```java
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package lab4.solution;

/**
 *
 * @author vanting
 */
public class FindShape {

    public static void main(String[] args) {
        int pos = max(new Shape[] { new Rectangle(3, 1), new Rectangle(2, 2), new
Triangle(5) });

        System.out.println("The biggest shape in [" + pos + "].");
    }

    public static int max(Shape[] shapes) {
        int max = 0;
        int maxSize = 0;

        for (int i = 0; i < shapes.length; i++) {
            int size = 0;
            if (shapes[i] instanceof Rectangle) {
                Rectangle r = (Rectangle) shapes[i];
                size = r.getWidth() * r.getHeight();
            } else if (shapes[i] instanceof Triangle) {
                Triangle t = (Triangle) shapes[i];
                size = (1 + (2 * t.getHeight() - 1)) * t.getHeight() / 2;
            }

            if (size > maxSize) {
                max = i;
                maxSize = size;
            }
        }
        return max;
    }

}
```

```java
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package lab4.solution;

/**
 * Rectangle.java
 *
 * A rectangle to be drawn using character in the middle (horizontally) of the
 * screen. Object-oriented style.
 *
 * @author vanting
 */
class Rectangle extends Shape {

    // === instance variables ===
    /**
     * Width of this Rectangle
     */
    private int width;
    /**
     * Height of this Rectangle
     */
    private int height;

    /**
     * Constructor with given width and height.
     */
    public Rectangle(int width, int height) {
        this.width = width;
        this.height = height;
    }

    // === static methods (utility methods) ===
    /**
     * Retrieves the width.
     *
     * @return the width.
     */
    public int getWidth() {
        return width;
    }

    /**
     * Changes the width.
     *
     * @param width the new width to set.
     */
    public void setWidth(int width) {
        this.width = width;
    }

    /**
     * Retrieves the height.
     *
     * @return the height.
     */
    public int getHeight() {
        return height;
    }

    /**
     * Changes the height.
     *
     * @param height the new height to set.
     */
    public void setHeight(int height) {
        this.height = height;
    }

    /**
     * Draws this Rectangle to the screen.
     */
    @Override
    public void draw() {
        for (int i = 0; i < height; i++) {
            drawChars(' ', MIDDLE - width / 2); // leading spaces
            // drawChars(drawingChar, width);
            drawChars(getDrawingChar(), width);
            System.out.println();
        }
    }

    /**
     * Gets a String representation of this Rectangle.
     *
     * @return a String representation of this Rectangle.
     */
    @Override
    public String toString() {
        return String.format("Rectangle (width: %d, height: %d, char: %c)", width,
height, getDrawingChar());
    }

    /**
     * The main method, creating a few Rectangles and testing the methods.
     */
    public static void main(String[] args) {
        Rectangle r1 = new Rectangle(1, 1);
        System.out.println(r1);
        r1.draw();

        Rectangle r2 = new Rectangle(1, 1);
        r2.setWidth(3);
        r2.setHeight(3);
        System.out.println(r2);
        r2.draw();

        Rectangle r3 = new Rectangle(10, 5);
        r3.setDrawingChar('#');
        System.out.println(r3);
        r3.draw();
    }
}

```

```java
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package lab4.solution;

/**
 * Shape.java
 *
 * A shape class to provide basic configuration of a shape.
 *
 * @author vanting
 */
public class Shape {

    /**
     * Character used to draw this Rectangle, default to '*'
     */
    private char drawingChar = '*';
    /**
     * The middle column on the screen
     */
    protected static final int MIDDLE = 40;

    /**
     * Draws a number of characters horizontally.
     *
     * @param ch  the character to draw
     * @param num the number of characters to draw
     * @remark This is a static method, as it is not specific to a particular
     *         instance of Rectangle.
     */
    protected static void drawChars(char ch, int num) {
        for (int i = 0; i < num; i++) {
            System.out.print(ch);
        }
    }

    /**
     * Retrieves the drawing character.
     *
     * @return the drawing character.
     */
    public char getDrawingChar() {
        return drawingChar;
    }

    /**
     * Changes the drawing character.
     *
     * @param drawingChar the new drawing character.
     */
    public void setDrawingChar(char drawingChar) {
        this.drawingChar = drawingChar;
    }

    /**
     * Dummy method with empty body. Is not invoked in general, but shall be
     * overriden by subclasses to draw the current shape to the screen depending on
     * the actual type represented by the subclass.
     */
    public void draw() {
        // should be implemented by subclass.
    }
}

```

```java
package lab4.solution;

import java.util.Scanner;

public class TestShape {

    /**
     * The main method, creating a few Rectangles and Triangle for testing.
     *
     * @param args
     */
    public static void main(String[] args) {

        // If we don't apply polymorphism, we may create 2 arrays...
        // Rectangle r[] = new Rectangle[NUM];
        // Triangle t[] = new Triangle[NUM];
        Scanner scanner = new Scanner(System.in);
        System.out.print("How many shapes to create? ");
        int num = Integer.parseInt(scanner.next());
        Shape shapes[] = new Shape[num];

        System.out.printf("Creating %d shapes.\n", shapes.length);
        for (int i = 0; i < shapes.length; i++) {
            System.out.printf("%d) Rectangle or Triangle ? (\"R\" or \"T\"): ", i +
1);
            String type = scanner.next();
            if (type.equalsIgnoreCase("R")) {
                System.out.printf("Enter the rectangle's width height (integer
integer): ");
                int width = scanner.nextInt();
                int height = scanner.nextInt();
                shapes[i] = new Rectangle(width, height);
            } else if (type.equalsIgnoreCase("T")) {
                System.out.printf("Enter the triangle's height (integer): ");
                int height = scanner.nextInt();
                shapes[i] = new Triangle(height);
            }
        }

        System.out.print("Enter drawing character for all shapes: ");
        String ch = scanner.next();

        // Now, we don't need to know the exact types inside the array.
        for (Shape s : shapes) {
            s.setDrawingChar(ch.charAt(0));
            System.out.println(s);
            s.draw(); // polymorphic
        }

        // For Q4.
        int pos = FindShape.max(shapes);
        System.out.println("The biggest shape is at [" + pos + "]:");
        System.out.println(shapes[pos]);

    }
}

/*
 * Reference example for creating array of objects
 *
 * int x[] = new int[10]; x[0] = 1;
 *
 * Shape s[] = new Shape[10]; s[0] = new Rectangle(1, 2);
 *
 */

```

```
/*
 * To change this template, choose Tools | Templates
 * and open the template in the editor.
 */
package lab4.solution;

/**
 * A triangle. Object-oriented style.
 *
 * @author vanting
 */
class Triangle extends Shape {

    private int height;

    public Triangle(int height) {
        this.height = height;
    }

    public int getHeight() {
        return height;
    }

    public void setHeight(int height) {
        this.height = height;
    }

    @Override
    public void draw() {
        for (int i = 0; i < height; i++) {
            drawChars(' ', MIDDLE - i); // print leading space
            drawChars(getDrawingChar(), 2 * i + 1); // print a line of Triangle
            System.out.println(); // print end-of-line
        }
    }

    @Override
    public String toString() {
        return String.format("Triangle (height: %d, char %c)", height,
getDrawingChar());
    }

    public static void main(String[] args) {
        Triangle t1 = new Triangle(5);
        System.out.println(t1);
        t1.draw();

        Triangle t2 = new Triangle(10);
        t2.setDrawingChar('@');
        System.out.println(t2);
        t2.draw();
    }

}

```