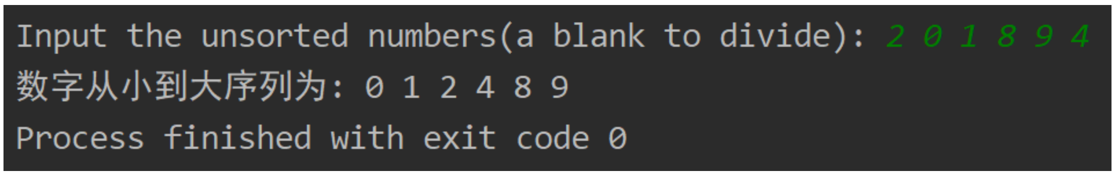
1: 第二章习题 7,编写 9 X 9 的乘法口诀表的程序

1）给出程序源码

2）给出程序运行结果

```
package coursework;


/**
  * @author Lolipop
  * @lastUpdate 2019/10/24
  */
public class NineNine {
        public static void main(String[] args){
                for(int i=1; i<=9; i++){
                        for(int n=1; n<=9; n++){
                                System.out.printf("%d*%d 得%d\t", i, n, i*n);
                        }
                        System.out.print("\n");
                }
        }
}
```



```
Input the unsorted numbers(a blank to divide): 2 0 1 8 9 4
数字从小到大序列为: 0 1 2 4 8 9
Process finished with exit code 0
```

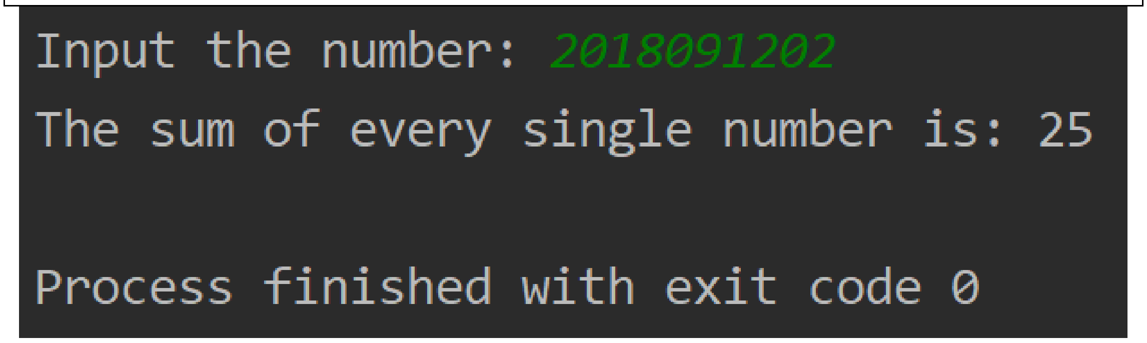
2: 编写程序，计算一个整数的各位数字之和，例如，整数 20160907，则计算并显示 2+0+1+6+0+9+0+7 的值。

1）给出程序源码

2）给出程序运行结果

```
package coursework;



import java.util.Scanner;
```

```
/**
 * @author Lolipop
 * @lastUpdate 2019/10/23
 */
public class NumberSum {
    public static void main(String[] args){
        int total = 0;
        Scanner scanner = new Scanner(System.in);
        System.out.print("Input the number: ");
        long num = scanner.nextLong();
        while (num != 0) {
            total += num % 10;
            num /= 10;
        }
        System.out.printf("The sum of every single number is: %d\n",
total);
    }
}
```



```
Input the number: 2018091202
The sum of every single number is: 25

Process finished with exit code 0
```

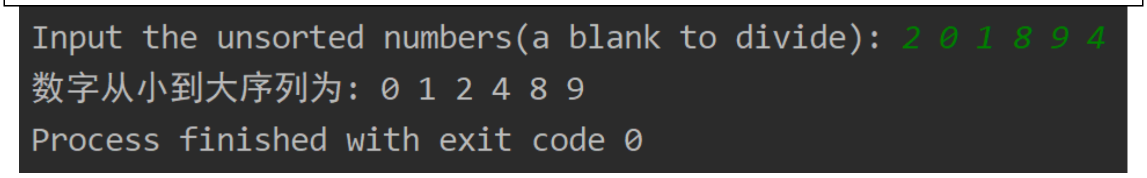
3：第三章习题 8，编写类 TestArray，只有一个 main 方法，该方法中，创建一个 int 类型的一维数组 sim，从键盘输入任意的数据，并实现数组 sim 元素从小到大排序，输出排序后的数组值。

1）给出程序源码

2）给出程序运行结果

```
package coursework;


import java.util.Arrays;

import java.util.Scanner;


/**

 * @author Lolipop

 * @lastUpdate 2019/10/24

 */

public class TestArray {

	public static void main(String[] args){

		Scanner scanner = new Scanner(System.in);


		//将输入的数字建立字符串数组

		System.out.print("Input the unsorted numbers(a blank to divide):
");
```

```
        String[] nums = scanner.nextLine().split(" ");
        int size = nums.length;

        //将字符串数组转化为 int 型赋给 sim 数组
        int[] sim = new int[size];
        for (int i=0; i<nums.length; i++){
            sim[i] = Integer.parseInt(nums[i]);
        }

        //排序 sim 数组并输出
        Arrays.sort(sim);
        System.out.print("数字从小到大序列为: ");
        for(int num: sim){
            System.out.printf("%d ", num);
        }
    }
}
```



```
Input the unsorted numbers(a blank to divide): 2 0 1 8 9 4
数字从小到大序列为: 0 1 2 4 8 9
Process finished with exit code 0
```

4：第四章习题 9，编写类 MyDate 具有属性年月日，要求一个构造函数初始化属性年月日，提供重置日期、增加日期（考虑闰年闰月）、

输出日期等成员函数。

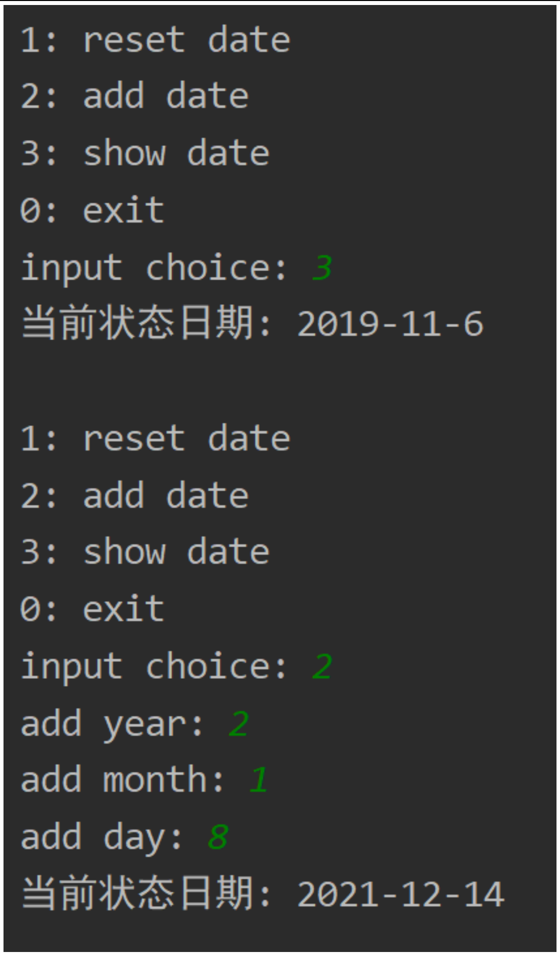
1）给出程序源码

2）给出程序运行结果

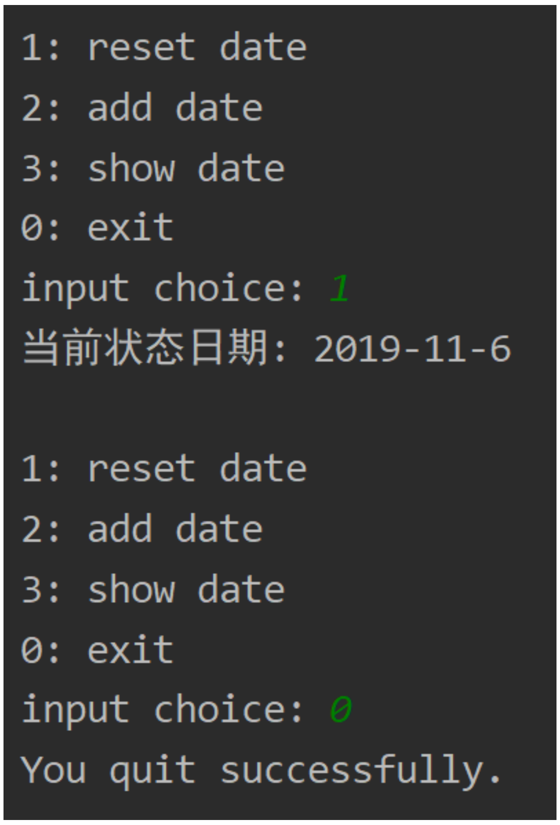
```
package coursework;


import java.util.Scanner;

import java.util.Calendar;


/**

 * @author Lolipop

 * @lastUpdate 2019/10/25

 */

public class MyDate {

    public static void main (String[] args) {

        Date date = new Date();

        date.initDate();


        Scanner scan = new Scanner(System.in);

        int choice, addYear, addMonth, addDay;


        do {

            System.out.print("\n1: reset date\n2: add date\n3: show
date\n0: exit\ninput choice: ");
                choice = scan.nextInt();
                switch (choice){
                    case 1: date.initDate(); date.printDate(); break;
                    case 2:
                        System.out.print("add year: ");
                        addYear = scan.nextInt();
                        System.out.print("add month: ");
                        addMonth = scan.nextInt();
                        System.out.print("add day: ");
                        addDay = scan.nextInt();
                        date.addDate(addYear, addMonth, addDay);
                        date.printDate();
                        break;
                    case 3: date.printDate(); break;
                    case 0: break;
                    default: System.out.println("Wrong code!");
                }
            } while (choice != 0);

            System.out.println("You quit successfully.");
        }
```

```
}

class Date {
	private int year;
	private int day;
	private int month;
	private Calendar cal = Calendar.getInstance();
	private Calendar calAdded = Calendar.getInstance();

	void initDate(){
		year = cal.get(Calendar.YEAR);
		//第一个月的值为 0，故应在月份上加一以表示客观的月份
		month = cal.get(Calendar.MONTH)+1;
		day = cal.get(Calendar.DATE);
		calAdded = Calendar.getInstance();
	}

	void addDate(int y, int m, int d){
		calAdded.add(Calendar.YEAR, y);
		calAdded.add(Calendar.MONTH, m);
		calAdded.add(Calendar.DATE, d);
```

```
            year = calAdded.get(Calendar.YEAR);

            month = calAdded.get(Calendar.MONTH)+1;

            day = calAdded.get(Calendar.DATE);

      }


      void printDate(){

            System.out.print(" 当 前 状 态 日 期 ： "+year+"-"+month+"-
"+day+"\n");

      }

}
```



```
1: reset date
2: add date
3: show date
0: exit
input choice: 3
当前状态日期: 2019-11-6

1: reset date
2: add date
3: show date
0: exit
input choice: 2
add year: 2
add month: 1
add day: 8
当前状态日期: 2021-12-14
```



```
1: reset date
2: add date
3: show date
0: exit
input choice: 1
当前状态日期: 2019-11-6

1: reset date
2: add date
3: show date
0: exit
input choice: 0
You quit successfully.
```

5: 第四章习题 10，编写类 ArraySort，该类有一个 int 类型一维数组 sim 的成员变量，一个 setOrder()的成员方法，一个带有一个参数的构造方法对 sim 数组初始化，方法 setOrder 没有参数和返回值，实现成员变量 sim 中的元素升序排序。

另外类 TestArray，只有一个 main 方法，该方法中，从键盘输入任意的数据但 int 类型的一维数组，从键盘输入任意的数据，并在创建一个 ArraySort 对象时，构造函数使用该数组做参数初始化 sim 成员变量， 并调用 setOrder 实现元素从小到大排序，并输出排序结果。

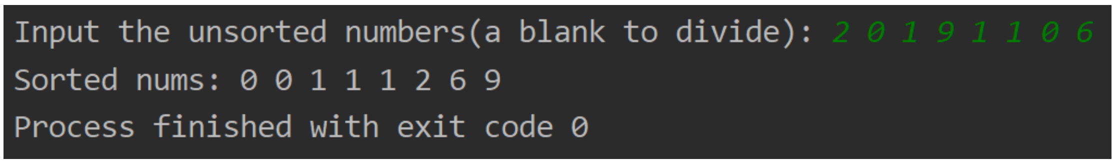
1）给出程序源码

2）给出程序运行结果

```
package coursework.classarray;
```

```
import java.util.Arrays;
import java.util.Scanner;

/**
 * @author Lolipop
 * @lastUpdate 2019/10/25
 */
public class TestArray {
	public static void main (String[] args) {
		ArraySort array = new ArraySort();

		//字符串数组赋值并传递给 array 数组
		Scanner scanner = new Scanner(System.in);
		System.out.print("Input the unsorted numbers(a blank to divide):
");
		String[] nums = scanner.nextLine().split(" ");
		int size = nums.length;
		array.sim = new int[size];
		for (int i=0; i<nums.length; i++){
			array.sim[i] = Integer.parseInt(nums[i]);
		}
```

```
            //调用 setOrder 方法并打印结果
            array.setOrder();
            System.out.print("Sorted nums: ");
            for (int i=0; i<nums.length; i++){
                System.out.printf("%d ", array.sim[i]);
            }
        }
}

class ArraySort {
        int[] sim;
        void setOrder() {
                Arrays.sort(sim);
        }
}
```



```
Input the unsorted numbers(a blank to divide): 2 0 1 9 1 1 0 6
Sorted nums: 0 0 1 1 1 2 6 9
Process finished with exit code 0
```

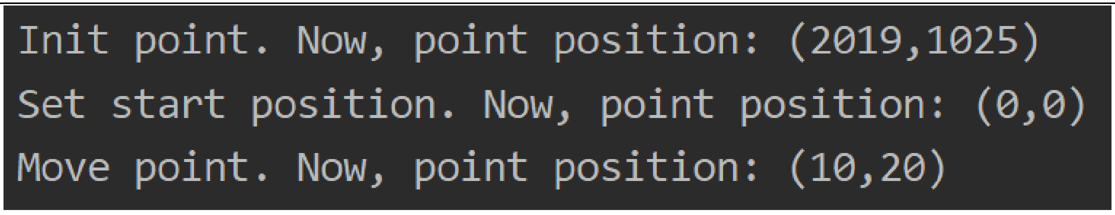
6: 第四章习题 10，创建一个类 Point,有成员变量 x,y,它们都是 int 类型，该类有四个成员方法 SetX(int),setY(int),getPoint() 和 movePoint(int,int).setx(int)和 setY(int)方法是设置成员变量 x 和 y 的

值，getPoint（）则是获得由 x，y 构成的坐标点，movePoint（int，int）带两个 int 参数，用来修改 x，y 构成的坐标点。point 类有一个构造方法，不带参数，为 x，y 设置原点值。另一个类为 TestPoint，有 main 方法用来对 point 类的实例进行测试。要求为其实例设置（0,0）坐标点，在移动到（10,20）坐标点上，并输出实例调用相应方法的结果。

1）给出程序源码

2）给出程序运行结果

```
package coursework;

import java.util.Scanner;

/**
 * @author Lolipop
 * @lastUpdate 2019/10/25
 */
public class TestPoint {
    public static void main (String[] args) {
        Point point = new Point();
        System.out.print("Init point. Now, ");
        point.getPoint();

        point.setX(0);
        point.setY(0);
        System.out.print("Set start position. Now, ");
        point.getPoint();

        point.movePoint(10, 20);
        System.out.print("Move point. Now, ");
        point.getPoint();
    }
}

class Point {
    private int x;
    private int y;
    Point () {
        x = 2019;
        y = 1025;
    }
    void setX(int positionX) {
        x = positionX;
    }
    void setY(int positionY) {
        y = positionY;
    }
    void getPoint() {
        System.out.printf("point position: (%d,%d)\n", x, y);
    }
    void movePoint(int moveX, int moveY) {
        x += moveX;
        y += moveY;
    }
}
```



```
Init point. Now, point position: (2019,1025)
Set start position. Now, point position: (0,0)
Move point. Now, point position: (10,20)
```

7: 编写程序，找出 1~n 以内的所有素数。要求使用数组元素的下标从 1~n 以内表示这些数值，数组元素的值作为素数的标志。其中用数组元素的值为 0 来表示该元素的下标的数值是素数，用 1 来表示该元素的下标的数值不是素数，并输出这些素数。

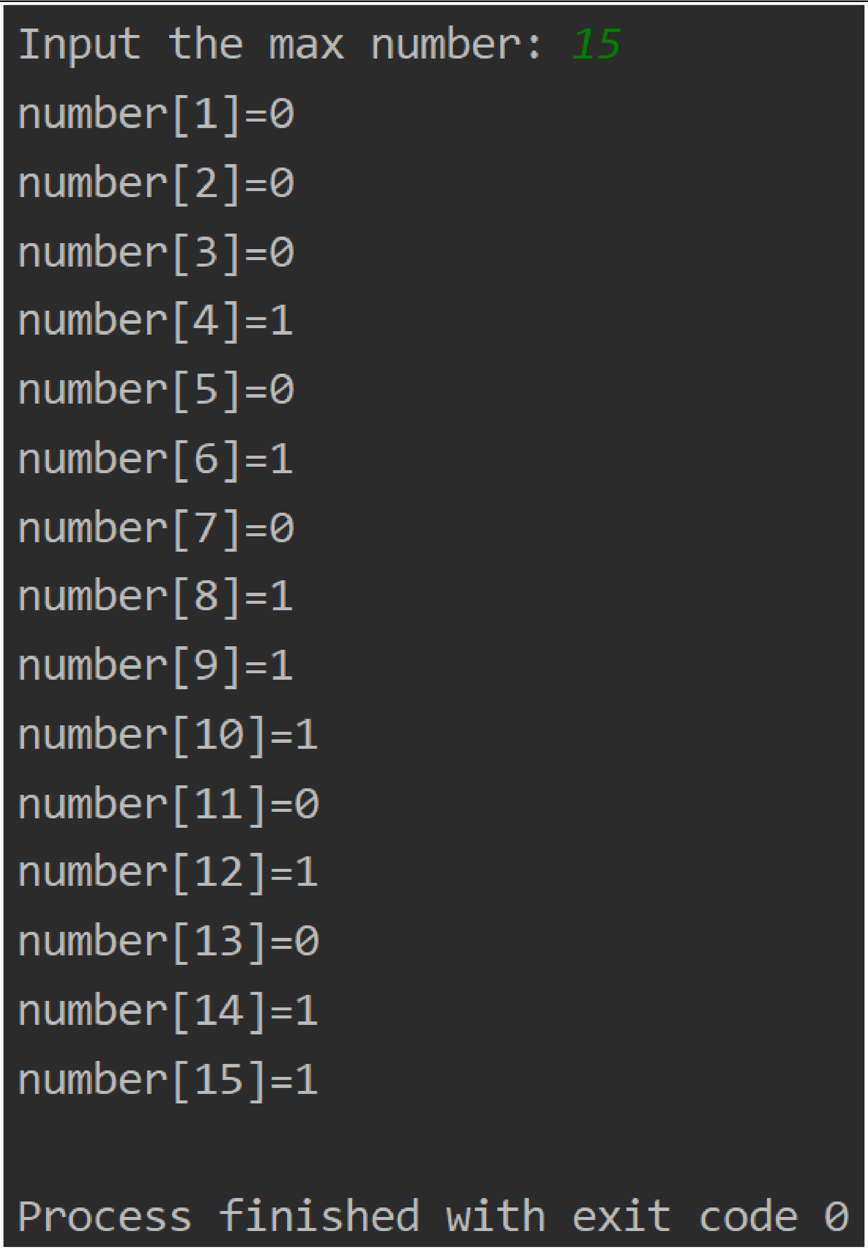
1）程序源码

2）实验结果

```
package coursework;


import java.util.Scanner;
```

```
/**
 * @author Lolipop
 * @lastUpdate 2019/10/25
 */
public class PrimeNumber {
	public static void main(String[] args){
		Scanner scanner = new Scanner(System.in);
		System.out.print("Input the max number: ");
		int n = scanner.nextInt();
		int[] number = new int[n+1];
		for (int i=1; i<number.length; i++) {
			//默认赋值为 0
			number[i] = 0;

			//非质数赋值为 1
			if (i>3) {
				for (int count = 2; count<=(i/2); count++) {
					if (i%count == 0) {
						number[i] = 1;
						break;
					}
				}
```

```
                    }


                    System.out.printf("number[%d]=%d\n", i, number[i]);
                }
            }
        }
```



```
Input the max number: 15
number[1]=0
number[2]=0
number[3]=0
number[4]=1
number[5]=0
number[6]=1
number[7]=0
number[8]=1
number[9]=1
number[10]=1
number[11]=0
number[12]=1
number[13]=0
number[14]=1
number[15]=1

Process finished with exit code 0
```

8：第五章习题 9，有类 Person 和 Student，它们之间存在继承关系，Person 有成员变量 name,sex,age，类型分别为 String,char,int，构造方法 Person（String,char,int）用来对成员变量进行初始化，成员方法 setData（String,char,int）设置成员变量 name,sex,age 的值，getData()是不带参数且返回值是 name,sex 和 age 的值构成的字符串的成员方法;

Student 是 Person 的子类，在 Student 中有 int 类型的 sID 和 classNo 用来表示学生的学号和班级号，它有带有 5 个参数的成员方法 setData()和不带参数的方法 getData()，setData()设置成员变量的值，getData()是返回五个成员变量值构成的字符串。

第五章习题 10，抽象类 Person 定义如下：

```
abstract class Person{
        String name；
        char sex；
        int age；
        abstract void setData(String name,char sex,int age);
        abstract String getData();
}
```

类 Student 和类 Teacher 均是抽象类 Person 的子类，类 Student 有成员变量 name,sex,age,sID,speciality，其中 sID 表示学生学号，speciality 表示学生专业；类 Teacher 有成员变量 name,sex,age,tID,department，其中 tID 表示教师的编号，department 表示教师所在部门，请编写类 Student 和类 Teacher 所需基本功能。

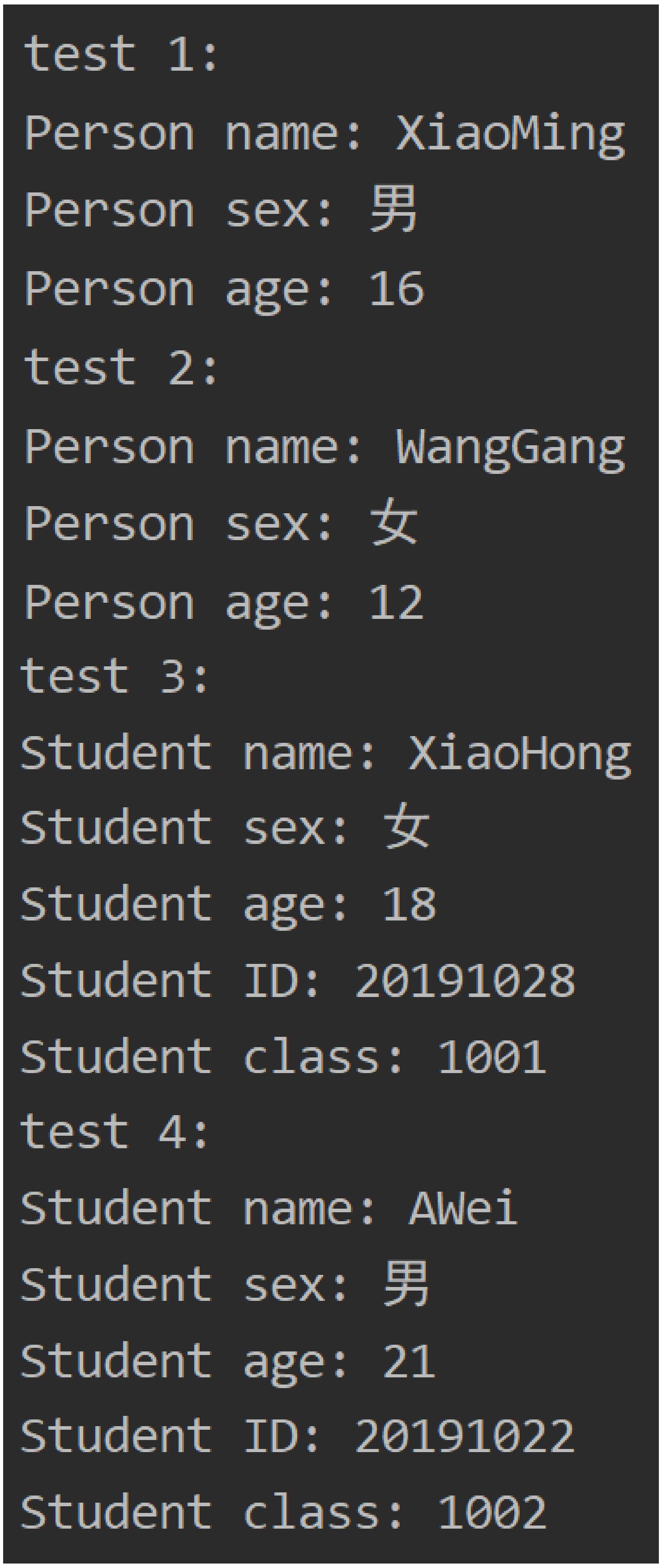
1）给出程序源码

2）给出程序运行结果

非抽象方法

```
package coursework;


/**
 * @author Lolipop
 * @lastUpdate 2019/10/28
 */
public class Person {
    String name;
    char sex;
    int age;

    Person (String testName, char testSex, int testAge) {
        this.name = testName;
        this.sex = testSex;
        this.age = testAge;
    }

    private void setData(String testName, char testSex, int testAge) {
        this.name = testName;
        this.sex = testSex;
        this.age = testAge;
    }

    protected String getData () {
        return "Person name: "+this.name+"\nPerson sex: "+this.sex+"\nPerson age: "+this.age+"\n";
    }

    public static void main (String[] args) {
        Person testPerson = new Person("XiaoMing", '男', 16);
        System.out.print("test 1:\n" + testPerson.getData());

        testPerson.setData("WangGang", '女', 12);
        System.out.print("test 2:\n" + testPerson.getData());

        Student testStudent = new Student("XiaoHong", '女', 18, 20191028, 1001);
        System.out.print("test 3:\n" + testStudent.getData());

        testStudent.setData("AWei", '男', 21, 20191022, 1002);
        System.out.print("test 4:\n" + testStudent.getData());
```

```
	}

}

class Student extends Person {
	private int sID;
	private int classNo;

	Student(String testName, char testSex, int testAge, int testsID, int testClassNo) {
		super(testName, testSex, testAge);
		this.sID = testsID;
		this.classNo = testClassNo;
	}

	void setData (String testName, char testSex, int testAge, int testsID, int testClassNo) {
		this.name = testName;
		this.sex = testSex;
		this.age = testAge;
		this.sID = testsID;
		this.classNo = testClassNo;
	}

	@Override
	protected String getData () {
		return "Student name: "+this.name+"\nStudent sex: "+this.sex+"\nStudent age: "+this.age+"\nStudent ID: "+this.sID+"\nStudent class: "+this.classNo+"\n";
	}
}
```



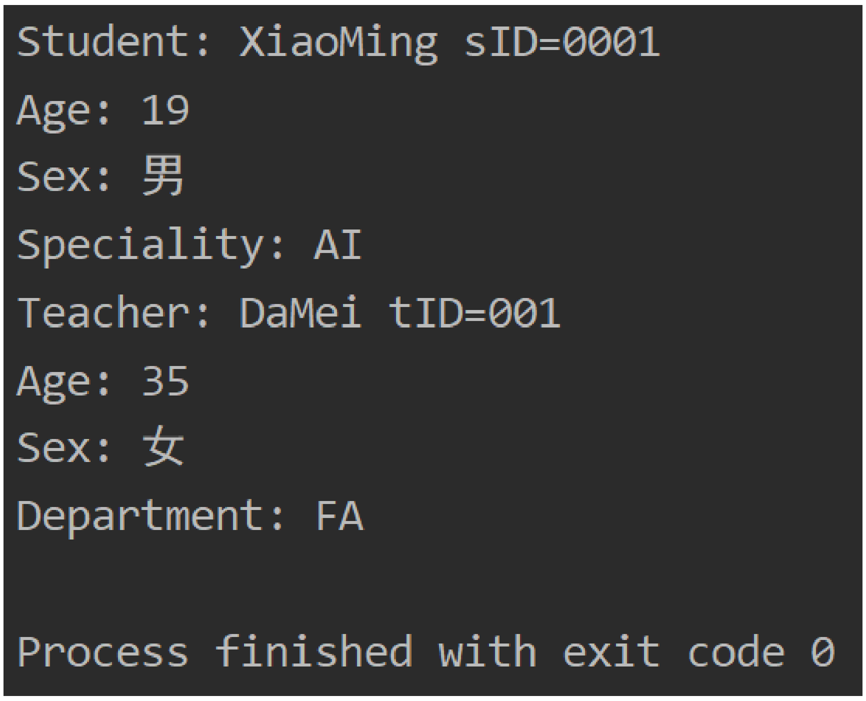
```
test 1:
Person name: XiaoMing
Person sex: 男
Person age: 16
test 2:
Person name: WangGang
Person sex: 女
Person age: 12
test 3:
Student name: XiaoHong
Student sex: 女
Student age: 18
Student ID: 20191028
Student class: 1001
test 4:
Student name: AWei
Student sex: 男
Student age: 21
Student ID: 20191022
Student class: 1002
```

抽象方法

```
package coursework.abstractperson;

/**
 * @author Lolipop
 * @lastUpdate 2019/10/25
 */
public class Person {
	public static void main (String[] args) {
		Student stu = new Student("0001", "AI");
		stu.setData("XiaoMing", '男', 19);
		System.out.print(stu.getData()+"\n");

		Teacher tea = new Teacher("001", "FA");
		tea.setData("DaMei", '女', 35);
		System.out.print(tea.getData()+"\n");
	}
}

abstract class BasePerson {
	String name;
	char sex;
	int age;

	/**
	 * set person data
	 * @param name: set person name
	 * @param sex: set person sex
	 * @param age: set person age
	 */
	abstract void setData (String name, char sex, int age);

	/**
	 * get person data
```

```
         * @return person data
         */
        abstract String getData();
}

class Student extends BasePerson {
        private String sID;
        private String speciality;

        Student (String sid, String sp) {
                this.sID = sid;
                this.speciality = sp;
        }

        @Override
        void setData(String name, char sex, int age) {
                this.name = name;
                this.sex = sex;
                this.age = age;
        }

        @Override
        String getData() {
                return    "Student:    "+this.name+"    sID="+this.sID+"\nAge:
"+this.age+"\nSex: "+this.sex+"\nSpeciality: "+this.speciality;
        }
}

class Teacher extends BasePerson {
        private String tID;
        private String department;

        Teacher (String tid, String de) {
```

```
        this.tID = tid;
        this.department = de;
    }

    @Override
    void setData(String name, char sex, int age) {
        this.name = name;
        this.sex = sex;
        this.age = age;
    }

    @Override
    String getData() {
        return "Teacher: "+this.name+" tID="+this.tID+"\nAge: "+this.age+"\nSex: "+this.sex+"\nDepartment: "+this.department;
    }
}
```



```
Student: XiaoMing sID=0001
Age: 19
Sex: 男
Speciality: AI
Teacher: DaMei tID=001
Age: 35
Sex: 女
Department: FA

Process finished with exit code 0
```

9: 第六章习题 7，创建一个接口 Print，在其中定义一个打印方法 print，再创建两个类分别实现这个接口。

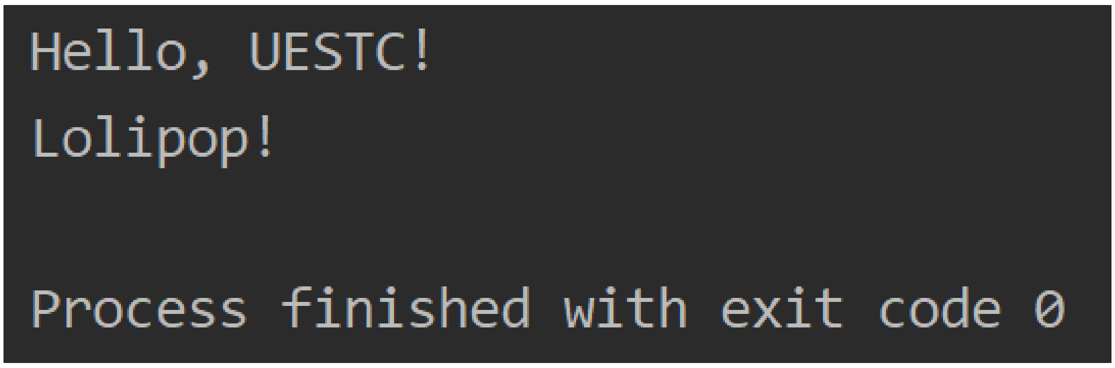
第六章习题 8，创建一个 Person 接口，它有方法 setData()和 getData()对属性 name,sex,birthday 赋值和获得这些属性组成的字符串信息；创建类 Student 实现 Person 接口，并重写 setData()成员方法，设置学生属性的成员变量 sID、speciality 设置值，重写 getData()获得学生成员变量值所组成的字符信息。

1）给出程序源码

2）给出程序运行结果

Print 接口

```
package coursework.interfacetest;

interface Print {
	/**
	  * print()：打印一些内容
	  */
	void print();
}

/**
  * @author Lolipop
  * @lastUpdate 2019/10/28
  */
public class PrintTest {
	public static void main (String[] args) {
		PrintSchool testPrintSchool = new PrintSchool();
		PrintMe testPrintMe = new PrintMe();
		testPrintSchool.print();
		testPrintMe.print();
	}
}

class PrintSchool implements Print {
	@Override
	public void print () {
		System.out.println("Hello, UESTC!");
	}
}

class PrintMe implements Print {
	@Override
	public void print () {
		System.out.println("Lolipop!");
	}
}
```

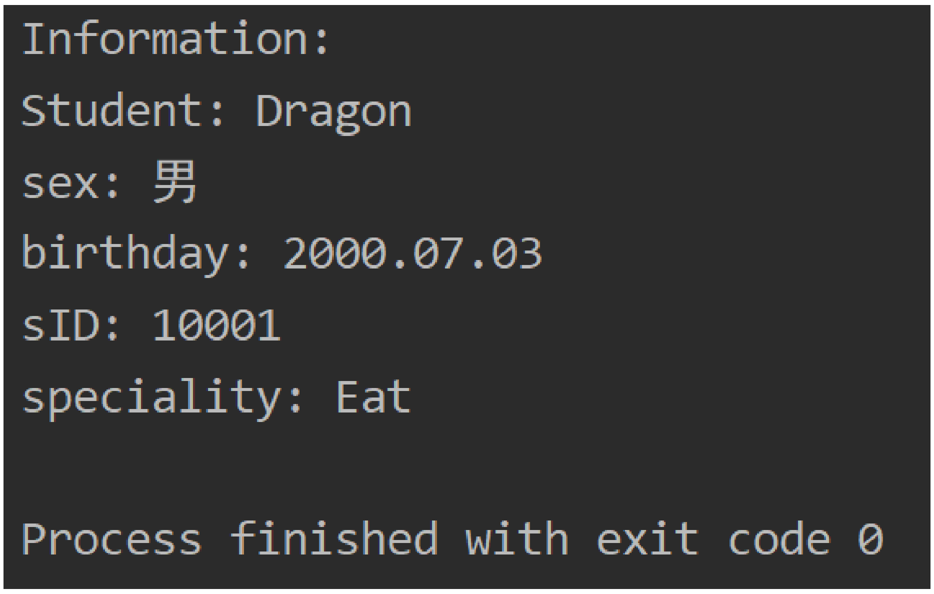


```
Hello, UESTC!
Lolipop!

Process finished with exit code 0
```

Person 接口

```
package coursework.interfacetest;

interface Person {
	/**
	 *  对属性 name,sex,birthday 赋值;
	 * @param name   设置姓名
	 * @param sex  设置性别
	 * @param birthday  设置生日
	 */
	void setData(String name, char sex, String birthday);

	/**
	 *  获得这些属性组成的字符串信息。
	 * @return name,sex,birthday 属性组成的字符串信息。
	 */
	String getData();
}

/**
 * @author Lolipop
 * @lastUpdate 2019/10/28
 */
public class PersonTest {
	public static void main (String[] args) {
		InfStudent student = new InfStudent();
		student.setData("Dragon", '男', "2000.07.03", 10001, "Eat");
		student.print();
	}
}

class InfStudent implements Person,Print{
	private String name;
	private char sex;
	private String birthday;
	private int sID;
	private String speciality;

	InfStudent () {
		name = "unset";
		sex = '男';
		birthday = "2000.01.01";
		sID = 10000;
		speciality = "unset";
	}

	@Override
	public void setData(String readName, char readSex, String
readBirthday) {
		this.name = readName;
		this.sex = readSex;
		this.birthday = readBirthday;
	}

	public void setData(String readName, char readSex, String
readBirthday, int readSID, String readSpeciality) {
		this.name = readName;
		this.sex = readSex;
		this.birthday = readBirthday;
		this.sID = readSID;
		this.speciality = readSpeciality;
	}

	@Override
	public String getData() {
		return "Student: "+this.name+"\nsex: "+this.sex+"\nbirthday:
"+this.birthday+"\nsID: "+this.sID+"\nspeciality: "+this.speciality;
	}

	@Override
	public void print() {
		String info = this.getData();
		System.out.println("Information:\n"+info);
	}
```

```
}
```



```
Information:
Student: Dragon
sex: 男
birthday: 2000.07.03
sID: 10001
speciality: Eat

Process finished with exit code 0
```

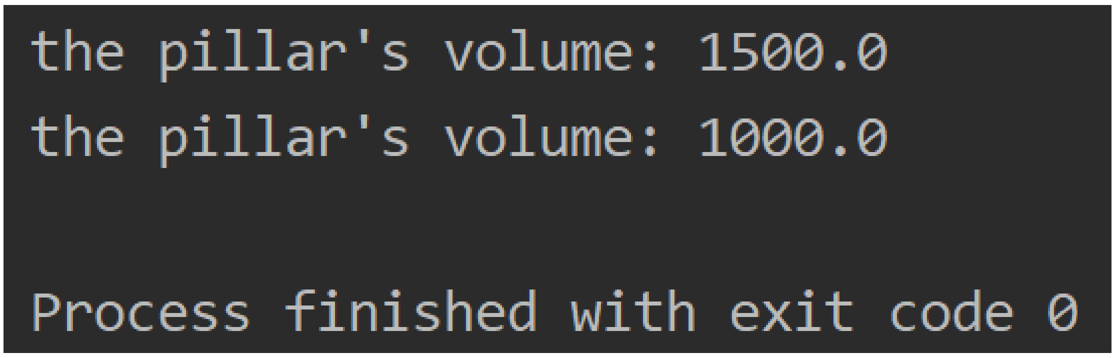
10： 有几何形状边数为 n 及可计算面积 area 的 Shape 类，其子类 Triangle 类及 Rectangle 类实现几何形状三角形和矩形面积 area 计算，利用前三个形状类实现柱体 Pillar 类的体积计算，并在 PillarTest 类中实现对某一柱体的体积计算。

1）给出程序源码

2）给出程序运行结果

```
package coursework;

/**
 * @author Lolipop
 * @lastUpdate 2019/10/30
 */
public class PillarTest {
	public static void main (String[] args) {
		Pillar pi1 = new Pillar();
		pi1.setPillar(3, 10, 15, 20);
		System.out.print("the pillar's volume: "+pi1.getVolume()+"\n");

		Pillar pi2 = new Pillar();
		pi2.setPillar(4, 20, 5, 10);
		System.out.print("the pillar's volume: "+pi2.getVolume()+"\n");
	}
}

class Shape {
	private int n;
	private double area;
	private double length;
	private double width;

	Shape () {
		n = 0;
		area = 0;
		length = 0;
		width = 0;
	}

	void setData (int sides, double l, double w) {
		this.n = sides;
		this.length = l;
		this.width = w;
	}

	static class Triangle {
		/**
		 * 计算三角形时，length 为底边长，width 为底边上的高
		 */
		double getTriangleArea (double l, double w) {
			return l*w/2;
		}
	}

	static class Rectangle {
        /**
         * 计算矩形时，length 为长，width 为宽
         */
        double getRectangleArea (double l, double w) {
            return l*w;
        }
    }

    double getArea () {
        int sides = this.n;
        switch (sides) {
            case 3: Triangle tr = new Triangle(); this.area = 
tr.getTriangleArea(this.length, this.width); break;
            case 4: Rectangle re = new Rectangle(); this.area = 
re.getRectangleArea(this.length, this.width); break;
            default: System.out.println("Wrong sides number!");
        }
        return this.area;
    }
}

class Pillar {
    private double height;
    private double volume;
    private Shape bottom = new Shape();

    Pillar () {
        height = 0;
        volume = 0;
    }

    void setPillar (int sides, double l, double w, int h) {
        bottom.setData(sides, l, w);
```

```
		this.height = h;
	}

	double getVolume () {
		this.volume = bottom.getArea() * this.height;
		return volume;
	}
}
```



```
the pillar's volume: 1500.0
the pillar's volume: 1000.0

Process finished with exit code 0
```

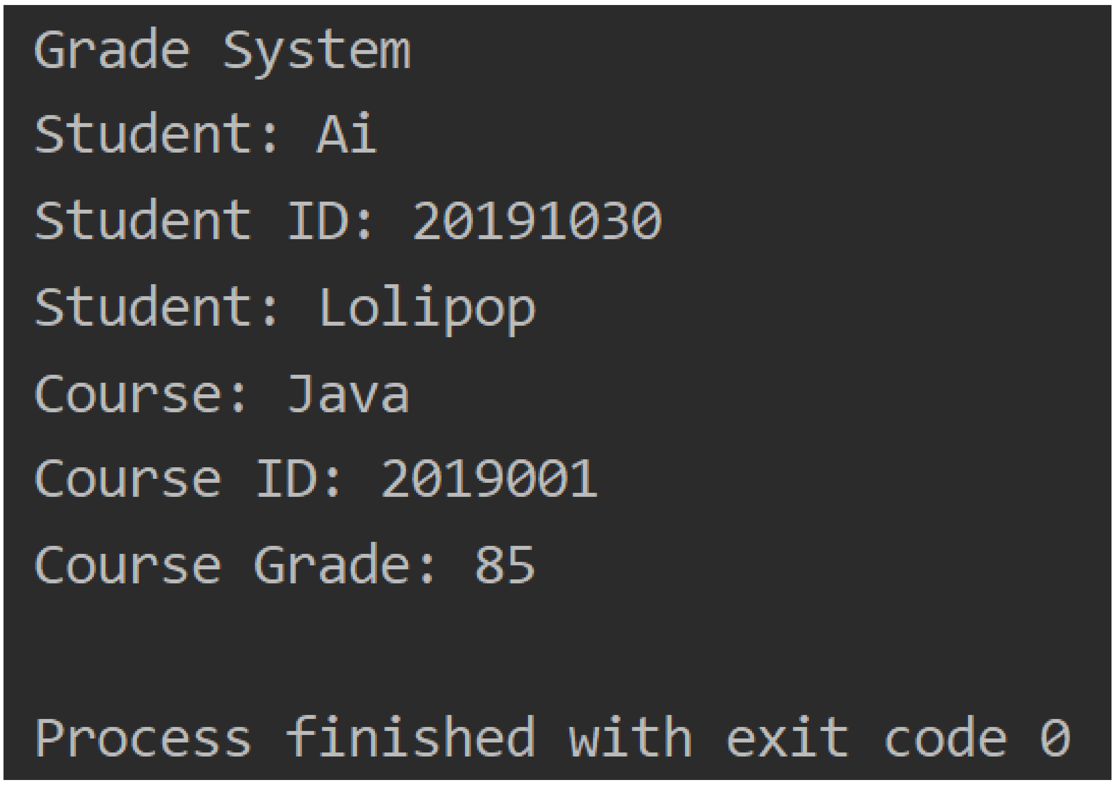
11: 编写程序，创建学生成绩中所涉及的类：Student 类、Teacher 类、Course 类，并由 Grade 类将 Student 类、Teacher 类和 Course 类关联起来，由 GradeTest 类对以上四个类进行测试。

1）程序源码

2）实验结果

```
package coursework.grade;


/**
 * @author Lolipop
 * @lastUpdate 2019/10/30
 */
public class GradeTest {
	public static void main (String[] args) {
		Grade gr = new Grade();
		gr.setData("Ai", 20191030, "Lolipop", "Java", 2019001, 85);
		gr.printGrade();
	}
```

```
}

class Student {
	private String sName;
	private int sId;

	Student () {
		this.sName = "unset";
		this.sId = 0;
	}

	void setData (String name, int id) {
		this.sName = name;
		this.sId = id;
	}

	String getData () {
		return "Student: "+this.sName+"\nStudent ID: "+this.sId+"\n";
	}
}

class Teacher {
	private String tName;

	Teacher () {
		this.tName = "unset";
	}

	void setData (String name) {
		this.tName = name;
	}

	String getData () {
		return "Student: "+this.tName+"\n";
	}
}

class Course {
	private String cName;
	private int cId;

	Course () {
		this.cName = "unset";
		this.cId = 0;
	}

	void setData (String name, int id) {
		this.cName = name;
		this.cId = id;
	}

	String getData () {
		return "Course: "+this.cName+"\nCourse ID: "+this.cId+"\n";
	}
}

class Grade {
	private Student st = new Student();
	private Teacher te = new Teacher();
	private Course co = new Course();
	private int grade;

	Grade () {
		this.grade = 0;
	}
	void setData (String sName, int sId, String tName, String cname, int cId, int g) {
		st.setData(sName, sId);
		te.setData(tName);
		co.setData(cname, cId);
		this.grade = g;
	}

	void printGrade () {
		System.out.print("Grade System\n"+st.getData()+te.getData()+co.getData()+"Course Grade: "+this.grade+"\n");
	}
}
```



```
Grade System
Student: Ai
Student ID: 20191030
Student: Lolipop
Course: Java
Course ID: 2019001
Course Grade: 85

Process finished with exit code 0
```

12： 第七章习题 6，编写一个含有 ArithmeticException、IndexOutOfBoundsException 和 NullPointerException 异常处理程序。

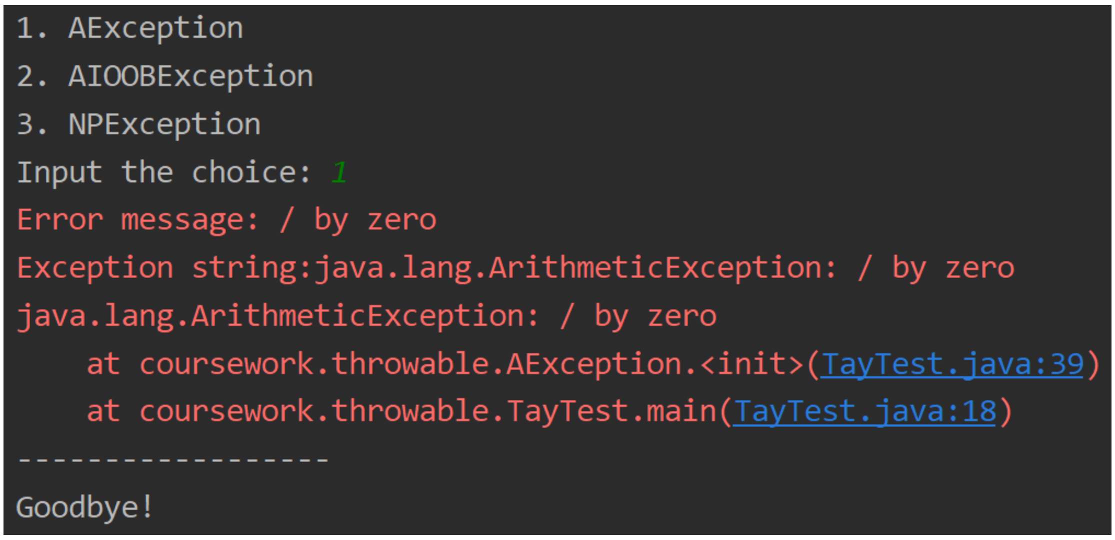
1）给出程序源码

2）给出程序运行结果

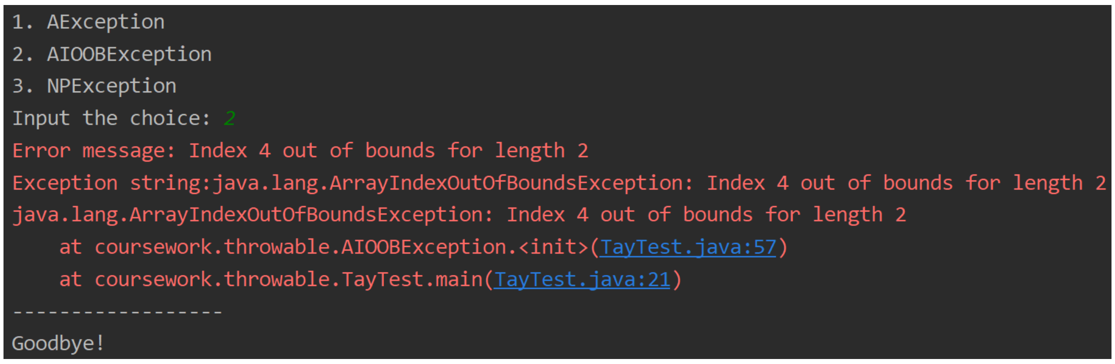
```
package coursework.throwable;

import java.util.Scanner;

/**
 * @author Lolipop
 * @lastUpdate 2019/11/6
 */
public class TayTest {
	public static void main (String[] args) {
		Scanner scan = new Scanner(System.in);

		System.out.print("1.    AException\n2.    AIOOBException\n3.
NPException\nInput the choice: ");
		int choice = scan.nextInt();

		switch (choice) {
			case 1:
				new AException();
				break;
			case 2:
				new AIOOBException();
				break;
			case 3:
				new NPException();
				break;
			default:
				System.out.println("Wrong code!");
		}
	}
}

/**
```

```
 * ArithmeticException: 算术错误情形
 */
class AException {
	AException () {
		try {
			int a = 3;
			a = a / 0;
		} catch (ArithmeticException e) {
			System.err.println("Error message: "+e.getMessage());
			System.err.println("Exception string:"+e.toString());
			e.printStackTrace();
		} finally {
			System.out.println("------------------\nGoodbye!");
		}
	}
}

/**
 * ArrayIndexOutOfBoundsException: 数组大小小于或大于实际的数组大小
 */
class AIOOBException {
	AIOOBException () {
		try {
			int[] a = new int[2];
			a[4] = 3;
		} catch (IndexOutOfBoundsException e) {
			System.err.println("Error message: "+e.getMessage());
			System.err.println("Exception string:"+e.toString());
			e.printStackTrace();
		} finally {
			System.out.println("------------------\nGoodbye!");
		}
```

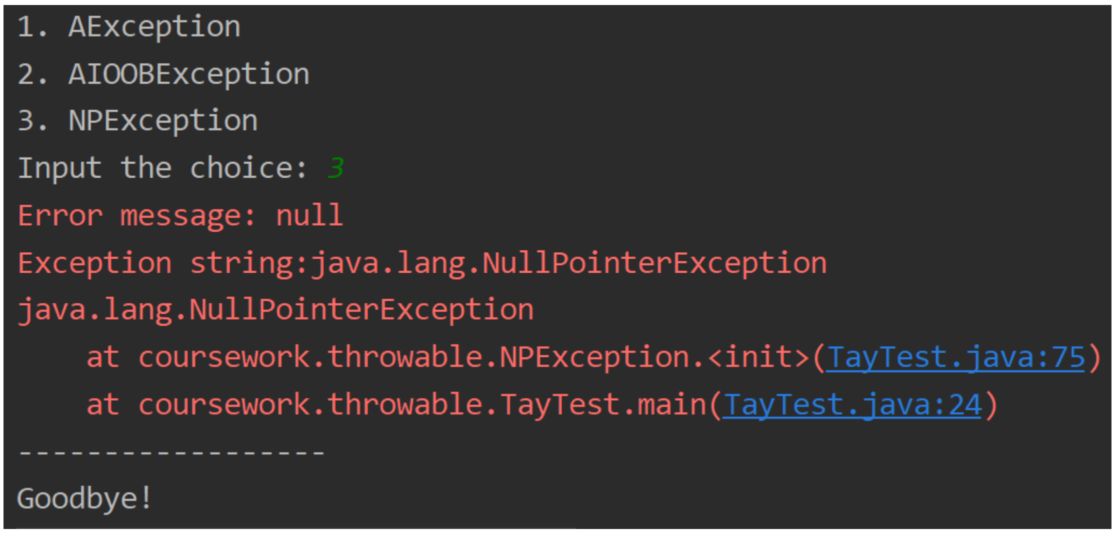
```
        }
}

/**
 * NullPointerException: 尝试访问 null 对象成员
 */
class NPException {
        NPException () {
                try {
                        String name = null;
                        if (name.equals("null")){
                                System.out.print(name);
                        }
                } catch (NullPointerException e) {
                        System.err.println("Error message: "+e.getMessage());
                        System.err.println("Exception string:"+e.toString());
                        e.printStackTrace();
                } finally {
                        System.out.println("------------------\nGoodbye!");
                }
        }
}
```



```
1. AException
2. AIOOBException
3. NPException
Input the choice: 1
Error message: / by zero
Exception string:java.lang.ArithmeticException: / by zero
java.lang.ArithmeticException: / by zero
    at coursework.throwable.AException.<init>(TayTest.java:39)
    at coursework.throwable.TayTest.main(TayTest.java:18)
------------------
Goodbye!
```



```
1. AException
2. AIOOBException
3. NPException
Input the choice: 2
Error message: Index 4 out of bounds for length 2
Exception string:java.lang.ArrayIndexOutOfBoundsException: Index 4 out of bounds for length 2
java.lang.ArrayIndexOutOfBoundsException: Index 4 out of bounds for length 2
    at coursework.throwable.AIOOBException.<init>(TayTest.java:57)
    at coursework.throwable.TayTest.main(TayTest.java:21)
------------------
Goodbye!
```



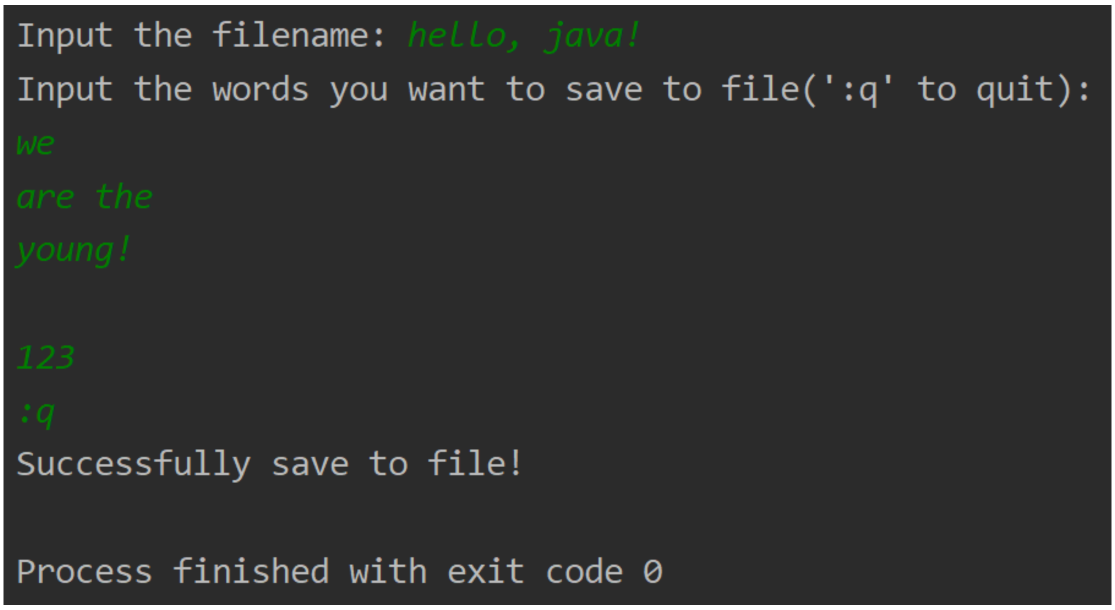
```
1. AException
2. AIOOBException
3. NPException
Input the choice: 3
Error message: null
Exception string:java.lang.NullPointerException
java.lang.NullPointerException
    at coursework.throwable.NPException.<init>(TayTest.java:75)
    at coursework.throwable.TayTest.main(TayTest.java:24)
------------------
Goodbye!
```

13：第九章习题 10，编写程序实现从键盘输入数据，保存到指定文件里。

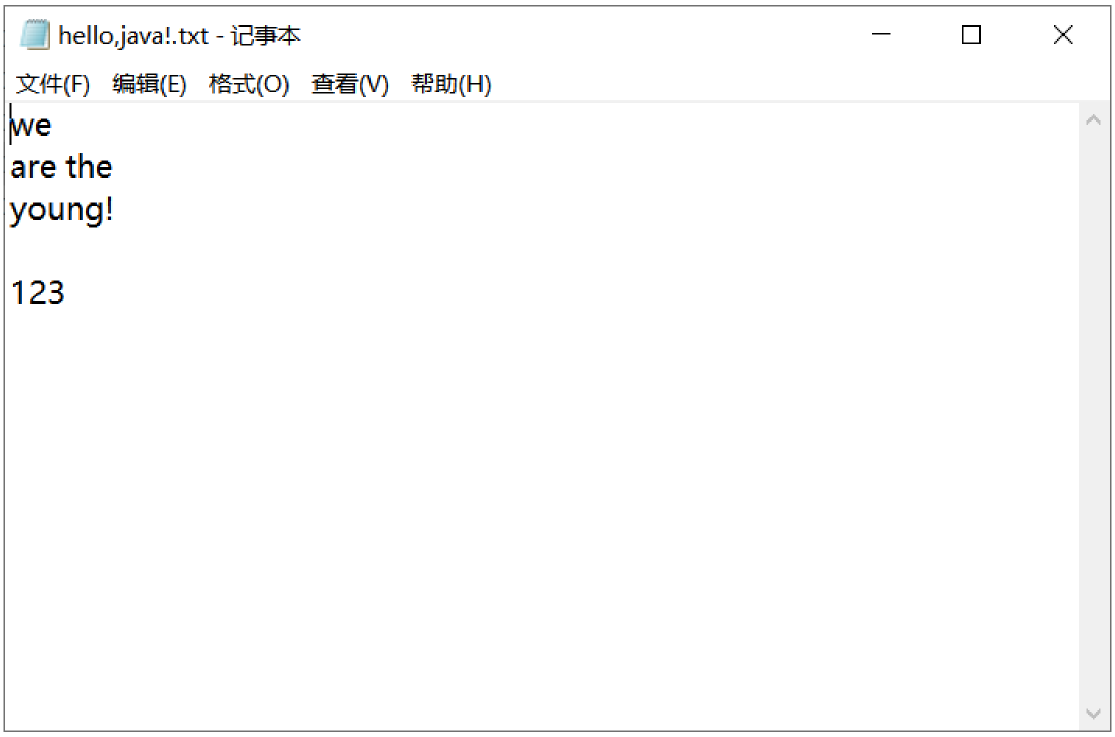
1）给出程序源码

2）给出程序运行结果

```
package coursework.savetofile;

import java.io.FileNotFoundException;
import java.util.Scanner;
import java.io.PrintWriter;

/**
 * @author Lolipop
 * @lastUpdate 2019/10/30
 */
public class SaveToFile {
    public static void main (String[] args) throws FileNotFoundException
{
        //获取文件名
        System.out.print("Input the filename: ");
        Scanner read = new Scanner(System.in);
        String filename = read.nextLine();

        //获取输入内容并保存
        System.out.println("Input the words you want to save to file(':q'
to quit):");
        PrintWriter write = new PrintWriter(filename+".txt");
        String line = read.nextLine();

        //输入':q'时结束录入
        while(!":q".equals(line))
        {
            write.println(line);
            line = read.nextLine();
        }

        write.close();
        read.close();

        System.out.println("Successfully save to file!");
    }
}
```



```
Input the filename: hello, java!
Input the words you want to save to file(':q' to quit):
we
are the
young!

123
:q
Successfully save to file!

Process finished with exit code 0
```



```
we
are the
young!

123
```

14: 第 11 章习题 7，编写程序，在面板上显示三个按钮，按钮上分别显示是：set red， set green，set blue，(1) 当按下 set red 按钮，窗口背景变为红色；（2）当按下 set green 按钮，窗口背景变为绿色；当按下 set blue 按钮，窗口背景变为蓝色。

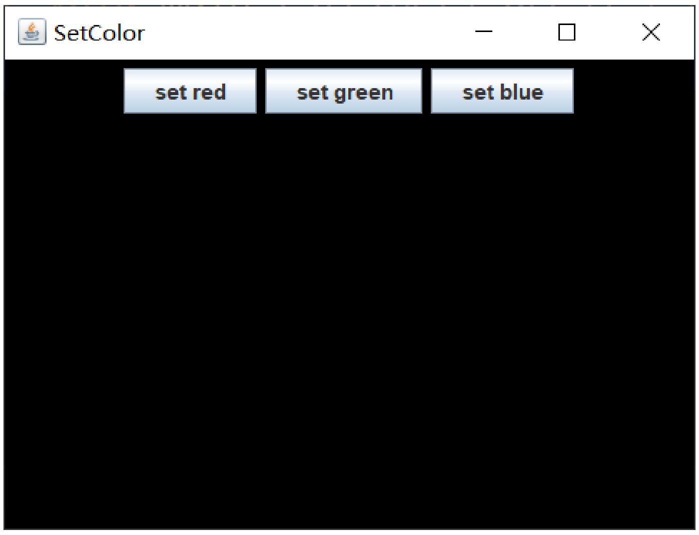
1）给出程序源码

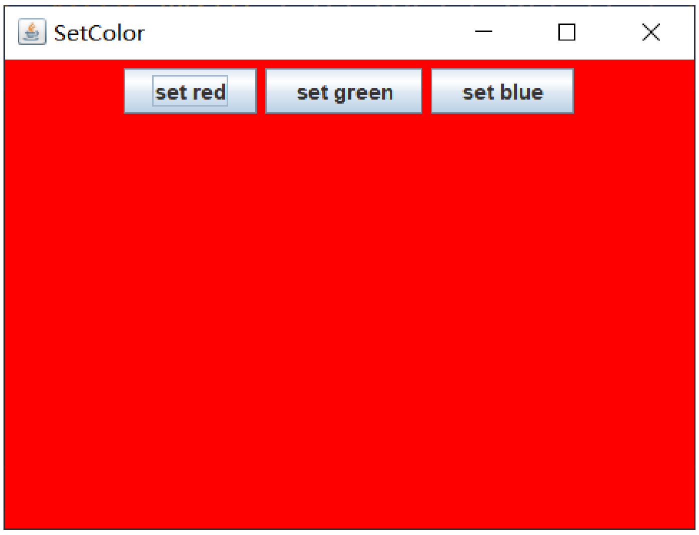
2）给出程序运行结果

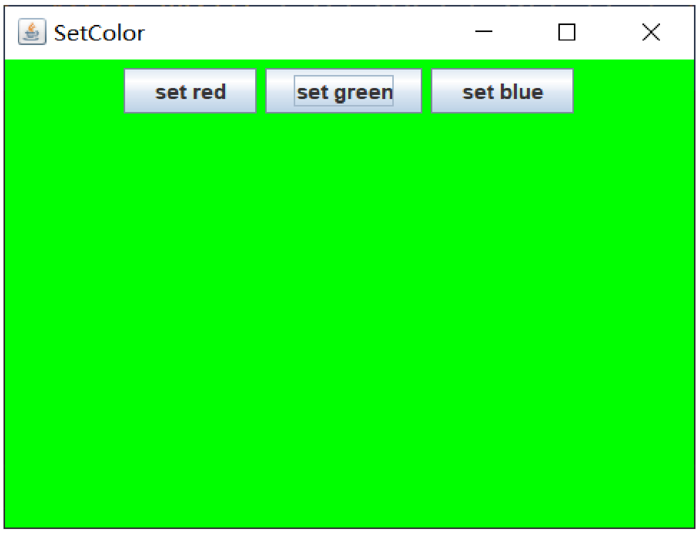
```
package coursework.gui;
```

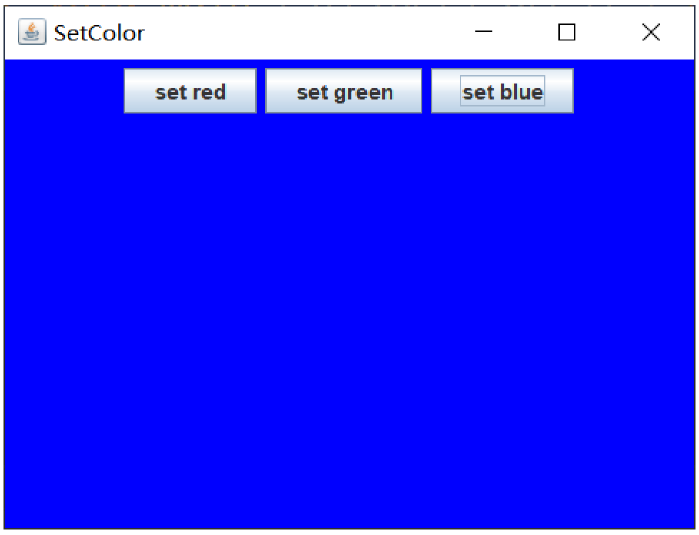
```
import javax.swing.*;
import java.awt.Color;

/**
 * @author Lolipop
 * @lastUpdate 2019/11/5
 */
public class SetColor {
	private void init() {
		// basic
		JFrame jf = new JFrame("SetColor");
		jf.setVisible(true);
		jf.setSize(400, 300);

jf.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
		JPanel jp = new JPanel();
		jp.setBackground(Color.black);

		// buttons
		JButton redBtn = new JButton("set red");
		JButton greenBtn = new JButton("set green");
		JButton blueBtn = new JButton("set blue");

		// button-event
		redBtn.addActionListener(e -> jp.setBackground(Color.red));
		greenBtn.addActionListener(e                                   ->
jp.setBackground(Color.green));
		blueBtn.addActionListener(e -> jp.setBackground(Color.blue));

		// Panel
		jp.add(redBtn);
		jp.add(greenBtn);
```

```
            jp.add(blueBtn);
            jf.setContentPane(jp);
        }

        public static void main (String[] args) {
            new SetColor().init();
        }
}
```









15: 第 11 章习题 10，编写类似 windows“记事本”的界面程序。

1）给出程序源码

2）给出程序运行结果

```
package coursework.gui;
```

```
import javax.swing.*;
import javax.swing.filechooser.FileNameExtensionFilter;
import java.awt.*;
import java.awt.datatransfer.*;
import java.awt.event.KeyEvent;
import java.io.*;
import java.nio.charset.StandardCharsets;

/**
 * @author Lolipop
 * @version 1.0.1
 * @lastUpdate 2019/11/6
 */
public class NoteBook {
	private JTextArea textArea;
	private File file = null;
	private JFrame frame = new JFrame("NoteBook");
	private          Clipboard          clipboard          =
frame.getToolkit().getSystemClipboard();

	private NoteBook() {
		// Frame
		frame.setSize(600, 400);

frame.setDefaultCloseOperation(WindowConstants.DISPOSE_ON_CLO
SE);

		// TextArea
		textArea = new JTextArea();
		textArea.setFont(new Font("黑体", Font.PLAIN, 20));

		// ScrollPane
```

```
        JScrollPane pane = new JScrollPane(textArea);
        frame.add(pane);

        // Menu
        // Menu: init menu body
        JMenuBar menu = new JMenuBar();
        frame.setJMenuBar(menu);

        // Menu: create menus
        JMenu fileMenu = new JMenu("File(F)");
        JMenu editMenu = new JMenu("Edit(E)");
        fileMenu.setMnemonic(KeyEvent.VK_F);
        editMenu.setMnemonic(KeyEvent.VK_E);

        // Menu: create menu items
        JMenuItem    newItem    =    new    JMenuItem("New(N)",
KeyEvent.VK_N);
        JMenuItem    openItem    =    new    JMenuItem("Open(O)",
KeyEvent.VK_O);
        JMenuItem    saveItem    =    new    JMenuItem("Save(S)",
KeyEvent.VK_S);
        JMenuItem    exitItem    =    new    JMenuItem("Exit(X)",
KeyEvent.VK_X);
        JMenuItem    cutItem    =    new    JMenuItem("Cut(T)",
KeyEvent.VK_T);
        JMenuItem    copyItem    =    new    JMenuItem("Copy(C)",
KeyEvent.VK_C);
        JMenuItem    pasteItem    =    new    JMenuItem("Paste(P)",
KeyEvent.VK_P);

        // fileMenu: set events
        // 新建 NoteBook 窗口
        newItem.addActionListener(e -> new NoteBook());
        // 打开文件
        openItem.addActionListener(e -> {
            JFileChooser fileChooser = new JFileChooser();
            if        (fileChooser.showOpenDialog(openItem)          ==
JFileChooser.APPROVE_OPTION) {
                File aimFile = fileChooser.getSelectedFile();

                // 打开新窗口并读取文件
                NoteBook newNoteBook = new NoteBook();
                readFile(aimFile, newNoteBook.textArea);
            }
        });

        // 保存文件
        saveItem.addActionListener(e -> {
            // 文件存在时（已经保存过）
            if (file != null) {
                saveFile(file.getPath());
            }

            // 文件不存在时（初次保存）
            else {
                JFileChooser fileChooser = new JFileChooser();

                // 后缀名过滤
                String extension = ".txt";
                fileChooser.setFileFilter(new
FileNameExtensionFilter("文本文件(*.txt)", extension));

                if      (fileChooser.showSaveDialog(saveItem)        ==
JFileChooser.APPROVE_OPTION) {
                    File newFile = fileChooser.getSelectedFile();
                    // 获取用户输入的文件名
                    String fName = fileChooser.getName(newFile);

                    // 若文件名不包含".txt"后缀则在最后加上
".txt"
                    if (!fName.contains(extension)) {
                        newFile                    =                    new
File(fileChooser.getCurrentDirectory(), fName+".txt");
                    }

                    // 保存文件
                    saveFile(newFile.getPath());

                    // 修改全局变量 file
                    file = newFile;

                    // 修改窗口 title
                    frame.setTitle(file.getName()+" - NoteBook");
                }
            }
        });

        // 退出 NoteBook
        exitItem.addActionListener(e -> {
            int   choice   =   JOptionPane.showConfirmDialog(null,
"Confirm exit NoteBook?",
                    "Exit", JOptionPane.YES_NO_OPTION);
            if (choice == 0) {
                frame.dispose();
            }
        });
        // editMenu: set events
        // 剪切
        cutItem.addActionListener(e -> {
            // 将选中的文本内容传递给剪切板
            StringSelection         cutText         =         new
StringSelection(textArea.getSelectedText());
            clipboard.setContents(cutText, null);

            // 删除选中文本
            int start = textArea.getSelectionStart();
            int end = textArea.getSelectionEnd();
            textArea.replaceRange("", start, end);
        });

        // 复制
        copyItem.addActionListener(e -> {
            StringSelection         copyText         =         new
StringSelection(textArea.getSelectedText());
            clipboard.setContents(copyText, null);
        });

        // 粘贴
        pasteItem.addActionListener(e -> {
            // 从剪切板获取内容保存到 contents 中
            Transferable contents = clipboard.getContents(null);

            // 设置 DataFlavor 映射剪切板 String 型数据
            DataFlavor flavor = DataFlavor.stringFlavor;

            // 若存在 String 型数据，则将数据粘贴到光标选中处
            if (contents.isDataFlavorSupported(flavor)) {
                try {
                    // 将 contents 数据转化成 String 格式保存到
text 中
                    String                text                    =
(String)contents.getTransferData(flavor);

                    // 替换选中内容
                    int start = textArea.getSelectionStart();
                    int end = textArea.getSelectionEnd();
                    textArea.replaceRange(text, start, end);
                } catch (UnsupportedFlavorException | IOException
ex) {
                    ex.printStackTrace();
                }
            }
        });

        // Menu: add items to menus
        fileMenu.add(newItem);
        fileMenu.add(openItem);
        fileMenu.add(saveItem);
        fileMenu.add(exitItem);
        editMenu.add(cutItem);
        editMenu.add(copyItem);
        editMenu.add(pasteItem);

        // Menu: add menus to menu body & set visible
        menu.add(fileMenu);
        menu.add(editMenu);
        menu.setVisible(true);

        // 设置界面可见
        frame.setVisible(true);
    }
    /**
     *  读取文件并在新的窗口显示出来
     * @param file  选择欲打开的文件
     * @param textArea  新建窗口的 textArea
     */
    private void readFile (File file, JTextArea textArea) {
        // init StringBuilder
        StringBuilder sBuilder = new StringBuilder();
        try {
            // init BufferedReader & str
            //  指定 GB2312 编码以显示文件的中文字符
            BufferedReader    bReader    =    new    BufferedReader(new
InputStreamReader(new FileInputStream(file),
                    "GB2312"));
            String str;

            // BufferedReader 所读取数据不为空行时，把 str 存储的
行内容传递给 StringBuilder
            while ((str = bReader.readLine()) != null) {
                sBuilder.append(str).append('\n');
            }

            //  将 StringBuilder 存储的数据显示在 textArea 上
            textArea.setText(sBuilder.toString());
        } catch (IOException e) {
            e.printStackTrace();
        }
    }

    /**
     *  将输入的内容保存为文本文件
     * @param path  文件存储的路径
     */
    private void saveFile (String path) {
        FileOutputStream os;
        try {
            // init FileOutputStream
            os = new FileOutputStream(path);

            // 将 textArea 域的内容转化为 UTF_8 编码格式的文本字符对象，并写入对应路径下的文件中
os.write(textArea.getText().getBytes(StandardCharsets.UTF_8));
            os.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }

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

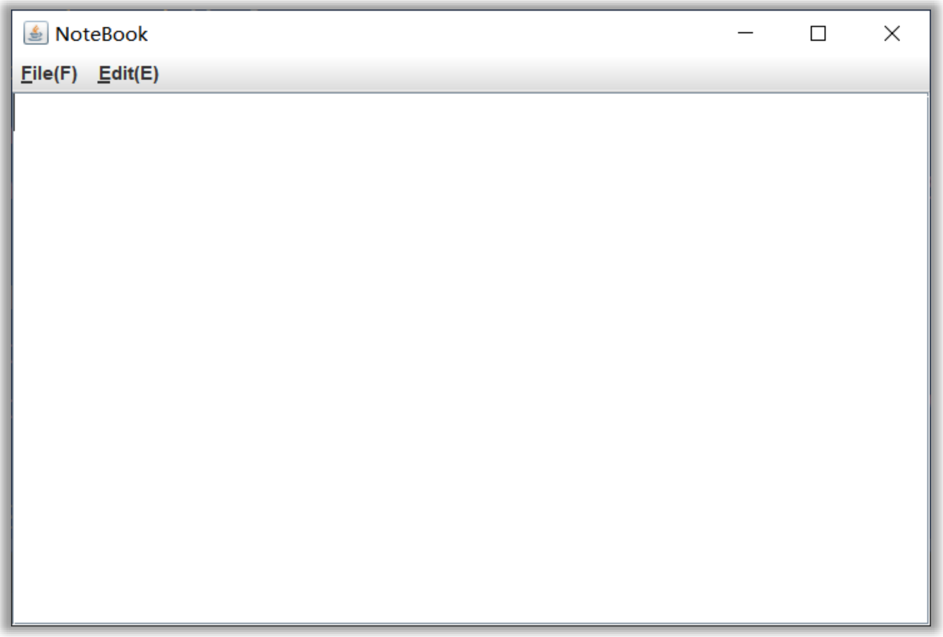


图 1：主界面

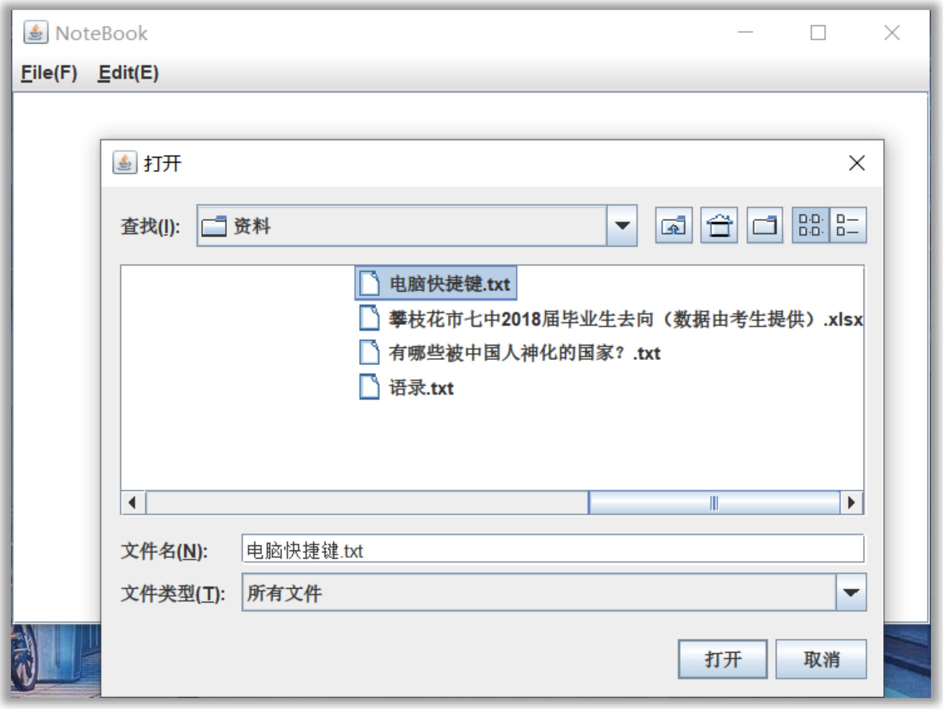
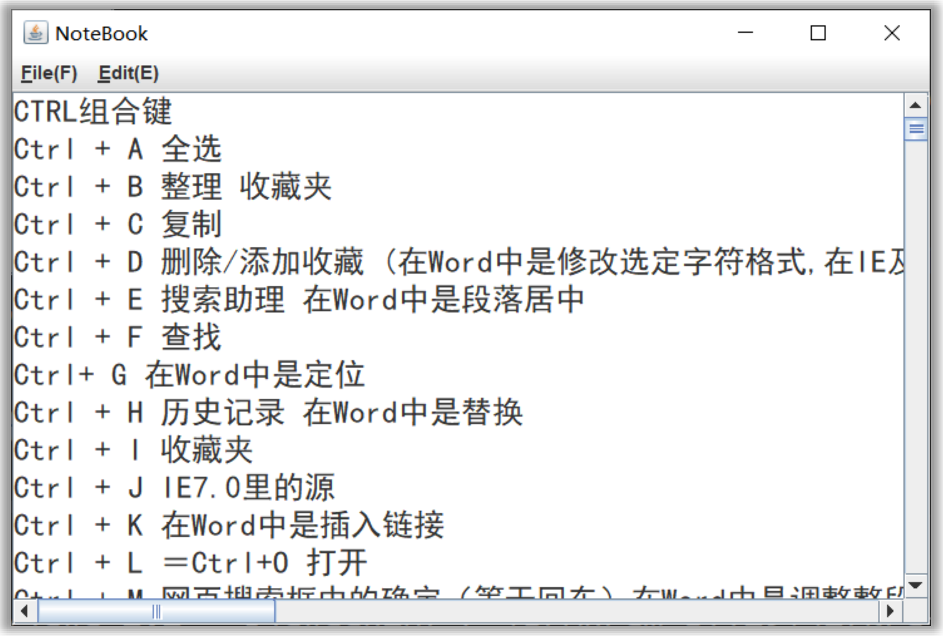
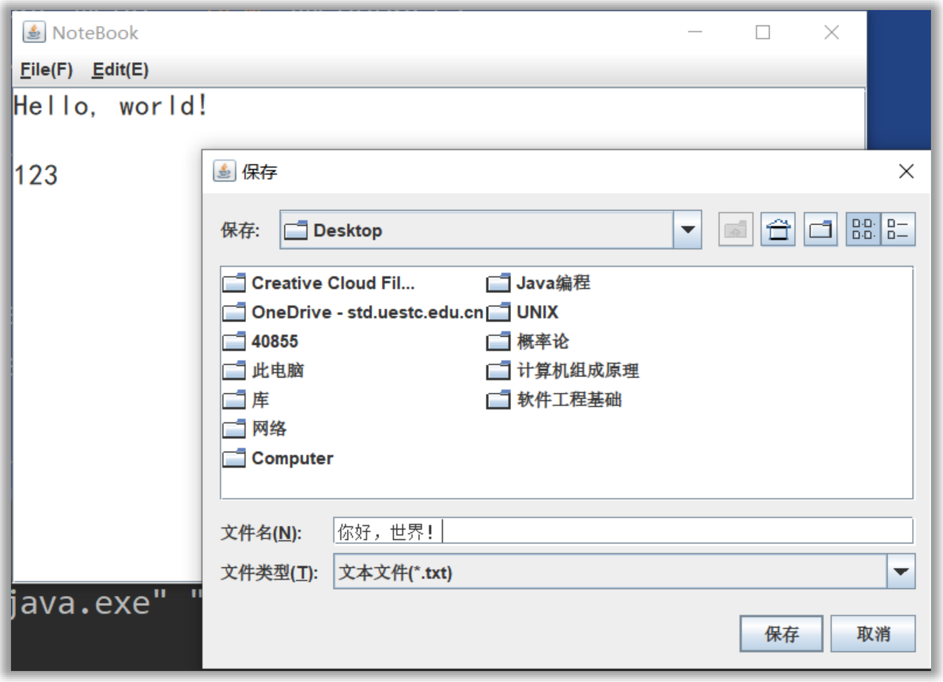

NoteBook
File(F) Edit(E)
打开
查找(I): 资料
电脑快捷键.txt
攀枝花市七中2018届毕业生去向（数据由考生提供）.xlsx
有哪些被中国人神化的国家？.txt
语录.txt
文件名(N): 电脑快捷键.txt
文件类型(T): 所有文件
打开
取消

图 2：打开文件



```
CTRL组合键
Ctrl + A 全选
Ctrl + B 整理 收藏夹
Ctrl + C 复制
Ctrl + D 删除/添加收藏 (在Word中是修改选定字符格式，在IE
Ctrl + E 搜索助理 在Word中是段落居中
Ctrl + F 查找
Ctrl+ G 在Word中是定位
Ctrl + H 历史记录 在Word中是替换
Ctrl + I 收藏夹
Ctrl + J IE7.0里的源
Ctrl + K 在Word中是插入链接
Ctrl + L =Ctrl+O 打开
```

图 3：打开结果

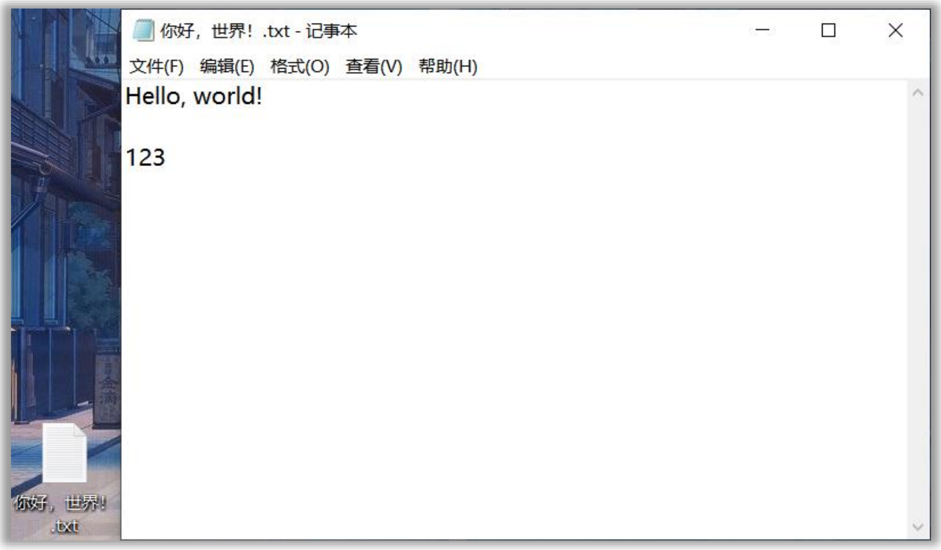


```
Hello, world!

123
```

```
java.exe" "
```

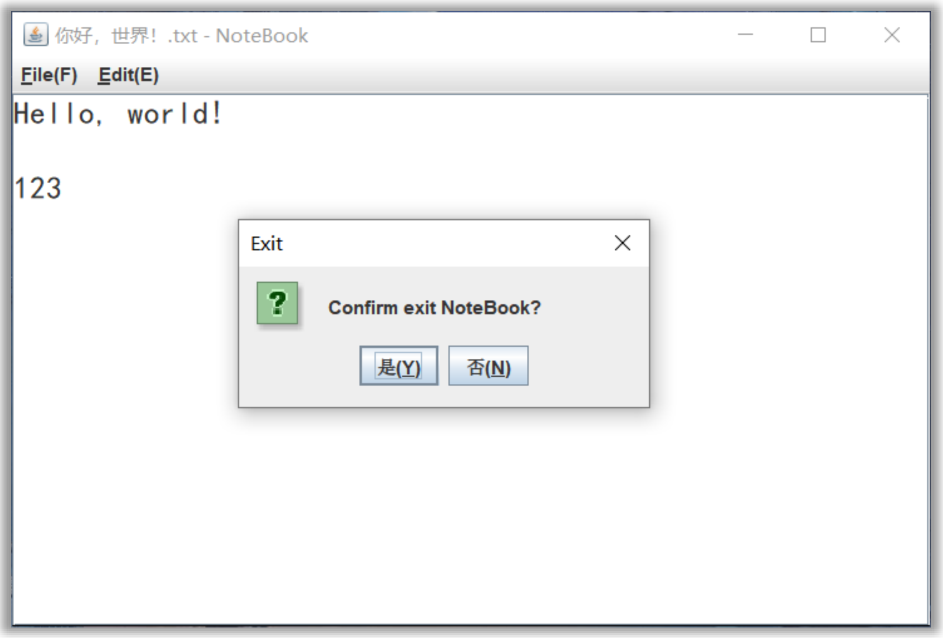
文件名: 你好，世界！

图 4：保存文件



```
Hello, world!

123
```

图 5：保存结果



```
Hello, world!

123
```

Confirm exit NoteBook?

图 6：退出程序