**เอกสารประกอบโปรแกรม**

**วาดกราฟ GrahpDrawing**

**ผู้เขียน นายไพรัตน์ อติชาต**

**พัฒนาด้วย**

Netbean IDE 8.2 with Java version "1.8.0\_111"

**ดาวน์โหลด** <https://www.oracle.com/technetwork/java/javase/downloads/jdk-netbeans-jsp-3413139-esa.html>

library เพิ่มเติม GSON

**ดาวน์โหลด** <https://jar-download.com/artifacts/com.google.code.gson/gson/2.8.2/source-code>

**Git Repository →** [**https://github.com/PTaati/GraphDrawing**](https://github.com/PTaati/GraphDrawing)

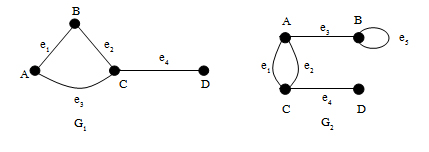
**แนวคิดพื้นฐาน**

**1. กราฟ**<https://www.scimath.org/lesson-mathematics/item/7334-2017-06-17-14-37-32>

**นิยาม** กราฟ G = (V(G),E(G)) ประกอบด้วย เซตจำกัด 2 เซต คือ

1. เซตที่ไม่เป็นเซตว่างของจุดยอด (Vertex) แทนด้วยสัญลักษณ์ V(G)
2. เซตของเส้นเชื่อม (Edge) ที่เชื่อมระหว่างจุดยอด แทนด้วยสัญลักษณ์ E(G)

**ตัวอย่าง**



จากกราฟ G1ที่กำหนดให้ จะได้ว่า

V(G1) = {A, B, C, D}

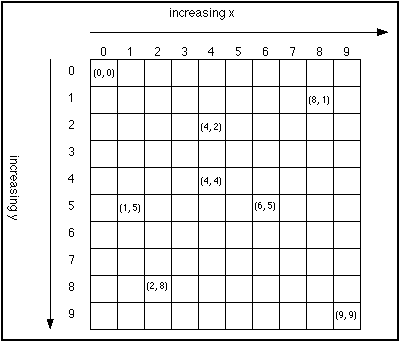
E(G1) = {e1, e2, e3, e4} = {AB, BC, AC, CD}

จากกราฟ G2ที่กำหนดให้ จะได้ว่า

V(G2) = {A, B, C, D}

E(G2) = { e1, e2, e3, e4, e5}

**2. การวาดภาพ 2 มิติด้วย java.awt.Graphics2D**

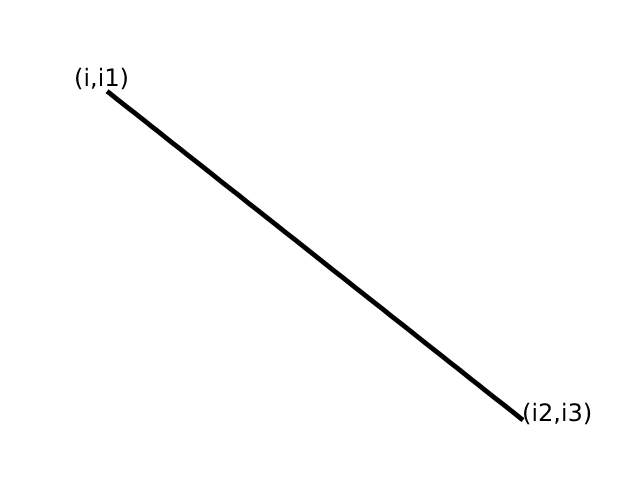
 **รูป แสดงการระบุพิกัดของคอมพิวเตอร์กราฟฟิคที่ใช้ในจาวา**

[**http://www.cafeaulait.org/course/week5/29.html**](http://www.cafeaulait.org/course/week5/29.html)

1. วาดเส้นตรง **drawLine(int i, int i1, int i2, int i3)**

*i , i1 คือ พิกัด x,y ของจุดปลายเส้นตรง*

*i2 , i3 คือ พิกัด x,y ของจุดปลายอีกด้านของเส้นตรง*

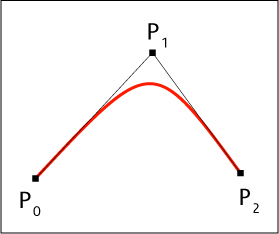
**

1. วาดเส้นโค้ง**draw(new QuadCurve2D.Float(float f, float f1, float f2, float f3, float f4, float f5)**

f,f1 คือ พิกัด x,y ของจุดปลายของเส้นโค้ง จากในรูปด้านล่าง อาจเป็นจุด P0 หรือ P2

f2,f3 คือ พิกัด x,y ของจุดควบคุมความโค้ง จากในรูปด้านล่าง คือ จุด P1

f4,f5 คือ พิกัด x,y ของจุดปลายของเส้นโค้ง จากในรูปด้านล่าง อาจเป็นจุด P0 หรือ P2



[**http://www.elml.uzh.ch/preview/cartouche/graphics/en/html/Curves\_learningObject2.html**](http://www.elml.uzh.ch/preview/cartouche/graphics/en/html/Curves_learningObject2.html)

1. วาดวงกลม **drawArc(int i, int i1, int i2, int i3, int i4, int i5)**

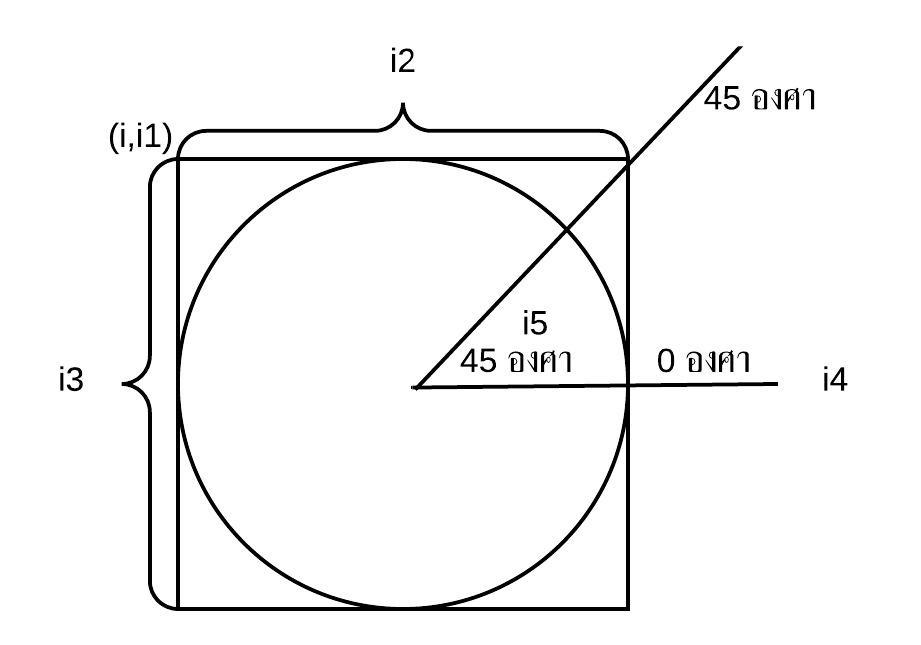
i,i1 คือ พิกัด x,y ของมุมกรอบด้านซ้ายบน

i2 คือ ขนาดความกว้าง

i3 คือ ขนาดความสูง

i4 คือ ตำแหน่งมุมเริ่มต้น

i5 คือ ขนาดมุม

****

1. วาดพื้นวงกลมด้วย fillOval(int i, int i1, int i2, int i3)

i, i1 คือ พิกัด x,y ของมุมกรอบด้านซ้ายบนดังรูปในข้อ 3

i2 คือ ขนาดความกว้าง

i3 คือ ขนาดความสูง

1. เขียนข้อความด้วย **drawString(String string, int i, int i1)**

string คือ ข้อความที่ต้องการวาด

I, i1 คือ พิกัด x,y มุมซ้ายบนของกรอบข้อความ

1. การเปลี่ยนสีก่อนวาดด้วย **setColor(Color color)**

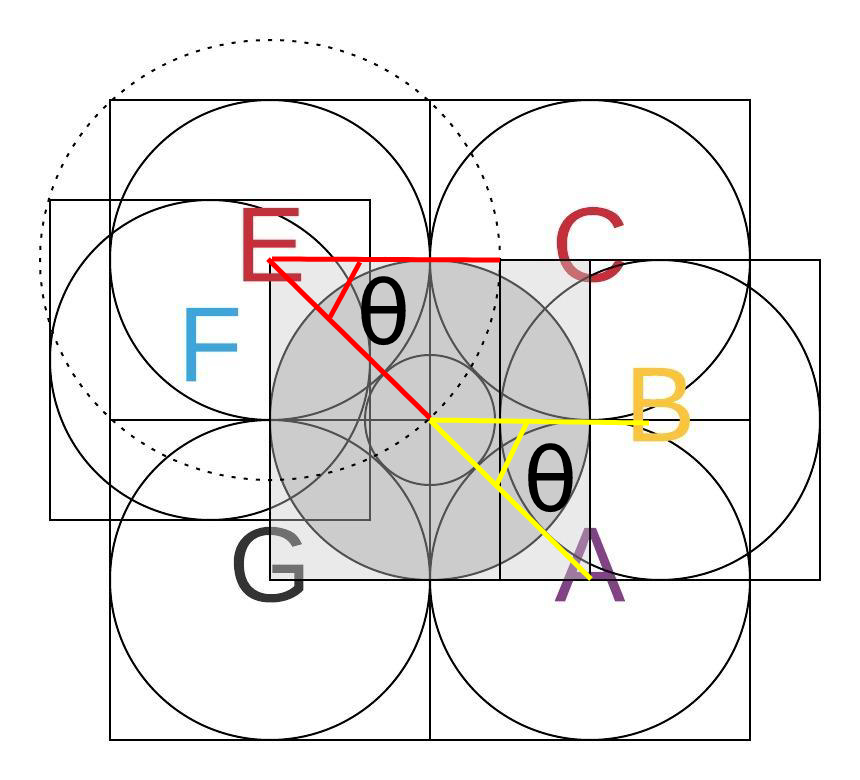
color คือ ค่าใน java.awt.Color

1. กำหนดขนาดเส้นด้วย **setStroke(new BasicStroke(float f))**

f คือ ขนาดความหนาของเส้น

1. การวาดวงวน (Loop)

จะใช้คำสั่ง **drawArc** วาดวงวน จากรูปด้านล่างถ้ากำหนดให้วงกลมสีเทาคือจุดยอด และวงกลม A,B,C,E,F,G คือ วงวน จะบนว่าการวาดวงวนรอบๆ จุดยอดนั้น ตำแหน่งมุมซ้ายบนของกรอบวงวนจะมีลักษณะเป็นรูปวงกลมเส้นประดังรูป และมุมระหว่างจุดศูนย์กลาง A และ B กับจุดยอดนั้นมีขนาดและตำแหน่งมุมที่เท่ากันกับ มุมระหว่างจุดขอบบนซ้ายของกรอบวงกลม A และ B กับจุดศูนย์กลางของวงกลมเส้นประ ทำให้เราสามารถรู้ตำแหน่งของจุดมุมกรอบวงวนได้ จากการหามุม และเทียบอัตราส่วนทางตรีโกณมิติ



**3. การใช้งาน GSON เพื่อบันทึกข้อมูลของวัตถุภายในโปรแกรมภาษาจาวา**

Gson เป็น library ของ java ที่ใช้แปลงข้อมูลระหว่าง Object และ JSON เพื่อเก็บค่าของ Object ขณะรันโปรแกรมเอาไว้ได้ และสามารถนำค่าที่เก็บไว้มาสามารถ Object ขึ้นอีกครั้งได้

**ขั้นตอนการใช้งานเก็บข้อมูลใน Object ลง JSON ด้วย GSON**

1. โหลด gson-2.8.2.jar มาจาก <https://jar-download.com/artifacts/com.google.code.gson/gson/2.8.2/source-code> จากนั้น Add jar เข้าไว้ใน Libraies ของโปรเจคที่สร้างไว้
2. import com.google.gson.Gson;

import com.google.gson.GsonBuilder;

GsonBuilder builder = new GsonBuilder();

Gson gson = builder.create();

1. สร้าง object ที่เราต้องการบันทึกค่า เช่น Backup backup = new Backup();
2. ใช้คำสั่ง toJson ดังนี้ gson.toJson(backup) นำตัวแปรอ้างอิงวัตถุที่ต้องการบันทึกค่าใส่เป็นพารามิเตอร์ของเมธอด toJson จากนั้น เมธอด toJson จะนำค่าใน Object แปลงเป็น String รูปแบบ json ให้ เราก็แค่นำString ที่ได้มาเขียนไฟล์เพื่อบันทึกค่า json ตามที่อยู่ที่ต้องการในตัวแปร path

FileWriter writer = new FileWriter(path);

writer.write(gson.toJson(backup));

writer.close();

**ขั้นตอนการใช้งานการนำข้อมูลใน JSON มาสร้างเป็น Object ด้วย GSON**

1. โหลด gson-2.8.2.jar มาจาก <https://jar-download.com/artifacts/com.google.code.gson/gson/2.8.2/source-code> จากนั้น Add jar เข้าไว้ใน Libraies ของโปรเจคที่สร้างไว้
2. อ่านไฟล์ json ตามที่อยู่ที่ต้องการในตัวแปร pathด้วย BufferedReader ดังนี้

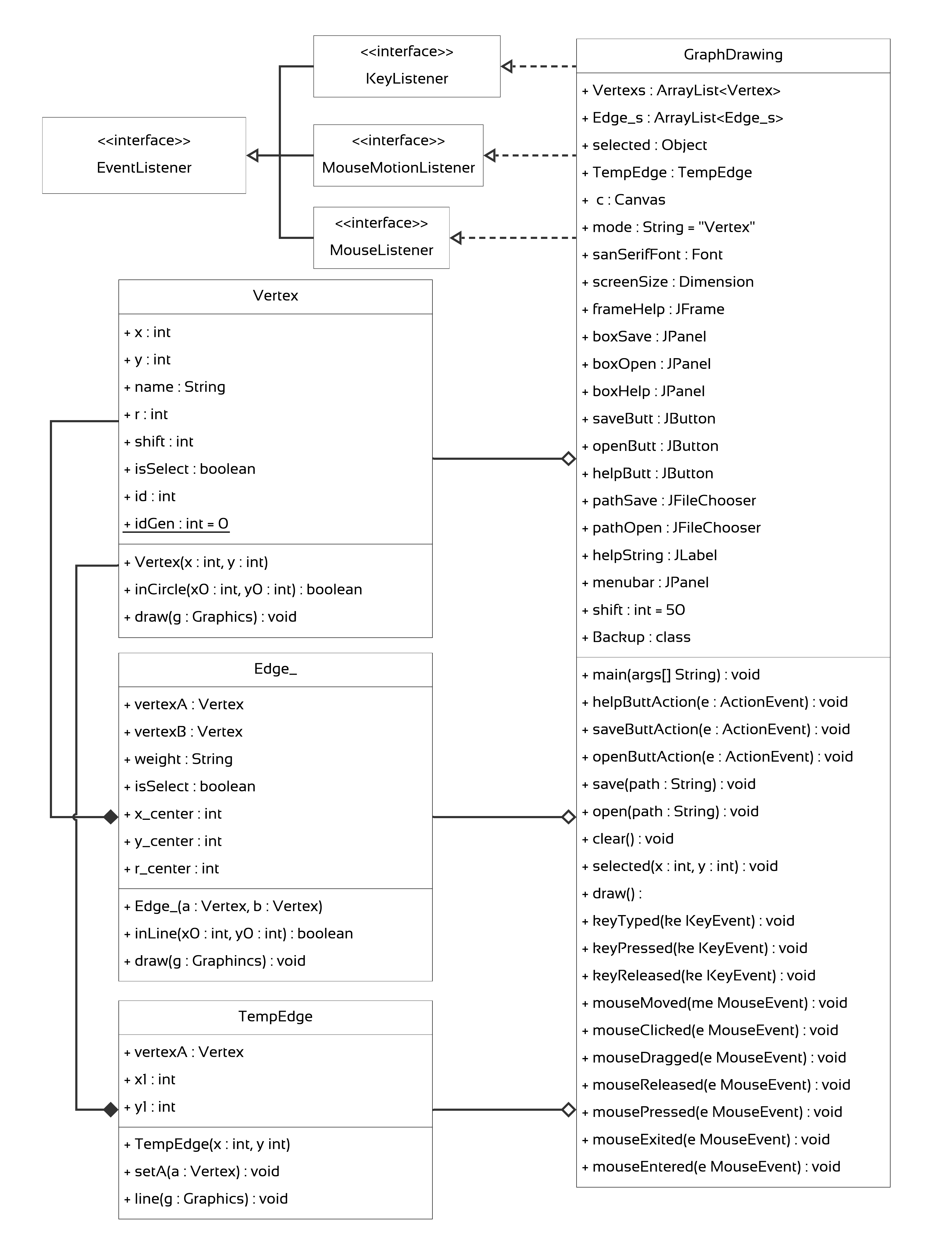
BufferedReader bufferedReader = new BufferedReader(new FileReader(path));

1. import com.google.gson.Gson;

import com.google.gson.GsonBuilder;

1. ใช้เมธอด fromJson โดยใส่พารามิเตอร์คือ bufferedReader ที่อ่าน json ไว้และ Object.class ที่ต้องการสร้าง Object โดย Object ที่ว่าต้องเป็น Object เดียวกับที่นำมาสร้าง json ไฟล์ดังขั้นตอนก่อนหน้า จากนั้น fromJson จะ return Object ที่เรากำหนกไว้เป็นพารามิเตอร์ให้

เช่น Backup backup = gson.fromJson(bufferedReader, Backup.class);

 **Class diagram GraphDrawing**

Class Vertex

|  |  |  |
| --- | --- | --- |
| Field / Method | Type / Return type | Description |
| x | int | พิกัดแกน x ของจุดศูนย์กลางของจุดยอด |
| y | int | พิกัดแกน y ของจุดศูนย์กลางของจุดยอด |
| name | String | ชื่อของจุดยอด |
| r | int | รัศมีถึงขอบนอกของจุดยอด |
| shift | int | รัศมีขอบในของจุดยอด |
| IsSelect | boolean | จุดยอดนี้ถูกเมาส์คลิ๊กเลือกไว้หรือไม่ |
| Id | int | เก็บ id เฉพาะของ Vertex |
| idGen | int | สร้าง id เฉพาะ Vertex |
| Vertex(x:int,y:int) | - | r = 36 , shift = 30 , name = “” , isSelect = false |
| inCircle(x0:int,y0:int) | boolean | ตวรจสอบว่า พิกัด (x,y) ที่รับเข้ามา อยู่ภายในจุดยอดนี้ไหม |
| draw(g:Graphics) | void | วาดรูปจุดยอดพร้อมชื่อจุดยอดลงบน Canvas |

Class Edge\_

|  |  |  |
| --- | --- | --- |
| Field / Method | Type / Return type | Description |
| VertexA | Vertex | เก็บตัวแปรอ้างอิงวัตถุของจุดยอด ที่เป็นจุดปลายเส้นเชื่อม |
| VertexB | Vertex | เก็บตัวแปรอ้างอิงวัตถุของจุดยอด ที่เป็นจุดปลายเส้นเชื่อม |
| weight | String | เก็บค่าน้ำหนักของเส้นเชื่อม เป็นข้อความ (ตัวเลข หรือตัวแปร) |
| isSelect | boolean | เส้นเชื่อมนี้ถูกเมาส์คลิ๊กเลือกไว้หรือไม่ |
| x\_center | int | พิกัดแกน x ของ weight [ จุดปรับความโค้ง หรือ จุดหมุนวงวน ] |
| y\_center | int | พิกัดแกน y ของ weight [ จุดปรับความโค้ง หรือ จุดหมุนวงวน ] |
| r\_center | int | รัศมีรอบข้อความ weight ที่สามารถคลิ๊กเพื่อเลือกเส้นได้ |
| Edge\_(a:Vertex,b:Vertex) | - | r\_center = 50 , weight = “1” , isSelect = false |
| inLine(x0:int,y0:int) | boolean | ตวรจสอบว่า พิกัด (x,y) ที่รับเข้ามา อยู่ภายในเส้นไหม (ภายในรัศมี r\_center รอบ weight ) |
| draw(g:Graphics2D) | void | วาดเส้นเชื่อม มี 2 กรณี คือเส้นเชื่อมระหว่าง 2 จุดยอด กับ เส้นเชื่อมวงวน |

Class TempEdge

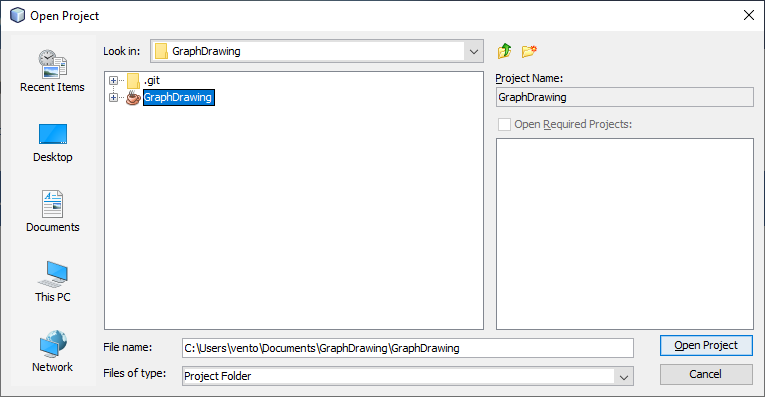
|  |  |  |
| --- | --- | --- |
| Field / Method | Type / Return type | Description |
| VertexA | Vertex | เก็บตัวแปรอ้างอิงวัตถุของจุดยอด ที่เป็นจุดปลายเส้นเชื่อม |
| x1 | int | พิกัดแกน x ของปลายเส้นเชื่อมปัจจุบันที่เมาส์ชี้อยู่ |
| y1 | int | พิกัดแกน y ของปลายเส้นเชื่อมปัจจุบันที่เมาส์ชี้อยู่ |
| TempEdge(x:int,y:int) | - | - |
| setA(a:Vertex) | void | กำหนดค่า จุดยอดเริ่มต้น |
| line(g:Graphics2D) | void | วาดเส้นเชื่อม คือเส้นเชื่อมระหว่าง จุดยอดกับจุดปัจจุบันที่เมาส์ชี้ อาจะเป็นเส้นเชื่อมงวนได้ |

Class GraphDrawing

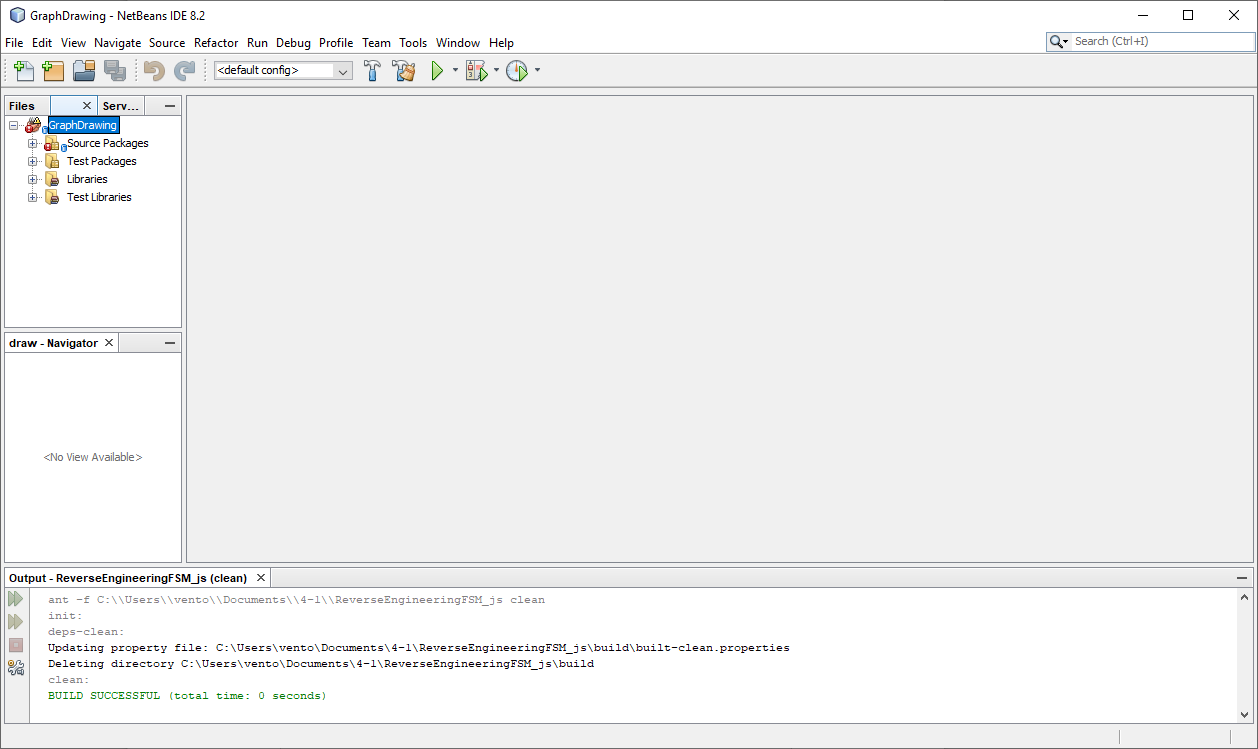
|  |  |  |
| --- | --- | --- |
| Field / Method | Type / Return type | Description |
| Vertexs | ArrayList<Vertex> | V(G) เอาไว้เก็บจุดยอดของกราฟ |
| Edge\_s | ArrayList<Edge\_s> | E(G) เอาไว้เก็บเส้นเชื่อมของกราฟ |
| Selected | Object | เอาไว้อ้างอิง จุดยอดหรือเส้มเชื่อมที่จะทำการแก้ไข โดยใช้เมาส์คลิ๊กเลือก |
| TempEdge | TempEdge | เส้นเชื่อมที่อยู่ในขั้นตอนการวาด ยังไม่ถูกเพิ่มลงใน E(G) |
| c | Canvas | พื้นที่สำหรับวาดกราฟ |
| mode | String | มีค่าระหว่าง “Vertex” กับ “Edge\_” ใช้เพื่อแบ่งโหมดการวาดกราฟ ระหว่างการจัดการเส้นเชื่อม และการจัดการจุดยอด ออกจากกัน กด space bar ค้างเพื่อให้อยู่ใน mode “Edge\_” นอกจากนั้นจะอยู่ใน mode “Vertex” |
| sanSeriFont | Font | รูปแบบตัวอักษรที่แสดงผล (“SanSerif”,Font.PLAIN,24) |
| ScreenSize | Dimension | ขนาดของหน้าจอปัจจุบัน |
| FrameHelp | Jframe | Frame สำหรับแสดงข้อความช่วยเหลือ |
| BoxSave | JPanel | Panel สำหรับการบันทึกไฟล์ |
| BoxOpen | JPanel | Panel สำหรับการเปิดไฟล์ |
| BoxHelp | JPanel | Panel สำหรับการแสดงข้อความช่วยเหลือ |
| SaveButt | JButton | ปุ่มกดเพื่อบันทึกกราฟการทำงานจะอยู่ใน method saveButtAction(e:ActionEvent) |
| OpenButt | JButton | ปุ่มกดเพื่ออ่านกราฟการทำงานจะอยู่ใน method openButtAction(e:ActionEvent) |
| HelpButt | JButton | ปุ่มกดเพื่อแสดงข้อความช่วยเหลือ การทำงานอยู่ใน method helpButtAction(e:ActionEvent ) |
| PathSave | JFileChooser | ใช้สำหรับกำหนดที่อยู่ไฟล์ json ที่ต้องการจะบันทึกข้อมูลกราฟ |
| PahtOpen | JFileChooser | ใช้สำหรับเลิือกไฟล์ json เพื่อนำเข้ากราฟเข้าโปรแกรม |
| HelpString | JLabel | เก็บข้อความช่วยเหลือ |
| Menubar | JPanel | แถบสำหรับวางปุ่ม save , open , help |
| Shift | int | ขนาดความกว้างของ menubar |
| Backup | class | Class สำหรับรวม Object ทั้งหมดของกราฟ ภายในมี  ArrayList<Vertex> VertexsBackup;  ArrayList<Edge\_> Edge\_sBackup; |
| Main(args[] : String) | void | Method main สำหรับรันโปรแกรม |
| helpButtAction(e:ActionEvent ) | void | การทำงานเมื่อกดปุ่ม help จะแสดง Frame ใหม่ขึ้นมาที่กลางจอภายใน frame จะประกอบไปด้วยข้อความอธิบายการใช้งานโปรแกรมเบื้องต้น |
| saveButtAction(e:ActionEvent) | void | การทำงานเมื่อกดปุ่ม save แสดงหน้าต่างให้กำหนดตำแหน่งไฟล์ที่จะบันทึกกราฟ จากนั้นจะเรียกใช้ method save(path) เพื่อบันทึกกราฟ |
| openButtAction(e:ActionEvent) | void | การทำงานเมื่อกดปุ่ม open จะแสดงหน้าต่างให้เลือกตำแหน่งไฟล์ที่กราฟที่จะนำเข้ามาเปิดในโปรแกรม จากนั้นจะเรียกใช้ method open(path) เพื่ออ่านไฟล์กราฟ |
| save(path:String) | void | ใช้ Gson ในการแปลง object Backup เป็น json และบันทึกไฟล์ลงในเครื่องตามตำแหน่งทีกำหนด |
| open(path:String) | void | อ่านไฟล์กราฟ แล้วใช้ Gson ในการแปลงjson เป็น object Backupและแสดงกราฟบนหน้าจอ |
| Clear() | void | ทำให้ canvas กลายเป็นสีขาวทั้งหมด |
| Selected(x:int,y:int) | void | รับพิกัด x,y ที่เมาส์คลิ๊ก ถ้าตำแหน่งที่คลิ๊กคือ vertex แล้ว vertex นั้นจะเป็นวัตถุที่ถูกเลือก ถ้าไม่มี vertex ที่ตำแหน่งนั้นก็จะไปตรวจสอบเส้นเชื่อมในแบบเดียวกัน ถ้าตำแหน่งนั้นไม่ใช่ทั้งเส้นเชื่อมและจุดยอด วัตถุที่ถูกเลือกจะถูกยกเลิกการเลือก |
| Draw() | void | วาดจุดยอดและเส้นเชื่อมทั้งหมดลงบน canvas  ถ้าเส้นเชื่อมหรือจุดยอดนั้นถูกเลือกจะแสดงสีน้ำเงิน ถ้าไม่ใช่จะเป็นสีดำ  (วาดเส้นเชื่อม ก่อน แล้วค่อยวาดจุดยอดทับ) |
| keyTyped(ke:KeyEvent) | void | เมื่อกดแป้นพิมพ์ เมธอดนี้จะทำงานโดยจะตรวจว่ากดอะไรเข้ามา ถ้ากด ctrl+s จะทำการบันทึกกราฟลงไฟล์ backup.json  ctrl+o จะเปิดกราฟจากไฟล์ backup.json  ถ้า vertexถูกเลือกอยู่ การพิมพ์จะเป็นการพิมพ์ชื่อให้แก่ vertex  ถ้า edge\_ ถูกเลือกอยู่ การพิมพ์จะเป็นการพิมพ์ค่าให้แก่ weight ของเส้น |
| keyPressed(ke:KeyEvent) | void | ถ้ากด spacebar ค้างในระขณะที่กดค้างโปรแกรมจะอยู่ใน mode “Edge\_” |
| keyReleased(ke:KeyEvent) | void | ถ้าเลิกกด keyboard โปรแกรมจะอยู่ใน mode “Vertex” |
| mouseMoved(me MoouseEvent) | void | - |
| mouseClicked(e:MouseEvent) | void | เมื่อคลิ๊กเมาส์ลงไป 1 ครั้งจะเป็นการเลือกวัตถุในกราฟ (vertex , edge)  ถ้า double click จะเป็นการสร้าง Vertex ใหม่ทีตำแหน่งที่ double click |
| mouseDragged(e:MouseEvent) | void | การกดเมาส์ค้าง  ถ้าอยู่ใน mode “Vertex” จะเป็นการเปลี่ยนตำแหน่งของเส้นเชื่อมหรือจุดยอดที่ถูกเลือกอยู่  ถ้าอยู่ใน mode “Edge\_” จะเป็นการวาดเส้นเชื่อม แต่จะวาดได้เมื่อเริ่มกดจากภายในจุดยอด ไปสู่จุดยอดเท่านั้น |
| mouseReleased(e:MouseEvent) | void | การออกจากการคลิ๊ก (เมื่อคลิ๊กเสร็จแล้ว)  ถ้าอยู่ใน mode “Vertex” จะยกเลิกการวาดเส้นเชื่อมที่ยังวาดไม่เสร็จ  ถ้าอยู่ใน mode “Edge\_” จะทำการสร้างเส้นเชื่อมขึ้นมา โดยมี 2 แบบ คือเส้นเชื่อมระหว่าง 2 จุดยอดกับ เส้นเชื่อมวงวน |
| mousePressed(e:MouseEvent) | void | เมื่อเริ่มกดคลิ๊กเมาส์ และอยู่ใน mode “Edge\_” จะสร้างเส้นเชื่อมชั่วคราวขึ้นมาโดยจุดยอดเริ่มต้นคือจุดยอดในตำแห่นงที่เมาส์เริ่มคลิ๊ก ถ้าตำแหน่งนั้นไม่มีจุดยอด จะไม่สร้างเส้นเชื่อมขึ้นมาให้ |
| mouseExited(e:MouseEvent) | void | - |
| mouseEntered(e:MouseEvent) | void | - |

**การ compile หรือ run โปรแกรม**

1. compile โปรแกรมภาษา java ผ่านโปรแกรม IDE Netbean8.2 โดย open Project ชื่อ GraphDrawing ซึ่งภายในมี source code อยู่ใน Folder src

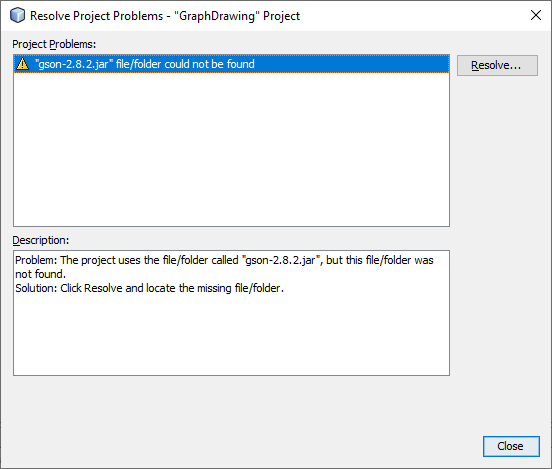


รูปที่ 1 แสดงการ open Project

2. ให้กดคลิ๊กขวาที่ ชื่อโปรเจคที่เปิดแล้วจากนั้นกด Resolve Project Problems ..เพื่อเพิ่ม GSON เข้ามาในโปรเจค

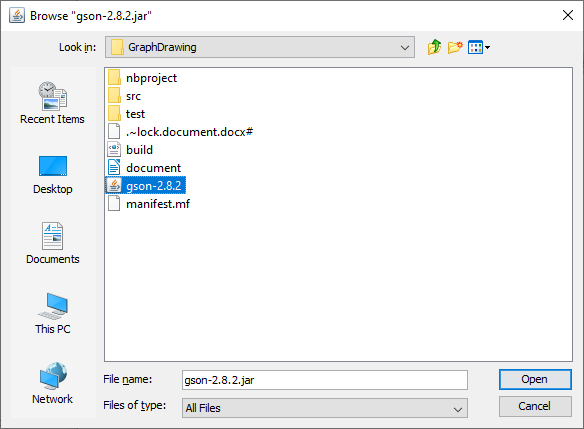
รูปที่ 2 แสดงการเพิ่ม GSON เข้ามาในโปรเจค

3. กด Resolve



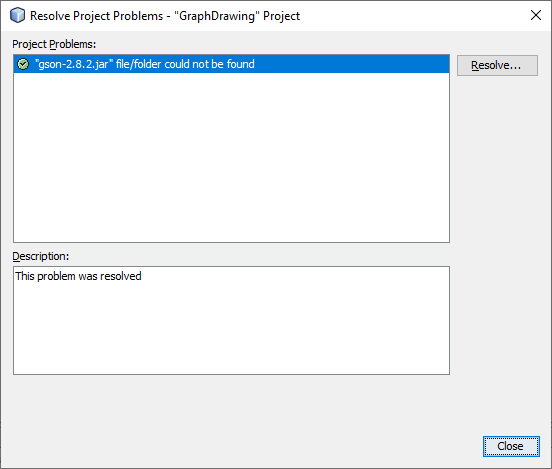
รูปที่ 3 แสดงหน้าต่าง Resolve

4. เข้าไปสู่ Directory ของ Project >> กดเลือกไฟล์ gson-2.8.2.jar >> กด open

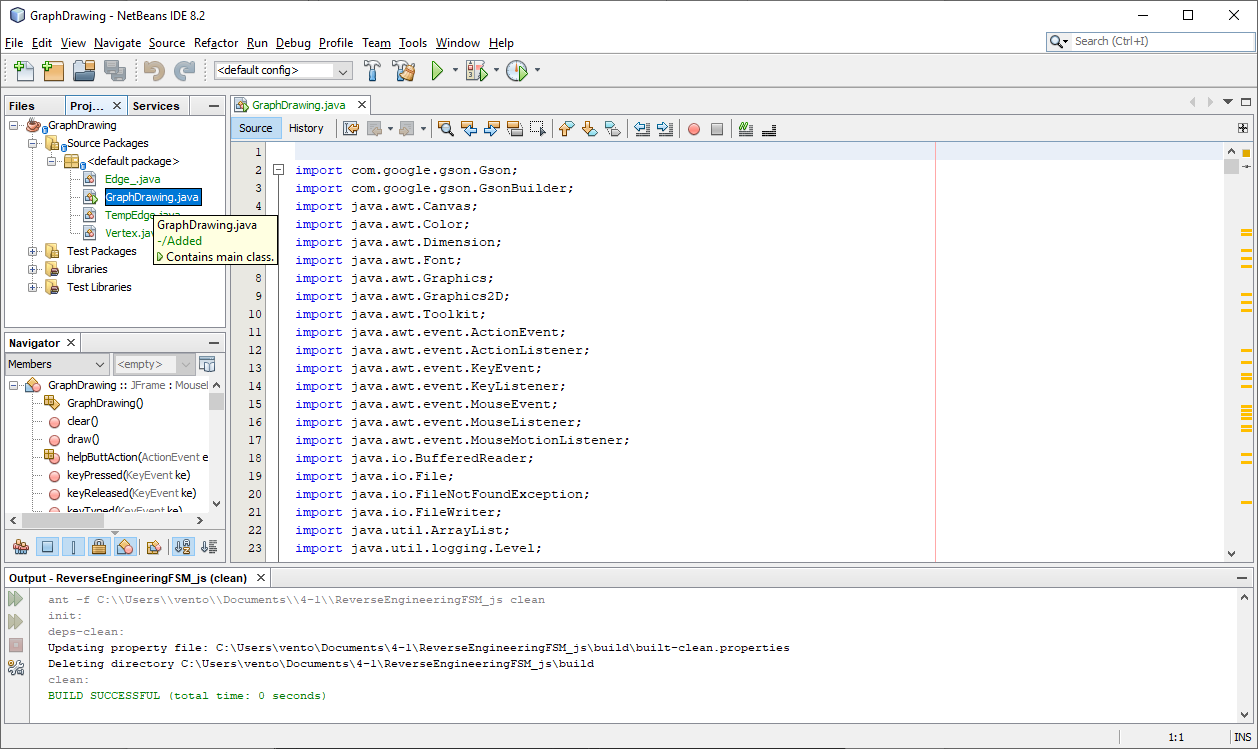


รูปที่ 4 แสดงการเปิดไฟล์ gson-2.8.2.jar

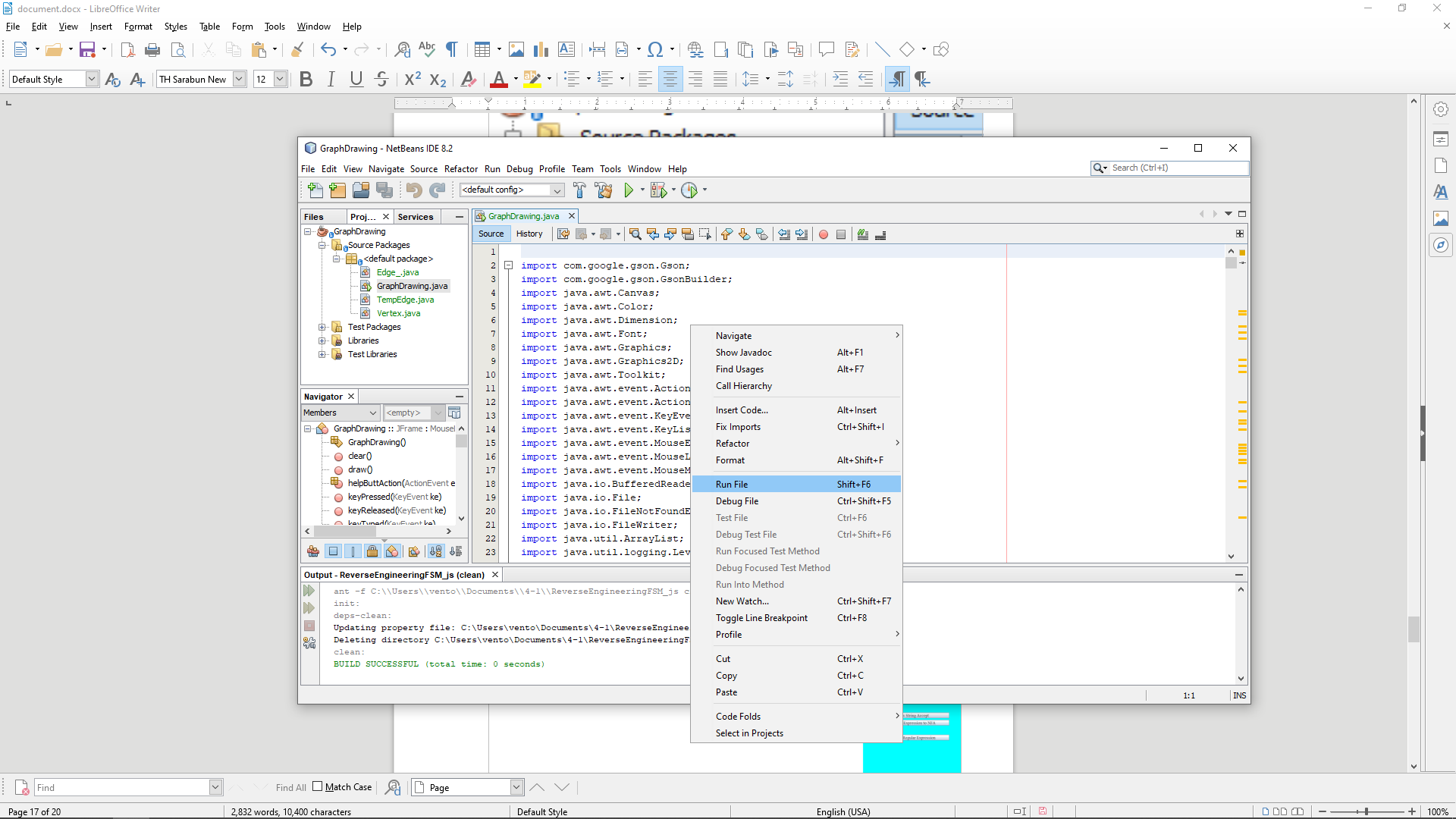
5 กด Close

รูปที่ 1 แสดงหน้าต่างหลังจากเพิ่ม GSON

6. เข้าสู่ Project GraphDrawingกดเข้าไปสู่ java ไฟล์ ดังนี้ GraphDrawing >> Source Packages >> <default Packages> >> GraphDrawing.java



รูปที่ 6 แสดงการเปิดไฟล์ GraphDrawing.java

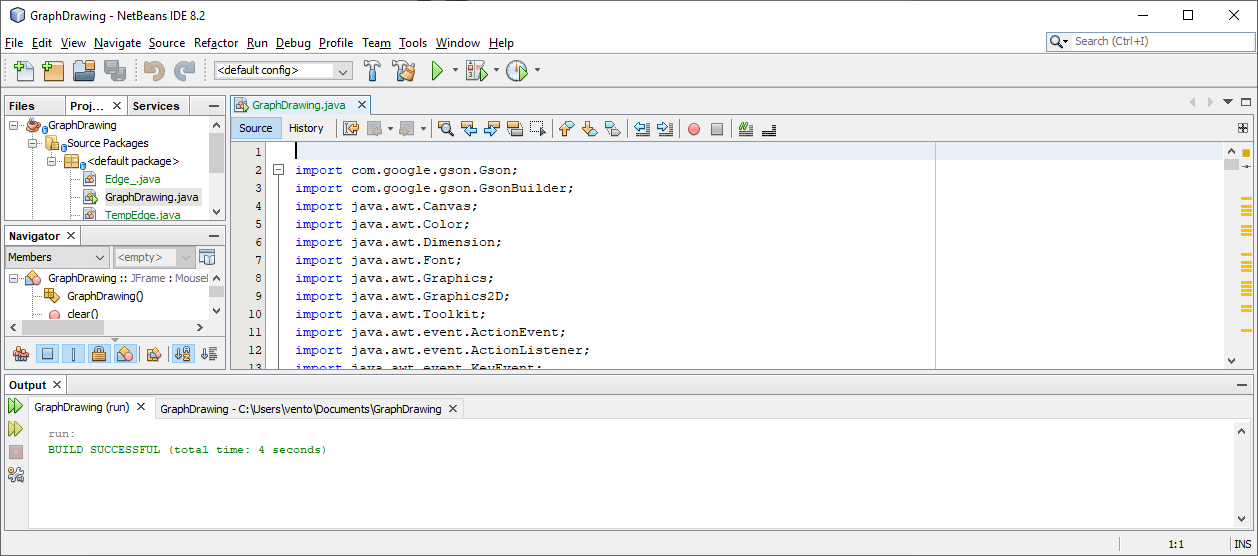
  
  
 7. คลิกขวาที่ส่วนของโค้ดแล้วกด Run File (Shift + F6)

รูปที่ 7 แสดงวิธีการกดรัน

8. โปรแกรมรันหน้า GUI ขึ้นมาดังนี้

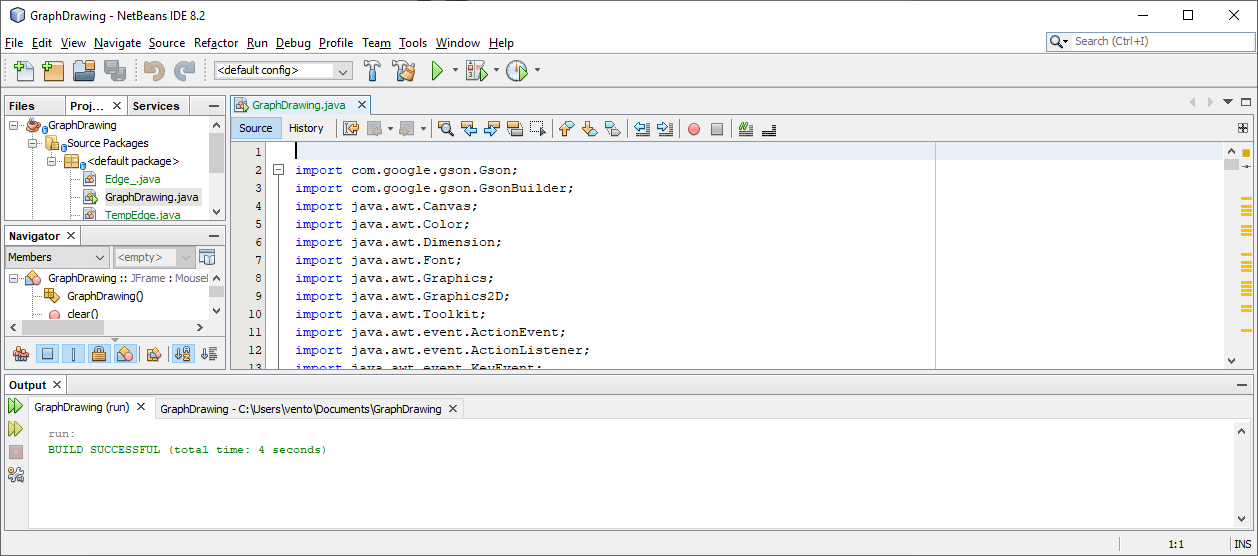


รูปที่ 8 แสดงตัวอย่างหน้า GUI



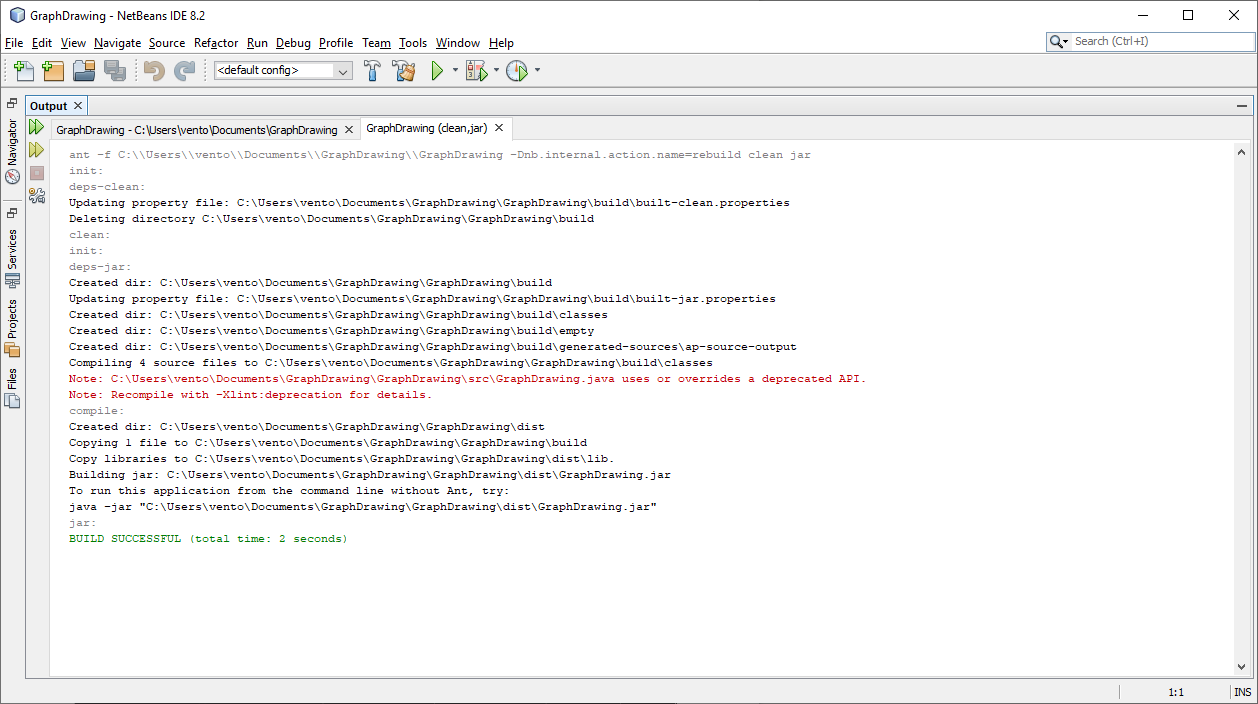
9. หากต้องการสร้างไฟล์ .jar ให้กด

Clean and Build Project (Shift + F11)



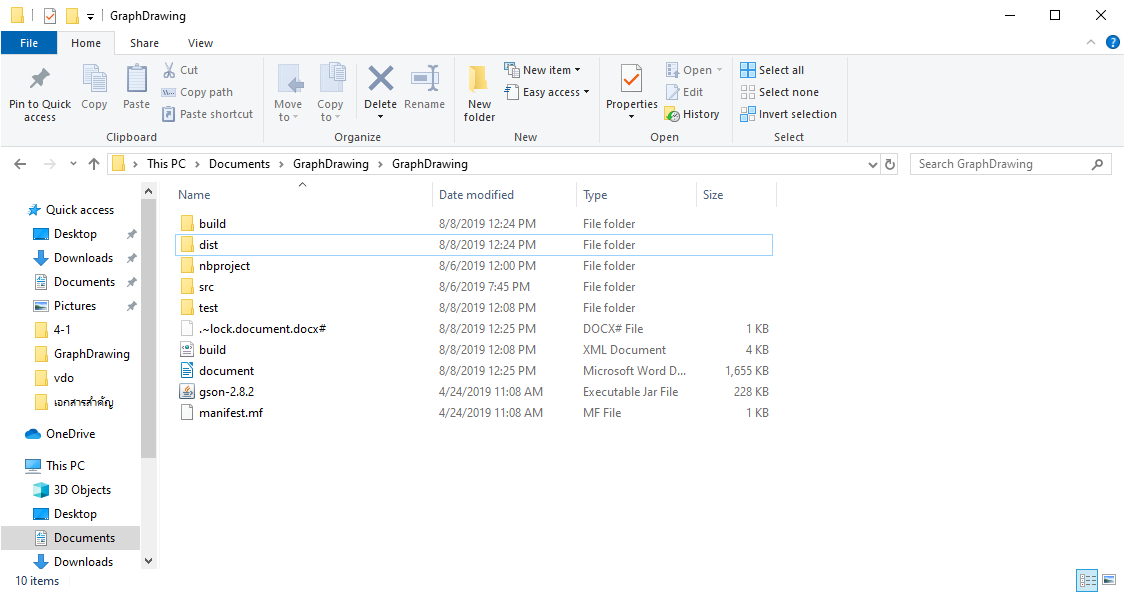
รูปที่ 9แสดงวิธีการสร้างไฟล์ .jar

10. แสดงสถานะการ Build Project



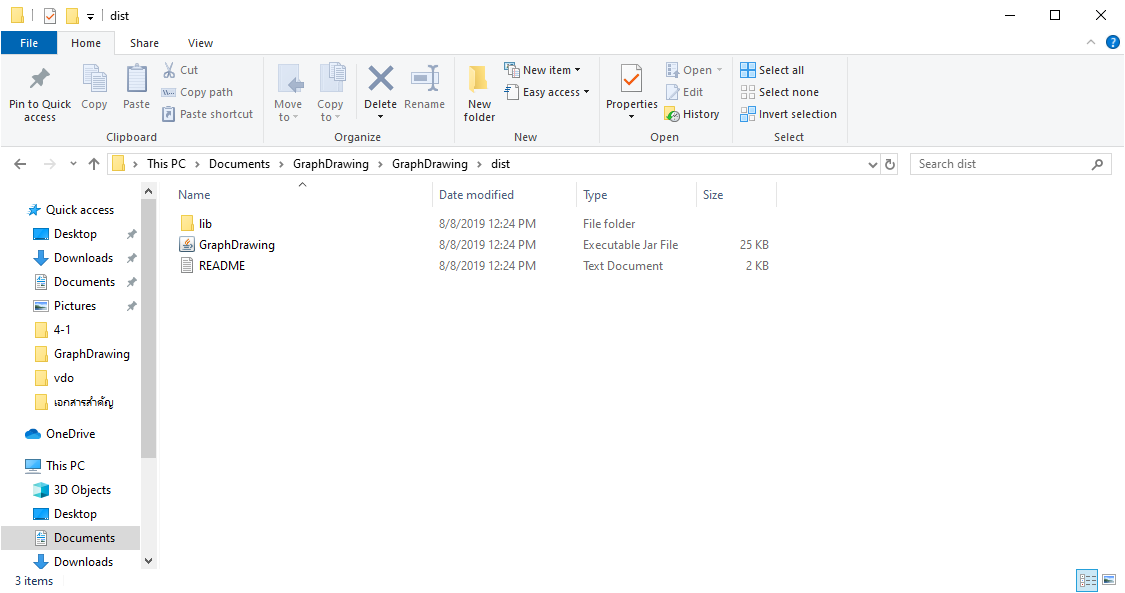
รูปที่ 10 แสดงสถานะการสร้าง

11. หลังจากขั้นตอน 10 จะมีโฟลเดอร์ dist ปรากฏขึ้นมา



รูปที่ 11 แสดงโฟลเดอร์เมื่อสร้างสำเร็จ

12. เมื่อเข้าไปใน Folder dist ก็จะพบการโปรแกรม GraphDrawing.jar และ Folder lib ที่เก็บ library GSON สำหรับใช้ save และ open ภายในโปรแกรม



รูปที่ 12 แสดงส่วนประกอบที่ถูกสร้างในโฟลเดอร์

13. สามารถ double click ไฟล์ GraphDrawing.jar เพื่อเปิดโปรแกรมได้ (ต้องลง JDK ก่อนถึงจะสามารถ run  ไฟล์ .jar ได้)



รูปที่ 13 แสดงการเปิดโปรแกรมเพื่อรันไฟล์ .jar