学生学号	0121910870616	实验课成绩	

武汉理工大学

学 生 实 验 报 告 书

实验课程名称	JAVA 语言程序设计 D
开 课 学 院	计算机科学与技术学院
指导教师姓名	祁明龙
学 生 姓 名	邓晟淼
学生专业班级	软件 1902

2020 -- 2021 学年 第 一 学期

实验项目名称	构造前 N 行杨辉三角形			实验成绩	
实 验 者	邓晟淼	专业班级	软件 1902	组　　别	无
同 组 者	无			实验日期	2020 年 10 月 17 日

第一部分：实验预习报告

实验目的：

构造前 N 行杨辉三角形，要求保存杨辉三角形的二维数组以及将 N 声明为类私有成员变量，系统的生成构造方法，getters 和 setters；

实验意义：

熟悉 java 的基本语法以及类的使用；

第二部分：实验过程记录

实验算法：

每一行的第一个数与最后一个数均为 1，当行数大于 2 的时候，所有非第一与最后一个元素的值 tri[m][n]，均等于其上的两个值的和 tri[m-1][n] + tri[m-1][n-1];可以通过此算法构造出杨辉三角形；

实验核心代码：

```
//TriYangHui.java
package yanghui;
public class TriYangHui {
    private int N; //杨辉三角形的行数
    private int[][] Tri;   //储存杨辉三角的二维数组
    public TriYangHui() {
        this.N = 0; //默认是 0 行
    }
    public TriYangHui(int Line) {
        if(Line <= 0) {
            if(Line < 0) System.out.println("不能设置负数行数，已将行数设置成 0!");
            this.N = 0;
            return;
        }
        this.N = Line;
        CreateTri();
    }
```

```
    public void setN(int Line) {
        this.N = Line;
        CreateTri();
    }
    public int getN() {
        return this.N;
    }
    public void setTri(int[][] temp) {
        this.Tri = temp;
    }
    public int[][] getTri() {
        return this.Tri;
    }
    protected void CreateTri() {
        Tri = new int[N][N];
        for(int i = 0;i < N;i++) {
            for(int j = 0;j < i + 1;j++) {
                if(i == j || j == 0) {
                    Tri[i][j] = 1; //开头元素或行尾元素赋值 1;
                    continue;
                }
                if(i >= 2) {
                    Tri[i][j] = Tri[i - 1][j - 1] + Tri[i - 1][j];
                }
            }
        }
    }
    public void ShowTri() {        //打印表格
        for(int[] temp:this.Tri) {
             for(int num:temp)
                if(num != 0) System.out.printf("%5d", num);
             System.out.println();
        }
    }
}

//demo.java
package demo;
import yanghui.*;
public class demo {
    public static void main(String[] args) {
        TriYangHui A,B;
        A = new TriYangHui(5);
        B = new TriYangHui();
```

```
        B.setN(15);
        System.out.println("杨辉三角形 A 的行数为" + A.getN());
        System.out.println("杨辉三角形 A:");
        A.ShowTri();
        System.out.println();

        System.out.println("杨辉三角形 B 的行数为" + B.getN());
        System.out.println("杨辉三角形 B:");
        B.ShowTri();
    }
}
```

第三部分 结果与讨论

一、实验结果分析

```
杨辉三角形A的行数为5
杨辉三角形A:
    1
    1    1
    1    2    1
    1    3    3    1
    1    4    6    4    1
```

```
杨辉三角形B的行数为15
杨辉三角形B:
    1
    1    1
    1    2    1
    1    3    3    1
    1    4    6    4    1
    1    5   10   10    5    1
    1    6   15   20   15    6    1
    1    7   21   35   35   21    7    1
    1    8   28   56   70   56   28    8    1
    1    9   36   84  126  126   84   36    9    1
    1   10   45  120  210  252  210  120   45   10    1
    1   11   55  165  330  462  462  330  165   55   11    1
    1   12   66  220  495  792  924  792  495  220   66   12    1
    1   13   78  286  715 1287 1716 1716 1287  715  286   78   13    1
    1   14   91  364 1001 2002 3003 3432 3003 2002 1001  364   91   14    1
```

二、小结、建议及体会

实验讨论：

通过这个作业，我了解了 java 的各种语法的书写格式，以及构造器的使用方法，加深了对 java 程序设计的理解，并且学习了如何使用 eclipse，相比于之前在命令指示符中的手动编译运行，理解了 IDE 的强大之处，在之后的 coding 过程中我一定会更加努力，学习更多的知识。

实验项目名称	矩阵和向量的乘积			实验成绩	
实 验 者	邓晟淼	专业班级	软件 1902	组　　别	无
同 组 者	无			实验日期	2020 年 10 月 17 日

第一部分：实验预习报告

实验目的：

矩阵和向量的乘积，要求矩阵和两个向量作为类的私有成员变量，系统的生成构造方法，getters 和 setters

实验意义：

掌握 java 的基本语法，IDE 的使用，二维数组的使用

第二部分：实验过程记录

实验算法：

首先我设置了构造方法允许使用已经存在的数组作为实参进行传递，用以构造矩阵和行列向量。其次，为了满足一般情况，我将 getter 和 setter 方法进行了重载，允许不提供默认参数，而是在构造之中进行赋值。下面代码将演示两种方法的使用。

实验核心代码：

```
//matrix.java
package matrix;
import java.util.Scanner;
public class matrix {
    private int[][] mat;          //矩阵
    private int[] row;       //行向量
    private int[] column;        //列向量

    public matrix(int m,int n) {          //m,n 表示行数和列数
        mat = new int[m][n];
        row = new int[m];
        column = new int[n];
    }
    /*
     * 私有成员变量 mat 的 setter 和 getter 方法
     * */
    public void setmat() {            //需要输入数据
        Scanner input = new Scanner(System.in);
        int temp;
        System.out.println("请输入数据构造矩阵:");
```

```
        for(int i = 0;i < mat.length;i++) {
            for(int j = 0;j < mat[i].length;j++) {
                System.out.print("请输入第"+ (i + 1) +"行第"+ (j + 1) +"列的数:");
                temp = input.nextInt();
                this.mat[i][j] = temp;
            }
        }
    }
    public void setmat(int[][] TempMat) {        //重载 setmat()；需要提供 2 维数组矩阵
        int i = 0,j = 0;
        for(int[] TempRow:TempMat) {
            for(int TempNum:TempRow) {
                this.mat[i][j] = TempNum;
                j++;
            }
            j = 0;
            i++;
        }
    }
    public int[][] getmat() {
        return this.mat;
    }
    /*
     * 私有成员变量 row 的 setter 和 getter 方法
     * */
    public void setrow() {            //需要输入数据
        Scanner input = new Scanner(System.in);
        System.out.println("请输入数据构造行向量:");
        int temp;
        for(int i = 0;i < row.length;i++) {
            System.out.print("请输入第"+ (i + 1) +"列的数:");
            temp = input.nextInt();
            this.row[i] = temp;
        }
    }
    public void setrow(int[] temp) {          //重载 setrow()；需要提供一维行矩阵
        int i = 0;
        for(int num:temp) {
            this.row[i++] = num;
        }
    }
    public int[] getrow() {
        return this.row;
    }
```

```
    /*
     * 私有成员变量 column 的 setter 和 getter 方法
     * */
    public void setcolumn() {       //需要输入数据
        Scanner input = new Scanner(System.in);
        System.out.println("请输入数据构造列向量:");
        int temp;
        for(int i = 0;i < column.length;i++) {
            System.out.print("请输入第"+ (i + 1) +"行的数:");
            temp = input.nextInt();
            this.column[i] = temp;
        }
    }
    public void setcolumn(int[] temp) {          //重载 setcolumn()；需要提供一维行矩
阵
        int i = 0;
        for(int num:temp) {
            this.column[i++] = num;
        }
    }
    public int[] getcolumn() {
        return this.column;
    }
    /*
     * 展示矩阵与行列向量的乘积
     * */
    public int[] showMatCol() {         //输出矩阵乘以列向量,并返回结果
        int[] temp = new int[this.mat.length];        //设置 temp 数组以储存相乘的结
果;
        int sum = 0;
        System.out.println("得到的列向量为:");
        for(int i = 0;i < this.mat.length;i++) {
            for(int j = 0;j < this.column.length;j++) {
                sum += mat[i][j] * column[j];
            }
            System.out.println(sum);
            temp[i] = sum;
            sum = 0;
        }
        return temp;
    }
    public int[] showRowMat() {       //输出行向量乘以矩阵,并返回结果
        int[] temp = new int[this.mat[0].length];          //设置 temp 数组以储存相乘的
结果;
```

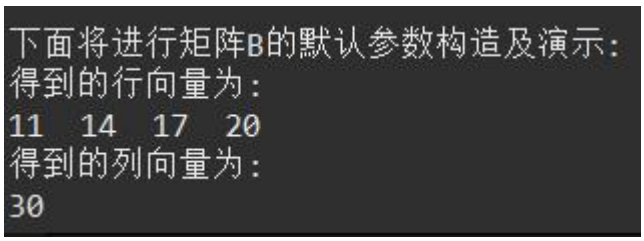
```
            int sum = 0;
            System.out.println("得到的行向量为:");
            for(int i = 0;i < this.mat[0].length;i++) {
                for(int j = 0;j < this.row.length;j++) {
                    sum += mat[j][i] * row[j];
                }
                System.out.print(sum + "    ");
                temp[i] = sum;
                sum = 0;
            }
            System.out.println();
            return temp;
        }
}
//demo.java
package demo;
import matrix.*;
public class demo {

    public static void main(String[] args) {
            int[][] B_mat = {{1,2,3,4},{5,6,7,8}};
            int[] B_row = {1,2};
            int[] B_column = {1,2,3,4};
            matrix A,B;
            /*
             * 输入构造矩阵
             * */
            System.out.println("下面将进行矩阵 A 的输入构造及演示:");
            A = new matrix(3,5);
            A.setmat();
            A.setrow();
            A.setcolumn();
            A.showRowMat();
            A.showMatCol();
            System.out.println();
            /*
             * 默认参数
             * */
            System.out.println("下面将进行矩阵 B 的默认参数构造及演示:");
            B = new matrix(2,4);
            B.setmat(B_mat);
            B.setrow(B_row);
            B.setcolumn(B_column);
            B.showRowMat();
```

```
            B.showMatCol();
        }
}
```

第三部分 结果与讨论

一、实验结果分析

(演示输入矩阵的元素进行构造)



(演示传递二维数组进行构造)

二、小结、建议及体会

实验讨论：

通过这个作业，我了解了 java 的各种语法的书写格式，以及构造器的使用方法，加深了对 java 程序设计的理解，学习了二维数组的构造使用，并且学习了如何使用 eclipse，相比于之前在命令指示符中的手动编译运行，理解了 IDE 的强大之处，在之后的 coding 过程中我一定会更加努力，学习更多的知识。