

Programming Fundamentals II

Lap8: - GUI + Event
 - Graphic

1.1 Mr.Smile

Lab 8.1 ให้นักศึกษาสร้าง Package และสร้าง Class ชื่อ Lab81DrawSmiley.java นำไปใส่ใน Package ตามที่ได้สร้างไว้ เติมโค้ดให้สมบูรณ์และเขียนโปรแกรมตามตัวอย่างโค้ดด้านล่าง (Class Lab81DrawSmiley สืบทอดมาจาก Class JPanel)
Lab81DrawSmiley.java

```
import java.awt.*;
import javax.swing.*;

public class Lab81DrawSmiley extends JPanel
{
    public void _____( Graphics g)
    {
        super. _____(g);

        //draw face
        g.setColor(Color.YELLOW);
        g.fillOval(10, 10, 200, 200);

        //draw eye
        g.setColor(Color.BLACK);
        g.fillOval(55, 65, 30, 30);
        g.fillOval(135, 65, 30, 30);

        //draw mouth
        g.fillOval(50, 110, 120 , 60);

        //draw smile
        g.setColor(Color.YELLOW);
        g.fillRect(50, 110, 120, 30);
        g.fillOval(50, 120, 120, 40);
    }
}
```

ให้นักศึกษาสร้าง Class มา Test โดยตั้งชื่อว่า Lab81DrawSmileyTest.java ให้ใส่ไว้ใน Package เดียวกัน

Lab81DrawSmileyTest.java

```
import javax.swing.*;

public class Lab81DrawSmileyTest
{
    public static void main(String[] args)
    {

    }
}
```

1.2 วาดรูปกันเถอะ

Lab 8.2 ให้นักศึกษาสร้าง Package และสร้าง Class ชื่อ Lab82PaintPanel.java นำไปใส่ใน Package ตามที่ได้สร้างไว้

Lab82PaintPanel.java

```
import java.awt.*;
import java.awt.event.*;
import java.util.*;
import javax.swing.*;

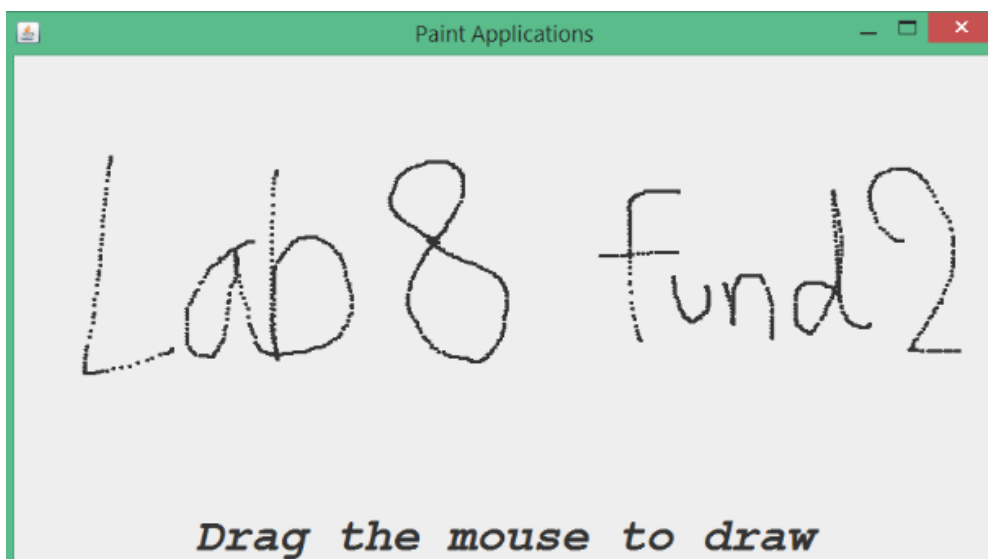
public class Lab82PaintPanel extends JPanel
{
    private final ArrayList<Point> points = new ArrayList<>();

    public PaintPanel()
    {
        addMouseListener(
            new MouseMotionAdapter()
            {
                public void mouseDragged(MouseEvent event)
                {
                    points.add(event.getPoint());
                    repaint();
                }
            }
        );
    }

    public void paintComponent(Graphics g)
    {
        super.paintComponent(g);

        for(Point point : points)
            g.fillOval(point.x, point.y, 4, 4);
    }
}
```

ให้นักศึกษาสร้าง Class มา Test โดยตั้งชื่อว่า Lab82PaintPanelTest.java ให้ใส่ไว้ใน Package เดียวกัน โดยให้มีผลลัพธ์ ดังตัวอย่างรูปที่แสดงข้างล่าง



Lab82PaintPanelTest.java

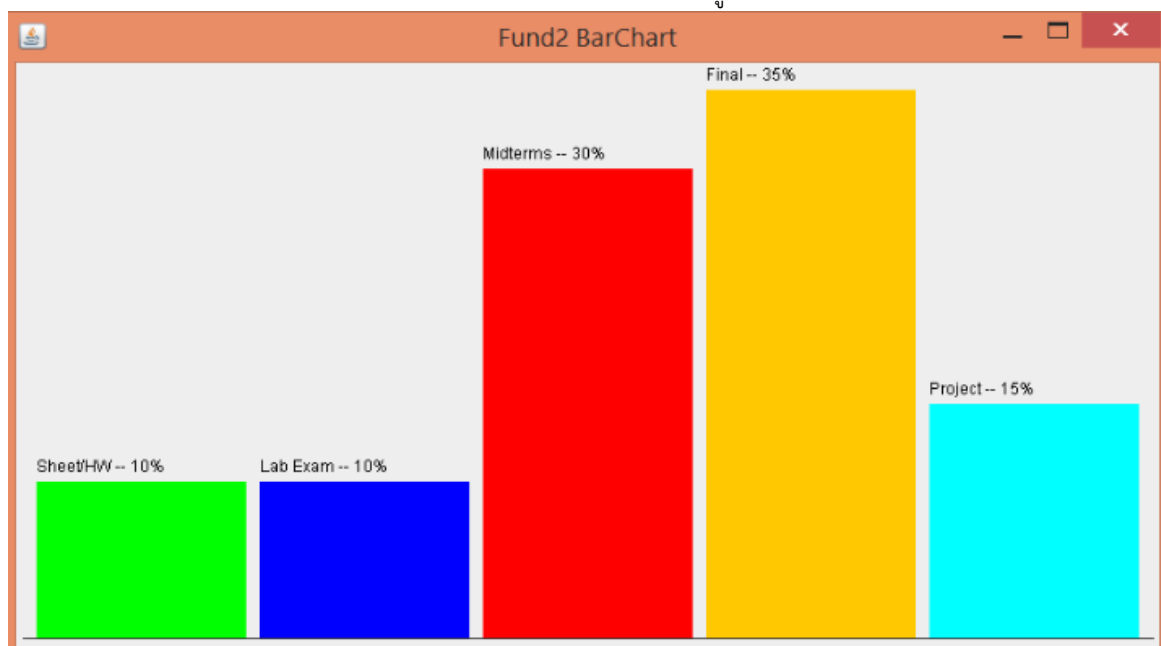
```
import java.awt.*;
import javax.swing.*;

public class Lab82PaintPanelTest
{
    public static void main(String[] args)
    {

    }
}
```

1.3 BarChart

ให้นักศึกษาการทำงาน ของ Class ที่ไว้วาด Bar Chart โดยให้ได้ผลลัพธ์ดังรูปด้านล่าง



Lab 8.3 ให้นักนิสิตสร้าง Package และสร้าง Class ชื่อ Lab83BarChart และเขียน Class ตามที่ได้แสดงตัวอย่างด้านล่าง

Lab83BarChart.java

```
import java.awt.*;
import javax.swing.*;

public class Lab83BarChart extends JPanel
{
    /**Sample data, and data names*/
    private double[] dataValue = {200, 140, 100, 60, 40};
    private String[] dataName = {"Fund1", "Fund2", "Data", "Skill", "Project"};
    private Color[] colors = {Color.green, Color.blue, Color.red,
        Color.orange, Color.cyan, Color.magenta, Color.yellow, Color.pink,
        Color.darkGray};

    public void paintComponent(Graphics g)
    {
        super.paintComponent(g);

        if (dataValue == null) {
            return;
        }

        // Find the maximum value in the data
        double max = dataValue[0];
        for (int i = 1; i < dataValue.length; i++)
        {
            max = Math.max(max, dataValue[i]);
        }

        int barWidth = (int) ((getWidth() - 20.0) / dataValue.length - 10);
        int maxBarHeight = getHeight() - 30;

        g.drawLine(5, getHeight() - 10, getWidth() - 5, getHeight() - 10);

        int x = 15;
        for (int i = 0; i < dataValue.length; i++)
        {
            g.setColor(colors[i % colors.length]);
            int newHeight = (int) (maxBarHeight * dataValue[i] / max);
            int y = getHeight() - 10 - newHeight;
            g.fillRect(x, y, barWidth, newHeight);
            g.setColor(Color.black);
            // Display name if exist
            if ((dataName != null) && (i < dataName.length))
            {
                g.drawString(dataName[i], x, y - 7);
            }
            x += barWidth + 10;
        }

        /**Set data*/
        public void setData(String[] dataName, double[] dataValue)
        {
            this.dataName = dataName;
            this.dataValue = dataValue;
            repaint();
        }
    }
}
```

ให้นักศึกษาสร้าง Class มา Test โดยตั้งชื่อว่า Lab83TestBarChart.java ให้ใส่ไว้ใน Package เดียวกัน โดยให้มีผลลัพธ์ ดังตัวอย่างรูปที่แสดงด้านบน

Lab83TestBarChart.java

```
import java.awt.*;
import javax.swing.*;

public class Lab83TestBarChart extends JFrame
{
    public Lab83TestBarChart()
    {
        Container container = getContentPane();

        container.setLayout(new GridLayout(1, 3, 20, 5));

        Lab83BarChart chart1 = new Lab83BarChart();

        double[] data1 = {10, 10, 30, 35, 15};

        String[] dataName1 = {"Sheet/HW -- 10%", "Lab Exam -- 10%", "Midterms -- 30%", "Final -- 35%", "Project -- 15%"};

        chart1.setData(data1, dataName1);

        container.add(chart1);
    }

    public static void main(String[] args)
    {
        JFrame frame = new JFrame("Fund2 BarChart");

        frame.setTitle("Fund2 BarChart");

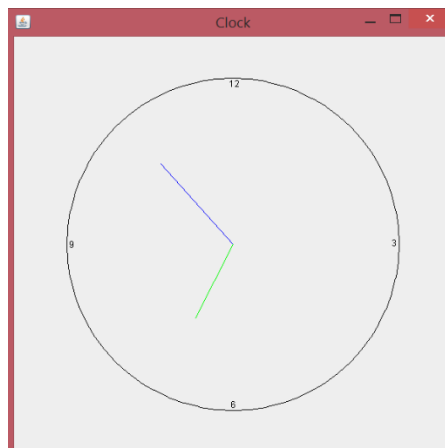
        frame.setSize(858, 480);

        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        frame.setVisible(true);
    }
}
```

1.4 Still Clock

ให้นักศึกษาศึกษาการทำงานของ Class ที่ชื่อว่า นาฬิกา Class Lab84StillClock เป็นคลาสสำหรับแสดงนาฬิกา และถูกเรียกใช้งานโดยคลาส Lab84TestStillClock เมื่อรัน Lab84TestStillClock จะได้ผลลัพธ์ดังแสดงในรูปต่อไปนี้



Lab84StillClock.java (Download ไฟล์ได้จาก <https://goo.gl/r4V3at>)

```
import java.awt.*;
import javax.swing.*;
import java.util.*;

public class Lab84StillClock extends JPanel {
    private int hour;
    private int minute;
    private int second;
    private boolean hourHandVisible = true;
    private boolean minuteHandVisible = true;
    private boolean secondHandVisible = true;

    /** Construct a default clock with the current time */
    public StillClock() {
        setCurrentTime();
    }

    /** Construct a clock with specified hour, minute, and second */
    public StillClock(int hour, int minute, int second) {
        this.hour = hour;
        this.minute = minute;
        this.second = second;
    }

    /** Return hour */
    public int getHour() {
        return hour;
    }

    /** Set a new hour */
    public void setHour(int hour) {
        this.hour = hour;
        repaint();
    }

    /** Return minute */
    public int getMinute() {
        return minute;
    }

    /** Set a new minute */
    public void setMinute(int minute) {
        this.minute = minute;
        repaint();
    }

    /** Return second */
    public int getSecond() {
        return second;
    }

    /** Set a new second */
    public void setSecond(int second) {
        this.second = second;
        repaint();
    }

    /** Draw the clock */
    protected void paintComponent(Graphics g) {
        super.paintComponent(g);

        // Initialize clock parameters
        int clockRadius =
            (int) (Math.min(getWidth(), getHeight()) * 0.8 * 0.5);
        int xCenter = getWidth() / 2;
        int yCenter = getHeight() / 2;

        // Draw circle
        g.setColor(Color.black);
        g.drawOval(xCenter - clockRadius, yCenter - clockRadius,
            2 * clockRadius, 2 * clockRadius);
        g.drawString("12", xCenter - 5, yCenter - clockRadius + 12);
        g.drawString("9", xCenter - clockRadius + 3, yCenter + 5);
        g.drawString("3", xCenter + clockRadius - 10, yCenter + 3);
        g.drawString("6", xCenter - 3, yCenter + clockRadius - 3);

        // Draw second hand
        if (secondHandVisible) {
            int sLength = (int) (clockRadius * 0.8);
            int xSecond = (int) (xCenter + sLength *
                Math.sin(second * (2 * Math.PI / 60)));
            int ySecond = (int) (yCenter - sLength *
                Math.cos(second * (2 * Math.PI / 60)));
            g.setColor(Color.red);
            g.drawLine(xCenter, yCenter, xSecond, ySecond);
        }

        // Draw minute hand
        if (minuteHandVisible) {
            int mLength = (int) (clockRadius * 0.65);
            int xMinute = (int) (xCenter + mLength *
                Math.sin(minute * (2 * Math.PI / 60)));
            int yMinute = (int) (yCenter - mLength *
                Math.cos(minute * (2 * Math.PI / 60)));
            g.setColor(Color.blue);
            g.drawLine(xCenter, yCenter, xMinute, yMinute);
        }
    }
}
```

```

// Draw hour hand
if (hourHandVisible) {
    int hLength = (int)(clockRadius * 0.5);
    int xHour = (int)(xCenter + hLength *
        Math.sin((hour % 12 + minute / 60.0) * (2 * Math.PI / 12)));
    int yHour = (int)(yCenter - hLength *
        Math.cos((hour % 12 + minute / 60.0) * (2 * Math.PI / 12)));
    g.setColor(Color.green);
    g.drawLine(xCenter, yCenter, xHour, yHour);
}
}

public void setCurrentTime() {
    // Construct a calendar for the current date and time
    Calendar calendar = new GregorianCalendar();

    // Set current hour, minute and second
    this.hour = calendar.get(Calendar.HOUR_OF_DAY);
    this.minute = calendar.get(Calendar.MINUTE);
    this.second = calendar.get(Calendar.SECOND);
}

public Dimension getPreferredSize() {
    return new Dimension(200, 200);
}

public boolean isHourHandVisible() {
    return hourHandVisible;
}

public boolean isMinuteHandVisible() {
    return hourHandVisible;
}

public boolean isSecondHandVisible() {
    return secondHandVisible;
}

public void setHourHandVisible(boolean hourHandVisible) {
    this.hourHandVisible = hourHandVisible;
    repaint();
}

public void setMinuteHandVisible(boolean minuteHandVisible) {
    this.minuteHandVisible = minuteHandVisible;
    repaint();
}

public void setSecondHandVisible(boolean secondHandVisible) {
    this.secondHandVisible = secondHandVisible;
    repaint();
}
}

```

ให้นักศึกษาสร้าง Class มา Test โดยตั้งชื่อว่า Lab84TestStillClock.java ให้ใส่ไว้ใน Package เดียวกัน โดยให้มีผลลัพธ์ ดังตัวอย่างรูปที่แสดงด้านบน

Lab84TestStillClock.java

```

import javax.swing.*;
import java.awt.*;

public class Lab84TestStillClock
{
    public static void main(String[] args)
    {
        JFrame frame = new JFrame("Clock");

        _____ clock = new _____;

        clock.setSecondHandVisible(_____);

        _____; // ตั้งเวลาตามปัจจุบัน ของคอมพิวเตอร์

        frame.add(clock);

        frame.setSize(600, 600);

        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

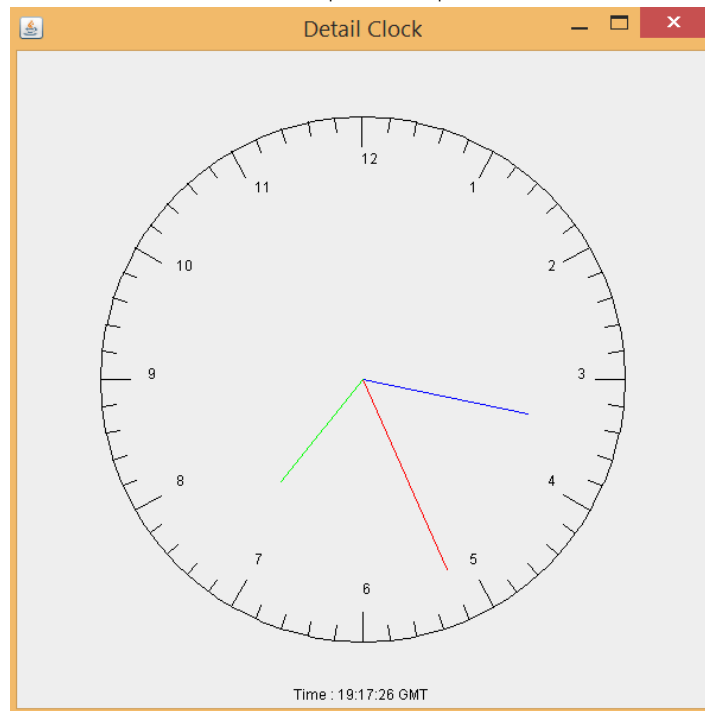
        frame.setVisible(true);

    }
}

```

1.5 Still Clock (Upgrade)

นิสิตจึงศึกษาการทำงานของทั้งสองคลาส เพื่อนำไปพัฒนาให้คลาส Lab84StillClock สามารถแสดงเส้นขีดของนาฬิกา และของวินาทีได้ ดังนั้นเมื่อรันคลาส Lab84TestStillClock จะได้รูปภาพที่มีความละเอียดขึ้นดังแสดงในรูปต่อไปนี้ (กำหนดให้แก้ไขได้ในเฉพาะส่วน Methode paintComponent())



ให้ตั้งชื่อใหม่จาก

Lab84StillClock.java -> Lab85StillClock.java

Lab84TestStillClock.java -> Lab85TestStillClock.java

ให้เขียน โค้ด ที่ได้แก้ไขเพิ่มเติมใน paintComponent() ใน Lab85StillClock.java ลงกล่องคำตอบด้านล่าง

```
/** Draw the clock */  
protected void paintComponent(Graphics g)  
{  
    super.paintComponent(g);
```



```
} // END paintComponent()
```

แบบทดสอบ Lab8

GUI#2

1. Mr. Hat

จากตัวอย่างโค้ดใน Lab81 จงเขียนโปรแกรมซึ่งแสดงรูปหน้ายิ้ม มีหนวด และสวมหมวก นิสิตสามารถเพิ่มองค์ประกอบอื่น ๆ ได้ตามอำเภอใจ

2. วาดรูปกันเถอะ

จากตัวอย่างใน Lab82 จงแก้ไขโปรแกรมให้วาดรูปโดยการลากเส้นต่อจุดแทนการวาดวงกลม

3. วาดรูประดับเซียน

จากตัวอย่างใน Lab82 จงแก้ไขโปรแกรมให้วาดรูปโดยเลือกได้ว่าจะใช้การลากเส้นต่อจุดหรือการวาดวงกลม โดยเปลี่ยนไปมาได้ด้วยการเลือกจาก **radio button** การจัดวาง **panel** สำหรับวาดและ **radio button** แนะนำให้ใช้ **BorderLayout** โดยให้ **panel** อยู่ตำแหน่ง **center** และ **radio button** อยู่ตำแหน่ง **south**

*** ทั้งนี้มีคะแนนพิเศษหากในรูปที่วาดสามารถผสมผสานได้ทั้งวงกลมและเส้น โดยเลือกเปลี่ยนไปมาในแต่ละส่วนที่วาดได้

4. จัดวางให้สวยงาม

จงเขียนโปรแกรมเพื่อจัดวาง GUI ให้ได้ใกล้เคียงกับตัวอย่างด้านล่างนี้ให้มากที่สุด โดยกำหนดให้เมื่อคลิกที่ **button** ต่าง ๆ จะเป็นเพียงการแสดงข้อความของปุ่มนั้นบน **text field** ด้านบน เช่น คลิกปุ่ม **Bck** ก็แสดงข้อความว่า **Bck** บน **text field** ยกเว้นปุ่ม **Cls** ซึ่งกำหนดให้ลบข้อความบน **text field** การจัดวางอาจต้องใช้ทั้ง **BorderLayout** และ **GridLayout** ร่วมกัน ส่วนที่ว่างไม่มี button อาจใช้ **panel** เปล่า ๆ วางแทนได้

