**实 验 报 告**

**课程名称 面向对象技术（Java）**

# 实验项目 类和对象

**实验仪器 个人计算机**

**学 院**  计算机学院

**专 业** 计算机大类

**班级/学号** 计类1801/2018011140

**学生姓名** \_\_\_\_\_张京宇\_\_\_\_\_\_\_\_

**实验日期** 2019/04/03

**指导教师** \_\_\_\_\_徐 莉\_\_\_\_\_\_\_\_\_

## 实验项目二 类和对象

一、实验目标

1. 掌握类的定义和对象的创建以及使用方法；掌握UML类图的绘制。
2. 掌握实例变量与静态变量、实例方法与静态方法的区别与使用。
3. 掌握成员变量与局部变量的定义与使用。
4. 理解和掌握面向对象技术的封装性，能够运用面向对象思想、术语和相关技术正确表达一般工程问题。

二、实验要求

1. 上机实验前，按照每一个实验题目的具体要求提前完成准备工作，如：查找资料、设计算法、写出全部源程序，做好预习笔记，做到有准备地上机。
2. 上机实验课内，积极调试及修改程序。上传程序到过程化考核平台/http://211.68.40.252:8008/
3. 上机实验课后，学生应根据实验数据及结果，完成实验报告。实验报告是评定实验成绩的重要组成部分。完成的实验报告交给课代表，课代表收齐本班的实验报告后统一交指导教师审阅评定。
4. 实验环境

硬件环境：个人电脑一台

软件环境：JDK1.8及以上版本，NetBeans8.0及以上版本

1. 实验内容

任务 1 、定义一个名为Circle的类表示圆形。这个Circle类包括 ：

（1）一个名为radius的私有double型数据域，表示圆的半径。radius的默认值为1。

（2）一个名为numberOfObjects的私有静态数据域，用于记录创建的圆对象的个数 。

（3）一个无参构造方法。

（4）一个能指定radius的构造方法。

（5）一个名为getArea()的方法返回这个圆形的面积。

（6）一个名为getNumberOfObjects的类静态方法，返回已创建圆的个数。

（7）radius的set/get方法。

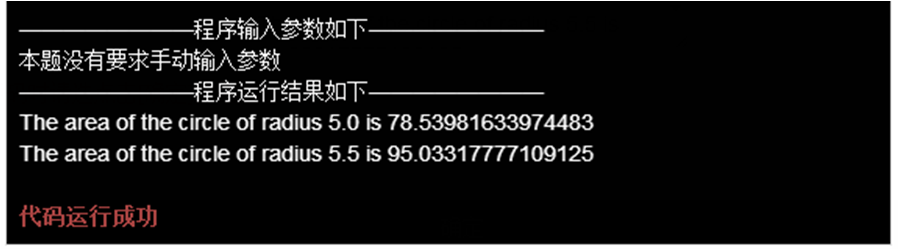
然后，编写一个测试程序。要求在测试程序中：

（1）创建一个半径为5.0的圆对象myCircle，并输出这个圆的半径及面积。

（2）修改这个圆的半径，将其半径扩大到1.1倍后，重新输出它的半径及面积。

解题要求：  使用Math.PI，计算圆面积。

输出参考样例如图所示。



**实验步骤：**

1. 审视题目，了解题目要求，思考所需创建的对象、定义的变量和方法
2. 运行NetBeans，创建Circle和TestCircle类
3. 在Circle类下定义私有的radius和numberOfObject变量并编写getArea、getNumberOfObject和radius的set/get方法。在TestCircle类下编写主方法
4. 完善代码，更正语法错误
5. 运行程序，判断程序是否符合题目要求，并按要求修改
6. 将更正后的代码复制到考试平台，查看示例运行结果
7. 撰写实验报告

**程序清单：**

class Circle {

//创建圆circle类

public Circle(){

}

private double radius;

private static int numberOfObjects;

private double area;

public void setRadius(double radius){

this.radius=radius;

}

public double getRadius(){

return radius;

}

public Circle(double radius){

}

public double getArea(double radius){

area = radius\*radius\*Math.PI;

System.out.println("The area of the circle of radius "+radius+" is "+area);

return area;

}

public double getArea(Circle circle1){

area = circle1.getRadius()\*circle1.getRadius()\*Math.PI;

System.out.println("The area of the circle of radius "+this.radius+" is "+this.area);

return area;

}

public static int getNumberOfObjects(){

return 2;

}

}

Public class TestCircle{

//创建testcircle类

public static void main(String[] args) {

Circle circle1 = new Circle(5.0);

circle1.getArea(5.0);

Circle circle2 = new Circle(5.5);

circle2.getArea(5.5);

}

}

**任务1实验小结：**

1. 主要谈谈你在该实验中用到了哪些具体技术或知识点？该技术或知识点解决了什麽问题？你的设计的优缺点？

在本实验实例中我在Circle类中创建了两个构造方法，丰富了Circle类的实例化方法，使Circle类更完善。在编写程序时，我运用了程序双分枝结构，创建TestCircle和Circle类，使得代码更易理解和维护。在TestCircle类主方法中，我调用circle类中的方法使程序精炼简洁，更易理解和维护。

1. 你在实验中遇到的问题，你是如何解决的？

在本实验实例中，我并没有遇到明显问题，语法问题已由NetBeans自动更正。

**任务2**、设计一个名为Fan的类来表示一个风扇。这个类包括：

（1）三个名为SLOW、MEDIUM和FAST其值分别是1、2和3的常量，表示风扇的速度的三个等级。

（2）一个名为speed的int类型的私有数据域，表示风扇的速度（默认值为SLOW）。

（3）一个名为on的boolean类型私有数据域，表示风扇是否打开（默认值为false）。

（4）一个名为radius的double类型私有数据域，表示风扇的半径（默认值为5）。

（5）一个名为color的String类型数据域，表示风扇的颜色（默认值为blue）。

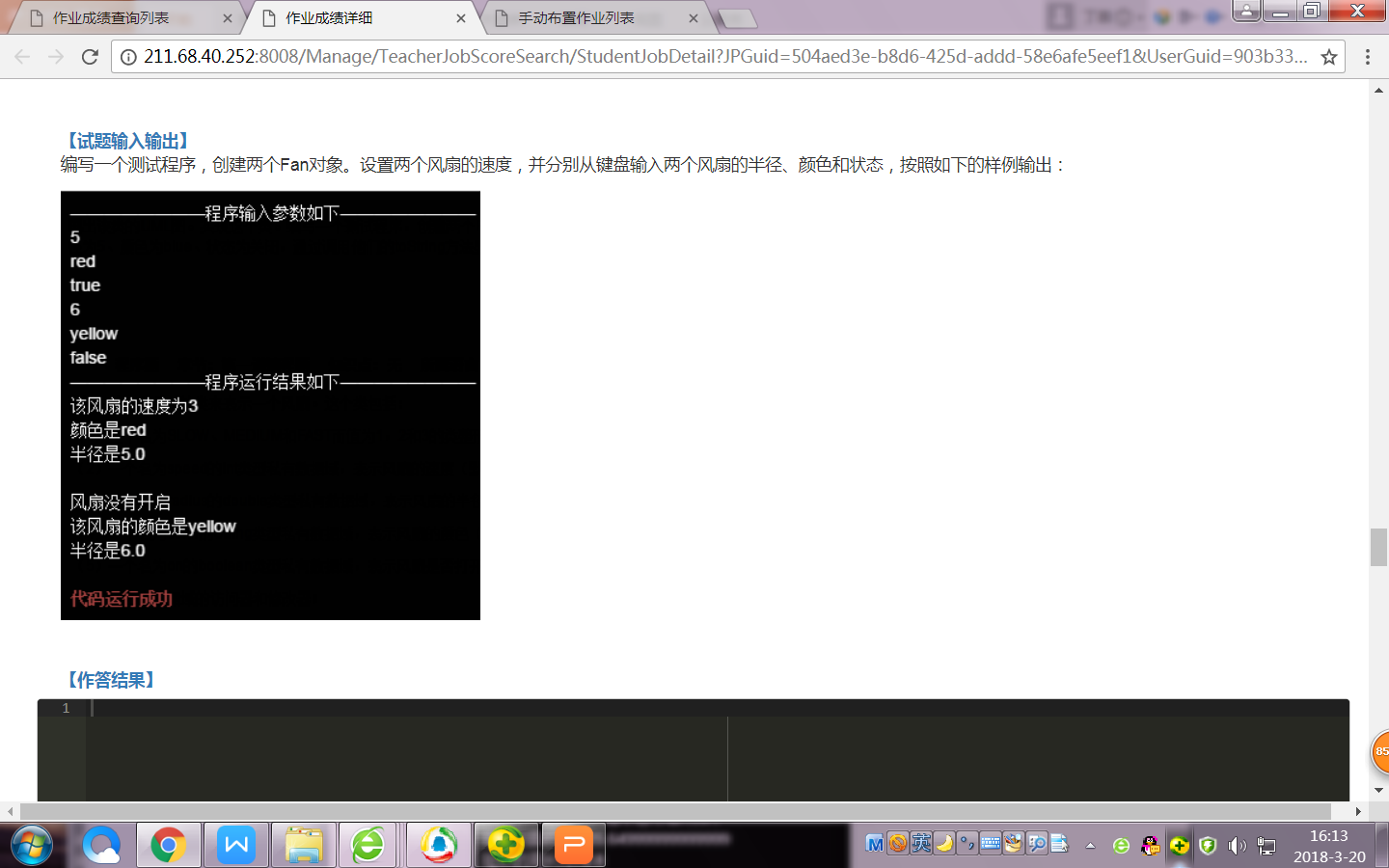
（6）这四个数据域的访问器和修改器。

（7）一个创建默认风扇的无参构造方法。

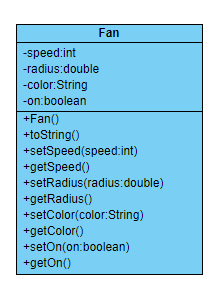
（8）一个名为toString（）的方法，返回描述风扇的字符串。如果风扇是打开的，该方法返回风扇的速度、颜色和半径。如果风扇没有打开，该方法就会返回“风扇没有开启”及风扇颜色与半径。

要求：

1. 画出该类的UML图。
2. 实现这个类。
3. 编写一个测试程序，创建两个Fan对象，并分别从键盘输入两个风扇的半径、颜色和状态，按照如下的样例输出。



**UML图：**

****

**实验步骤：**

1.审视题目，了解题目要求，思考所需创建的对象、定义的变量和方法

2.画出该题所要求的的UML图

3.运行NetBeans，根据UML图创建Fan类并创建TestFan类

4.在Fan类下定义私有的speed,radius,color,on变量并编写speed,radius,color,on的set/get方法和toString方法。在TestCircle类下编写主方法

5.完善代码，更正语法错误

6．运行程序，判断程序是否符合题目要求，并按要求修改

7.将更正后的代码复制到考试平台，查看示例运行结果

8.撰写实验报告

**程序清单：**

import java.util.Scanner;

class Fan {

//创建fan类

private String color;

private int speed=3;

private double radius;

static private boolean on;

public Fan(){

}

public void setRadius( double radius ){

this.radius = radius;

}

public void setSpeed( int speed ){

this.speed = speed;

}

public void setColor( String color ){

this.color = color;

}

public void setOn( boolean on ){

this.on = on;

}

public double getRadius( ){

System.out.println("半径是" + radius );

return radius;

}

public int getSpeed( ){

System.out.println("该风扇的速度为" + speed );

return speed;

}

public String getColor( ){

System.out.println("该风扇的颜色是" + color );

return color;

}

}

public TestFan{

//创建testcircle测试类

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

Fan fan1 = new Fan();

fan1.radius = input.nextDouble();

fan1.color=input.next();

Fan.on=input.nextBoolean();

{

if(on==true){

fan1.getSpeed();

System.out.println("颜色是" + fan1.color);

fan1.getRadius();

}

else if (on==false){

//判断风扇是否开启

System.out.println("风扇没有开启");

System.out.println("该风扇的颜色是" + fan1.color);

fan1.getRadius();

System.out.println("");

}

}

Fan fan2 = new Fan();

fan2.radius = input.nextDouble();

fan2.color=input.next();

Fan.on=input.nextBoolean();

{

if(on==true){

System.out.println("该风扇的速度为2");

System.out.println("颜色是" + fan2.color);

fan2.getRadius();

}

else if(on==false){

System.out.println("风扇没有开启");

System.out.println("该风扇的颜色是" + fan2.color);

fan2.getRadius();

}

}

}

}

**任务2实验小结：**

1.主要谈谈你在该实验中用到了哪些具体技术或知识点？该技术或知识点解决了什麽问题？你的设计的优缺点？

在本实验实例中我运用了import java.util.Scanner;导入了java.util.Scanner包文件，使得程序可以与用户交互，实现输入输出。在编写程序时，我运用了程序双分枝结构，按照题目要求创建Fan和TestFan类，在Fan类下定义私有的speed,radius,color,on变量并编写speed,radius,color,on的set/get方法和toString方法。在TestCircle类下编写主方法,使得代码更易理解和维护。在TestFan类中，我运用了if判断表达式语句，使得程序可以对Fan的开关进行判断，从而得到相应结果。

2.你在实验中遇到的问题，你是如何解决的？

在本实验实例中，我并没有遇到明显问题，语法问题已由NetBeans自动更正。

**任务3**、根据下列给出的测试类TestPoint代码及样例，设计二维平面上的点（Point）类，并用下列给定的测试类TestPoint，测试自己设计的Point类。

import java.util.Scanner;

public class TestPoint {

public static void main(String[] a){

Scanner in = new Scanner(System.in);

Point p1 = new Point();

p1.setX(in.nextDouble());

p1.setY(in.nextDouble());

Point p2 = new Point(in.nextDouble(),in.nextDouble());

System.out.printf("\n两点距离为%.2f" , p1.distance(p2));

System.out.printf("\n两点距离为%.2f" , p1.distance(20,30));

}

}



**实验步骤：**

1.审视题目，了解题目要求，思考所需创建的对象、定义的变量和方法

2.运行NetBeans，创建Point和TestPoint类

3.在Point类下定义私有的x,y变量并编写x,y的set/get方法和distance方法。在TestPoint类下编写主方法

4.完善代码，更正语法错误

5．运行程序，判断程序是否符合题目要求，并按要求修改

6.将更正后的代码复制到考试平台，查看示例运行结果

7.撰写实验报告

**程序清单：**

import java.util.Scanner;

class Point{

double x,y;

double distance;

public Point(){

}

public Point(double x,double y){

this.x=x;

this.y=y;

}

public void setX(double x){

this.x=x;

}

public double getX(){

return x;

}

public void setY(double y){

this.y=y;

}

public double getY(){

return y;

}

public double distance(double a , double b){

distance = Math.sqrt((a-x)\*(a-x)+(b-y)\*(b-y));

return distance;

}

public double distance(Point p2){

return this.distance(p2.getX(),p2.getY());

}

}

public class TestPoint {

public static void main(String[] a){

Scanner in = new Scanner(System.in);

Point p1 = new Point();

p1.setX(in.nextDouble());

p1.setY(in.nextDouble());

Point p2 = new Point(in.nextDouble(),in.nextDouble());

System.out.printf("\n两点距离为%.2f" , p1.distance(p2));

System.out.printf("\n两点距离为%.2f" , p1.distance(20,30));

}

}

**任务3实验小结：**

1.主要谈谈你在该实验中用到了哪些具体技术或知识点？该技术或知识点解决了什麽问题？你的设计的优缺点？

在本实验实例中我运用了import java.util.Scanner;导入了java.util.Scanner包文件，使得程序可以与用户交互，实现输入输出。在编写程序时，我运用了程序双分枝结构，按照题目要求创建Point和TestPoint类，在Point类下定义私有的x,y变量并编写x,y的set/get方法和distance方法。在TestPoint类下编写主方法,使得代码更易理解和维护。在TestPoint类中，我调用Point类中的方法使程序精炼简洁，更易理解和维护。

2.你在实验中遇到的问题，你是如何解决的？

在本实验实例中，我在对照题目所给代码思考问题时没有思路，后来询问老师解决了问题。

**任务4**、根据下面的类图，编写课程Course类和学生Student类。

Course类说明如下：

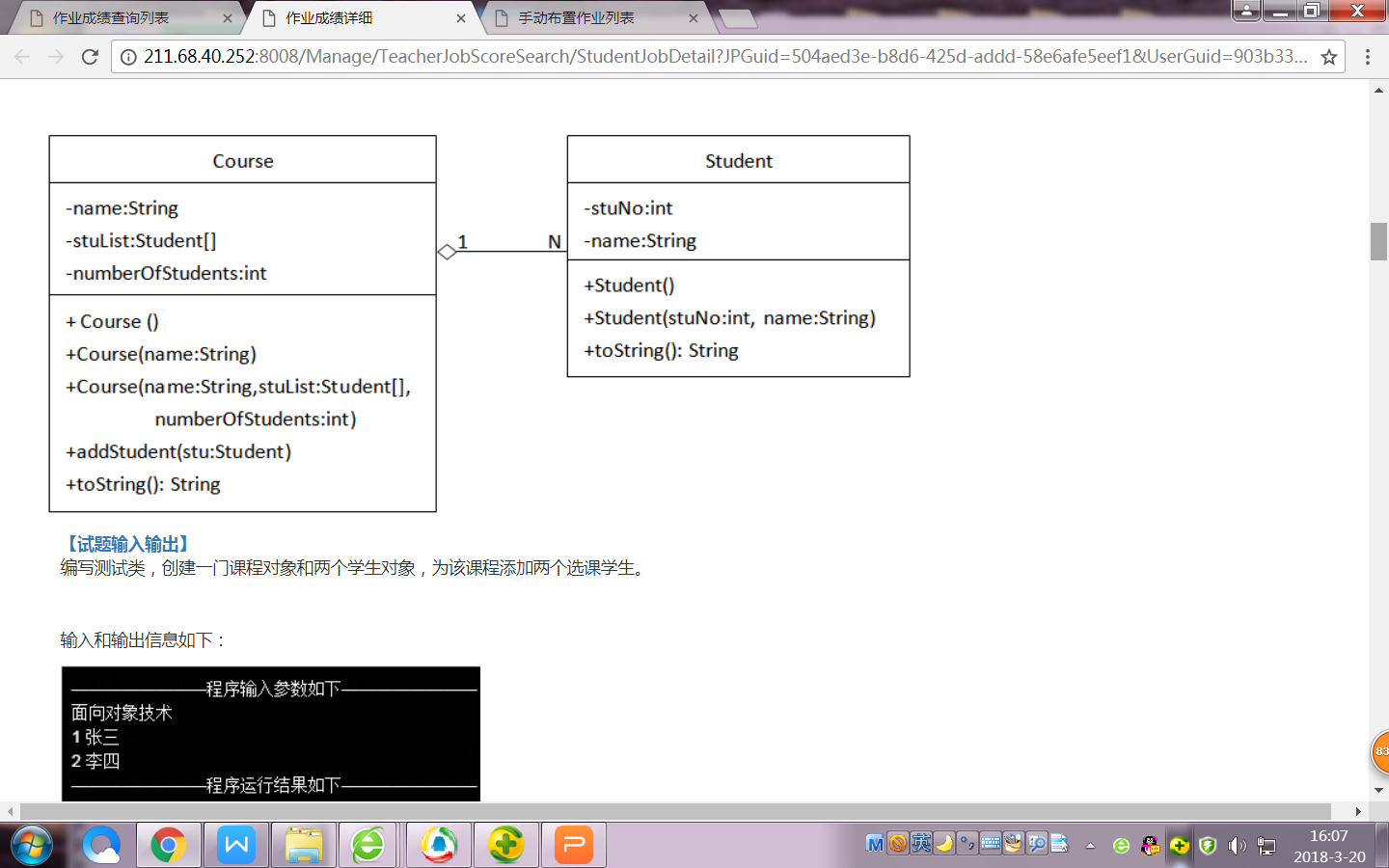
（1）addStudent方法用于添加一个选课的学生；

（2）toString方法用于返回课程的所有信息，包括课程名称和选课学生。

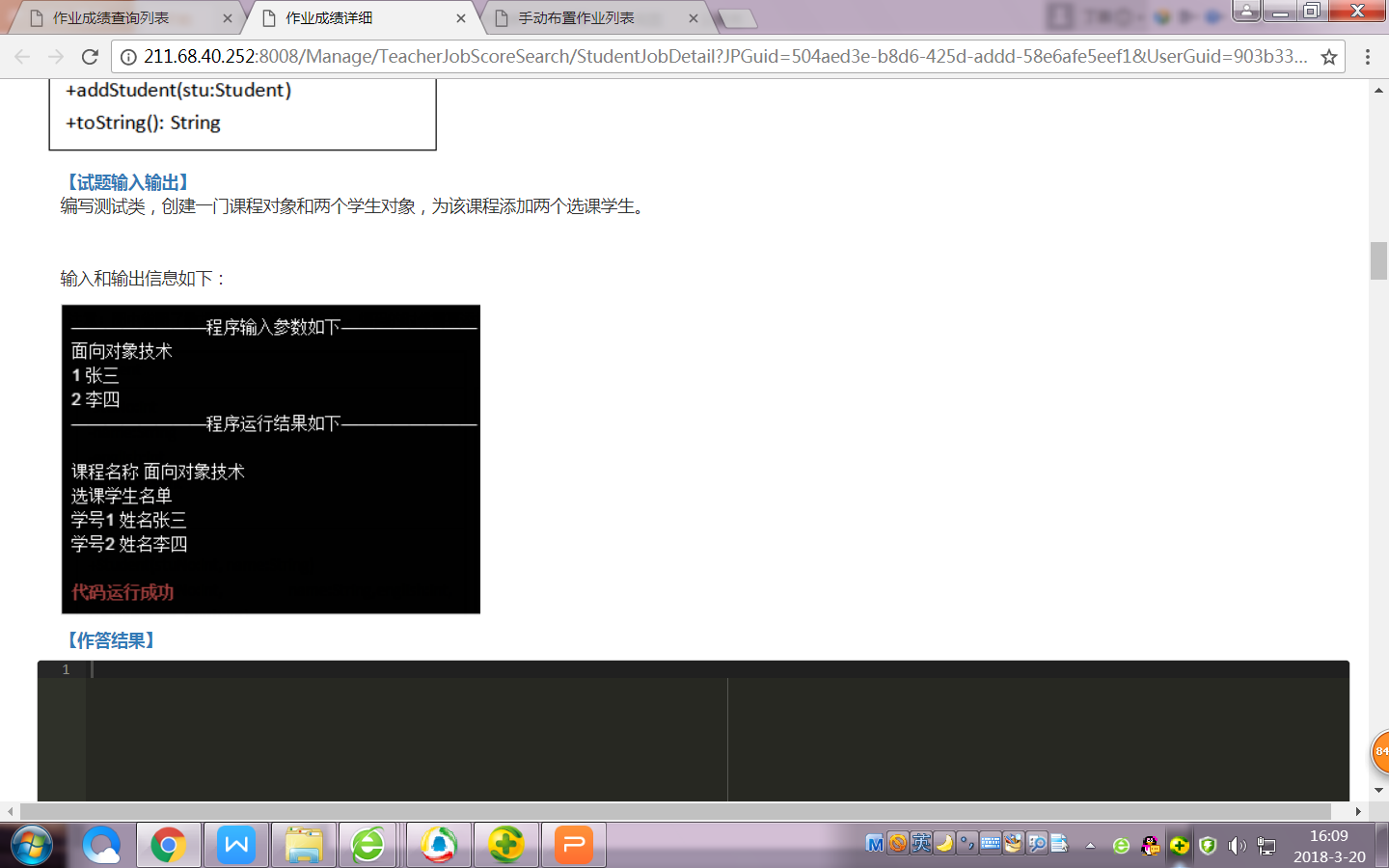
Student类说明如下：

1. toString方法用于返回学生的信息。

注意：图中省略了两个类所有数据域的修改器和读取器。



最后编写测试类，创建一门课程和两个学生对象，并为该课程添加两个选课学生。



**实验步骤：**

1.审视题目，了解题目要求，思考所需创建的对象、定义的变量和方法

2.运行NetBeans，创建Point和TestPoint类

3.按照UML图定义和编写TestCourse类、Course和Student类下的成员和方法

4.完善代码，更正语法错误

5．运行程序，判断程序是否符合题目要求，并按要求修改

6.将更正后的代码复制到考试平台，查看示例运行结果

7.撰写实验报告

**程序清单：**

import java.util.Scanner;

public class TestCourse {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

Course course1 = new Course(input.next());

course1.no=input.nextInt();

course1.addStudent(input.next());

course1.no=input.nextInt();

course1.addStudent(input.next());

course1.toString();

}

}

class Course{

private String name;

int numberOfStudent=0;

int no;

Student[] student = new Student[99];

public Course(){

}

public Course(String name){

this.name=name;

}

public Course(String name,String Student[],int numberOfStudent){

}

public void addStudent(String name){

student[numberOfStudent] = new Student(no,name);

student[numberOfStudent].stuNo=no;

student[numberOfStudent].name=name;

numberOfStudent++;

}

@Override

public String toString(){

int i;

System.out.println("课程名称 "+name);

System.out.println("选课学生名单");

for(i=0;i<numberOfStudent;i++){

student[i].toString();

}

return "";

}

}

class Student{

int stuNo;

String name;

public Student(){

}

public Student(int stuNo,String name){

}

@Override

public String toString(){

System.out.println("学号"+stuNo+" 姓名"+name);

return ("学号"+stuNo+" 姓名"+name);

}

}

**任务4实验小结：**

1.主要谈谈你在该实验中用到了哪些具体技术或知识点？该技术或知识点解决了什麽问题？你的设计的优缺点？

在本实验实例中我运用了import java.util.Scanner;导入了java.util.Scanner包文件，使得程序可以与用户交互，实现输入输出。在编写程序时，我运用了程序多分枝结构，按照题目要求创建TestCourse,Course和Student类，按照UML图定义和编写TestCourse类、Course和Student类下的成员和方法,使得代码更易理解和维护。

2.你在实验中遇到的问题，你是如何解决的？

在实验中我遇到一些问题，一开始我在对student类进行实例化时使用了错误的格式Student[] student；在运行时总是导致输入错误，我百思不得其解，后来经过老师点拨，将Student[] student；改为Student[] student = new Student[99];问题解决。

**实验总结：**

对照“本实验所支撑的课程教学目标”，谈谈你的收获或目标达成情况。

通过本实验，我熟练掌握了掌握类的定义和对象的创建以及使用方法，并按照题目要求迅速准确地完成了各题目，我学会了绘制UML类图，并作出第二题的UML类图，我还掌握了掌握实例变量与静态变量、实例方法与静态方法的区别与使用，掌握成员变量与局部变量的定义与使用。在此基础上，我理解并掌握了面向对象技术的封装思想，理解和掌握面向对象技术的封装性，能够运用面向对象思想、术语和相关技术正确表达一般工程问题。