```
import javax.swing.JOptionPane;
/**
 * @author      Benjamin Tu
 * @author      Nathan Chen
 * @teacher     Coglianese
 * @period      2
 * @version     12-10-18
 *
 * Driver12 makes star designs
 */
public class Driver12
{
    //Instance variables
    public static int myRows;
    /**
     * Default constructor
     */
    public static void main(String[] args)
    {
        while(true)
        {
            myRows = Integer.parseInt(
                JOptionPane.showInputDialog(
                    "How many rows? (-1 to quit)"));
            if(myRows == -1)
            {
                System.out.println("Dave, this conversation can serve no pur
pose anymore. Goodbye.");
                System.exit(0);
            }
            String message = "What type? (1-4)";
            message += "\n1. Standard";
            message += "\n2. Backward";
            message += "\n3. Pyramid";
            message +=  "\n4. Box with X";
            int type = Integer.parseInt(JOptionPane.showInputDialog(message)
);
            switch(type)
            {
                case 1: standard(myRows);
                break;
                case 2: backward(myRows);
                break;
                case 3: pyramid(myRows);
                break;
                case 4: box(myRows);
                break;
                default: System.out.println("It can only be attributable to
human error.");
```

```
                break;
            }
            System.out.println("That's a very nice rendering, Dave. \nI thin
k you've improved a great deal.");
        }
    }

    /**
     * Makes a design as specified
     *
     * @param       n       Amount of rows
     */
    public static void standard(int n)
    {
        String s="";
        for(int i=1;i<=n;i++){
            s=s+"*";
            System.out.println(s);}
    }

    /**
     * Makes a design as specified
     *
     * @param       n       Amount of rows
     */
    public static void backward(int n)
    {
        for(int i=1;i<=n;i++){
            for(int j=1;j<=n-i;j++)
                System.out.print(" ");
            for(int j=1;j<=i;j++)
                System.out.print("*");
            System.out.println();}
    }

    /**
     * Makes a design as specified
     *
     * @param       n       Amount of rows
     */
    public static void pyramid(int n)
    {
        for(int i=1;i<=n;i++){
            for(int j=1;j<=n-i;j++)
                System.out.print(" ");
            for(int j=1;j<=i;j++)
                System.out.print("*");
            for(int j=1;j<=i-1;j++)
                System.out.print("*");
```

```
            System.out.println();}
    }

    /**
     * Makes a design as specified
     *
     * @param       n       Amount of rows
     */
    public static void box(int n)
    {
        for (int row = 1; row <= n; row++){
            for (int col = 1; col <= n; col++)
                if((col==1)||(col==n)||(row==1)||(row==n)||(row==col)||(row+
col-1==n))
                    System.out.print("*");
                else
                    System.out.print(" ");
            System.out.println();
        }
    }
}
```