

# เอกสารประกอบโปรแกรม วาดกราฟ GraphDrawing

ผู้เขียน นายไพรัตน์ อติชาติ

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-es.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>

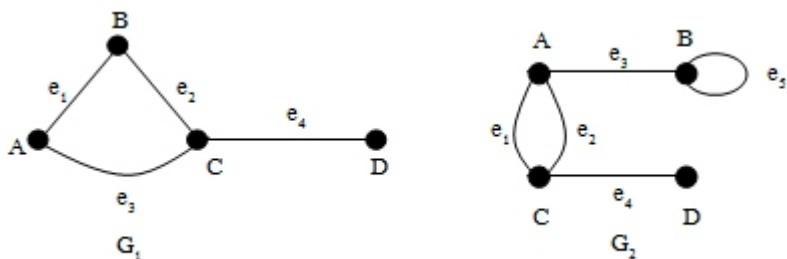
## แนวคิดพื้นฐาน

### 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  $G_1$  ที่กำหนดให้ จะได้ว่า

$$V(G_1) = \{A, B, C, D\}$$

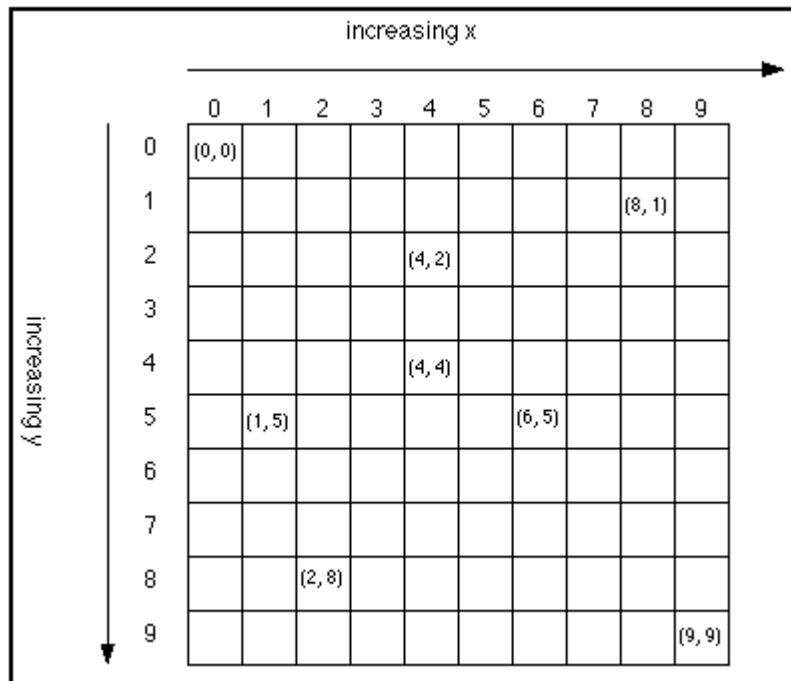
$$E(G_1) = \{e_1, e_2, e_3, e_4\} = \{AB, BC, AC, CD\}$$

จากกราฟ  $G_2$  ที่กำหนดให้ จะได้ว่า

$$V(G_2) = \{A, B, C, D\}$$

$$E(G_2) = \{e_1, e_2, e_3, e_4, e_5\}$$

## 2. การวาดภาพ 2 มิติด้วย java.awt.Graphics2D



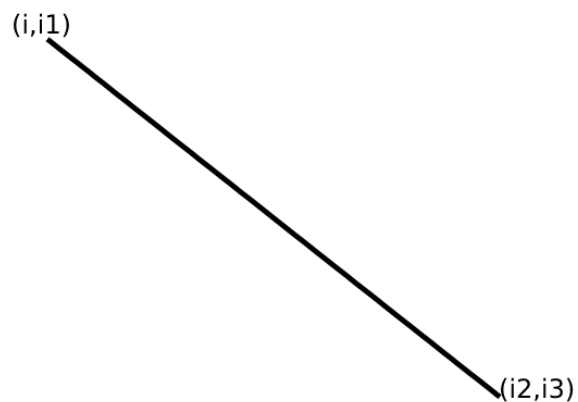
รูป แสดงการระบุพิกัดของคอมพิวเตอร์กราฟฟิคที่ใช้ในจาวา

<http://www.cafeulait.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

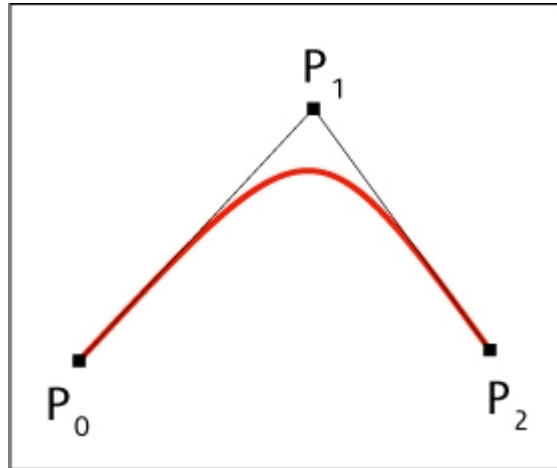


2. วาดเส้นโค้ง `draw(new QuadCurve2D.Float(float f, float f1, float f2, float f3, float f4, float f5)`

$f, f1$  คือ พิกัด  $x, y$  ของจุดปลายของเส้นโค้ง จากในรูปด้านล่าง อาจเป็นจุด  $P_0$  หรือ  $P_2$

$f2, f3$  คือ พิกัด  $x, y$  ของจุดควบคุมความโค้ง จากในรูปด้านล่าง คือ จุด  $P_1$

$f4, f5$  คือ พิกัด  $x, y$  ของจุดปลายของเส้นโค้ง จากในรูปด้านล่าง อาจเป็นจุด  $P_0$  หรือ  $P_2$



[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)

3. วาดวงกลม `drawArc(int i, int i1, int i2, int i3, int i4, int i5)`

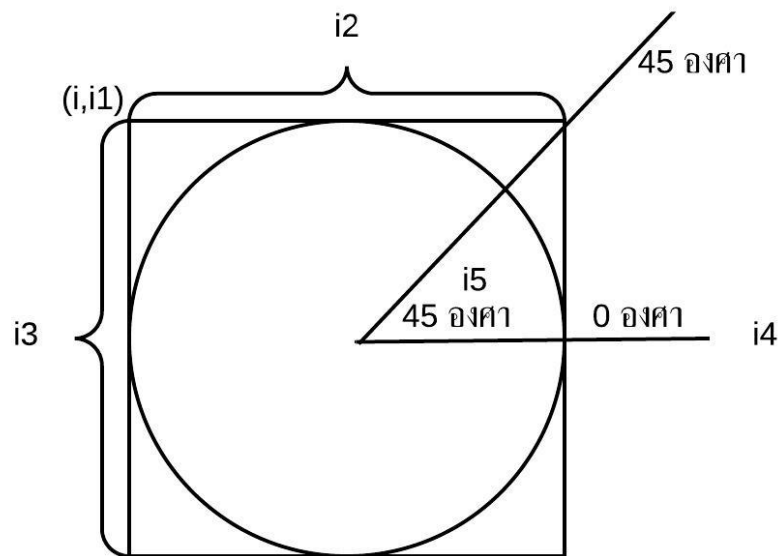
$i, i1$  คือ พิกัด  $x, y$  ของมุมกรอบด้านซ้ายบน

$i2$  คือ ขนาดความกว้าง

$i3$  คือ ขนาดความสูง

$i4$  คือ ตำแหน่งมุมเริ่มต้น

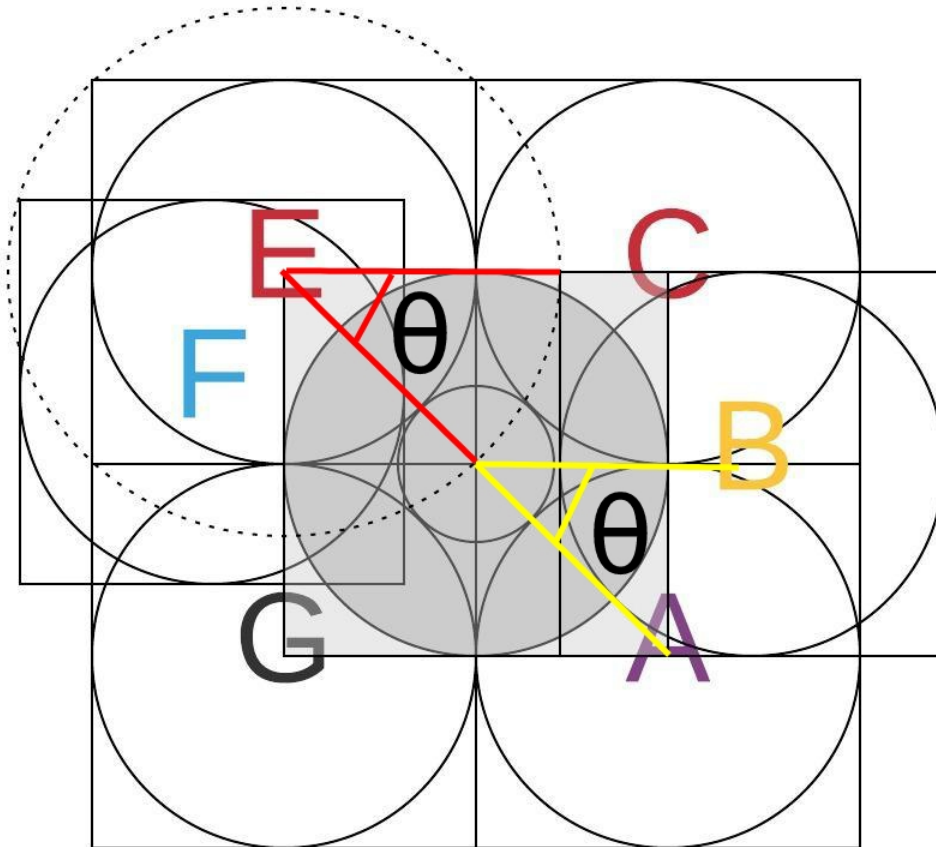
$i5$  คือ ขนาดมุม



4. วาดพื้นวงกลมด้วย `fillOval(int i, int i1, int i2, int i3)`  
 $i, i1$  คือ พิกัด  $x, y$  ของมุมกรอบด้านซ้ายบนดังรูปในข้อ 3  
 $i2$  คือ ขนาดความกว้าง  
 $i3$  คือ ขนาดความสูง
5. เขียนข้อความด้วย `drawString(String string, int i, int i1)`  
 $string$  คือ ข้อความที่ต้องการวาด  
 $i, i1$  คือ พิกัด  $x, y$  มุมซ้ายบนของกรอบข้อความ
6. การเปลี่ยนสีก่อนวาดด้วย `setColor(Color color)`  
 $color$  คือ ค่าใน `java.awt.Color`
7. กำหนดขนาดเส้นด้วย `setStroke(new BasicStroke(float f))`  
 $f$  คือ ขนาดความหนาของเส้น

## 8. การวาดวงวน (Loop)

จะใช้คำสั่ง **drawArc** วาดวงวน จากรูปด้านล่างถ้ากำหนดให้วงกลมสีเทาคือจุดยอด และวงกลม A,B,C,E,F,G คือ วงวน จะพบว่า การวาดวงวนรอบๆ 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();
```
3. สร้าง object ที่เราต้องการบันทึกค่า เช่น Backup backup = new Backup();
4. ใช้คำสั่ง toJson ดังนี้ gson.toJson(backup) นำตัวแปรอ้างอิงวัตถุที่ต้องการบันทึกค่าใส่เป็นพารามิเตอร์ของเมธอด toJson จากนั้น เมธอด toJson จะนำค่าใน Object แปลงเป็น String รูปแบบ json ให้ เราก็แค่นำ String ที่ได้มาเขียนไฟล์เพื่อบันทึกค่า json ตามที่อยู่ที่ต้องการในตัวแปร path

```
FileWriter writer = new FileWriter(path);
writer.write(gson.toJson(backup));
writer.close();
```

#### ขั้นตอนการใช้งานการนำข้อมูลใน JSON มาสร้างเป็น Object ด้วย GSON

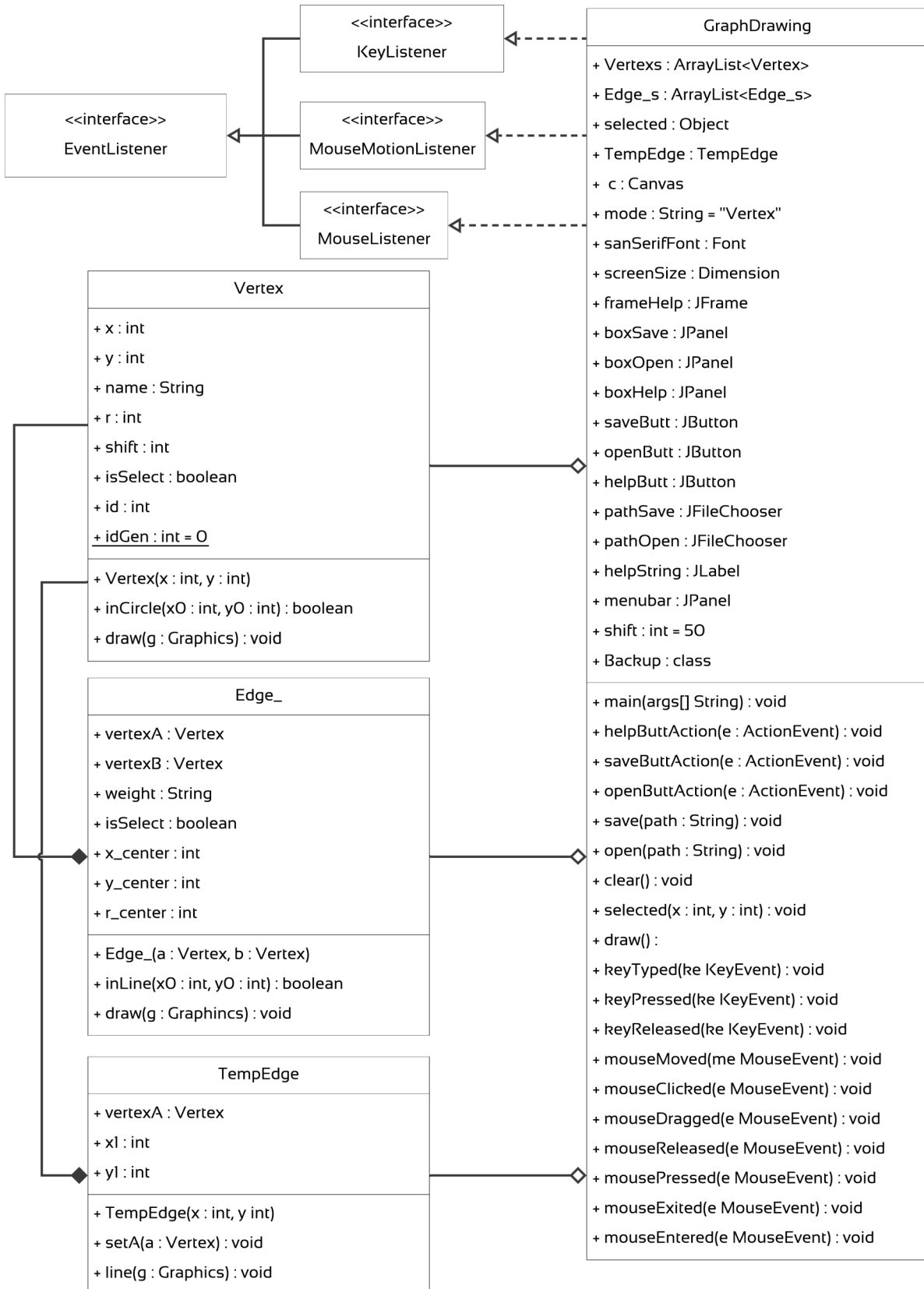
1. โหลด gson-2.8.2.jar มาจาก <https://jar-download.com/artifacts/com.google.code.gson/gson/2.8.2/source-code> จากนั้น Add jar เข้าไว้ใน Libraies ของโปรเจคที่สร้างไว้
2. อ่านไฟล์ json ตามที่อยู่ที่ต้องการในตัวแปร path ด้วย BufferedReader ดังนี้

```
BufferedReader bufferedReader = new BufferedReader(new FileReader(path));
```
3. 

```
import com.google.gson.Gson;
import com.google.gson.GsonBuilder;
```
4. ใช้เมธอด fromJson โดยใส่พารามิเตอร์คือ bufferedReader ที่อ่าน json ไว้และ Object.class ที่ต้องการสร้าง Object โดย Object ที่ว่าต้องเป็น Object เดียวกับที่นำมาสร้าง json ไฟล์ตั้งขั้นตอนก่อนหน้า จากนั้น fromJson จะ return Object ที่เรากำหนดไว้เป็นพารามิเตอร์ให้

```
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
isSelected	boolean	จุดยอดนี้ถูกเมาส์คลิกเลือกไว้หรือไม่
id	int	เก็บ id เฉพาะของ Vertex
idGen	int	สร้าง id เฉพาะ Vertex
Vertex(x:int,y:int)	-	r = 36 , shift = 30 , name = "" , isSelected = false
inCircle(x0:int,y0:int)	boolean	ตรวจสอบว่า 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)
isSelected	boolean	เส้นเชื่อมนี้ถูกเมาส์คลิกเลือกไว้หรือไม่
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 ที่สามารถคลิกเพื่อเลือกเส้นได้
Edge_(a:Vertex,b:Vertex)	-	r_center = 50 , weight = "1" , isSelected = false
inLine(x0:int,y0:int)	boolean	ตรวจสอบว่า 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 ของปลายเส้นเชื่อมปัจจุบันที่เมาส์ขี้อยู่
y1	int	พิกัดแกน y ของปลายเส้นเชื่อมปัจจุบันที่เมาส์ขี้อยู่
TempEdge(x:int,y:int)	-	-
setA(a:Vertex)	void	กำหนดค่า จุดยอดเริ่มต้น
line(g:Graphics2D)	void	วาดเส้นเชื่อม คือเส้นเชื่อมระหว่าง จุดยอดกับจุดปัจจุบันที่เมาส์ชี้ อาจจะเป็นเส้นเชื่อมวงก็ได้

### Class GraphDrawing

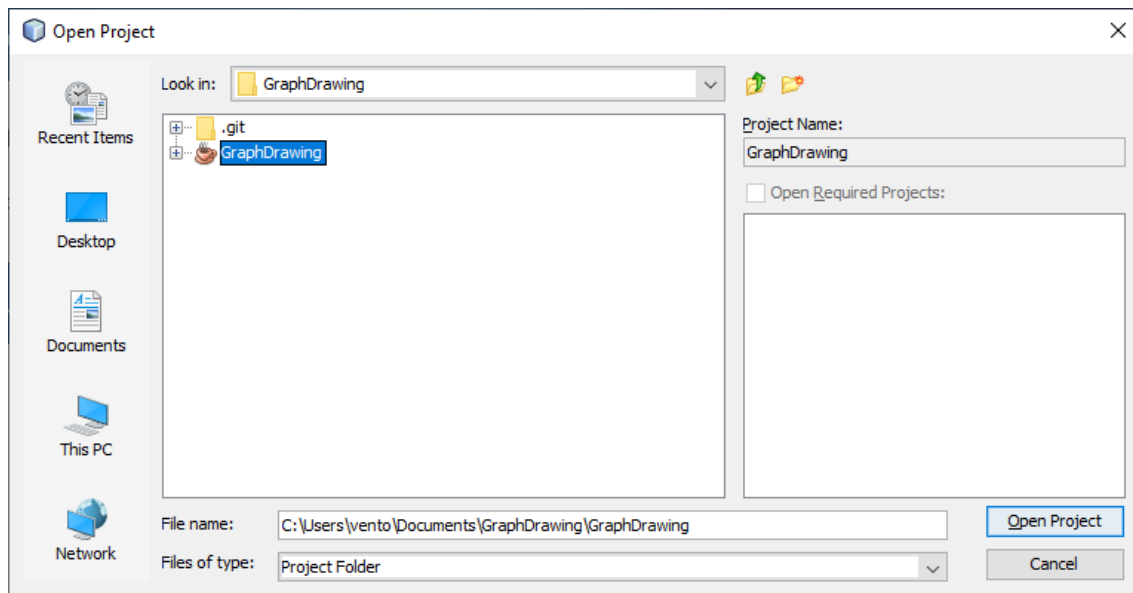
Field / Method	Type / Return type	Description
Vertexs	ArrayList<Vertex>	V(G) เอาไว้เก็บจุดยอดของกราฟ
Edge_s	ArrayList<Edge_s>	E(G) เอาไว้เก็บเส้นเชื่อมของกราฟ
Selected	Object	เอาไว้อ้างอิง จุดยอดหรือเส้นเชื่อมที่จะทำการแก้ไข โดยใช้เมาส์คลิกเลือก
TempEdge	TempEdge	เส้นเชื่อมที่อยู่ในขั้นตอนการวาด ยังไม่ถูกเพิ่มลงใน E(G)
c	Canvas	พื้นที่สำหรับวาดกราฟ
mode	String	มีค่าระหว่าง “Vertex” กับ “Edge_” ใช้เพื่อแบ่งโหมดการวาดกราฟ ระหว่างการจัดการเส้นเชื่อม และการจัดการจุดยอด ออกจากกัน กด space bar ค้างเพื่อให้อยู่ใน mode “Edge_” นอกจากนั้นจะอยู่ใน mode “Vertex”
sanSerifFont	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	ใช้สำหรับเลือกไฟล์ 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 แสดงหน้าต่างให้กำหนดตำแหน่งไฟล์ที่จะบันทึก กราฟ จากนั้นจะเรียกใช้ method save(path) เพื่อบันทึกกราฟ
openButtAction(e:ActionEvent)	void	การทำงานเมื่อกดปุ่ม open จะแสดงหน้าต่างให้เลือกตำแหน่งไฟล์ที่กราฟที่ จะนำเข้ามาเปิดในโปรแกรม จากนั้นจะเรียกใช้ method open(path) เพื่อ อ่านไฟล์กราฟ
save(path:String)	void	ใช้ Gson ในการแปลง object Backup เป็น json และบันทึกไฟล์ลงใน เครื่องตามตำแหน่งที่กำหนด
open(path:String)	void	อ่านไฟล์กราฟ แล้วใช้ Gson ในการแปลง json เป็น object Backup และ แสดงกราฟบนหน้าจอ
Clear()	void	ทำให้ canvas กลายเป็นสีขาวทั้งหมด

Selected(x:int,y:int)	void	รับพิกัด x,y ที่เมาส์คลิก ถ้าตำแหน่งที่คลิกคือ 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 vertex ถูกเลือกอยู่ การพิมพ์จะเป็นการพิมพ์ชื่อให้แก่ vertex ถ้า edge_ ถูกเลือกอยู่ การพิมพ์จะเป็นการพิมพ์ค่าให้แก่ weight ของเส้น
keyPressed(ke:KeyEvent)	void	ถ้ากด spacebar ค้างในระลอกที่กดค้างโปรแกรมจะอยู่ใน mode “Edge_”
keyReleased(ke:KeyEvent)	void	ถ้าเลิกกด keyboard โปรแกรมจะอยู่ใน mode “Vertex”
mouseMoved(me:MouseEvent)	void	-
mouseClicked(e:MouseEvent)	void	เมื่อคลิกเมาส์ลงไป 1 ครั้งจะเป็นการเลือกวัตถุในกราฟ (vertex , edge) ถ้า double click จะเป็นการสร้าง Vertex ใหม่ที่ตำแหน่งที่ double click
mouseDragged(e:MouseEvent)	void	การกดเมาส์ค้าง ถ้าอยู่ใน mode “Vertex” จะเป็นการเปลี่ยนตำแหน่งของเส้นเชื่อมหรือจุดยอดที่ถูกเลือกอยู่ ถ้าอยู่ใน mode “Edge_” จะเป็นการวาดเส้นเชื่อม แต่จะวาดได้เมื่อเริ่มกดจากภายในจุดยอด ไปสู่จุดยอดเท่านั้น
mouseReleased(e:MouseEvent)	void	การออกจากคลิก (เมื่อคลิกเสร็จแล้ว) ถ้าอยู่ใน mode “Vertex” จะยกเลิกการวาดเส้นเชื่อมที่ยังวาดไม่เสร็จ ถ้าอยู่ใน mode “Edge_” จะทำการสร้างเส้นเชื่อมขึ้นมา โดยมี 2 แบบ คือ เส้นเชื่อมระหว่าง 2 จุดยอดกับ เส้นเชื่อมวงวน
mousePressed(e:MouseEvent)	void	เมื่อเริ่มกดคลิกเมาส์ และอยู่ใน mode “Edge_” จะสร้างเส้นเชื่อมชั่วคราวขึ้นมาโดยจุดยอดเริ่มต้นคือจุดยอดในตำแหน่งที่เมาส์เริ่มคลิก ถ้าตำแหน่งนั้นไม่มีจุดยอด จะไม่สร้างเส้นเชื่อมขึ้นมาให้
mouseExited(e:MouseEvent)	void	-
mouseEntered(e:MouseEvent)	void	-

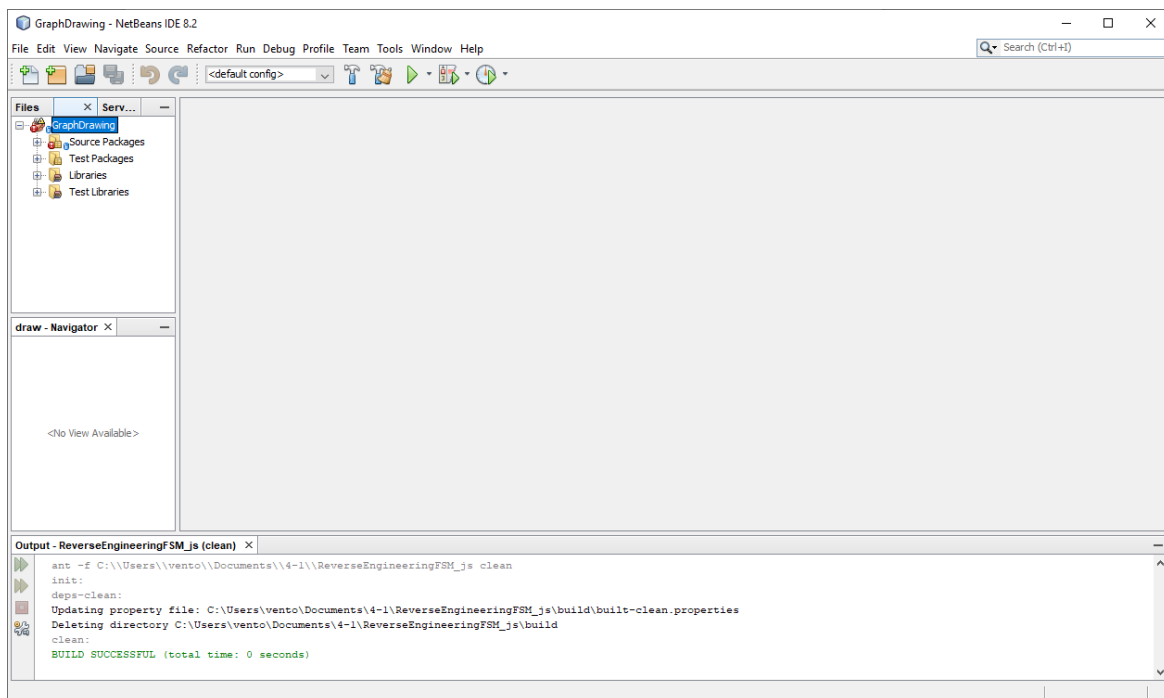
## การ compile หรือ run โปรแกรม

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



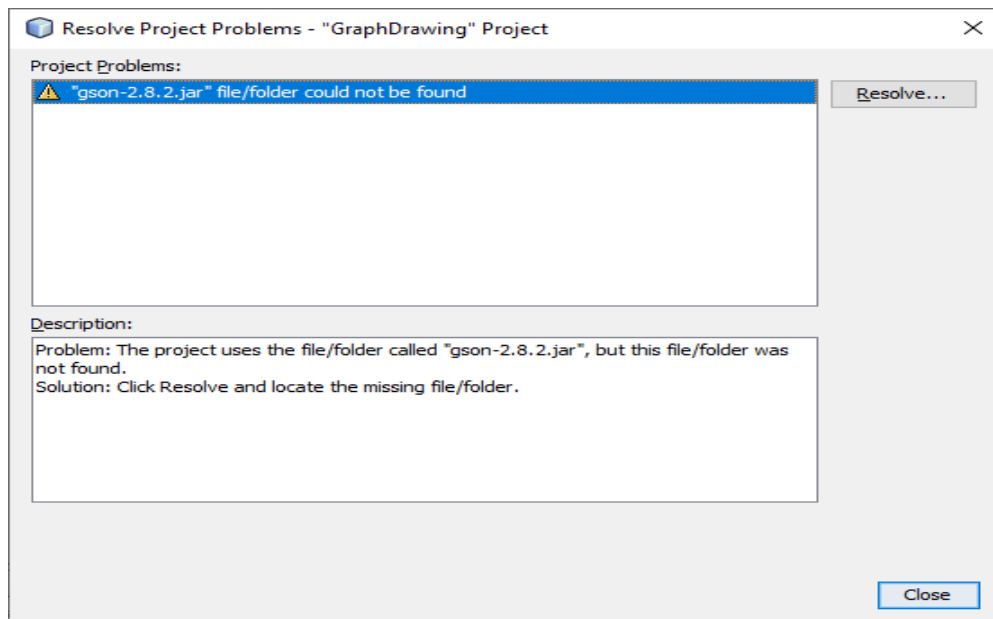
รูปที่ 1 แสดงการ open Project

2. ให้กดคลิกขวาที่ ชื่อโปรเจกต์ที่เปิดแล้วจากนั้นกด Resolve Project Problems ..เพื่อเพิ่ม GSON เข้ามาในโปรเจกต์



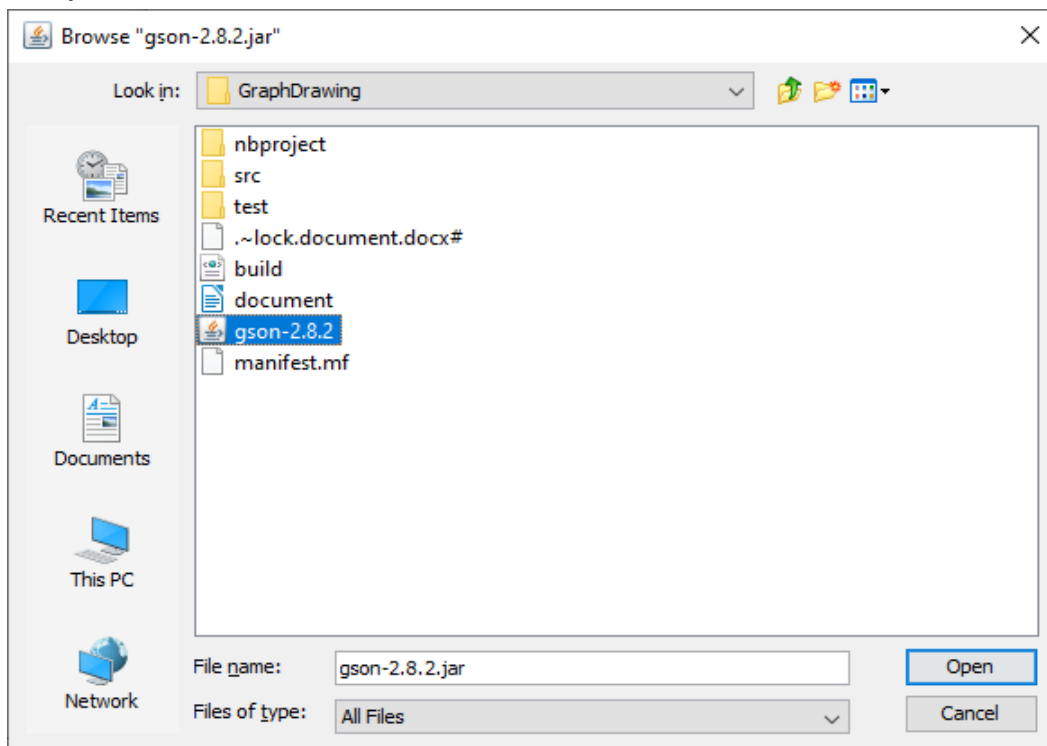
รูปที่ 2 แสดงการเพิ่ม GSON เข้ามาในโปรเจกต์

### 3. กด Resolve



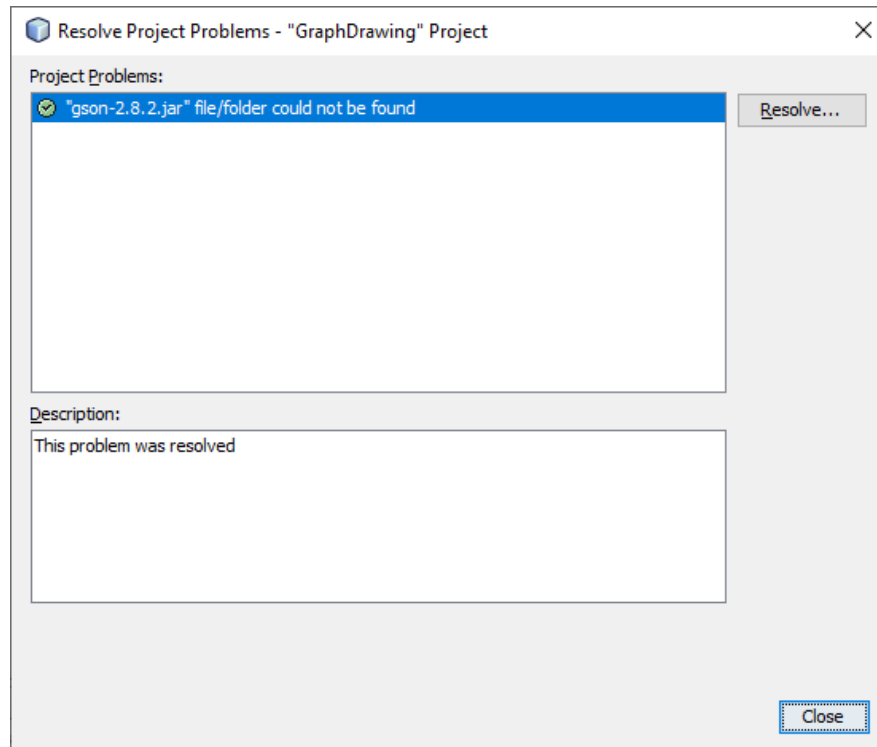
รูปที่ 3 แสดงหน้าต่าง Resolve

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



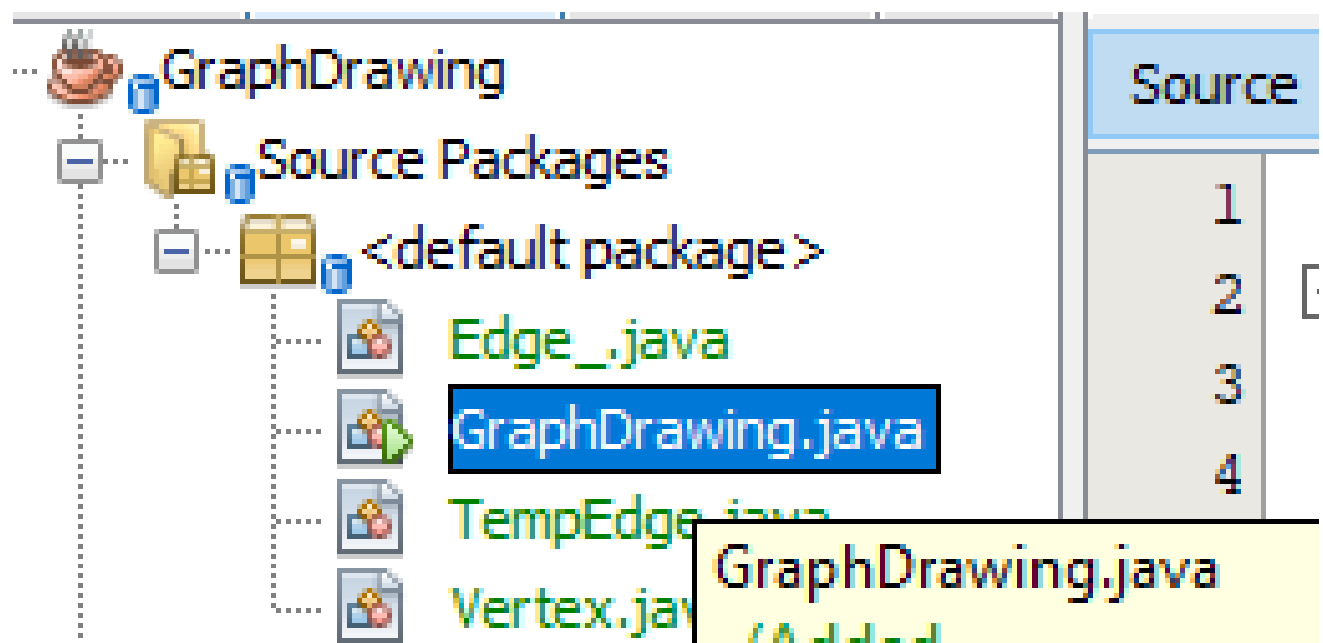
รูปที่ 4 แสดงการเปิดไฟล์ gson-2.8.2.jar

5 กด Close



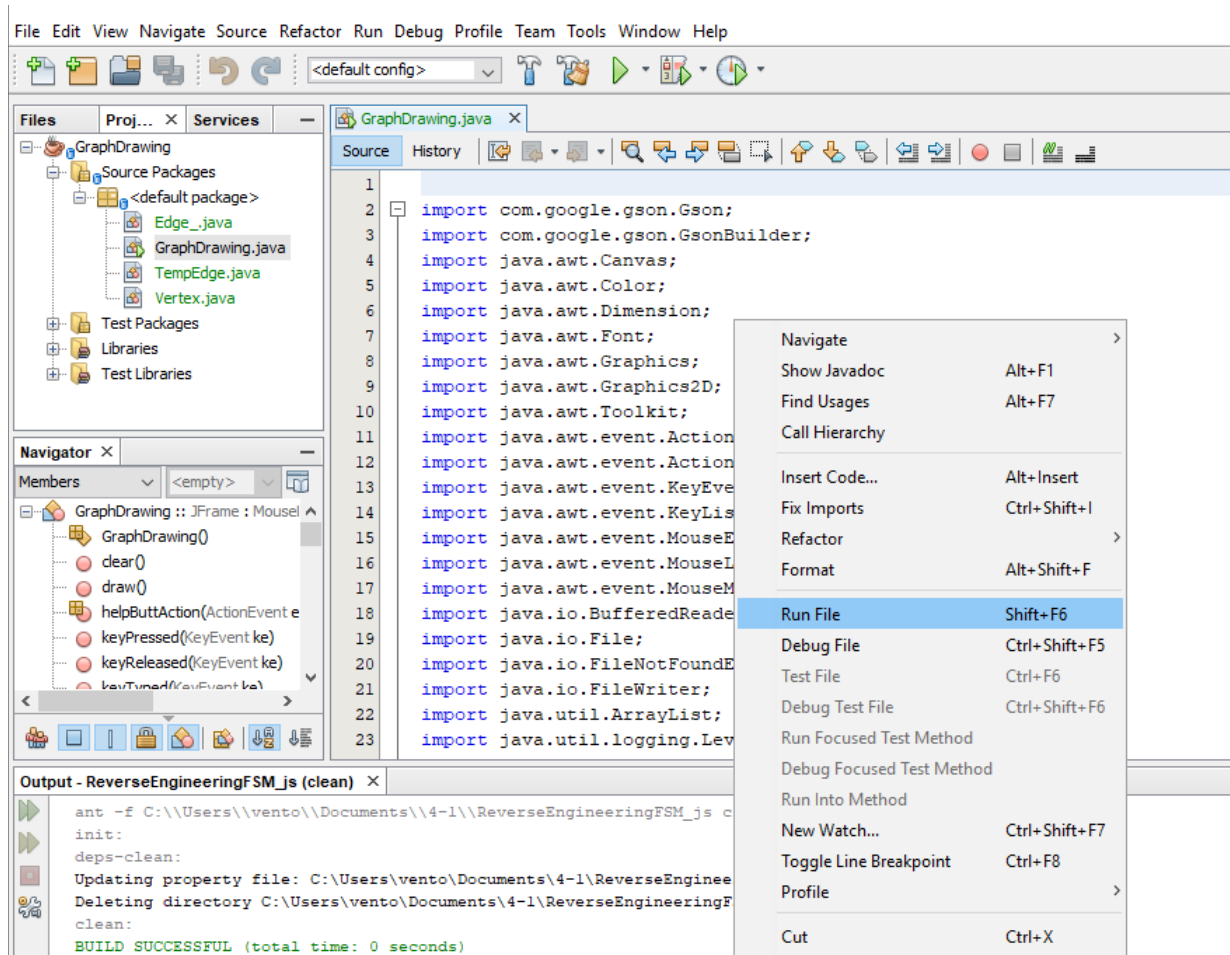
รูปที่ 1 แสดงหน้าต่างหลังจากเพิ่ม GSON

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



รูปที่ 6 แสดงการเปิดไฟล์ GraphDrawing.java

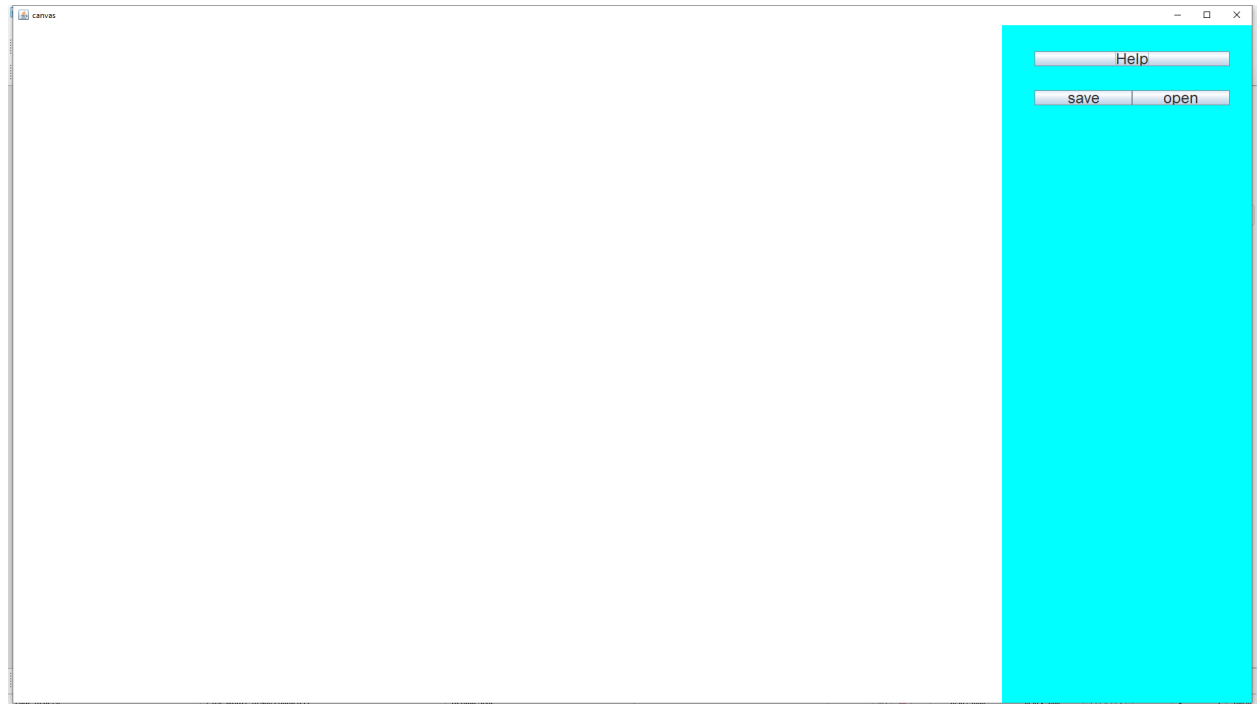
## 7. คลิกขวาที่ส่วนของโค้ดแล้วกด Run File (Shift + F6)



รูปที่ 7 แสดงวิธีการกดรัน



## 8. โปรแกรมรันหน้า GUI ขึ้นมาดังนี้

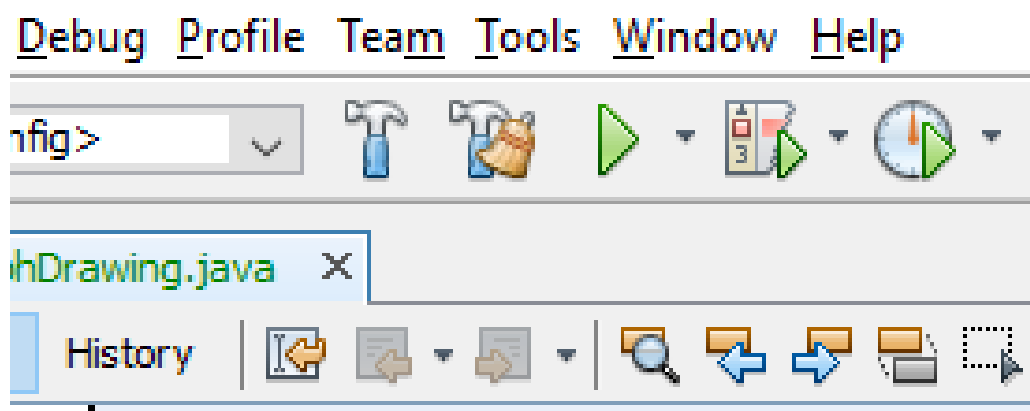


รูปที่ 8 แสดงตัวอย่างหน้า GUI

## 9. หากต้องการสร้างไฟล์ .jar ให้กด

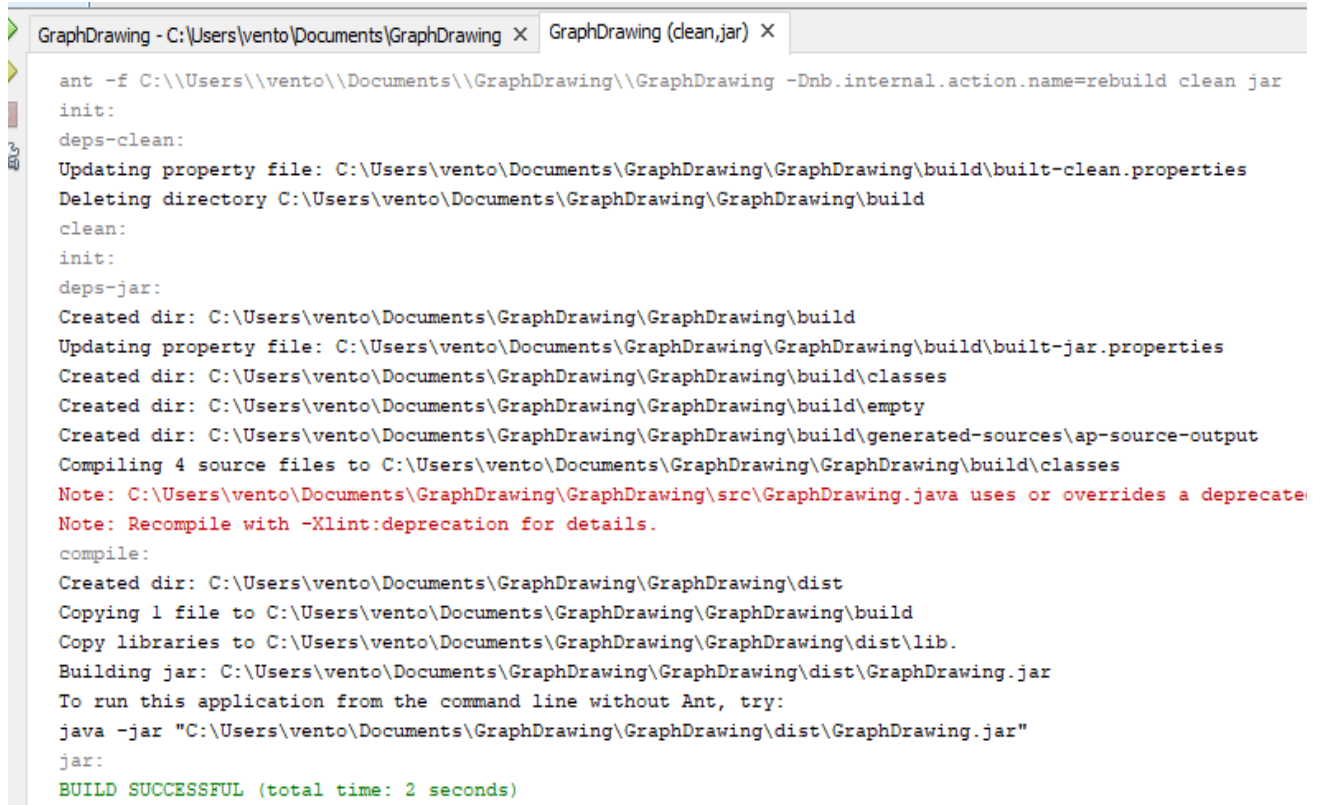


Clean and Build Project (Shift + F11)



รูปที่ 9 แสดงวิธีการสร้างไฟล์ .jar

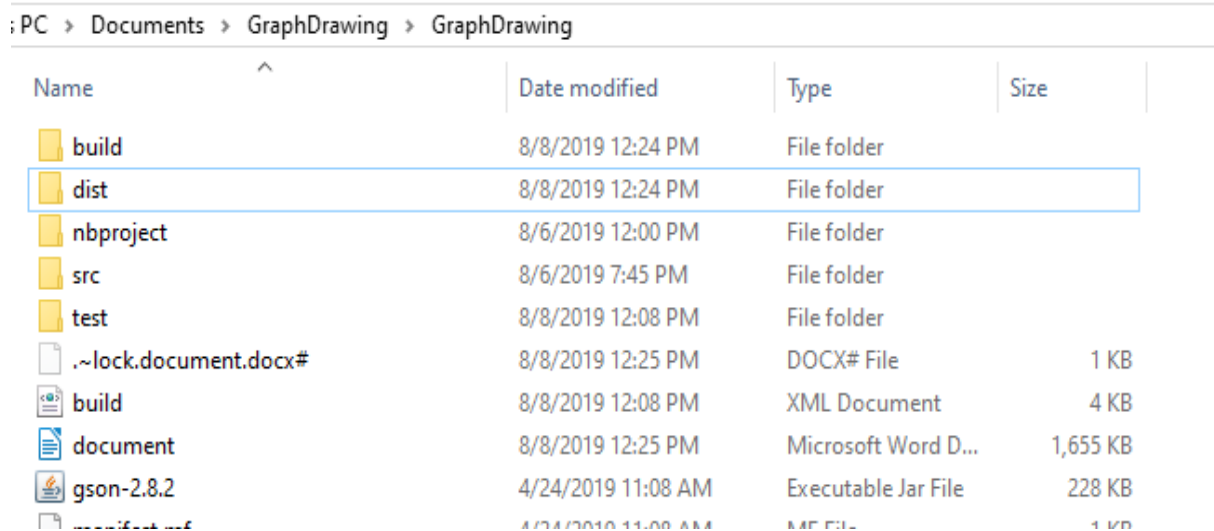
## 10. แสดงสถานะการ Build Project



```
GraphDrawing - C:\Users\vento\Documents\GraphDrawing X GraphDrawing (clean,jar) X
ant -f C:\Users\vento\Documents\GraphDrawing\GraphDrawing -Dnb.internal.action.name=rebuild clean jar
init:
deps-clean:
Updating property file: C:\Users\vento\Documents\GraphDrawing\GraphDrawing\build\build-clean.properties
Deleting directory C:\Users\vento\Documents\GraphDrawing\GraphDrawing\build
clean:
init:
deps-jar:
Created dir: C:\Users\vento\Documents\GraphDrawing\GraphDrawing\build
Updating property file: C:\Users\vento\Documents\GraphDrawing\GraphDrawing\build\build-jar.properties
Created dir: C:\Users\vento\Documents\GraphDrawing\GraphDrawing\build\classes
Created dir: C:\Users\vento\Documents\GraphDrawing\GraphDrawing\build\empty
Created dir: C:\Users\vento\Documents\GraphDrawing\GraphDrawing\build\generated-sources\ap-source-output
Compiling 4 source files to C:\Users\vento\Documents\GraphDrawing\GraphDrawing\build\classes
Note: C:\Users\vento\Documents\GraphDrawing\GraphDrawing\src\GraphDrawing.java uses or overrides a deprecated
Note: Recompile with -Xlint:deprecation for details.
compile:
Created dir: C:\Users\vento\Documents\GraphDrawing\GraphDrawing\dist
Copying 1 file to C:\Users\vento\Documents\GraphDrawing\GraphDrawing\build
Copy libraries to C:\Users\vento\Documents\GraphDrawing\GraphDrawing\dist\lib.
Building jar: C:\Users\vento\Documents\GraphDrawing\GraphDrawing\dist\GraphDrawing.jar
To run this application from the command line without Ant, try:
java -jar "C:\Users\vento\Documents\GraphDrawing\GraphDrawing\dist\GraphDrawing.jar"
jar:
BUILD SUCCESSFUL (total time: 2 seconds)
```

รูปที่ 10 แสดงสถานะการสร้าง

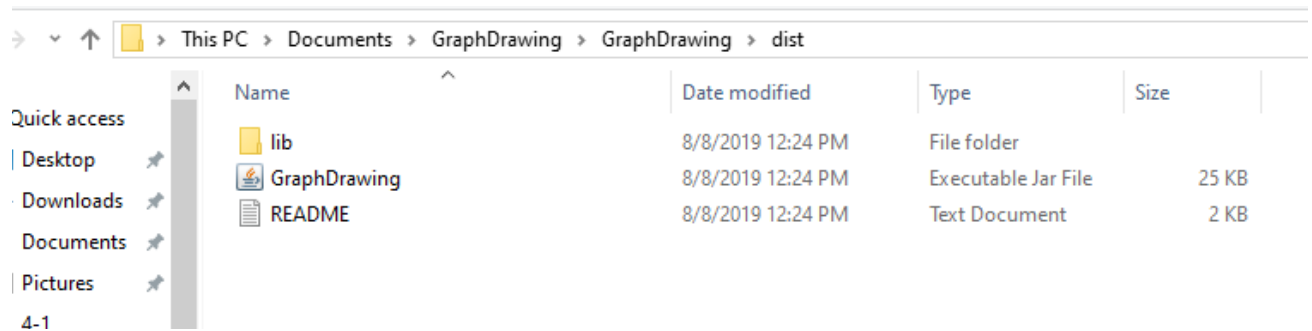
## 11. หลังจากขั้นตอน 10 จะมีโฟลเดอร์ dist ปรากฏขึ้นมา



; PC > Documents > GraphDrawing > GraphDrawing				
Name	^	Date modified	Type	Size
build		8/8/2019 12:24 PM	File folder	
dist		8/8/2019 12:24 PM	File folder	
nbproject		8/6/2019 12:00 PM	File folder	
src		8/6/2019 7:45 PM	File folder	
test		8/8/2019 12:08 PM	File folder	
..lock.document.docx#		8/8/2019 12:25 PM	DOCX# File	1 KB
build		8/8/2019 12:08 PM	XML Document	4 KB
document		8/8/2019 12:25 PM	Microsoft Word D...	1,655 KB
gson-2.8.2		4/24/2019 11:08 AM	Executable Jar File	228 KB
manifest.mf		4/24/2019 11:08 AM	MF File	1 KB

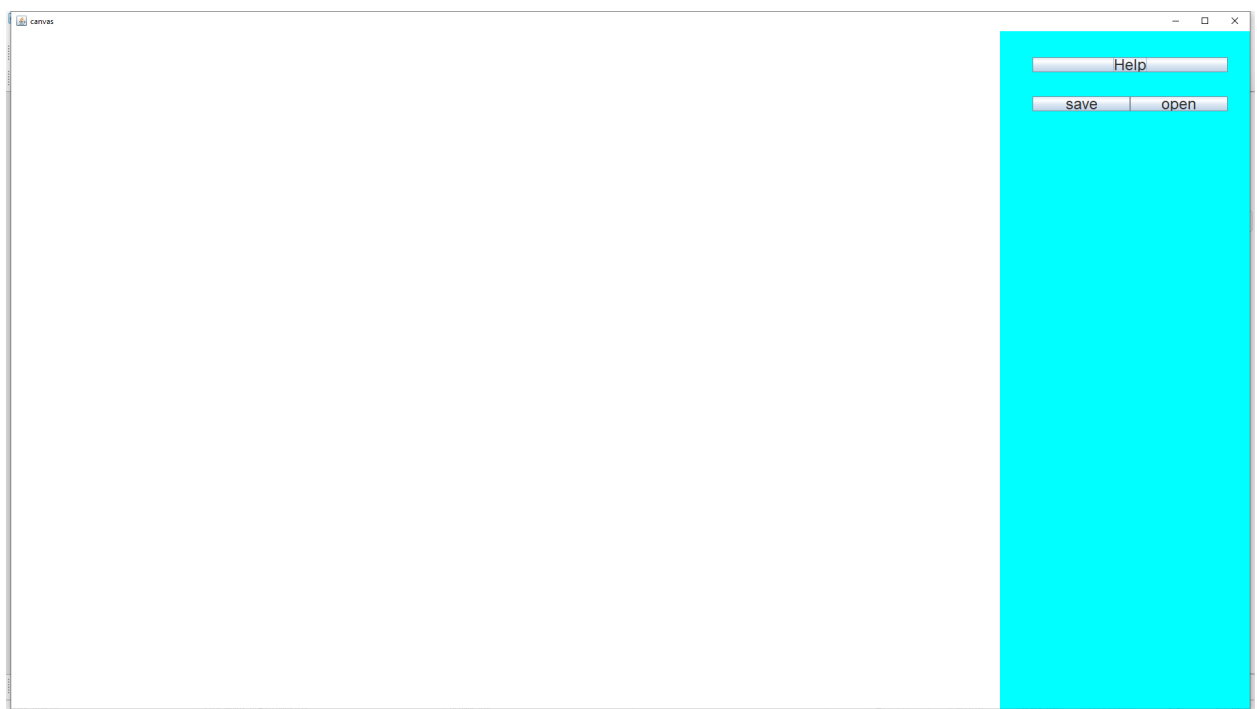
รูปที่ 11 แสดงโฟลเดอร์เมื่อสร้างสำเร็จ

12. เมื่อเข้าไปใน Folder dist ก็จะมีการโปรแกรม 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