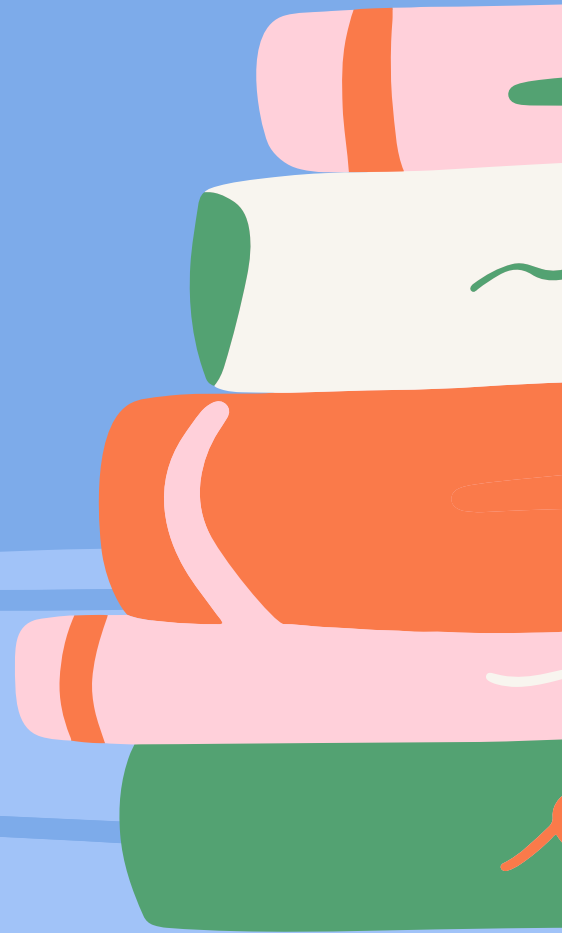


ชั้นคอมมี้

เลขชงัฒัฒรฒมม

กลุ่ม 1



ความสำคัญและที่มา



- ความยากในการอ่านรูปสรุปผลการแข่งขัน แล้วดูว่า ใครชนะใคร
- ต้องการความสะดวกในผู้ที่สนใจการแข่งขันในการแข่งขันหนึ่ง
- โดยใช้เป็นโปรแกรมที่รับ Input เป็นชื่อผู้เข้าแข่งขัน แล้ว Return ชื่อผู้เข้าแข่งขันที่ชนะมา
- ข้อมูล Tournament ใช้ข้อมูลจาก Janken Tournament 2020 (แข่งเป่ายิ้งฉุบ)



การประยุกต์ใช้โครงสร้างข้อมูล

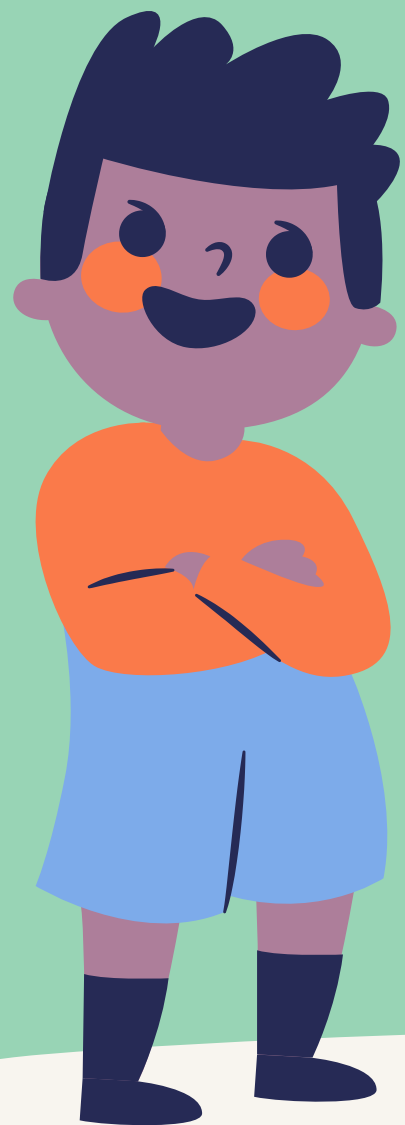
- ใช้การ Implement จาก Graph แบบ Directed Graph
- ที่ไม่ได้ใช้ Tree เป็นเพราะต้องการแสดงให้เห็นถึงทิศทางว่า หัวลูกศรชี้ไปที่ใคร โดยนั้นจะหมายความว่า คนที่ชี้ไป ชนะ คนชี้ถูกชี้
- Node ของแต่ละ Graph จะเก็บข้อมูล ดังนี้
 - Incoming เก็บเป็น Array List โดยเป็น Array ของ Node ที่ชี้มาหาตัวเอง
 - Data เก็บเป็นข้อมูลของ Node ตัวเอง โดยในงานนี้ เก็บเป็นข้อมูล Member
 - Outcoming เก็บเป็น Array List โดยเป็น Array ของ Node ที่ชี้ไปหา Node ตัวอื่น

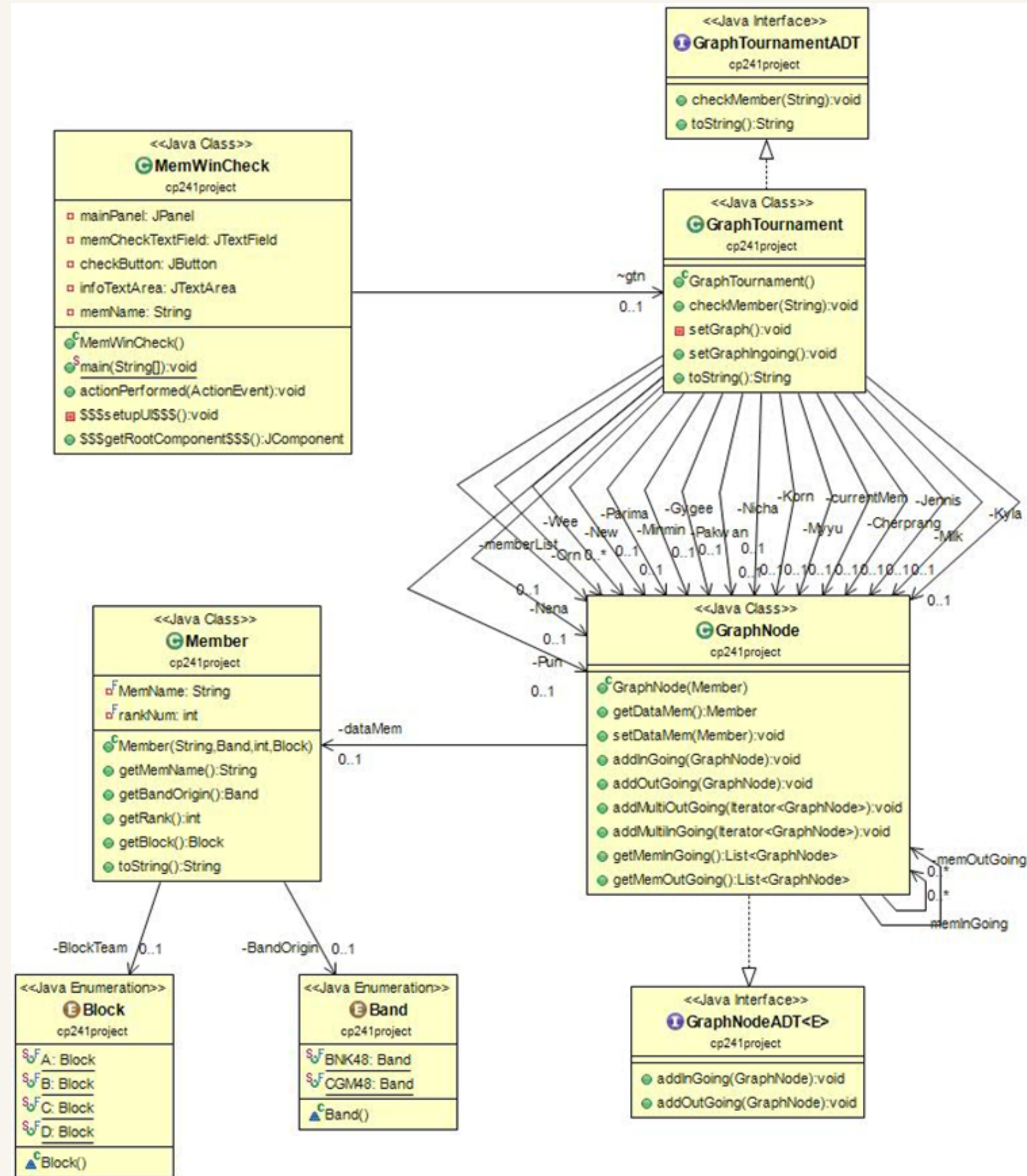




การดำเนินการ ด้วยภาษาจาวา

- Graph Node
- การสร้าง Graph
- Data
- ทำงานหลัก (Driver Class)





The background is a light cream color with various abstract elements. In the top left, there is a dark grey circle, a large green circle, a thin black circle, and a series of concentric blue circles. In the top right, there are four diagonal brush strokes in orange, pink, blue, and grey. In the bottom left, there are three diagonal brush strokes in pink, blue, and purple. In the bottom right, there are several grey and blue circles of different sizes, some with outlines, and a small blue and orange brush stroke. At the very bottom, there is a solid blue horizontal bar with some small, dark blue and orange shapes on it.

Graph Node

GraphNodeADT.java



```
1 package cp241project;
2
3 import java.util.Iterator;
4
5 public interface GraphNodeADT<E> {
6     public void addInGoing(GraphNode data);
7     public void addOutGoing(GraphNode data);
8 }
```

//เป็นการ Import เพื่อใช้ไลบรารี Iterator

//ประกาศ Method addInGoing โดยเก็บข้อมูล GraphNode เป็น data
//ประกาศ Method addOutGoing โดยเก็บข้อมูล GraphNode เป็น data



GraphNode.java



```
1 package cp241project;
2
3 import java.util.*;
4
5 public class GraphNode implements GraphNodeADT{
6     private Member dataMem;
7     private List<GraphNode> memOutGoing;
8     private List<GraphNode> memInGoing;
9     public GraphNode(Member memIn){
10         dataMem = memIn;
11         memOutGoing = new ArrayList<>();
12         memInGoing = new ArrayList<>();
13     }
14
15     public Member getDataMem(){
16         return dataMem;
17     }
18     public void setDataMem(Member memIn){
19         this.dataMem = memIn;
20     }
21     @Override
22     public void addInGoing(GraphNode gNode) {
23         memInGoing.add(gNode);
24     }
25
26     @Override
27     public void addOutGoing(GraphNode gNode) {
28         memOutGoing.add(gNode);
29     }
30 }
```

//เป็นการ Import เพื่อใช้ไลบรารี util
//ประกาศ Data Member ของ Class GraphNode
//ประกาศตัวแปรชนิด Member เป็น dataMem
//ประกาศตัวแปรที่เก็บค่าว่าชนะใครมาบ้าง โดยใช้ list ชื่อว่า GraphNode
//ประกาศตัวแปรที่เก็บค่าว่าแพ้ใครมาบ้าง โดยใช้ list ชื่อว่า GraphNode
//สร้าง Constructor ของ GraphNode โดยที่มี Data เป็น Member
//กำหนดให้ตัวแปร dataMem เก็บข้อมูลจาก memIn
//สร้าง Array List ชื่อว่า memOutGoing
//สร้าง Array List ชื่อว่า memInGoing

//ประกาศ Method getDataMem ของ Class GraphNode
//ส่งค่า dataMem ของโหนดนี้ คืนให้ตัวแปรที่เรียกใช้

//ประกาศ method setDataMem ให้ดึงค่า Member memIn
//กำหนดให้ตัวแปร this.dataMem เก็บข้อมูลจาก memIn

//ประกาศ Method addInGoing ที่รับ Parameter เป็น GraphNode ชื่อ gNode
/*นำ gNode ที่รับเข้ามา เพิ่มไปยัง Array List ชื่อ memInGoing เพื่อเก็บข้อมูลว่า
คนนี้แพ้ใครมาบ้าง*/

//ประกาศ Method addOutGoing ที่รับ Parameter เป็น GraphNode ชื่อ gNode
/*นำ gNode ที่รับเข้ามา เพิ่มไปยัง Array List ชื่อ memOutGoing เพื่อเก็บข้อมูลว่า
คนนี้ชนะใครมาบ้าง*/



GraphNode.java

```
31
32 public void addMultiOutGoing(Iterator<GraphNode> MemOut) {
33     while (MemOut.hasNext()){
34         addOutGoing(MemOut.next());
35     }
36
37 }
38
39
40 public void addMultiInGoing(Iterator<GraphNode> MemIn) {
41     while (MemIn.hasNext()){
42         addInGoing(MemIn.next());
43     }
44 }
45
46 public List<GraphNode> getMemInGoing(){
47     return memInGoing;
48 }
49
50 public List<GraphNode> getMemOutGoing(){
51     return memOutGoing;
52 }
53 }
```


//ประกาศ Method addMultiOutGoing รับค่าข้อมูลเป็นชนิด Iterator<GraphNode>
//สร้างลูป while loop โดยดึง MemOut มาทีละตัว และทำซ้ำไปเรื่อย ๆ จนครบทุกตัว
/*เพิ่ม MemOut ที่ GraphNode ในส่วน Out Going Node (ตัวที่ชี้ไปหา Node อื่น)
ด้วยการเรียก Method addOutGoing เพื่อเพิ่มข้อมูลเข้า ArrayList memOutGoing
ทีละตัว ตามแต่ละรอบลูป*/

//ประกาศ Method addMultiInGoing รับค่าข้อมูลเป็นชนิด Iterator<GraphNode>
//สร้างลูป while loop โดยดึง MemIn มาทีละตัว และทำซ้ำไปเรื่อย ๆ จนครบทุกตัว
/*ทำการเพิ่ม MemIn ที่ GraphNode ในส่วน In Going Node
(ตัวที่ Node อื่นชี้มาหาเรา) ด้วยการเรียก Method addInGoing
เพื่อเพิ่มข้อมูลเข้า ArrayList memInGoing ทีละตัว ตามแต่ละรอบลูป*/

//ประกาศ Method getMemInGoing() โดยให้คืนค่าเป็นชนิด List<GraphNode>
//คืนค่าข้อมูล memInGoing ที่เก็บค่าขณะนี้แพ้นใครมา

//ประกาศ Method getMemOutGoing() โดยให้คืนค่าเป็นชนิด List<GraphNode>
//คืนค่าข้อมูล memOutGoing ที่เก็บค่าขณะนี้แพ้นใครมา





การสร้าง Graph



GraphTournamentADT.java

```
1 package cp241project;
2
3 public interface GraphTournamentADT {
4     public void checkMember(String nameIn) throws NullPointerException;
5
6     public String toString();
7 }
```

/*ประกาศ Method checkMember ตาม String ของ nameIn และ throw ชื่อว่า
NullPointerException*/
//ประกาศชื่อ Method ที่ Return เป็น String โดยใช้ชื่อ toString



GraphTournament.java

```
1 package cp241project;
2
3 import java.util.*;
4
5 2 usages
6 public class GraphTournament implements GraphTournamentADT {
7     private final GraphNode Gygee = new GraphNode(new Member("Gygee", Band.BNK48, 1, Block.A));
8     private final GraphNode Wee = new GraphNode(new Member("Wee", Band.BNK48, 2, Block.C));
9     private final GraphNode Orn = new GraphNode(new Member("Orn", Band.BNK48, 3, Block.B));
10    private final GraphNode Cherprang = new GraphNode(new Member("Cherprang", Band.BNK48, 4, Block.D));
11    private final GraphNode Kyla = new GraphNode(new Member("Kyla", Band.CGM48, 5, Block.D));
12    private final GraphNode Myyu = new GraphNode(new Member("Myyu", Band.BNK48, 6, Block.B));
13    private final GraphNode Milk = new GraphNode(new Member("Milk", Band.CGM48, 7, Block.C));
14    private final GraphNode Pun = new GraphNode(new Member("Pun", Band.BNK48, 8, Block.A));
15    private final GraphNode Korn = new GraphNode(new Member("Korn", Band.BNK48, 9, Block.A));
16    private final GraphNode Nicha = new GraphNode(new Member("Nicha", Band.CGM48, 10, Block.C));
17    private final GraphNode Minmin = new GraphNode(new Member("Minmin", Band.BNK48, 11, Block.B));
18    private final GraphNode Pakwan = new GraphNode(new Member("Pakwan", Band.BNK48, 12, Block.D));
19    private final GraphNode New = new GraphNode(new Member("New", Band.BNK48, 13, Block.D));
20    private final GraphNode Parima = new GraphNode(new Member("Parima", Band.CGM48, 14, Block.A));
21    private final GraphNode Nena = new GraphNode(new Member("Nena", Band.CGM48, 15, Block.C));
22    private final GraphNode Jennis = new GraphNode(new Member("Jennis", Band.BNK48, 16, Block.B));
23    private GraphNode currentMem;
24    private final GraphNode[] memberList = {Gygee, Wee, Orn, Cherprang, Kyla, Myyu, Milk, Pun, Korn, Nicha,
25        Minmin, Pakwan, New, Parima, Nena, Jennis};
26
27    public GraphTournament(){
28        setGraph();
29    }
```

//ประกาศ + สร้างข้อมูล GraphNode ของกราฟนี้

//ประกาศตัวแปร GraphNode เพื่อเตรียมเอาไปใช้ในการชี้ตัวที่เราสนใจ
//สร้าง Array ของ GraphNode เพื่อเลือกสมาชิกของ currentMem

//สร้าง Constructor ของ GraphTournament
//เรียก method setGraph มา





GraphTournament.java

```
30 public void checkMember(String nameIn) throws NullPointerException{
31     currentMem = null;
32     for (GraphNode member : memberList) {
33         if (nameIn.equalsIgnoreCase(member.getDataMem().getMemName())) {
34             currentMem = member;
35             break;
36         }
37     }
38     if(currentMem == null){
39         throw new NullPointerException( "Not found or not in top-16: Please type again");
40     }
41 }
42 }
```

/*ประกาศ Method checkMember โดยรับข้อมูลเป็น String จาก nameIn + throws NullPointerException เพื่อดักจับข้อมูลที่เป็น null หรืออยู่นอกเหนือจาก Array ของ memberList */
//กำหนดให้ currentMem มีค่าเป็น null ก่อน
/*ดึงข้อมูลใน Array memberList มาใส่ใน member เพื่อกำหนด currentMem*/
/*ถ้ามีที่เหมือนกัน ก็ทำการกำหนด currentMem ชี้ไปที่ Node ของ Member ที่เจอ แล้วหลุดจากลูปด้วยการ break*/

//ทำการตรวจสอบเงื่อนไขหลังจากออกจากลูปว่า currentMem ยังเป็น null อยู่หรือไม่
/*ถ้ายังเป็น null (ที่พิมพ์ไป ไม่ตรงกับชื่อ Member ที่เก็บไว้ใน Array memberList) ก็ให้โปรแกรมโยน NullPointerException ไป*/





GraphTournament.java

```
43 private void setGraph(){ //Set relation between winner and loser.
44     ArrayList<GraphNode> GygeeWin = new ArrayList<>(Arrays.asList(Wee,Orn,Pun,Parima));
45     ArrayList<GraphNode> WeeWin = new ArrayList<>(Arrays.asList(Cherprang,Milk,Nena));
46     ArrayList<GraphNode> CherprangWin = new ArrayList<>(Arrays.asList(Kyla,New));
47     ArrayList<GraphNode> MilkWin = new ArrayList<>(Arrays.asList(Nicha));
48     ArrayList<GraphNode> OrnWin = new ArrayList<>(Arrays.asList(Myyu,Minmin));
49     ArrayList<GraphNode> MyyuWin = new ArrayList<>(Arrays.asList(Jennis));
50     ArrayList<GraphNode> KylaWin = new ArrayList<>(List.of(Pakwan));
51     ArrayList<GraphNode> PunWin = new ArrayList<>(List.of(Korn));
52     Gygee.addMultiOutGoing(GygeeWin.iterator());
53     Wee.addMultiOutGoing(WeeWin.iterator());
54     Cherprang.addMultiOutGoing(CherprangWin.iterator());
55     Milk.addMultiOutGoing(MilkWin.iterator());
56     Orn.addMultiOutGoing(OrnWin.iterator());
57     Myyu.addMultiOutGoing(MyyuWin.iterator());
58     Kyla.addMultiOutGoing(KylaWin.iterator());
59     Pun.addMultiOutGoing(PunWin.iterator());
60     setGraphIngoing();
61 }
62
```

//ประกาศเมธอด setGraph

/*ประกาศ ArrayList เป็น GraphNode ที่กำหนดค่าเป็น GraphNode ของ Member ที่แต่ละ Member ชนะมา*/

/*ให้แต่ละโหนด กำหนด Outgoing Node (คนที่ชนะ) ไปแต่ละตัว ตาม ArrayList ที่กำหนดไว้บรรทัดก่อนหน้านี้*/

/*เรียกใช้ Method setGraphIngoing เพื่อกำหนดคนที่แพ้ของแต่ละโหนด*/





GraphTournament.java

//กำหนดตัวชี้ไปที่โหนดชื่อของคนที่แพ้แต่ละคน

```
63     public void setGraphIngoing() {  
64         Wee.addIngoing(Gygee);  
65         Cherprang.addIngoing(Wee);  
66         Milk.addIngoing(Wee);  
67         Orn.addIngoing(Gygee);  
68         Myyu.addIngoing(Orn);  
69         Kyla.addIngoing(Cherprang);  
70         Pun.addIngoing(Gygee);  
71         Pakwan.addIngoing(Kyla);  
72         New.addIngoing(Cherprang);  
73         Nena.addIngoing(Wee);  
74         Nicha.addIngoing(Milk);  
75         Jennis.addIngoing(Myyu);  
76         Minmin.addIngoing(Orn);  
77         Korn.addIngoing(Pun);  
78         Parima.addIngoing(Gygee);  
79     }  
80
```

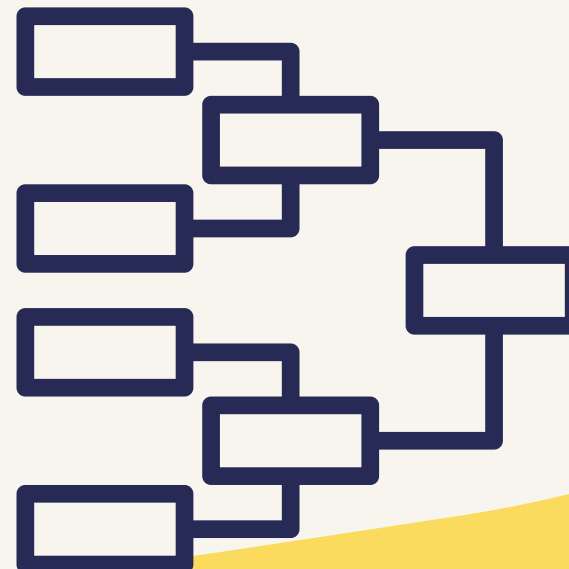




GraphTournament.java

```
81 public String toString(){
82     StringBuilder prepareReturn = new StringBuilder();
83     prepareReturn.append(String.format("\t%s\tfrom%s\tBlock%2s\tis rank%3d\n", currentMem.getDataMem().getMemName(),
84         currentMem.getDataMem().getBandOrigin(), currentMem.getDataMem().getBlock(),
85         currentMem.getDataMem().getRank()));
86     prepareReturn.append("\n\tHas win with\n");
87     List<GraphNode> MemWin = currentMem.getMemOutGoing();
88     if(MemWin.isEmpty()){
89         prepareReturn.append("\tNo One in the rank of top-16\n");
90     } else{
91         for(GraphNode x:MemWin){
92             prepareReturn.append(String.format("\t- %s\tfrom%s\tBlock%2s\tRank%3d\n", x.getDataMem().getMemName(),
93                 x.getDataMem().getBandOrigin(), x.getDataMem().getBlock(), x.getDataMem().getRank()));
94         }
95     }
96 }
```

//ประกาศเมธอด toString() โดยห้คืนค่าข้อมูลเป็น String
//สร้าง StringBuilder ที่ชื่อว่า prepareReturn
/*ให้ prepareReturn เพิ่มข้อมูล ชื่อสมาชิก ชื่อวง สายการแข่งขัน
และลำดับ ที่ชื่ออยู่ในปัจจุบัน*/
//ให้ prepareReturn เพิ่มข้อมูล "Has win with"
//สร้างตัวแปร MemWin ที่ชนิดข้อมูลเป็น List<GraphNode> ให้ดึง
ข้อมูลว่าคนที่ชื่ออยู่ในปัจจุบันชนะใครมาบ้าง
//prepareReturn เพิ่มเพิ่มข้อมูลว่า ไม่ได้ชนะกับใครใน 16 อันดับแรก
/*กรณีที่เป็นเท็จ ให้ทำงานในส่วน else ที่เป็นการดึงข้อมูลสมาชิกแต่ละ
คน มาแสดงผล*/
//สร้าง for loop โดยให้วนลูปโดยดึงค่าจาก MemWin ที่ละตัว
/*ให้ตัวแปร prepareReturn เพิ่มข้อมูล ชื่อสมาชิก ชื่อวง สายการแข่งขัน
และลำดับ ของ MemWin แต่ละตัว*/





GraphTournament.java

```
96
97     List<GraphNode> Memlose = currentMem.getMemInGoing();
98
99     if(Memlose.isEmpty()){
100         prepareReturn.append("\n\twinner\n");
101     } else{
102         prepareReturn.append("\n\tHas lose with\n");
103
104         for(GraphNode x:Memlose){
105             prepareReturn.append(String.format("\t- %s\tfrom%6s\tBlock%2s\tRank%3d\n", x.getDataMem().getMemName(),
106                 x.getDataMem().getBandOrigin(), x.getDataMem().getBlock(), x.getDataMem().getRank()));
107         }
108     }
109
110     return prepareReturn.toString();
111 }
112 }
```

//สร้าง Memlose ที่มีชนิดเป็น List<GraphNode> ให้ดึง
ข้อมูลว่าคนที่ชื่อยู่ในปัจจุบันแพ้ใครมาบ้าง
//สร้างเงื่อนไข if ว่าถ้า Memlose ว่าง ก็จะเป็นจริง ให้ทำเงื่อนไข
ด้านใน if
//ให้ตัวแปร prepareReturn เพิ่มข้อมูลว่า เป็น Winner ของ
Tournament นี้
/*กรณีที่เป็นเท็จ ให้ทำงานในส่วน else ที่เป็นการดึงข้อมูล
สมาชิกแต่ละคน มาแสดงผล*/
//ให้ตัวแปร prepareReturn เพิ่มข้อมูล "Has lose with"
//สร้าง for loop ใหวนลูปโดยดึงค่าจาก Memlose ที่ละตัว
/*ให้ตัวแปร prepareReturn เพิ่มข้อมูล ชื่อสมาชิก ชื่อวง
สายการแข่งขัน และลำดับ ของ Memlose แต่ละตัว*/
//คืนค่าตัวแปร prepareReturn โดยแปลงให้เป็น String ก่อน



An abstract graphic design featuring a central light blue rounded rectangle with a dark blue border. Inside this rectangle, the word "Data" is written in a bold, black, sans-serif font. The background is a light cream color, decorated with various geometric shapes: a large green circle, a dark grey circle, a teal circle, a series of concentric blue circles, and several diagonal brushstrokes in orange, blue, pink, and purple. In the bottom right corner, there is a cluster of grey and blue circles, some with blue outlines. At the very bottom, a pink horizontal band contains a blue oval, an orange oval, and a dark blue rectangle with a pink top section and two horizontal lines.

Data

Member.java

```
1 package cp241project;
2 enum Band {BNK48, CGM48}
3 enum Block {A, B, C, D}
4 public class Member {
5     private final String MemName;
6     private final Band BandOrigin;
7     private final int rankNum;
8     private final Block BlockTeam;
9     public Member(String name, Band bandIn, int rank, Block BlockTeamIn){
10         MemName = name;
11         rankNum = rank;
12         BandOrigin = bandIn;
13         BlockTeam = BlockTeamIn;
14     }
15     public String getMemName(){
16         return MemName;
17     }
18     public Band getBandOrigin(){
19         return BandOrigin;
20     }
21     public int getRank(){
22         return rankNum;
23     }
24     public Block getBlock(){
25         return BlockTeam;
26     }
27     public String toString(){
28         return getMemName()+" from "+getBandOrigin()+" Block "+getBlock()+" is rank "+getRank();
29     }
30 }
31
32
```

//ประกาศ Band เป็น Enum โดยใน Band จะมีตัวเลือก BNK48, CGM48
//ประกาศ Block เป็น Enum โดยใน Block จะมีตัวเลือก A, B, C, D
//ประกาศคลาส Member
//ประกาศ Data Member เป็น final String ชื่อ MemName
//ประกาศ Data Member เป็น final Band ชื่อ BandOrigin
//ประกาศ Data Member เป็น final int ชื่อ rankNum
//ประกาศ Data Member เป็น final Block ชื่อ BlockTeam
/*สร้าง Constructor ของ Member ให้รับ name, Band, rank,
BlockTeamIn แล้วส่งค่านั้น ต่อให้ Data Member*/
//ประกาศ Method getMemName ที่ return ค่าเป็น String
//คืนค่า MemName ที่เก็บค่าเป็นชื่อของเมมเบอร์
//ประกาศ Method getBandOrigin ที่ return ค่าเป็น Band
//คืนค่า Band ที่เก็บค่าเป็นชื่อวง
//ประกาศ Method getRank ที่ return ค่าเป็น int
//คืนค่า rankNum ที่เก็บค่าเป็นอันดับของเมมเบอร์
//ประกาศ Method getBlock ที่ return ค่าเป็น Block
//คืนค่า Block ที่เก็บค่าเป็น Block ที่อยู่
//ประกาศ Method toString ที่ return ค่าเป็น String
//คืนค่า getMemname ที่เก็บค่าเป็นชื่อสมาชิก, getBandOrigin ที่เก็บค่าเป็นชื่อ
วง, getBlock ที่เก็บค่าเป็นสายการแข่งขัน และ getRank ที่เก็บค่าเป็นลำดับ



การทำงานหลัก (Driver Class)

MemWinCheck.java

```
1 package cp241project;
2
3 import com.intellij.uiDesigner.core.GridConstraints;
4 import com.intellij.uiDesigner.core.GridLayoutManager;
5 import com.intellij.uiDesigner.core.Spacer;
6
7 import javax.swing.*;
8 import java.awt.*;
9 import java.awt.event.ActionEvent;
10 import java.awt.event.ActionListener;
11
12 public class MemWinCheck extends JFrame implements ActionListener {
13     private JPanel mainPanel;
14     private JTextField memCheckTextField;
15     private JButton checkButton;
16     private JTextArea infoTextArea;
17     private String memName;
18     GraphTournament gtn = new GraphTournament();
19
20     public MemWinCheck() {
21
22         setContentPane(mainPanel);
23         setTitle("Janken Tournament Winner Check");
24         setSize(800, 600);
25         setResizable(false);
26         setLocationRelativeTo(null);
27         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
28         checkButton.addActionListener(this);
29         memCheckTextField.addActionListener(this);
30     }
```

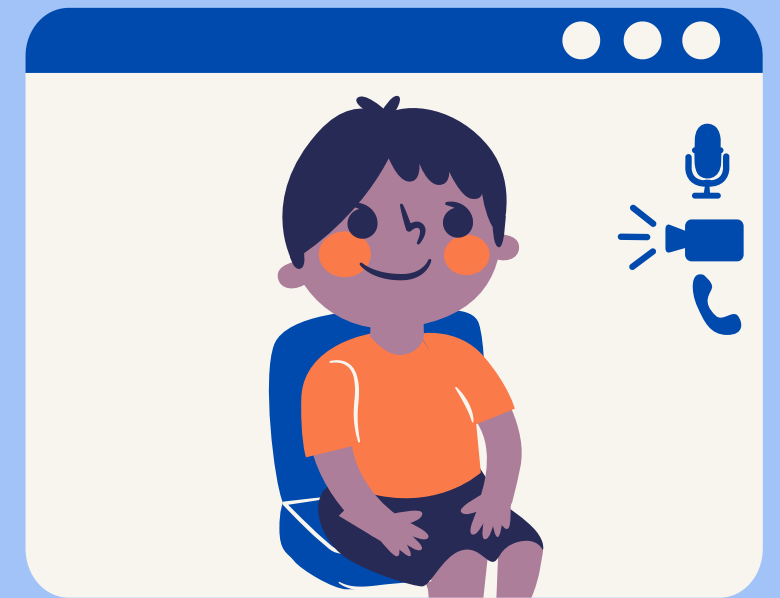
// เป็นการ Import เพื่อใช้ไลบรารี swing
// เป็นการ Import เพื่อใช้ไลบรารี awt

// ประกาศตัวแปร mainPanel ให้มีชนิดข้อมูลเป็น JPanel
// ประกาศตัวแปร memCheckTextField ให้มีชนิดข้อมูลเป็น JTextField
// ประกาศตัวแปร checkButton ให้มีชนิดข้อมูลเป็น JButton
// ประกาศตัวแปร infoTextArea ให้มีชนิดข้อมูลเป็น JTextArea
// ประกาศตัวแปร memName ให้มีชนิดข้อมูลเป็น String
// สร้าง GraphTournament ให้มีชื่อว่า gtn

// สร้าง Constructor ของ Class นี้

// กำหนด GUI

MemWinCheck.java



```
31
32 public static void main(String[] args) {
33     MemWinCheck mwc = new MemWinCheck();
34     mwc.setVisible(true);
35 }
36
37 @Override
38 public void actionPerformed(ActionEvent e) {
39     memName = memCheckTextField.getText();
40     memCheckTextField.setText("");
41     try {
42         gtn.checkMember(memName);
43         infoTextArea.setText(gtn.toString());
44     } catch (Exception ex) {
45         infoTextArea.setText(ex.getMessage());
46     }
47 }
```

```
// ประกาศเมธอด main
// สร้าง MemWinCheck ที่มีชื่อตัวแปรว่า mwc
// กำหนดให้แสดง mwc
```

```
//สร้างตัวเมธอด actionPerformed เพื่อให้(ActionEvent) ทำงานเมื่อคลิก
//กำหนดให้ตัวแปร memName รับค่าเป็น text จาก memCheckTextField
//ให้ช่อง text จาก memCheckTextField ว่างเอาไว้
// สร้าง try-catch เพื่อดักจับข้อผิดพลาด
// ตรวจสอบข้อมูลตรง memName

// แสดงข้อความเมื่อรับค่า null บน infoTextArea
```


MemWinCheck.java

```
48 {
49 // GUI initializer generated by IntelliJ IDEA GUI Designer
50 // >>> IMPORTANT!! <<<
51 // DO NOT EDIT OR ADD ANY CODE HERE!
52 $$$setupUI$$$();
53 }
54
55
56 /**
57 * Method generated by IntelliJ IDEA GUI Designer
58 * >>> IMPORTANT!! <<<
59 * DO NOT edit this method OR call it in your code!
60 *
61 * @noinspection ALL
62 */
63 private void $$$setupUI$$$() {
64     mainPanel = new JPanel();
65     mainPanel.setLayout(new GridLayoutManager(5, 5, new Insets(8, 8, 8, 8), -1, -1));
66     mainPanel.setMinimumSize(new Dimension(888, 688));
67     final JLabel label1 = new JLabel();
68     label1.setIcon(new ImageIcon(getClass().getResource("/image/JankenBanner.png")));
69     label1.setText("");
70     mainPanel.add(label1, new GridConstraints(0, 0, 1, 5, GridConstraints.ANCHOR_CENTER, GridConstraints.FILL_NONE, GridConstraints.SIZEPOLICY_FIXED, GridConstraints.SIZEPOLICY_FIXED, null, null, null, 0, false));
71     memCheckTextField = new JTextField();
72     memCheckTextField.setText("");
73     mainPanel.add(memCheckTextField, new GridConstraints(2, 1, 1, 1, GridConstraints.ANCHOR_WEST, GridConstraints.FILL_HORIZONTAL, GridConstraints.SIZEPOLICY_WANT_GROW, GridConstraints.SIZEPOLICY_FIXED, null, new Dimension(158, 18), null, 0, false));
74     final JLabel label2 = new JLabel();
75     label2.setText("Janken Tournament 2028: Please Enter the member name (in English) you want to check. (Top 16 Only.)");
76     mainPanel.add(label2, new GridConstraints(1, 1, 1, 1, GridConstraints.ANCHOR_WEST, GridConstraints.FILL_NONE, GridConstraints.SIZEPOLICY_FIXED, GridConstraints.SIZEPOLICY_FIXED, null, new Dimension(5, 18), null, 0, false));
77     infoTextArea = new JTextArea();
78     infoTextArea.setEditable(false);
79     infoTextArea.setMargin(new Insets(5, 5, 5, 5));
80     mainPanel.add(infoTextArea, new GridConstraints(4, 1, 1, 3, GridConstraints.ANCHOR_CENTER, GridConstraints.FILL_BOTH, GridConstraints.SIZEPOLICY_WANT_GROW, GridConstraints.SIZEPOLICY_WANT_GROW, null, new Dimension(158, 58), null, 0, false));
81     final Spacer spacer1 = new Spacer();
82     mainPanel.add(spacer1, new GridConstraints(4, 4, 1, 1, GridConstraints.ANCHOR_CENTER, GridConstraints.FILL_VERTICAL, 1, GridConstraints.SIZEPOLICY_WANT_GROW, null, null, null, 0, false));
83     final Spacer spacer2 = new Spacer();
84     mainPanel.add(spacer2, new GridConstraints(2, 4, 2, 1, GridConstraints.ANCHOR_CENTER, GridConstraints.FILL_VERTICAL, 1, GridConstraints.SIZEPOLICY_WANT_GROW, null, null, null, 0, false));
85     final Spacer spacer3 = new Spacer();
86     mainPanel.add(spacer3, new GridConstraints(3, 1, 1, 1, GridConstraints.ANCHOR_CENTER, GridConstraints.FILL_HORIZONTAL, GridConstraints.SIZEPOLICY_WANT_GROW, 1, null, null, null, 0, false));
87     checkButton = new JButton();
88     checkButton.setText("Check");
89     mainPanel.add(checkButton, new GridConstraints(2, 3, 2, 1, GridConstraints.ANCHOR_CENTER, GridConstraints.FILL_HORIZONTAL, GridConstraints.SIZEPOLICY_CAN_SHRINK | GridConstraints.SIZEPOLICY_CAN_GROW, GridConstraints.SIZEPOLICY_FIXED, null, new Dimension(54, 38), null, 0, false));
90     final Spacer spacer4 = new Spacer();
91     mainPanel.add(spacer4, new GridConstraints(4, 0, 1, 1, GridConstraints.ANCHOR_CENTER, GridConstraints.FILL_VERTICAL, 1, GridConstraints.SIZEPOLICY_WANT_GROW, null, null, null, 0, false));
92 }
93
94 /**
95 * @noinspection ALL
96 */
97 public JComponent $$$getRootComponent$$$() {
98     return mainPanel;
99 }
100
101 }
```

//เป็น Auto Generate จาก IntelliJ IDEA GUI Designer ในการสร้าง GUI

MemWinCheck.form

MemWinCheck.form

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <form xmlns="http://www.intellij.com/uiDesigner/form/" version="1" bind-to-class="cp241project.MemWinCheck">
3   <grid id="27dc6" binding="mainPanel" layout-manager="GridLayoutManager" row-count="5" column-count="5" same-size-horizontally="false" same-size-vertically="false" hgap="-1" vgap="-1">
4     <margin top="0" left="0" bottom="0" right="0"/>
5     <constraints>
6       <xy x="20" y="20" width="800" height="600"/>
7     </constraints>
8     <properties>
9       <minimumSize width="800" height="600"/>
10    </properties>
11    <border type="none"/>
12    <children>
13      <component id="41991" class="javax.swing.JLabel">
14        <constraints>
15          <grid row="0" column="0" row-span="1" col-span="5" vsize-policy="0" hsize-policy="0" anchor="0" fill="0" indent="0" use-parent-layout="false"/>
16        </constraints>
17        <properties>
18          <icon value="image/JankenBanner.png"/>
19          <text value=""/>
20        </properties>
21      </component>
22      <component id="49492" class="javax.swing.JTextField" binding="memCheckTextField">
23        <constraints>
24          <grid row="2" column="1" row-span="1" col-span="1" vsize-policy="0" hsize-policy="6" anchor="8" fill="1" indent="0" use-parent-layout="false">
25            <preferred-size width="150" height="10"/>
26          </grid>
27        </constraints>
28        <properties>
29          <text value=""/>
30        </properties>
31      </component>
32      <component id="6c36" class="javax.swing.JLabel">
33        <constraints>
34          <grid row="1" column="1" row-span="1" col-span="1" vsize-policy="0" hsize-policy="0" anchor="8" fill="0" indent="0" use-parent-layout="false">
35            <preferred-size width="5" height="10"/>
36          </grid>
37        </constraints>
38        <properties>
39          <text value="Janken Tournament 2020: Please Enter the member name (in English) you want to check. (Top 16 Only.)"/>
40        </properties>
41      </component>
42      <component id="67b06" class="javax.swing.JTextArea" binding="infoTextArea">
43        <constraints>
44          <grid row="4" column="1" row-span="1" col-span="3" vsize-policy="6" hsize-policy="6" anchor="0" fill="3" indent="0" use-parent-layout="false">
45            <preferred-size width="150" height="50"/>
46          </grid>
47        </constraints>
48        <properties>
49          <editable value="false"/>
```

MemWinCheck.form

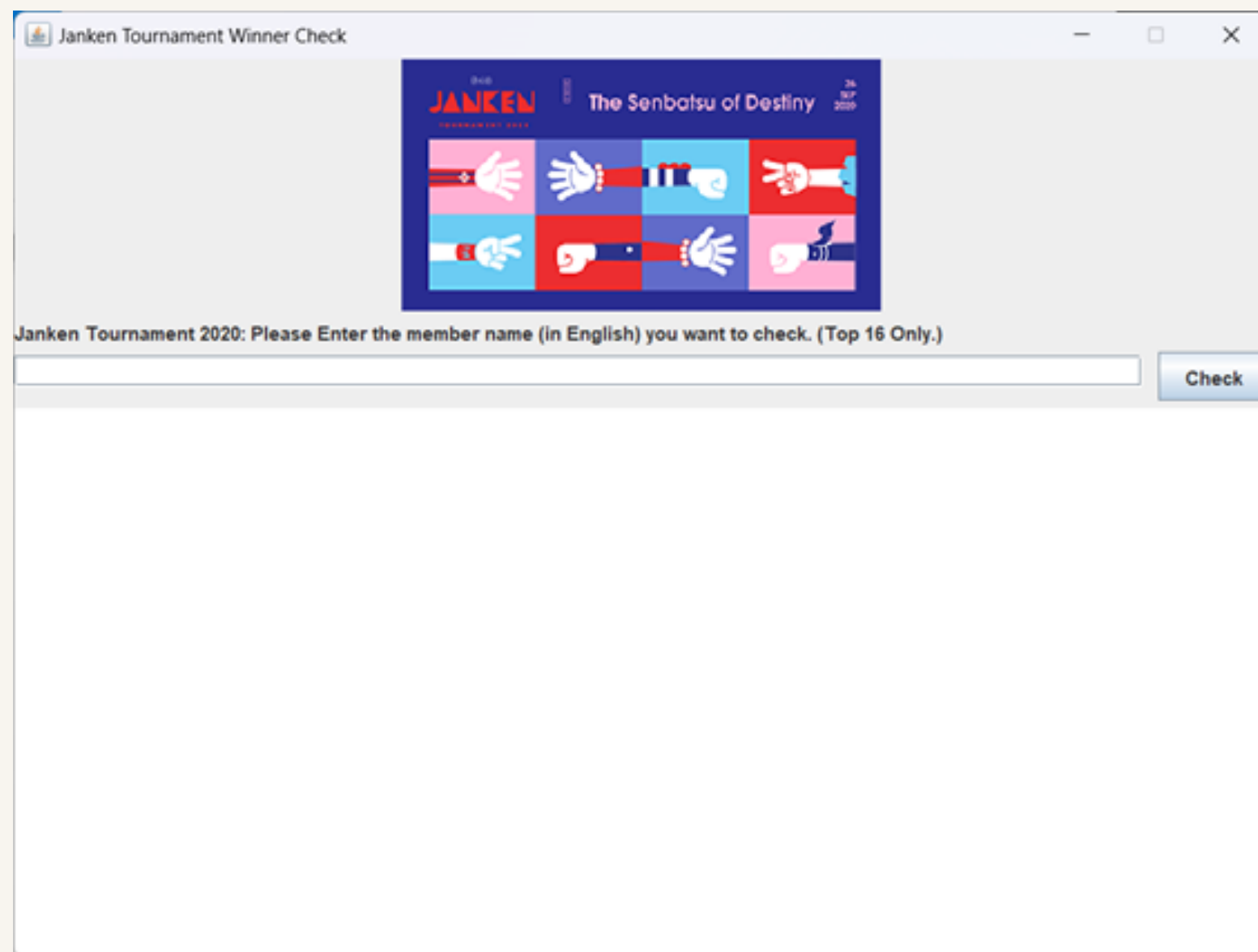
```
50      <margin top="5" left="5" bottom="5" right="5"/>
51      </properties>
52    </component>
53    <vspacer id="33d02">
54      <constraints>
55        <grid row="4" column="4" row-span="1" col-span="1" vsize-policy="6" hsize-policy="1" anchor="0" fill="2" indent="0" use-parent-layout="false"/>
56      </constraints>
57    </vspacer>
58    <vspacer id="2663">
59      <constraints>
60        <grid row="2" column="4" row-span="2" col-span="1" vsize-policy="6" hsize-policy="1" anchor="0" fill="2" indent="0" use-parent-layout="false"/>
61      </constraints>
62    </vspacer>
63    <hspacer id="6258e">
64      <constraints>
65        <grid row="3" column="1" row-span="1" col-span="1" vsize-policy="1" hsize-policy="6" anchor="0" fill="1" indent="0" use-parent-layout="false"/>
66      </constraints>
67    </hspace>
68    <component id="68b1c" class="javax.swing.JButton" binding="checkButton">
69      <constraints>
70        <grid row="2" column="3" row-span="2" col-span="1" vsize-policy="0" hsize-policy="3" anchor="0" fill="1" indent="0" use-parent-layout="false">
71          <preferred-size width="54" height="30"/>
72        </grid>
73      </constraints>
74      <properties>
75        <text value="Check"/>
76      </properties>
77    </component>
78    <vspacer id="b3c99">
79      <constraints>
80        <grid row="4" column="0" row-span="1" col-span="1" vsize-policy="6" hsize-policy="1" anchor="0" fill="2" indent="0" use-parent-layout="false"/>
81      </constraints>
82    </vspacer>
83  </children>
84 </grid>
85 </form>
```

//เป็น Auto Generate จาก IntelliJ IDEA GUI Designer ในการสร้าง GUI

การดำเนินงานของโปรแกรม

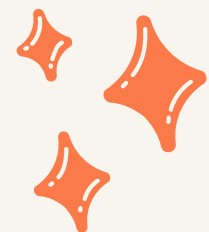


ภาพรวมของหน้าต่าง



หน้าต่าง MemWinCheck เป็นหน้าต่างของโปรแกรมนี้ สำหรับให้ผู้มาใช้งานได้กรอกชื่อของคนที่อยากจะรู้รายละเอียดของสมาชิกที่เข้าแข่งขัน Janken Tournament 2020 ว่าได้ผลแพ้ชนะอย่างไรบ้าง

- ส่วนบนจะเป็นรูปภาพโปรมทของการแข่ง
- มีช่องสำหรับใส่ชื่อสมาชิก
- ทางด้านขวามีปุ่ม Check สำหรับกดดูผลการแข่งขัน และส่วนพื้นที่ว่างด้านล่างเป็นพื้นที่สำหรับหน้าจอแสดงผลการแข่งขัน



Janken Tournament Winner Check

Janken The Senbatsu of Destiny

Janken Tournament 2020: Please Enter the member name (in English) you want to check. (Top 16 Only.)

Cherprang

Check

วิธีการใช้โปรแกรม

ขั้นที่ 1 : ใส่ชื่อของสมาชิกที่อยากจะรู้ว่าผลการ
แข่งขันเป็นอย่างไร ชนะใครมาบ้าง หรือมีแพ้
บ้างไหม





Janken Tournament Winner Check

JANKEN The Senbatsu of Destiny 24 SEP 2019

Janken Tournament 2020: Please Enter the member name (in English) you want to check. (Top 16 Only.)

Check

Cherprang	from BNK48	Block D	is rank 4
Has win with			
- Kyla	from CGM48	Block D	rank 5
- New	from BNK48	Block D	rank 13
Has lose with			
- Wee	from BNK48	Block C	rank 2

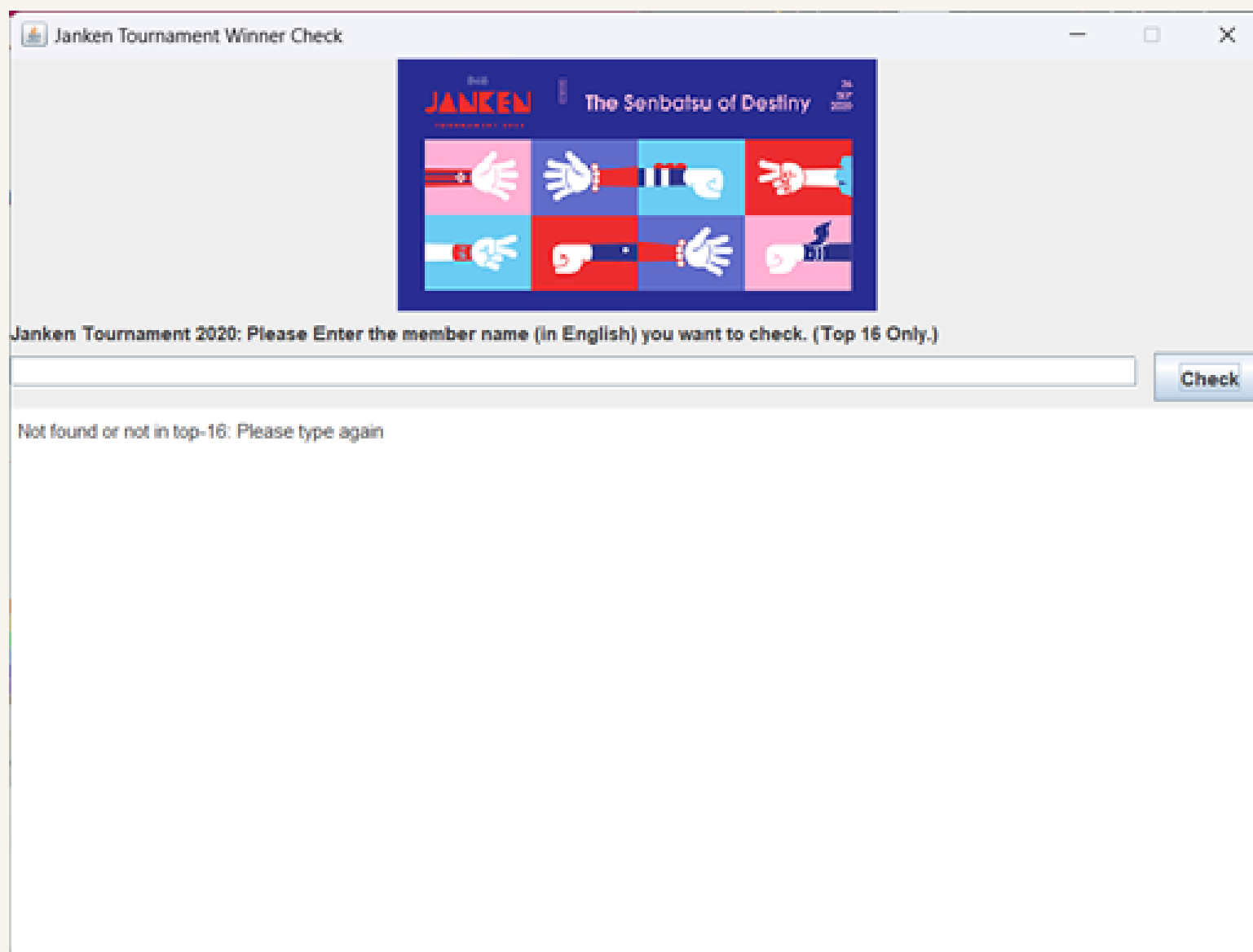
วิธีการใช้โปรแกรม

ขั้นที่ 2 : เมื่อใส่ชื่อเสร็จเรียบร้อยแล้วให้กดปุ่ม Check จากนั้นหน้าต่างที่วางตรงส่วนด้านล่างของช่องใส่ชื่อจะปรากฏข้อมูลของชื่อของคนที่ยกรอกไปว่าผลการแข่งขันเป็นอย่างไร



กรณีป้อนข้อมูลผิดพลาด

หากผู้ป้อนข้อมูลเป็นชื่อของสมาชิกที่ไม่ได้เข้าร่วม
แข่งขันหรือไม่ติด 16 อันดับแรก หรือมีการสะกดผิด
จะแสดงหน้าจอแจ้งในโปรแกรม



The screenshot shows a window titled "Janken Tournament Winner Check". At the top, there is a banner for "JANKEN The Senbatsu of Destiny". Below the banner is a 2x4 grid of eight janken (rock-paper-scissors) hand icons in various colors. Underneath the grid, the text reads: "Janken Tournament 2020: Please Enter the member name (in English) you want to check. (Top 16 Only.)". There is a text input field and a "Check" button. Below the input field, the message "Not found or not in top-16: Please type again" is displayed.

สรุปและข้อเสนอแนะ



จากการทำโครงงาน ฉันคนนี้ เคยชนะใครมานะ สามารถนำความรู้ Data Structure ที่เกี่ยวกับ Graph มาประยุกต์ใช้สร้างโปรแกรม ฉันคนนี้ เคยชนะใครมานะ เพื่อให้ผู้ใช้สามารถเช็คข้อมูลว่าคนที่กรอกมีผลการแข่งขันใน Janken Tournament 2020 เป็นอย่างไรได้ โดยในการดำเนินการทางจาวา เพื่อเขียนโปรแกรมนี้ขึ้นมาจะประกอบไปด้วย 4 ส่วนหลัก ๆ ได้แก่ ส่วนของ Graph Node สำหรับเก็บข้อมูลแพ้-ชนะของแต่ละคน, ส่วนของการสร้าง Graph สำหรับประกาศข้อมูลเพื่อชี้ว่าใครชนะใครและใครแพ้ใคร, ส่วนของ Data สำหรับเก็บข้อมูลผู้เข้าแข่งขันที่ติด 16 อันดับแรกทั้งหมด และส่วนการทำงานหลัก (Driver Class) ซึ่งจะเป็นตัวแสดงหน้าจอโปรแกรมขึ้นมา ซึ่งจากการเขียนโปรแกรมนั้น ทำให้ได้เรียนรู้ว่าโครงสร้างข้อมูลแบบ Graph นั้นสามารถนำมาใช้งานได้อย่างไร และเข้าใจถึงวิธีการทำงาน โดยโปรแกรมก็จะเป็นประโยชน์ให้แก่ผู้ที่ติดตาม BNK48 ได้ และสำหรับข้อเสนอของโครงงานนี้ ได้แก่

1. เพิ่มการตกแต่งตัวหน้าจอแสดงผลให้มีลูกเล่นหรือสีสันมากขึ้น
2. เพิ่มการแสดงผลชื่อสมาชิกแต่ละคนที่เข้าแข่งขันใน Janken Tournament 2020 เพื่อให้ผู้ใช้โปรแกรมนี้สามารถกรอกชื่อสมาชิกได้ถูกต้อง

การแบ่งหน้าที่ในโครงการ



64102010077

นายธนีสร ธนาวิชญ์

ทำหน้าที่ : เขียนโค้ด

64102010294

นางสาวบวรรัตน์ หลิมเจริญ

ทำหน้าที่ : เขียนโค้ด

64102010297

นางสาวภัทธานิษฐ์ ชัยกันทะ

ทำหน้าที่ : ทำรายงาน

64102010299

นายวิทวัส มาตรการมี

ทำหน้าที่ : ทำสไลด์



บรรณานุกรม

- ‘BNK48 JANKEN TOURNAMENT 2020’ ส่องเวียนเต๋อดที่ต้องมีดวง. (28 สิงหาคม 2563). เรียกใช้เมื่อ 30 พฤศจิกายน 2565 จาก กรุงเทพฯธุรกิจ: <https://www.bangkokbiznews.com/social/895547>**
- Agarwal, K. (2020, May 11). How to implement a graph in Java using ArrayList. Retrieved November 30, 2022, from YouTube: <https://www.youtube.com/watch?v=UhFfdBdHCJM>**
- Chakraborty, A. (2020, August 11). Tournament Trees, Winner Trees and Loser Trees in Data Structure. Retrieved September 7, 2022, from <https://www.tutorialspoint.com/tournament-trees-winner-trees-and-loser-trees-in-data-structure>**
- Chaturvedi, A. (2022, October 2). Tournament Trees and Binary Heaps. Retrieved November 30, 2022, from Coding Studio: <https://www.codingninjas.com/codestudio/library/tournament-trees-and-binary-heaps>**
- Class ArrayList<E>. (n.d.). Retrieved November 30, 2022, from Java™ Platform Standard Ed. 8: <https://docs.oracle.com/javase/8/docs/api/java/util/ArrayList.html>**
- Janken Tournament. (2022, February 4). Retrieved November 28, 2022, from Stage48 Wiki: http://stage48.net/wiki/index.php/Janken_Tournament**

บรรณานุกรม

- moomanowwww. (28 กุมภาพันธ์ 2560). รวมศัพท์น่ารู้ฉบับ BNK48 เพื่อการตามดูอย่างเข้าใจว่าเหล่า "โอตะ" เขาคุยอะไรกัน. เรียกใช้เมื่อ 28 พฤศจิกายน 2565 จาก Mango Zero: <https://www.mangozero.com/interesting-vocab-about-48g-girls/reviewaraidee>. (27 กันยายน 2563). [ผลเป่ายิงฉุบ] BNK48 Janken Tournament 2020 (Janken Queen). เรียกใช้เมื่อ 30 พฤศจิกายน 2565 จาก reviewaraidee: <https://www.reviewaraidee.com/janken-queen-bnk48-janken-tournament-2020/>
- Tournament Tree (Winner Tree) and Binary Heap. (2022, June 17). Retrieved November 30, 2022, from geeksforgeeks: <https://www.geeksforgeeks.org/tournament-tree-and-binary-heap/>
- Trakunphutthirak, R. (2022). Graph. Learning Materials.
- Trakunphutthirak, R. (2022). Tree Data Structure. Learning Materials.

