Lösningsförslag till övning 4.

Uppgift 1

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
public class Reaktion extends JFrame implements ActionListener {
    private JButton tryck;
    private JLabel talLabel;
    private long start;
    private boolean first = true;

    public Reaktion() {
        tryck = new JButton("TRYCK");
        talLabel = new JLabel("", JLabel.CENTER);
        getContentPane().setLayout(new GridLayout(2, 1, 5, 5));
        getContentPane().add(talLabel);
        getContentPane().add(tryck);
        tryck.setBackground(Color.yellow);
        talLabel.setBackground(Color.white);
        talLabel.setOpaque(true);
        tryck.addActionListener(this);
        Font font = new Font("Times", Font.PLAIN, 36);
        tryck.setFont(font);
        talLabel.setFont(font);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
        setSize(500,150);
        setVisible(true);
    }//konstruktor

    public void actionPerformed(ActionEvent e) {
        if (e.getSource() == tryck) {
            if (first) {
                first = false;
                start = System.currentTimeMillis();
            }
            else {
                long stopp = System.currentTimeMillis();
                first = true;
                long tid = stopp - start;
                String text = "Reaktionstiden är " + tid + " ms!";
                talLabel.setText(text);
            }
        }
    }//actionPerformed

    public static void main(String[] arg) {
        Reaktion v = new Reaktion();

    }//main
}//Reaktion
```

Uppgift 2

```
import javax.swing.*;
import java.awt.*;
public class CircelButton  extends JButton {
    private boolean isBlack = true;
    public CircelButton() {
        setBackground(new Color(0,130,0));
    }//konstruktor

    public void paintComponent(Graphics pen) {
        super.paintComponent(pen);
        if (isBlack)
            pen.setColor(Color.black);
        else
            pen.setColor(Color.white);
        int h = getHeight();
        int w = getWidth();
        pen.fillOval(0,0,w,h);
    }//paintComponent

    public void setWhite() {
        isBlack = false;
    }//setWhite

    public void setBlack() {
        isBlack = true;
    }//setBlack

    public void changeColor() {
        isBlack = !isBlack;
    }//changeColor
}//CircelButton
```

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.util.*;
public class Niner extends JPanel implements ActionListener {
    private CircelButton[][] buttons;
    public Niner() {
        buttons = new CircelButton[3][3];
        setLayout(new GridLayout(3, 3));
        Random rand = new Random();
        for (int r = 0; r < buttons.length; r = r + 1) {
            for (int k = 0; k < buttons[r].length; k = k + 1) {
                buttons[r][k] = new CircelButton();
                buttons[r][k].addActionListener(this);
                if (rand.nextInt(2) == 0)
                    buttons[r][k].setBlack();
                else
                    buttons[r][k].setWhite();
                add(buttons[r][k]);
            }
        }
    }//konstruktor

    public void actionPerformed(ActionEvent e) {
        for (int r = 0; r < buttons.length; r = r + 1) {
            for (int k = 0; k < buttons[r].length; k = k + 1) {
                if (e.getSource() == buttons[r][k])
                    changeColor(r,k);
            }
        }
        repaint();
    }//actionPerformed
```

```
    private void changeColor(int r, int k) {
        buttons[r][k].changeColor();
        if (r == 0 && k == 1 || r == 2 && k == 1) {
            buttons[r][k-1].changeColor();
            buttons[r][k+1].changeColor();
        }
        if (r == 1 && k == 0 || r == 1 && k == 2) {
            buttons[r-1][k].changeColor();
            buttons[r+1][k].changeColor();
        }
        if (r == 0 && k == 0) {
            buttons[r][k+1].changeColor();
            buttons[r+1][k].changeColor();
            buttons[r+1][k+1].changeColor();
        }
        if (r == 0 && k == 2) {
            buttons[r][k-1].changeColor();
            buttons[r+1][k].changeColor();
            buttons[r+1][k-1].changeColor();
        }
        if (r == 2 && k == 0) {
            buttons[r][k+1].changeColor();
            buttons[r-1][k].changeColor();
            buttons[r-1][k+1].changeColor();
        }
        if (r == 2 && k == 2) {
            buttons[r][k-1].changeColor();
            buttons[r-1][k].changeColor();
            buttons[r-1][k-1].changeColor();
        }

        if (r == 1 && k == 1){
            buttons[r][k-1].changeColor();
            buttons[r][k+1].changeColor();
            buttons[r-1][k].changeColor();
            buttons[r+1][k].changeColor();
        }
    }//changeColor

    public static void main(String[] args) {
        JFrame window = new JFrame();
        Niner  inv = new Niner();
        window.setSize(300, 300);
        window.getContentPane().add(inv);
        window.setLocation(50,50);
        window.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        window.setVisible(true);
    }//main
}// Niner
```

Uppgift 3

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
public class RotatingSquare extends JPanel  implements ActionListener{
    private int vinkel= 0;
    private javax.swing.Timer t = new javax.swing.Timer(100, this);
    public RotatingSquare() {
        t.restart();
    }
    public void paintComponent(Graphics pen) {
        super.paintComponent(pen);
        int x0 = getWidth()/2;
        int y0 = getHeight()/2;
        int radie = Math.min(getWidth(), getHeight())/4;
        Polygon p = new Polygon();
        for (int i = 0; i <= 4; i = i + 1) {
            int x1 = x0 + (int) (radie*Math.cos(Math.toRadians(vinkel + i*90)));
            int y1 = y0 + (int) (radie*Math.sin(Math.toRadians(vinkel + i*90)));
            p.addPoint(x1,y1);
        }
        pen.setColor(Color.blue);
        pen.fillPolygon(p);
    }//paintComponent

    public void actionPerformed(ActionEvent e) {
        vinkel = vinkel + 1 % 360;
        repaint();
    }//actionPerformed

    public static void main(String[] args){
        JFrame w  = new JFrame();
        RotatingSquare rs = new RotatingSquare();
        w.add(rs);
        w.setSize(300, 300);
        w.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        w.setVisible(true);
    }//main
}// RotatingSquare
```