金字塔（pyramid）练习：

//使用for 循环完成下面的案例  
//请编写一个程序，可以接收一个整数,表示层数（totalLevel），打印出金字塔。



这是金字塔的代码

import java.util.Scanner;  
public class MultiplyCycleExercise02 {  
 public static void main(String[]args){  
 //金字塔练习 见wor7.18  
 //使用for 循环完成下面的案例  
 //请编写一个程序，可以接收一个整数,表示层数（totalLevel），打印出金字塔。  
 //思路分析（化繁为简）; 先死后活(改常量为变量)  
 /\*  
 第一步：先打印一个矩形  
 \*\*\*\*\*  
 \*\*\*\*\*  
 \*\*\*\*\*  
 \*\*\*\*\*  
 \*\*\*\*\*  
 第二步：打印半个金字塔  
 \* //第1层1个\*  
 \*\* //第2层2个\*  
 \*\*\* //第3层3个\*  
 \*\*\*\* //第4层4个\*  
 \*\*\*\*\* //第5层5个\*  
 第三步：打印整个金字塔  
 \* //第1层1个\* ---2\*层数-1 ---前面有(层数-1)个空格  
 \*\*\* //第2层3个\*  
 \*\*\*\*\* //第3层5个\*  
 \*\*\*\*\*\*\* //第4层7个\*  
 \*\*\*\*\*\*\*\*\* //第5层9个\*  
 第四步：打印空心金字塔  
 \* //第1层1个\* --- 中间0个空格 ---2\*层数-1 ---前面有(层数-1)个空格  
 \* \* //第2层2个\* --- 中间1个空格 ---2\*层数-3个空格  
 \* \* //第3层2个\* --- 中间2个空格  
 \* \* //第4层2个\* --- 中间5个空格  
 \*\*\*\*\*\*\*\*\* //第5层9个\* --- 中间0个空格  
 第五步：先死后活  
 把层数做成一个变量 storey：楼层  
 \*/  
 Scanner myScanner = new Scanner(System.*in*);  
 System.*out*.println("请输出金字塔的层数");  
 int storey = myScanner.nextInt();  
 for (int i = 1; i <=storey ; i++) { //i表示层数  
 //在输出\*之前 先输出 对应的空格=层数-1  
 for (int j = 1; j <= (storey-i) ; j++) {  
 //System.out.print("\s");这两都可以表示空格  
 System.*out*.print(" ");  
 }  
 //控制打印每层\*的个数  
 for (int j = 1; j <= (i\*2-1) ; j++) {  
 if (j==1 || j==(i\*2-1) || i==storey) {  
 System.*out*.print("\*");  
 }else {  
 System.*out*.print(" ");  
 }  
 }  
 //每打完一层\*后 就要换行  
 //System.out.println("\n"); 这两都可以表示换行  
 System.*out*.println("");  
 }  
 }  
}

这是自己写的空心菱形的代码：

import java.util.Scanner;  
  
public class MultiplyCycleExerciseOfDiamond {  
 public static void main(String[]args){  
 //空心菱形练习 见wor7.18  
 /\*转化：  
 变为空心的  
 第一步：化繁为简 先打印一个5\*10层的矩形  
 一直转化 然后去把常量改为变量  
 设置层数菱形 由于输入的可能是奇数或偶数 中间层数会有不同的变化 我这里用  
 if设置了一个中间层数的变量 来进行纠正  
 总结：第一次写这么多代码并且不断去优化 有成就感 但还要继续加快学习  
 \*/  
 Scanner myScanner = new Scanner(System.*in*);  
 System.*out*.println("请输入菱形的层数");  
 int storey = myScanner.nextInt();  
 int middleStorey =0;  
 if (storey%2==0){  
 middleStorey = (storey/2);  
 }else {  
 middleStorey = (storey/2)+1;  
 }  
 for (int i = 1; i <= storey ; i++) { //i为层数 i=1时 进入第一层为1个\* i=10为1个\* i=5时为5个\*  
 if (i <= middleStorey){  
 for (int j = 1; j <=(middleStorey-i) ; j++) {  
 System.*out*.print(" ");  
 }  
 for (int j = 1; j <= (i\*2-1); j++) {//j\*i表示打几个\*  
 if (j==1 || j==(i\*2-1) ){  
 System.*out*.print("\*");  
 }else System.*out*.print(" ");  
 }  
 System.*out*.println("");  
 }else if (i > middleStorey){  
 for (int j = 1; j <=(i-middleStorey) ; j++) {  
 System.*out*.print(" ");  
 }  
 for (int j = 1; j <= ((storey\*2)-(i\*2-1)); j++) {//j\*i表示打几个\*  
 if (j==1 || j==((storey\*2)-(i\*2-1)) ){  
 System.*out*.print("\*");  
 }else System.*out*.print(" ");  
 }  
 System.*out*.println("");  
 }  
 }  
 /\* 解析：  
 第一步：化繁为简 先打印一个5\*10层的矩形  
 转化：  
 \*  
 \*\*  
 \*\*\*  
 \*\*\*\*  
 \*\*\*\*\*  
 \*\*\*\*\*\*  
 \*\*\*\*\*\*\*  
 \*\*\*\*\*\*\*\*  
 \*\*\*\*\*\*\*\*\*  
 \*\*\*\*\*\*\*\*\*\*  
 在转化成下面这样  
 \* 1  
 \*\* 2  
 \*\*\* 3  
 \*\*\*\* 4  
 \*\*\*\*\* 5  
 \*\*\*\* 4 6--4  
 \*\*\* 3 7--3  
 \*\* 2 8--2  
 \* 1 9--1  
 转化：  
 \* 1 4空  
 \*\*\* 2 3空  
 \*\*\*\*\* 3 2空  
 \*\*\*\*\*\*\* 4 1空  
 \*\*\*\*\*\*\*\*\* 5 0空 5--9 2\*层数-1  
  
 \*\*\*\*\*\*\* 4 1空 6--7 6--11 最高层\*2-（层数\*2-1）  
 \*\*\*\*\* 3 2空 7--5 7--13  
 \*\*\* 2 3空 8--3 8--15  
 \* 1 4空 9--1 9--17  
  
 \*/  
 }  
}