

# 《Java 程序设计》实验报告

年级、专业、班级		姓名	
实验题目	基于标准输入/输出的绘图程序基础设计与实现		
实验时间	2020. 3. 17~2020. 4. 1	实验地点	DS1408
实验成绩		实验性质	<input type="checkbox"/> 验证性 <input type="checkbox"/> 设计性 <input checked="" type="checkbox"/> 综合性
<p>教师评价：</p> <p><input type="checkbox"/> 算法/实验过程正确； <input type="checkbox"/> 源程序/实验内容提交   <input type="checkbox"/> 程序结构/实验步骤合理；</p> <p><input type="checkbox"/> 实验结果正确；   <input type="checkbox"/> 语法、语义正确；   <input type="checkbox"/> 报告规范；</p> <p>其他：</p> <p>评价教师签名：</p>			
<p>一、实验目的</p> <p>运用面向对象程序设计思想，设计并实现基于标准输入/输出的绘图程序。</p>			
<p>二、实验项目内容</p> <p>1. 其中 Shape 类是图形类的抽象父类, 它包含一个抽象方法 draw(), 在他的派生类中都实现了 draw() 方法 (只需要将本实例的类名称和坐标属性输出到标准输出即可)、各自的属性和属性的修改方法；</p> <p>2. Graphic 是用来存储当前已有的对象、绘制已有的对象实例和改变某个实例对象形状（例如位置坐标）等功能，其内部有存储 shape 类极其子类实例的容器集合（例如 shape 数组），add() 方法负责把 shape 类极其子类实例添加内部容器集合中，draw() 用于调用内部容器集合中的所有实例的 draw() 方法，还可以添加其它方法（改变某个实例对象形状的方法）；</p> <p>3. OpenGLApp 是用来测试这些集合类的，方法 initGL() 是用来创建和初始化 Graphics 类实例，方法 display() 绘制所有保存在 Graphics 类实例中图形的，reshape() 是用来修改 Graphics 类实例中图形的区域属性值；Point 类用来存储 Shape 类集合中点属性的；</p> <p>4. 在类 OpenGLApp 定义 main 函数，在 main 函数中创建 OpenGLApp 的实例并调用相关的成员方法；把 Graphics 和 shape 类及其派生类放在包 graph 中，把类 OpenGLApp 放到 app 包中，并在 OpenGLApp import 包 graph 中的类</p>			

报告创建时间：

### 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）

#### （一）类的设计与算法设计

##### （1）Shape 类：

是一个抽象类，包含抽象方法 `draw()` 用于输出图形和 `getCN()` 用于返回图形类别，使得所有继承于 Shape 类的子类必须实现这两个方法。

##### （2）Point 类：

是一个自定义数据结构类，有私有成员变量 `xyz` 代表坐标的三个分量和 `name` 代表点的名称；定义了构造函数和 `get` 与 `set` 方法，还实现了 `connect` 方法，用于模拟输出一个点到另一个点的连线图形。

##### （3）Triangle 类：

成员变量：三个点 `abc`；

成员函数：`getCN()` 用于返回图形的类型，`get` 和 `set` 方法，其中 `set` 方法有多重参数表的重载，`reshape` 函数用于更新对象，`draw` 方法用于模拟输出三角形形状。

##### （4）Rectangle 类：

成员变量：两个点 `ab`、宽度 `width` 和高度 `height`；

成员函数：`getCN()` 用于返回图形的类型，`get` 和 `set` 方法，其中 `set` 方法有多重参数表的重载，`reshape` 函数用于更新对象，`draw` 方法用于模拟输出四边形形状。

##### （5）Cube 类：

成员变量：体中心 `v1`，宽度 `width`；

成员函数：`getCN()` 用于返回图形的类型，`get` 和 `set` 方法，其中 `set` 方法有多重参数表的重载，`reshape` 函数用于更新对象，`draw` 方法用于模拟输出正方体形状。

##### （6）Graphic 类：

成员变量：`LinkedList` 类型的 `shapes` 集合，用于存放 Shapes 的类对象；

成员函数：`draw()` 方法：调用 `shapes` 集合中存放的所有类对象的 `draw` 方法，实现模拟绘图所有的 Shape 类对象，`add(Shape s)` 方法，将 Shape 的类对象加入 `shapes` 集合中，`reshape` 方法，修改选定图形的属性。

##### （7）OpenGLApp 类：

相当于一个控制器类，成员变量有一个 Graphic 类的类对象，成员函数有 `initGL` 函数，用于初始化 Graphic 类对象，`display` 函数用于模拟绘图所有的图形，`reshape` 函数用于改变特定图像的属性。

#### （二）实现流程设计

OpenGLApp 类中，Main 函数是程序的入口；

Main 函数中可以调用 OpenGLApp 类的 `initGL` 方法，从而将 OpenGLApp

的 Graphic 类型的成员变量初始化。初始化之后，即可以对其进行操作。可以在 Main 函数中实例化一些各种 Shape 类的类对象，然后使用 add 方法将其加入 Graphic 类对象的 Shape 数组中以供后续操作。可以在 Main 函数中调用 display 方法，从而将所有图像全部画出，也可以在 Main 函数中调用 reshape 方法，在通过命令行输入新参数，对对象进行更新修改。

### 个人使用的测试流程：

(1) 首先用不带参数的构造函数实例化一个 Triangle 对象 t1，然后调用 t1 的 draw 方法，观察**构造函数**与 **draw 方法**的正确性；

(2) 然后调用 t1 的 set 方法，对其的成员变量进行更新，接着再次调用 t1 的 draw 方法，观察 **set 方法**的正确性；

(3) 对类 Rectangle 和 Cube 重复如上两个步骤，分别验证他们的方法正确性；

(4) 用不带参数的构造函数实例化 Graphic 类的对象 shapes，然后调用 shapes 的 add 方法将 t1、r1 和 c1 添加进 shapes，然后调用 shapes 的 draw 方法，观察 **add 方法**和 **draw 方法**的正确性

(5) 使用不带参数的构造函数实例化 OpenGLApp 类的对象 controller，调用 controller 的 initGL 方法，然后调用 controller.graphic 的 add 方法将 t1、r1 和 c1 加入，再调用 controller 的 display 方法，验证 **initGL 方法**和 **display 方法**的正确性；

(6) 调用 controller 的 reshape 方法，根据提示进行输入，然后再次调用 display 方法，观察 **reshape 方法**的正确性。

使用如上的测试流程即可观察所有类的所有方法的正确性，测试的结果以及调试过程将在第三部分中展示。

### (三) 关键代码说明

源文件中对每一条关键代码都进行了注释说明，为了保证编码正确的问题，采用英文进行注解，为了使得语法清晰易懂且学术化，采用了最简单鲜明的语法，便于理解阅读。

### (四) 源程序代码

Source codes 均打包于 eclipse 工程文件中

## 四、实验结果展示及分析和（或）源程序调试过程

(1) 首先用不带参数的构造函数实例化一个 Triangle 对象 t1，然后调用 t1 的 draw 方法，观察**构造函数**与 **draw 方法**的正确性；

(2) 然后调用 t1 的 set 方法，对其的成员变量进行更新，接着再次调用 t1 的 draw 方法，观察 **set 方法**的正确性；

```

I'm drawing the triangle.
a(0.0,0.0,0.0)----->b(0.0,0.0,0.0);b(0.0,0.0,0.0)----->c(0.0,0.0,0.0);
After update
I'm drawing the triangle.
a(1.0,1.0,1.0)----->b(2.0,2.0,2.0);b(2.0,2.0,2.0)----->c(3.0,3.0,3.0);

```

```

I'm drawing the rectangle.
Point:a(0.0,0.0,0.0)
width0.0
Height0.0
Point:b(0.0,0.0,0.0)

```

After update

```

I'm drawing the rectangle.
Point:a(1.0,1.0,0.0)
width1.0
Height1.0
Point:b(2.0,2.0,0.0)

```

```

I'm drawing the Cube.
Point:v1(0.0,0.0,0.0)
width0.0

```

After update

```

I'm drawing the Cube.
Point:v1(1.0,1.0,1.0)
width2.0

```

说明构造函数与 draw 方法和 set 方法正确。

(3) 对类 Rectangle 和 Cube 重复如上两个步骤，分别验证他们的方法正确性；

截图同上

(4) 用不带参数的构造函数实例化 Graphic 类的对象 shapes，然后调用 shapes 的 add 方法将 t1、r1 和 c1 添加进 shapes，然后调用 shapes 的 draw 方法，观察 add 方法和 draw 方法的正确性

```

I'm drawing the triangle.
a(1.0,1.0,1.0)----->b(2.0,2.0,2.0);b(2.0,2.0,2.0)----->c(3.0,3.0,3.0);c(3.0,3.0,3.0)----->a(1.0,1.0,1.0)
I'm drawing the rectangle.
Point:a(1.0,1.0,0.0)
width1.0
Height1.0
Point:b(2.0,2.0,0.0)
I'm drawing the Cube.
Point:v1(1.0,1.0,1.0)
width0.0

```

说明 add 方法和 draw 方法正确

(5)使用不带参数的构造函数实例化 OpenGLApp 类的对象 controller，调用 controller 的 initGL 方法，然后调用 controller.graphic 的 add 方法将 t1、r1 和 c1 加入，再调用 controller 的 display 方法，验证 initGL 方法和 display 方法的正确性；

```
I'm drawing the triangle.
a(1.0,1.0,1.0)----->b(2.0,2.0,2.0);b(2.0,2.0,2.0)----->c(3.0,3.0,3.0);c(3.0,3.0,3.0)----->
I'm drawing the rectangle.
Point:a(1.0,1.0,0.0)
width1.0
Height1.0
Point:b(2.0,2.0,0.0)
I'm drawing the Cube.
Point:v1(1.0,1.0,1.0)
width0.0
```

说明 initGL 方法和 display 方法正确

(6) 调用 controller 的 reshape 方法，根据提示进行输入，然后再次调用 display 方法，观察 reshape 方法的正确性。

```
Index of chosen shape
0
You are reshaping a triangle.
New coordinates of a(x y z):
5 5 5
New coordinates of b(x y z):
6 6 6
New coordinates of c(x y z):
7 7 7
I'm drawing the triangle.
a(5.0,5.0,5.0)----->b(6.0,6.0,6.0);b(6.0,6.0,6.0)----->c(7.0,7.0,7.0);c(7.0,7.0,7.0)----->a(5
I'm drawing the rectangle.
Point:a(1.0,1.0,0.0)
width1.0
Height1.0
Point:b(2.0,2.0,0.0)
I'm drawing the Cube.
Point:v1(1.0,1.0,1.0)
width0.0
```

发现已被改变，所以说明 reshape 方法正确

通过如上测试可知，本次实现的程序各方法均逻辑正确，功能正常，设计正确。

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： $90 \times 5 = 450$ ）。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。