

## แบบฝึกหัดปฏิบัติการคาบที่ 9: GUI and Event Handling

### คำสั่ง

1. ให้ศึกษาหลักการ GUI ต่อไปนี้
  - 1.1. คลาสที่ทำหน้าที่ในการจัดการเกี่ยวกับหน้าจอหรือ User interface จะอยู่ในแพ็คเกจของจาวาที่ชื่อ Abstract Windows Toolkit (AWT) ทุก ๆ platform ที่รันภาษาจาวา components ของ package AWT จะถูกเรียกอัตโนมัติโดยการเลือกคอมโพเนนต์ที่ตรงกับระบบปฏิบัติการแต่ละประเภท
  - 1.2. แพ็คเกจ AWT จะช่วยในการสร้างโปรแกรม GUI ประเภท Look and Feel ที่ขึ้นอยู่กับแพลตฟอร์มที่ใช้งาน ในแพ็คเกจ java.awt มีคลาสและอินเตอร์เฟซที่สำคัญดังนี้ Component Container LayoutManager Graphics Color Font
  - 1.3. การเขียนโปรแกรม GUI นั้นจะเป็นการสร้างออบเจกต์ต่าง ๆ ที่เป็นออบเจกต์ของคลาสที่เป็นส่วนประกอบกราฟฟิค (Graphic component) โดยคลาสที่เป็นส่วนประกอบกราฟฟิคจะสืบทอดมาจากคลาสที่ชื่อว่า Component
  - 1.4. คลาสที่เป็น Subclass ของคลาส Component จะแบ่งเป็นสองกลุ่มคือ
    - คลาสที่เป็นคลาสประเภท Container เป็นคลาสที่ใช้ในการใส่ส่วนประกอบกราฟฟิคต่าง ๆ
    - คลาสที่เป็นส่วนประกอบกราฟฟิคอื่น ๆ เช่น Button Choice และ List เป็นต้น
  - 1.5. ใน Class Component จะมี method ที่ใช้วาด object ต่าง ๆ ที่สร้างจาก class นี้ได้แก่ paint () และ repaint () ในส่วนของแพ็คเกจ Swing จะมีคลาส JComponent เป็น super-class ของ Swing component
  - 1.6. ส่วน Class Container เป็นส่วนของพื้นที่ที่ใช้ในการแสดงผลทาง graphics เมธอดที่ใช้กันบ่อย ได้แก่
    - add () สำหรับใช้ในการเพิ่ม graphical object
    - setLayout () สำหรับใช้ในการกำหนดโครงสร้างและช่วยในการกำหนดตำแหน่งและขนาดของ components
  - 1.7. โปรแกรม GUI จะต้องมีการสร้างออบเจกต์ของคลาสประเภท Container อย่างน้อย 1 ออบเจกต์ขึ้นมาก่อน เพื่อใช้ในการใส่ออบเจกต์ของคลาสที่เป็นส่วนประกอบกราฟฟิคอื่น ๆ คลาสประเภท Container ที่อยู่ในแพ็คเกจ AWT มีดังนี้ Frame Panel Dialog Applet

### การใช้ Panels เป็น sub-Containers

1. Panels ทำหน้าที่เป็น sub-containers สำหรับรวมกลุ่มของ user interface components ใน container ตัวอื่น เราสามารถรวม user interface components ใน panels ก่อนที่จะเอาไปใส่ frame หรือสามารถเพิ่มหลาย ๆ panel ใน panel ก็ได้ ในการ add ตัวคอมโพเนนต์ลงใน JFrame เราสามารถเพิ่มใน content pane ของ JFrame การเพิ่มคอมโพเนนต์ ลงใน panel เราสามารถเพิ่มได้โดยตรงลงใน panel โดยการใส่เมธอด add

2. การใช้งาน Panel สามารถใช้ new JPanel() สำหรับสร้าง panel โดยมี Layout manager คือ FlowLayout หรือจะกำหนด Layout manager ตามความต้องการโดยใช้ new JPanel(LayoutManager) หลังจากนั้นสามารถใช้ add(Component) สำหรับเพิ่ม component ลงไปที่ panel โดยใช้คำสั่ง

- ▶ `JPanel p = new JPanel();`
- ▶ `p.add(new JButton("OK")); import javax.swing.*;`

### การใช้ JFrame เป็น Containers

```
public class MyFrameWithComponents {
    public static void main(String[] args) {
        JFrame frame = new JFrame("MyFrameWithComponents");
        JButton jb1OK = new JButton("OK");
        frame.getContentPane().add( new JButton("OK"));
        frame.add(jb1OK);
        frame.setSize(400, 300);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setLocationRelativeTo(null); // Center the frame
        frame.setVisible(true);
    }
}
```

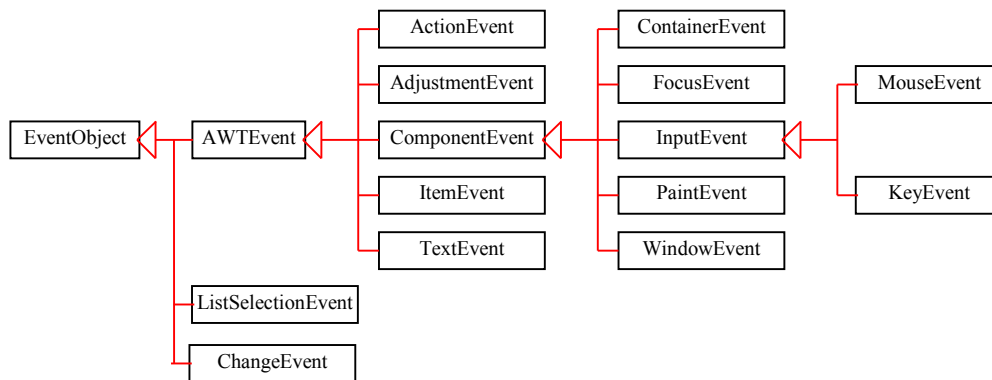
## 2. ให้ศึกษาหลักการการจัดการเหตุการณ์ต่อไปนี้

การเขียนโปรแกรมในรูปแบบ Procedural programming จะประมวลผลโปรแกรมตามลำดับของการเขียน (procedural order) ส่วนการเขียนโปรแกรมในรูปแบบ Event-driven programming จะประมวลผลเมื่อมีเหตุการณ์เกิดขึ้น

### เหตุการณ์ (Events)

เหตุการณ์(Event) เป็นสถานการณ์ที่เกิดขึ้นในขณะรันโปรแกรม เช่น การใช้เมาส์หรือคีย์บอร์ดติดต่อกับโปรแกรม GUI การเกิดเหตุการณ์ในโปรแกรมภาษาจาวาจะเป็นการสร้างวัตถุของคลาสประเภท Event ชนิดต่าง ๆ ขึ้นมาตามประเภทของเหตุการณ์ เช่น

เมื่อเลื่อนเมาส์ในเฟรมจะเกิดวัตถุของคลาส MouseEvent ขึ้นมา  
 เมื่อปิดเฟรมจะเกิดวัตถุของคลาส WindowEvent ขึ้นมา  
 เมื่อกดปุ่มที่อยู่ในเฟรมจะเกิดวัตถุของคลาส ActionEvent ขึ้นมา  
 เมื่อพิมพ์ข้อความใน TextField จะเกิดวัตถุของคลาส KeyEvent ขึ้นมา  
 เหตุการณ์ที่เกิดกับ OS เช่น เหตุการณ์ของ timer



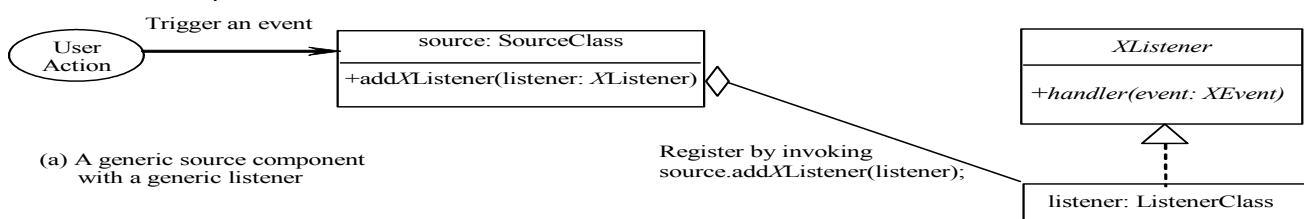
### องค์ประกอบของ Event

Event คือวัตถุที่เกิดขึ้นตามประเภทของเหตุการณ์

Event Source คือ ส่วนที่ทำให้เกิดเหตุการณ์

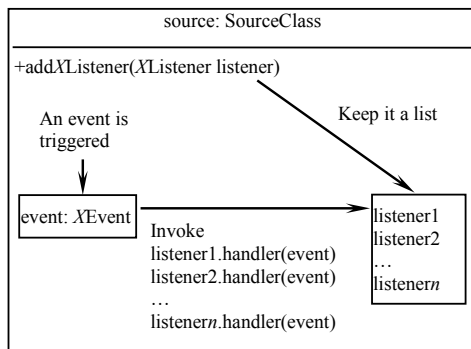
Event Handler คือวัตถุที่ทำหน้าที่จัดการกับเหตุการณ์ที่เกิดขึ้นโดยมีเมธอดที่จะรับวัตถุชนิด Event ดังกล่าว และมีคำสั่งในการจัดการกับเหตุการณ์เพื่อโต้ตอบกับผู้ใช้ เช่น การกดเมาส์จะมี MouseListener เป็นตัวรับฟังเหตุการณ์ และมีเมธอดที่จะทำงานภายในคลาสดังกล่าว

### การจัดการกับเหตุการณ์

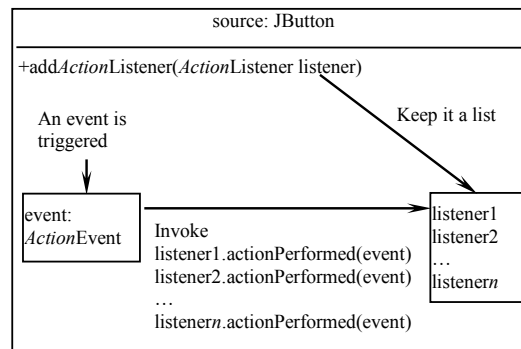


ภาษาจาวาจะมีวิธีการจัดการกับเหตุการณ์ที่เรียกว่า Delegation Model โดยจะมีหลักการดังนี้

- ▶ วัตถุของส่วนประกอบกราฟิกใด ๆ สามารถเป็นออบเจ็กต์ประเภทของ Event Source ได้ เช่น ออบเจ็กต์ของคลาส Button สามารถเป็น Event Source ของ ActionEvent ได้
- ▶ คลาสใด ๆ สามารถรับฟังเหตุการณ์ใด ๆ ก็ได้ถ้าคลาสนั้น implements อินเตอร์เฟซประเภท Listener ที่สอดคล้องกัน เช่นคลาสที่ต้องการรับฟังเหตุการณ์ ActionEvent จะต้อง implements อินเตอร์เฟซที่ชื่อ ActionListener
- ▶ ออบเจ็กต์ประเภท Event ที่เกิดจาก Event Source จะถูกส่งไปยังออบเจ็กต์ของคลาสที่สามารถรับฟังเหตุการณ์ประเภทนั้น



(a) Internal function of a generic source object



(b) Internal function of a JButton object

```
JButton jbt = new JButton("OK");

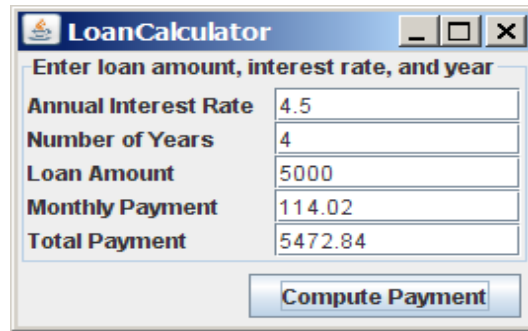
ActionListener listener = new OKListener();

jbt.addActionListener(listener);
```

อินเตอร์เฟซและเมธอดสำหรับการดักเหตุการณ์ประเภทต่าง ๆ ดังนี้

User Action, Source Object, Event Type, Listener Interface, and Handler				
User Action	Source Object	Event Type Fired	Listener Interface	Listener Interface Methods
Click a button	JButton	ActionEvent	ActionListener	actionPerformed(ActionEvent e)
Press Enter in a text field	JTextField	ActionEvent	ActionListener	actionPerformed(ActionEvent e)
Select a new item	JComboBox	ActionEvent	ActionListener	actionPerformed(ActionEvent e)
		ItemEvent	ItemListener	itemStateChanged(ItemEvent e)
Check or uncheck	JRadioButton	ActionEvent	ActionListener	actionPerformed(ActionEvent e)
		ItemEvent	ItemListener	itemStateChanged(ItemEvent e)
Check or uncheck	JCheckBox	ActionEvent	ActionListener	actionPerformed(ActionEvent e)
		ItemEvent	ItemListener	itemStateChanged(ItemEvent e)
Select a new item	JComboBox	ActionEvent	ActionListener	actionPerformed(ActionEvent e)
		ItemEvent	ItemListener	itemStateChanged(ItemEvent e)
Mouse pressed	Component	MouseEvent	MouseListener	mousePressed(MouseEvent e)
Mouse released				mouseReleased(MouseEvent e)
Mouse clicked				mouseClicked(MouseEvent e)
Mouse entered				mouseEntered(MouseEvent e)
Mouse exited				mouseExited(MouseEvent e)
Mouse moved			MouseMotionListener	mouseMoved(MouseEvent e)
Mouse dragged				mouseDragged(MouseEvent e)
Key pressed	Component	KeyEvent	KeyListener	keyPressed(KeyEvent e)
Key released				keyReleased(KeyEvent e)
Key typed				keyTyped(KeyEvent e)

3. ให้ศึกษาและทดลองพิมพ์ตัวอย่างการสร้าง GUI จากตัวอย่างต่อไปนี้ ให้เขียนอธิบายการทำงานในแต่ละบรรทัด



```

1 import java.awt.*;
2 import java.awt.event.*;
3 import javax.swing.*;
4 import javax.swing.border.TitledBorder;
5 public class LoanCalculator extends JFrame {
6     // Create text fields for interest rate, years
7     // loan amount, monthly payment, and total payment
8     private JTextField jtfAnnualInterestRate = new JTextField();
9     private JTextField jtfNumberOfYears = new JTextField();
10    private JTextField jtfLoanAmount = new JTextField();
11    private JTextField jtfMonthlyPayment = new JTextField();
12    private JTextField jtfTotalPayment = new JTextField();
13
14    // Create a Compute Payment button
15    private JButton jbtComputeLoan = new JButton("Compute Payment");
16    public LoanCalculator() {
17        // Panel p1 to hold labels and text fields
18        JPanel p1 = new JPanel(new GridLayout(5, 2));
19        p1.add(new JLabel("Annual Interest Rate"));
20        p1.add(jtfAnnualInterestRate);
21        p1.add(new JLabel("Number of Years"));
22        p1.add(jtfNumberOfYears);
23        p1.add(new JLabel("Loan Amount"));
24        p1.add(jtfLoanAmount);
25        p1.add(new JLabel("Monthly Payment"));
26        p1.add(jtfMonthlyPayment);
27        p1.add(new JLabel("Total Payment"));
28        p1.add(jtfTotalPayment);
29        p1.setBorder(new
30            TitledBorder("Enter loan amount, interest rate, and years"));
31        // Panel p2 to hold the button
32        JPanel p2 = new JPanel(new FlowLayout(FlowLayout.RIGHT));
33        p2.add(jbtComputeLoan);
34        // Add the panels to the frame
35        add(p1, BorderLayout.CENTER);
36        add(p2, BorderLayout.SOUTH);
37
38        // Register listener
39        jbtComputeLoan.addActionListener(new ButtonListener());
40    }
41    /** Handle the Compute Payment button */
42    private class ButtonListener implements ActionListener {
43        @Override
44        public void actionPerformed(ActionEvent e) {
45            // Get values from text fields
46            double interest =
47                Double.parseDouble(jtfAnnualInterestRate.getText());
48            int year = Integer.parseInt(jtfNumberOfYears.getText());
49            double loanAmount =
50                Double.parseDouble(jtfLoanAmount.getText());
51

```

import คำสั่ง

กำหนด Class นาม: Extends JFrame ทำให้มันเป็นหน้าต่าง

กำหนดค่าในฟิลด์เพื่อสำหรับป้อนข้อมูล และแสดงผลข้อมูล

ปุ่มชื่อ Compute Payment

Layout แบบ Grid มี 5 row, 2 col

เพิ่มปุ่มชื่อและใส่จำนวนที่ป้อน panel p1

ตั้งค่ากรอบที่มีชื่อสำหรับ p1

สร้าง panel p2 แบบชิดขวา

เพิ่มปุ่มคำนวณที่ป้อน panel p2

เพิ่ม panel p1, p2 ไปยังกรอบหลัก โดย p1 อยู่ตรงกลาง และ p2 อยู่ด้านล่าง

Listener สำหรับเหตุการณ์คลิกของปุ่มคำนวณ

เริ่มต้น Class ButtonListener เป็นคลาสที่สืบทอดจาก ActionListener

Override Method

ดึงค่าจากฟิลด์ของนามและแปลงเป็นประเภทข้อมูลที่เหมาะสม

```

52      // Create a loan object
53      Loan loan = new Loan(interest, year, loanAmount); — สร้าง Object Loan
54
55      // Display monthly payment and total payment
56      jtfMonthlyPayment.setText(String.format("%.2f",
57          loan.getMonthlyPayment()));
58      jtfTotalPayment.setText(String.format("%.2f",
59          loan.getTotalPayment()));
60  }
61  }
62  public static void main(String[] args) {
63      LoanCalculator frame = new LoanCalculator();
64      frame.pack();
65      frame.setTitle("LoanCalculator");
66      frame.setLocationRelativeTo(null); // Center the frame
67      frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
68      frame.setVisible(true);
69  }
70  }

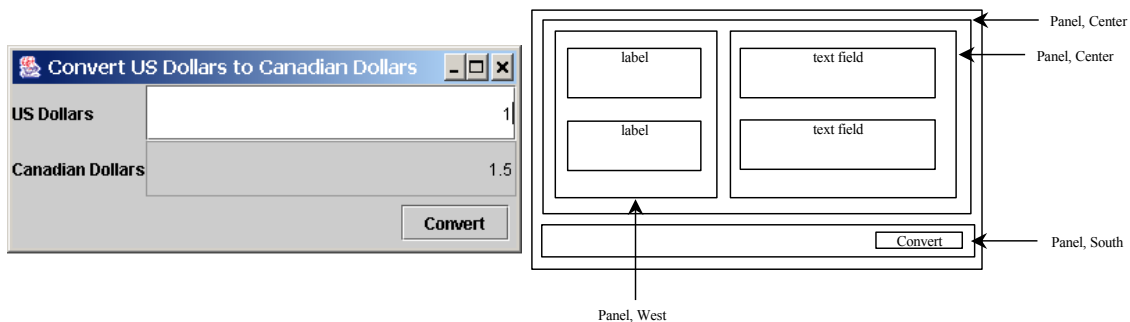
1  public class Loan {
2      private double annualInterestRate;
3      private int numberOfYears;
4      private double loanAmount;
5      private java.util.Date loanDate;
6
7      /** Default constructor */
8      public Loan() {
9          this(2.5, 1, 1000);
10     }
11
12     /** Construct a loan with specified annual interest rate,
13         number of years, and loan amount
14     */
15     public Loan(double annualInterestRate, int numberOfYears,
16         double loanAmount) {
17         this.annualInterestRate = annualInterestRate;
18         this.numberOfYears = numberOfYears;
19         this.loanAmount = loanAmount;
20         loanDate = new java.util.Date();
21     }
22
23     /** Return annualInterestRate */
24     public double getAnnualInterestRate() {
25         return annualInterestRate;
26     }
27
28     /** Set a new annualInterestRate */
29     public void setAnnualInterestRate(double annualInterestRate) {
30         this.annualInterestRate = annualInterestRate;
31     }
32     /** Return numberOfYears */
33     public int getNumberOfYears() {
34         return numberOfYears;
35     }
36     /** Set a new numberOfYears */
37     public void setNumberOfYears(int numberOfYears) {
38         this.numberOfYears = numberOfYears;
39     }
40 }

```

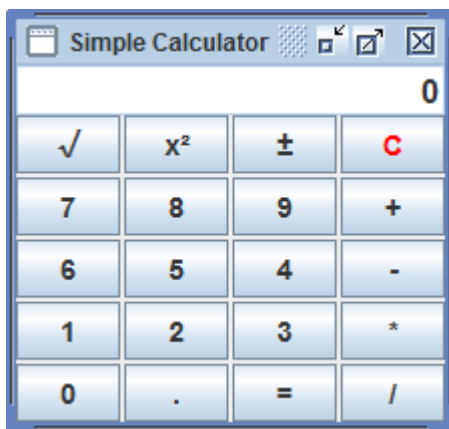
แสดงการชำระเงินรายเดือน และการชำระรวม

แสดงหน้าต่าง ทั้งชื่อกล่อง และขนาดให้พอดีกับส่วนประกอบ  
ทั้งตำแหน่งของ เมื่อคลิกปุ่มปิด

4. จาก GUI ของโปรแกรมสำหรับการแปลงเงิน US Dollars เป็น Canadian Dollars จงเขียนโปรแกรมสำหรับการแปลงเงิน US Dollars เป็น Canadian Dollars ต่อไปนี้

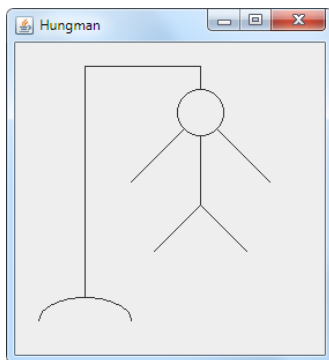


5. จาก GUI ของเครื่องคิดเลข ให้เพิ่มคำสั่งที่ทำให้เครื่องคิดเลขสามารถคำนวณได้



6. จาก GUI ของเกมส์ Hangman อธิบายผลลัพธ์ที่เกิดจากการใช้คำสั่งต่อไปนี้ในเมธอด

`public void paintComponent(Graphics g)` และให้เขียนโปรแกรมที่ทำให้ Hangman สามารถขยับตัวซ้ายขวา



```

g.drawArc(20, 220, 80, 40, 0, 180); — วาดครึ่งวงกลม
g.drawLine(20 + 40, 220, 20 + 40, 20); — วาดเส้นทแยงมุม
g.drawLine(20 + 40, 20, 20 + 40 + 100, 20); — วาดเส้นทแยงมุม
g.drawLine(20 + 40 + 100, 20, 20 + 40 + 100, 40); — วาดเส้นที่ไขว้กันตอน
int radius = 20;
g.drawOval(20 + 40 + 100 - radius, 40, 2 * radius, 2 * radius);
L วาดกลม (หัว)
g.drawLine(20 + 40 + 100 - (int)(radius *
Math.cos(Math.toRadians(45))), — วาดแขนซ้าย
40 + radius + (int)(radius * Math.sin(Math.toRadians(45))),
20 + 40 + 100 - 60, 40 + radius + 60);

g.drawLine(20 + 40 + 100 + (int)(radius *
Math.cos(Math.toRadians(45))), — วาดแขนขวา
40 + radius + (int)(radius * Math.sin(Math.toRadians(45))),
20 + 40 + 100 + 60, 40 + radius + 60);

g.drawLine(20 + 40 + 100, 40 + 2 * radius,
20 + 40 + 100, 40 + radius + 80); — วาดตัว

g.drawLine(20 + 40 + 100, 40 + radius + 80, 20 + 40 + 100 - 40,
40 + radius + 80 + 40); — วาดขาซ้าย

g.drawLine(20 + 40 + 100, 40 + radius + 80, 20 + 40 + 100 + 40,
40 + radius + 80 + 40); — วาดขาขวา
    
```

## 7. [Application]

จาก GUI สำหรับระบบเก็บข้อมูลผู้เปิดบัญชีธนาคารจากคลาส Account ให้เพิ่มคำสั่งที่ทำให้โปรแกรมสามารถบันทึกข้อมูลได้

8. ให้ออกแบบคลาส AngryBirds และออกแบบ GUI ของเกมส์ตามหน้าจอดังนี้



โดยเงื่อนไขของการทำงานคือ

1. สร้าง Object ของนกจากคลาสที่ได้สร้างไว้แล้วบน panel ที่มี Background ด้านหลัง
2. สุ่มตำแหน่งการวางของนกที่ตำแหน่ง  $x = 250$  (การคำนวณให้ถือว่า  $x$  มีตำแหน่งเป็น 0) ส่วนตำแหน่ง  $y$  ให้รับจากผู้ใช้งานผ่าน text area
3. บ่อนความเร็วของนกเช่น 50
4. บ่อนมุมของการยิงของนกหน่วยเป็นองศาเช่น 30 องศา
5. สร้าง Object ของหนูจากคลาสที่ได้สร้างไว้แล้วบน panel ที่มี Background ด้านหลัง
6. สุ่มตำแหน่งการวางของหนูที่ตำแหน่งใด ๆ โดยโปรแกรม
7. เมื่อผู้ใช้กดปุ่ม Ok โปรแกรมจะทำการตรวจสอบจุด  $x, y$  ของการเคลื่อนที่สุดท้ายของนกว่ามีโอกาสตกอยู่ในบริเวณของหนูหรือไม่ หากการเคลื่อนที่ของนกอยู่ในบริเวณตำแหน่งของหนู  $+20$  pixel ให้เพิ่มคะแนนที่ Score 100 คะแนน

การคำนวณ เนื่องจากการเคลื่อนที่เป็นแบบโปรเจกไทล์จะมีวิธีการคำนวณดังนี้

1. หาเวลาที่ตกพื้นในแนวดิ่ง เมื่อระยะทางแนวดิ่งเริ่มต้นเป็น  $S_y$  เช่น  $S_y = -100$
2. กำหนดความเร็ว  $U = 50$  m/s กำหนดมุม 30 องศา
3. หาระยะทางจากหน้าผาถึงจุดตกจากการเคลื่อนที่แนวราบโดย  $S_x = U \times T$
4. คำนวณหาระยะทางในแนวแกน  $y$  จากสมการ  $S_y = U_y T + \frac{1}{2} g t^2$
5. จะได้  $-100 = 50 (\sin 30) t + \frac{1}{2} (-10) t^2$
6. แก้สมการหา  $t$  จาก  $t^2 - 5t - 20 = 0$
7. จะได้  $t = 7.6$  วินาที
8. หาระยะทางของการตกของวัตถุในแนวแกน  $x$  จะได้  $S_x = 50 \cos 30 \times 7.6 = 326$
9. เพราะฉะนั้นพิกัดของการตกของวัตถุคือ 326, 0

9. [Algorithms] การชนกันของอนุภาคสองอนุภาคใด ๆ (Midterm 1/57)

อนุภาคแบบแข็งทำพิเศษจำนวน  $N$  อนุภาควางเรียงกัน เราจะเรียกอนุภาคดังกล่าวว่าอนุภาคที่ 1, 2, ...,  $N$  ตามลำดับ อนุภาคแต่ละอนุภาคจะมีค่าพลังงานสะสมอยู่ กล่าวคือ อนุภาคที่  $i$  จะมีพลังงานสะสมเท่ากับ  $X_i$  หน่วย

อนุภาคสองอนุภาคใด ๆ เมื่อนำมาชนกันจะสลายตัวและปล่อยพลังงานออกมา โดยพลังงานที่ปล่อยออกมานั้น มีค่าเท่ากับผลต่างของพลังงานสะสมของอนุภาคทั้งสอง

หัวหน้าห้องปฏิบัติการวานให้คุณทดลองนำอนุภาคทั้ง  $N$  อันมาชนกัน โดยหัวหน้าได้ระบุคำสั่งไว้ดังนี้ 1. ให้เลือกอนุภาคสองอนุภาคที่ติดกัน ที่ชนกันแล้วสลายตัวให้พลังงานมากที่สุด ถ้ามีหลายทางเลือก

ให้เลือกคู่ของอนุภาคที่ประกอบด้วยอนุภาคที่มีหมายเลขน้อยที่สุด

2. นำอนุภาคทั้งสองมาชนกัน ทำไปเรื่อยๆ จนกระทั่งอนุภาคหมด หรือเหลือแค่ 1 อนุภาค (ไม่สามารถชนกับใครได้อีก)

สังเกตว่าเมื่ออนุภาคชนกันแล้วจะสลายไปทั้งคู่ ทำให้อนุภาคคู่อื่น ๆ ที่เมื่อเริ่มต้นไม่ได้มีตำแหน่งติดกัน มีลำดับอยู่ติดกันได้

ตัวอย่างการดำเนินการเป็นดังนี้ สมมติมีอนุภาค 7 อนุภาคที่มีพลังงานสะสมดังนี้



1 2 4 3 1 2 3

คุณเลือกขนอนูภาคที่ 2 กับ 3 (สังเกตว่า คู่ของอนุภาค 3 กับ 1 ก็มีผลต่างเท่ากับ 2 เหมือนกัน แต่เราไม่เลือกเนื่องจากอนุภาคที่ 2 มีหมายเลขน้อยกว่า) ได้พลังงาน 2 หน่วย

หลังจากนั้น เราจะเหลืออนุภาค 5 อนุภาค

1 3 1 2 3

เลือกคู่อนุภาค 1 กับอนุภาค 4 ได้พลังงาน 2 หน่วย

1 2 3

เลือกคู่อนุภาค 5 กับอนุภาค 6 ได้พลังงาน 1 หน่วย

3

เมื่อเหลืออนุภาคเดียวเราจะไม่สามารถชนได้อีก รวมแล้วได้พลังงานทั้งหมด 5 หน่วย

จงเขียนโปรแกรมแบบ OOP โดยนิยาม class ที่ประกอบไปด้วย attribute และ method

รับข้อมูลพลังงานสะสมของอนุภาค จากนั้นคำนวณพลังงานทั้งหมดที่ได้รับจากการชนอนุภาคด้วยวิธีการตามหัวหน้าห้องปฏิบัติการระบุ

ข้อมูลนำเข้า

บรรทัดแรก ระบุจำนวนเต็ม  $N$  ( $1 \leq N \leq 1,000$ ) แทนจำนวนอนุภาค

อีก  $N$  บรรทัด ระบุพลังงานสะสมของแต่ละอนุภาค กล่าวคือ บรรทัดที่  $1 + i$  จะระบุจำนวนเต็ม  $X_i$  ( $1 \leq X_i \leq 1,000,000$ ) แทนพลังงานสะสมของอนุภาคที่  $i$

ข้อมูลส่งออก

มีบรรทัดเดียว คือ พลังงานรวมทั้งหมดที่ได้รับ

ตัวอย่างข้อมูลนำเข้า	ตัวอย่างข้อมูลส่งออก
7 1 2 4 3 1 2 3	5