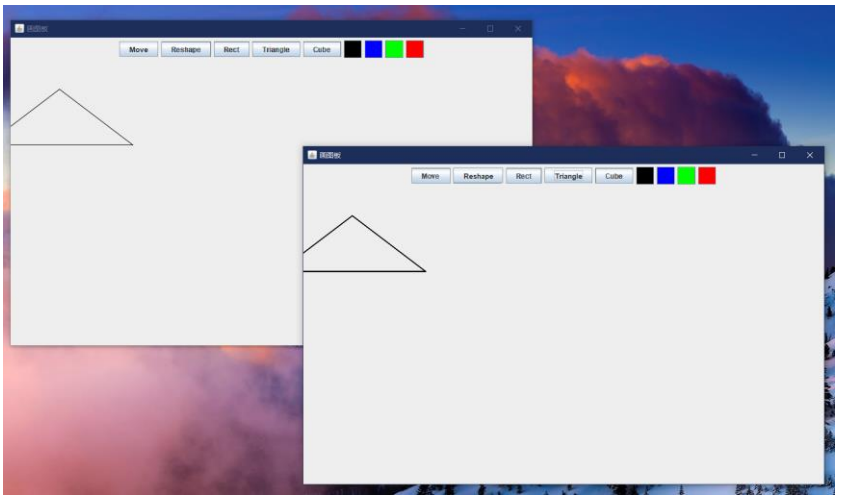
缩小了的正方体



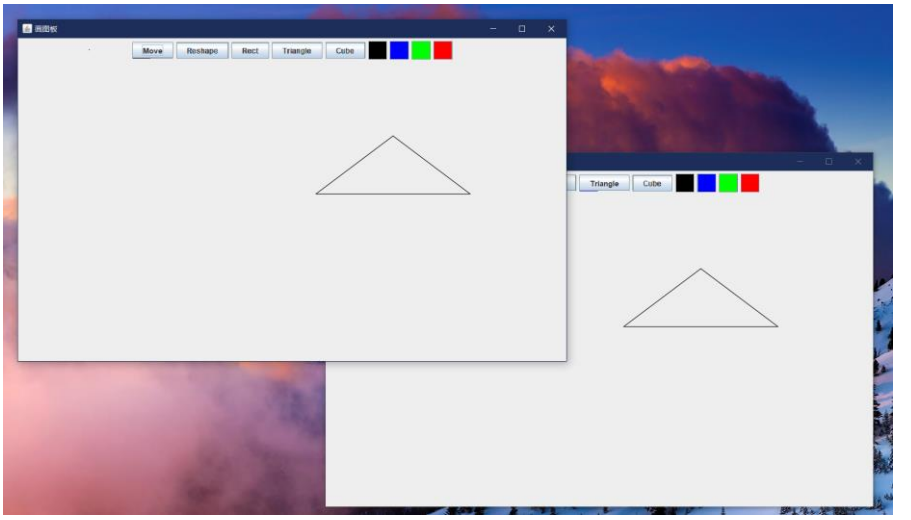
4.5 移动图形

4.5.1 移动三角形

4.5.1.1 移动三角形前

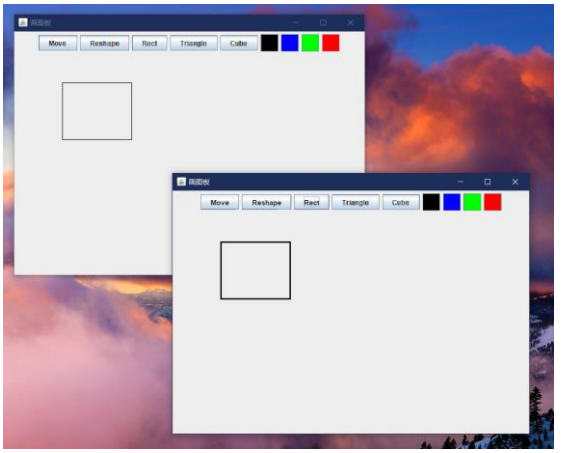


4.5.1.2 移动三角形后

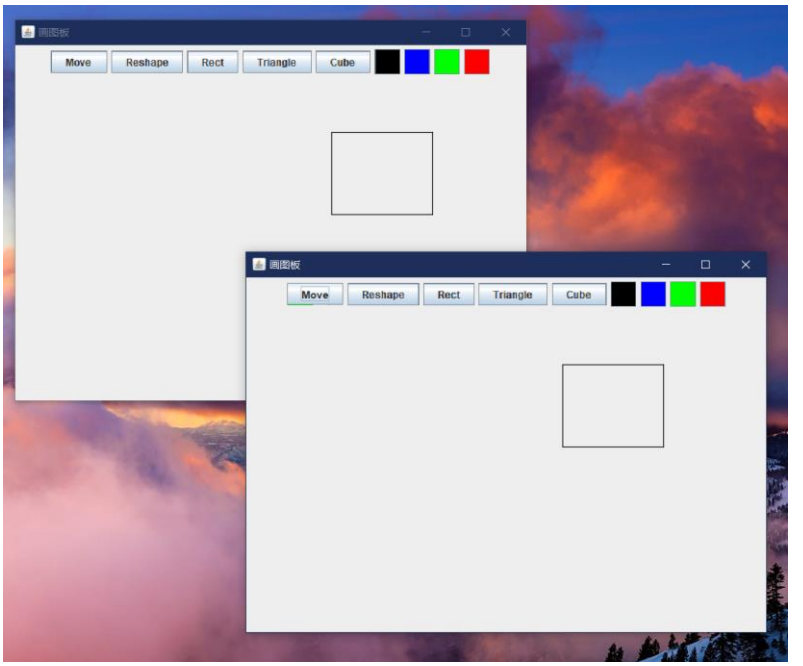


4.5.2 移动矩形

4.5.2.1 移动矩形前

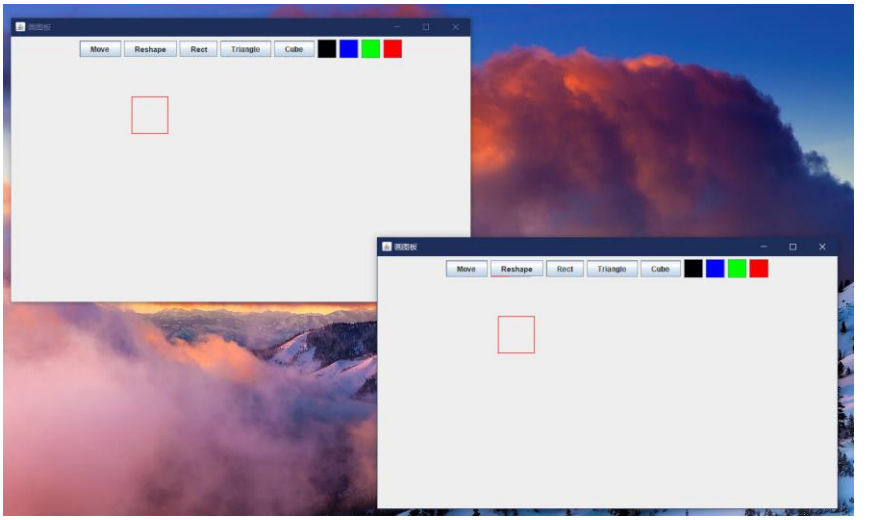


4.5.2.2 移动矩形后

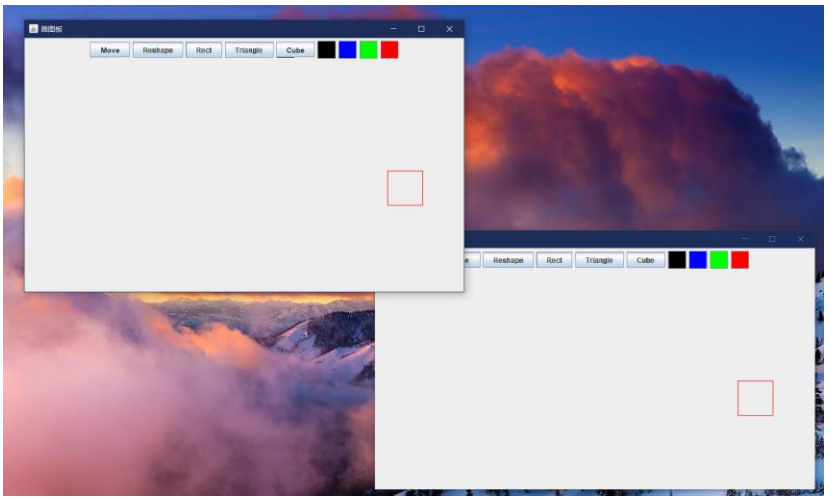


4.5.3 移动正方体

4.5.3.1 移动正方体前

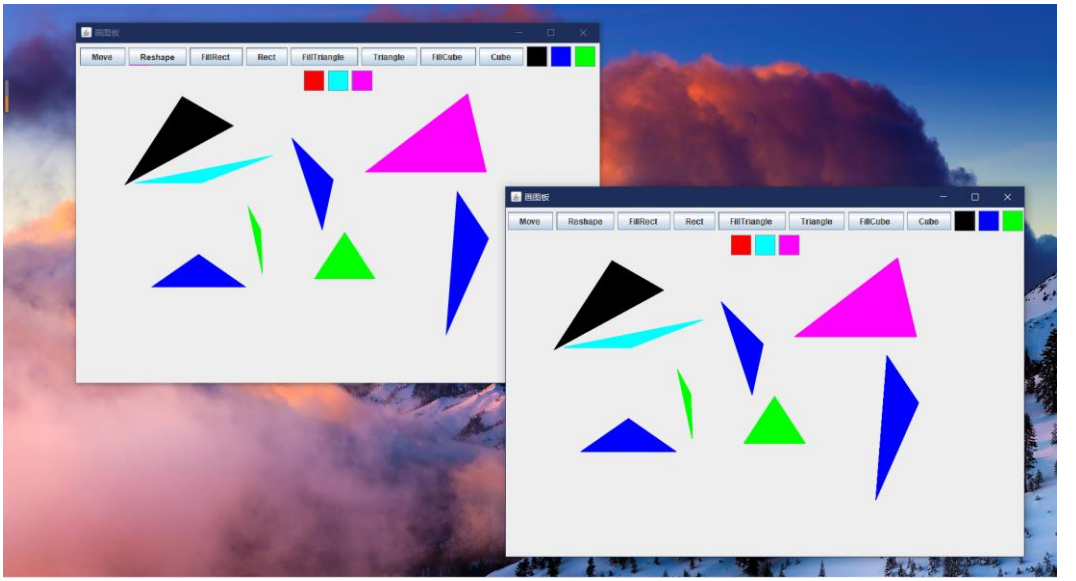


4.5.3.2 移动正方体后

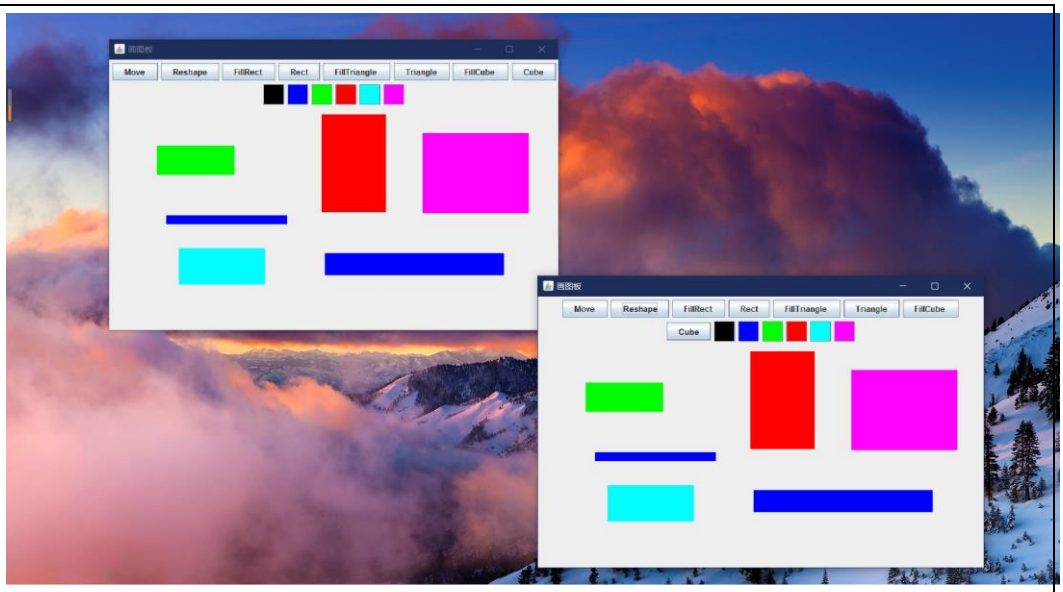


4.6 填充图形

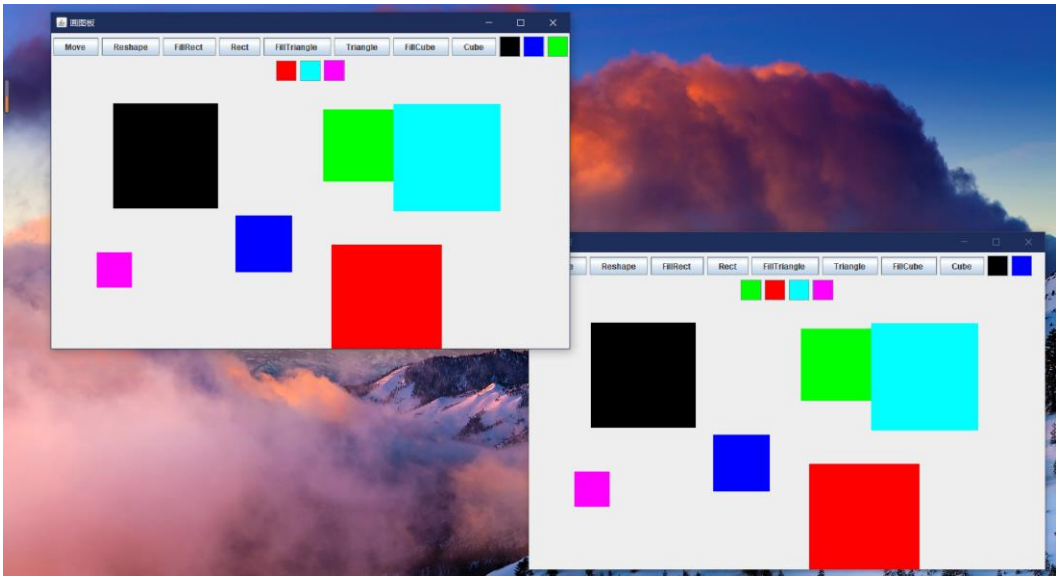
4.6.1 填充三角形



4.6.2 填充矩形



4.6.3 填充正方体



备注：

1、教师在布置需撰写实验报告的实验前，应先将报告书上的“实验题目”、“实验性质”、“实验目的”、“实验项目内容”等项目填写完成，然后再下发给学生。

2、教师在布置需撰写报告的实验项目时，应告知学生提交实验报告的最后期限。

3、学生应按照要求正确地撰写实验报告：

1) 在实验报告上正确地填写“实验时间”、“实验地点”等栏目。
2) 将实验所涉及的源程序文件内容（实验操作步骤或者算法）填写在**“实验过程或算法（源程序）”**栏目中。
3) 将实验所涉及源程序调试过程（输入数据和输出结果）或者实验的分析内容填写在**“实验结果及分析和（或）源程序调试过程”**栏目中。
4) 在实验报告页脚的“报告创建时间：”处插入完成实验报告时的日期和时间。
5) 学生将每个实验完成后，按实验要求的文件名通过网络提交（上载）到指定的服务器所规定的共享文件夹中。每个实验一个电子文档，如果实验中有多个电子文档（如源程序或图形等），则用 WinRAR 压缩成一个压缩包文档提交，压缩包文件名同实验报告文件名（见下条）。
6) 提交的实验报告电子文档命名为：“组号（2 位数字）年级（两位数字不要“级”字）专业（缩写：计算机科学与技术专业（计科）、网络工程专业（网络）、信息安全专业（信息）、物联网工程（物联网））项目组成员（学号（八位数字）姓名）实验序号（一位数字）. doc。如第 1 组完成第 1 个 Project，专业为“计算机科学与技术”专业，项目组成员有：张三（学号 20115676），李四（学号 20115676），王五（学号 20115676），完成的课程设计报告命名为：01_10 计科_20115676 张三_20115676 李四_20115676 王五 1.doc，以后几次实验的报告名称以此类推。

4、教师（或助教）在评价学生实验时，应根据其提交的其他实验相关资料（例如源程序文件等）对实验报告进行仔细评价。评价后应完成的项目有:

1) 在“成绩”栏中填写实验成绩。每个项目的实验成绩按照五级制（优、良、中、及格、不及格）方式评分，实验总成绩则通过计算每个项目得分的平均值获得（平均值计算时需将五级制转换为百分制优=95、良=85、中=75、及格=65、不及格=55）。

2) 在“教师评价”栏中用符号标注评价项目结果（用√表示正确，用×表示错误，用≈表示　半对半错）。

3) 在“教师评价”栏中“评价教师签名”填写评价教师（或助教）姓名。将评价后的实验报告转换为 PDF 格式文件归档。

4) 课程实验环节结束后，任课教师将自己教学班的实验报告文件夹进行清理。在提交文件夹中，文件总数为实验次数×教学班学生人数（如，教学班人数为 90 人，实验项目为 5，其文件数为：90×5=450）。任课教师一定要认真清理，总数相符，否则学生该实验项目不能得分。最后将学生提交的实验报告刻光盘连同实验成绩一起放入试卷袋存档。