一、实验目的

1) 理解类、对象的基本概念；

2) 掌握类的设计、对象的创建、类的封装、构造方法的定义和使用；

3) 掌握类的静态成员与实例成员。

二、实验环境

Windows 7,Eclipse+jdk8 ,PC

三、实验内容与步骤

1．编写一个 Java 应用程序，模拟家庭买一台电视，即家庭将电视作为自己的一个成员，即通过调用一个方法将某个电视的引用传递给自己的电视成员。具体要求如下：

- 有三个源文件：TV.java、Family.java 和 MainClass.java，其中 TV.java 中的 TV 类负责创建“电视”对象，amily.java 中的 amily 类负责创建“家庭”对象，MainClass.java 是主类。
- 在主类的 Main()方法中首先使用 TV 类创建一个对象 haierTV，然后使用 Family 类再创建一个对象 zhangSanFamily，并将先前 TV 类的实例 haierTV 的引用传递给 zhangSanFamily 对象的成员变量 homeTV。

Family 类组合 TV 类的实例的 UML 图如图 1 所示。

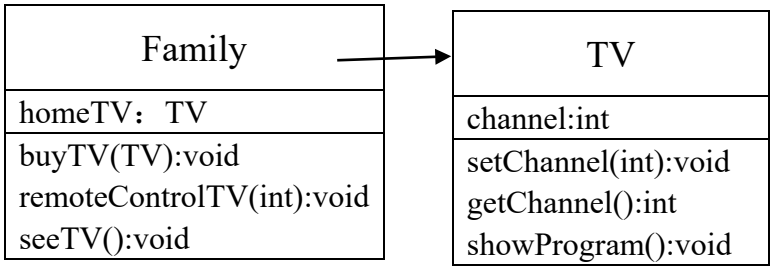

图 1 Family 类组合 TV 类的实例的 UML 图

2 编写复数类（Complex）和一个测试类（ComplexDemo），实现复数的四则运算（加、减、乘、除）和输入、输出等操作。

每个运算提供两种调用方法，以加法为例：

c3=c1.add(c2);

c3=Complex.add(c1,c2);

3．编写一个 Java 应用程序。设计一个学生类 Students，包括属性有：序号，学号，姓名，性别，专业，三门课程成绩（数学，计算机，英语）；包括方法有：求成绩总分，成绩平均分，除序号外各个属性的赋值方法，各个属性值的获取方法等等。说明：每创建一个 Students 对象，则序号值增 1；第一个学生对象的序号值为 1。

编写一个主类 StudentsDemo.java，利用 Students 类输入 5 名同学的三门课成绩，按以下要求输出相关信息：

（1）输入一个学生的学号或者姓名，输出该学生的所有信息。

（2）数学课程最高分同学序号、学号、姓名和课程分数

（3）总分最高/最低同学序号、学号、姓名和课程分数

（4）有没有这样的学生存在：总分大于所有学生的平均成绩，但是至少有一门课不及格（低于 60 分）。若有，按行输出这些学生的全部信息；若没有，不输出任何信息。

四、实验代码与实验结果及分析

1、

实验代码：

文件 1：MainClass.java

```
public class MainClass {
	public static void main(String args[]) {
		TV haierTV=new TV();
		Family zhangSanFamily=new Family();
		zhangSanFamily.buyTV(haierTV);
		zhangSanFamily.seeTV();
	}

}
```

文件 2：Family.java

```
import java.util.Scanner;
public class Family {
	TV homeTV;
	Scanner reader=new Scanner(System.in);
	void buyTV(TV tv) {
		homeTV=tv;
	}
	void remoteControlTV(int s) {
		homeTV.setChannel(s);

	}
	void seeTV() {
		System.out.println("Opening...");
		homeTV.showProgram();
		System.out.print("Now,we watch TV,choose channel:");
		remoteControlTV(reader.nextInt());
		homeTV.showProgram();
	}
}
```

文件 3：TV.java

```
public class TV {
	int channel=1;
	void setChannel(int c) {
		channel=c;
	}
	int getChannel() {
		return channel;
	}
	void showProgram() {
		System.out.println("channel: "+getChannel());
```

```
            System.out.print("TVprogram: ");
            switch(getChannel()) {
                case 1 :System.out.println("Synthetical channel");break;
                case 2 :System.out.println("economical channel");break;
                case 3 :System.out.println("synthesis skill channel");break;
                case 4 :System.out.println("Chinese international channel");break;
                case 5 :System.out.println("sports channel");break;
                case 6 :System.out.println("movie channel");break;
                case 7 :System.out.println("soap opera channel");break;
                case 8 :System.out.println("English international channel");break;
                case 9 :System.out.println("drama channel");break;
                case 10 :System.out.println("society and law channel");break;
            }
        }

}
```

实验结果及分析：

实验结果：

```
Opening...
channel: 1
TVprogram: Synthetical channel
Now,we watch TV,choose channel:6↙
channel: 6
TVprogram: movie channel
```

分析：实验达到预期效果。

2、

实验代码：

文件：ComplexDemo.java

```
import java.util.*;
public class ComplexDemo {
    public static void main(String args[]) {
        System.out.println("Test...");
        Complex c1=new Complex(0,0);
        System.out.println("set c1...");
        c1.set();
        System.out.print("c1=");
        c1.output();
        Complex c2=new Complex(0,0);
        System.out.println("set c2...");
        c2.set();
        System.out.print("c2=");
        c2.output();
        //new
```

```
            Complex c3=c1.add(c2);
            System.out.print("way1:c1+c2=");
            c3.output();
            c3=Complex.add(c1, c2);
            System.out.print("way2:c1+c2=");
            c3.output();
            //add
            c3=c1.cut(c2);
            System.out.print("way1:c1-c2=");
            c3.output();
            c3=Complex.cut(c1, c2);
            System.out.print("way2:c1-c2=");
            c3.output();
            //cut
            c3=c1.multi(c2);
            System.out.print("way1:c1*c2=");
            c3.output();
            c3=Complex.multi(c1, c2);
            System.out.print("way2:c1*c2=");
            c3.output();
            //multi
            c3=c1.devision(c2);
            System.out.print("way1:c1/c2=");
            c3.output();
            c3=Complex.devision(c1, c2);
            System.out.print("way2:c1/c2=");
            c3.output();
        }
}
class Complex {
        double realPart=0.0;
        double imaginaryPart=0.0;
        Complex(double temp1,double temp2){
            realPart=temp1;
            imaginaryPart=temp2;
        }
        Complex add(Complex c) {
            double temp1=realPart+c.realPart;
            double temp2=imaginaryPart+c.imaginaryPart;
            return new Complex(temp1,temp2);
        }
        static Complex add(Complex c1,Complex c2) {
            double temp1=c1.realPart+c2.realPart;
            double temp2=c1.imaginaryPart+c2.imaginaryPart;
```

```
        return new Complex(temp1,temp2);

    }
    //add
    Complex cut(Complex c) {
        double temp1=realPart-c.realPart;
        double temp2=imaginaryPart-c.imaginaryPart;
        return new Complex(temp1,temp2);
    }
    static Complex cut(Complex c1,Complex c2) {
        double temp1=c1.realPart-c2.realPart;
        double temp2=c1.imaginaryPart-c2.imaginaryPart;
        return new Complex(temp1,temp2);
    }
    //cut
    Complex multi(Complex c) {
        double temp1=realPart*c.realPart-imaginaryPart*c.imaginaryPart;
        double temp2=imaginaryPart*c.realPart+realPart*c.imaginaryPart;
        return new Complex(temp1,temp2);
    }
    static Complex multi(Complex c1,Complex c2) {
        double temp1=c1.realPart*c2.realPart-c1.imaginaryPart*c2.imaginaryPart;
        double temp2=c1.imaginaryPart*c2.realPart+c1.realPart*c2.imaginaryPart;
        return new Complex(temp1,temp2);
    }
    //multi
    Complex devision(Complex c) {
        double temp3=c.realPart*c.realPart+c.imaginaryPart*c.imaginaryPart;
        double temp1=(realPart*c.realPart+imaginaryPart*c.imaginaryPart)/temp3;
        double temp2=(imaginaryPart*c.realPart-realPart*c.imaginaryPart)/temp3;
        return new Complex(temp1,temp2);
    }
    static Complex devision(Complex c1,Complex c2) {
        double temp3=c2.realPart*c2.realPart+c2.imaginaryPart*c2.imaginaryPart;
        double temp1=(c1.realPart*c2.realPart+c1.imaginaryPart*c2.imaginaryPart)/temp3;
        double temp2=(c1.imaginaryPart*c2.realPart-c1.realPart*c2.imaginaryPart)/temp3;
        return new Complex(temp1,temp2);
    }
    //devision
    void output() {
        if(imaginaryPart==0.0) System.out.println(realPart);
        else if(imaginaryPart<0.0) System.out.println(realPart+""+imaginaryPart+"i");
        else System.out.println(realPart+"+"+imaginaryPart+"i");
```

```
    }
    void set() {
        Scanner reader=new Scanner(System.in);
        System.out.print("realPart:");
        realPart=reader.nextDouble();
        System.out.print("imaginaryPart:");
        imaginaryPart=reader.nextDouble();
    }
}
```

实验结果及分析：

实验结果：

```
Test...
set c1...
realPart:5↙
imaginaryPart:7↙
c1=5.0+7.0i
set c2...
realPart:2↙
imaginaryPart:9↙
c2=2.0+9.0i
way1:c1+c2=7.0+16.0i
way2:c1+c2=7.0+16.0i
way1:c1-c2=3.0-2.0i
way2:c1-c2=3.0-2.0i
way1:c1*c2=-53.0+59.0i
way2:c1*c2=-53.0+59.0i
way1:c1/c2=0.8588235294117647-0.36470588235294116i
way2:c1/c2=0.8588235294117647-0.36470588235294116i
```

分析：两种调用方法（way1，way2）都能得出正确答案，实验达到预期效果。

3、

实验代码：

文件：StudentsDemo.java

```
import java.util.Scanner;
import java.lang.Object;
public class StudentsDemo {
    public static void main(String args[]) {
        int i;
        Students.count=1;
        Scanner reader = new Scanner(System.in);
        System.out.print("How many students do you want?->");
        int N=reader.nextInt();
        Students stu[]=new Students[N];
        for(i=0;i<N;i++) {
```

```
        System.out.println("Student"+(i+1));
        stu[i]=new Students();
        stu[i].newStudent();
}//new student
System.out.print("Searching by name:");
String name_temp=reader.nextLine();//get /n
name_temp=reader.nextLine();
int count=0;
for(i=0;i<N;i++) {
        if(stu[i].name.equals(name_temp)) {
                stu[i].outPutStudent();
                count++;
        }
}
if(count==0) {
        System.out.println("NOT FOUND!");
}//Searching by name
System.out.print("Searching by student number:");
long studentNumber_temp=reader.nextLong();
count=0;
for(i=0;i<N;i++) {
        if(stu[i].studentNumber==studentNumber_temp) {
                stu[i].outPutStudent();
                count++;
        }
}
if(count==0) {
        System.out.println("NOT FOUND!");
}//Searching by student number
System.out.println("The best students in math£º");
double mathMax=stu[0].math;
for(i=1;i<N;i++) {
        if(stu[i].math>mathMax) {
                mathMax=stu[i].math;
        }
}
for(i=0;i<N;i++) {
        if(stu[i].math==mathMax) {
                stu[i].outPutStudent();
        }
}//math max
System.out.println("The best students in total£º");
double totalMax=stu[0].getTotalScore();
for(i=1;i<N;i++) {
```

```
            if(stu[i].getTotalScore()>totalMax) {
                totalMax=stu[i].getTotalScore();
            }
        }
        for(i=0;i<N;i++) {
            if(stu[i].getTotalScore()==totalMax) {
                stu[i].outPutStudent();
            }
        }//total Max
        System.out.println("The worst students in total£º");
        double totalMin=stu[0].getTotalScore();
        for(i=1;i<N;i++) {
            if(stu[i].getTotalScore()<totalMin) {
                totalMin=stu[i].getTotalScore();
            }
        }
        for(i=0;i<N;i++) {
            if(stu[i].getTotalScore()==totalMin) {
                stu[i].outPutStudent();
            }
        }//total Min
        System.out.println("The stuent whose total score is higher than the average scores of all
students, but at least one class fails:");
        double Score=0;
        for(i=0;i<N;i++) {
            Score+=stu[i].getTotalScore();
        }
        Score/=N;
        count=0;
        for(i=0;i<N;i++) {
            if(stu[i].getTotalScore()>Score) {
                if(stu[i].getMath()<60.0) {
                    stu[i].outPutStudent();
                    count++;
                }
                else if(stu[i].getEnglish()<60.0) {
                    stu[i].outPutStudent();
                    count++;
                }
                else if(stu[i].getComputer()<60.0) {
                    stu[i].outPutStudent();
                    count++;
                }
            }
```

```
            }
            if(count==0) {
                System.out.println("NOT FOUND!");
            }//Biased undergraduates
        }

}
class Students {
        Scanner reader = new Scanner(System.in);
        static int count;
        int serialNumber;
        long studentNumber;
        String name;
        String sex;
        String major;
        double math;
        double computer;
        double english;
        Students(){
            serialNumber=count++;
        }
        void setName() {
            name=reader.nextLine();
        }
        void setStudentNumber() {
            studentNumber=reader.nextLong();
        }
        void setSex(){
            sex=reader.nextLine();
        }
        void setMajor() {
            major=reader.nextLine();
        }
        void setMath() {
            math=reader.nextDouble();
        }
        void setComputer() {
            computer=reader.nextDouble();
        }
        void setEnglish(){
            english=reader.nextDouble();
        }
        int getSerialNumber() {
            return serialNumber;
```

```
}
String getName() {
	return name;
}
long getStudentNumber() {
	return studentNumber;
}
String getSex() {
	return sex;
}
String getMajor() {
	return major;
}
double getMath() {
	return math;
}
double getComputer() {
	return computer;
}
double getEnglish() {
	return english;
}
double getTotalScore() {
	return math+computer+english;
}
double getAverageScore() {
	return (math+computer+english)/3.0;
}
void newStudent() {
	System.out.print("name:");
	setName();
	System.out.print("sex:(M/F):");
	setSex();
	System.out.print("major:");
	setMajor();
	System.out.print("student number:");
	setStudentNumber();
	System.out.print("math scores:");
	setMath();
	System.out.print("computer scores:");
	setComputer();
	System.out.print("english scores:");
	setEnglish();
}
```

```
    void outPutStudent() {
        System.out.println("serial number:"+getSerialNumber());
        System.out.println("student number:"+getStudentNumber());
        System.out.println("name:"+getName());
        System.out.println("sex:"+getSex());
        System.out.println("major:"+getMajor());
        System.out.println("math scores:"+getMath());
        System.out.println("computer scores:"+getComputer());
        System.out.println("english scores:"+getEnglish());
        System.out.println("average score:"+getAverageScore());
        System.out.println("total score:"+getTotalScore());
    }
}
```

实验结果及分析：

How many students do you want?->**5**↙
Student1
name:**Tom**
sex:(M/F):**M**↙
major:math
student number:**18111303011**↙
math scores:**99**↙
computer scores:**60**↙
english scores:**59**↙
Student2
name:**Jack**↙
sex:(M/F):**M**↙
major:**math**
student number:**18111303012**↙
math scores:**100**↙
computer scores:**88**↙
english scores:**77**↙
Student3
name:**Alice**↙
sex:(M/F):**F**↙
major:**english**↙
student number:**18111303013**↙
math scores:**77**↙
computer scores:**60**↙
english scores:**99**↙
Student4
name:**Anne**↙
sex:(M/F):**F**↙
major:**english**↙
student number:**18111303014**↙

math scores:**60↙**
computer scores:**20↙**
english scores:**100**
Student5
name:**Cindy↙**
sex:(M/F):**M↙**
major:**computer↙**
student number:**18111303015↙**
math scores:**88↙**
computer scores:**100↙**
english scores:**75↙**
Searching by name:**Anne↙**
serial number:4
student number:18111303014
name:Anne
sex:F
major:english
math scores:60.0
computer scores:20.0
english scores:100.0
average score:60.0
total score:180.0
Searching by student number:**18111303012↙**
serial number:2
student number:18111303012
name:Jack
sex:M
major:math
math scores:100.0
computer scores:88.0
english scores:77.0
average score:88.33333333333333
total score:265.0
The best students in math：
serial number:2
student number:18111303012
name:Jack
sex:M
major:math
math scores:100.0
computer scores:88.0
english scores:77.0
average score:88.33333333333333
total score:265.0