



ฉันทน์นี้ เคยชนะใครมานะ

เสนอ

อาจารย์ ดร. เรืองศักดิ์ ตระกูลพุทธิรักษ์

โดย

ตอนเรียน B01 กลุ่มที่ 1

นายธนีสร ธนาวิษฐ์	64102010077
นางสาวบรรรัตน์ หลิมเจริญ	64102010294
นางสาวภัทรานิษฐ์ ชัยกันทะ	64102010297
นายวิหวัส มาตรคำมี	64102010299

โครงการนี้เป็นส่วนหนึ่งของการศึกษารายวิชา CP241 โครงสร้างข้อมูล

มหาวิทยาลัยศรีนครินทรวิโรฒ

ภาคการศึกษาที่ 1 ปีการศึกษา 2565

คำนำ

โครงการนี้เป็นส่วนหนึ่งของรายวิชา คพ241 โครงสร้างข้อมูล (Data Structure) ชั้นปีที่ 2 คณะวิทยาศาสตร์ สาขาวิทยาการคอมพิวเตอร์ มหาวิทยาลัยศรีนครินทรวิโรฒ จัดทำขึ้นเพื่อศึกษาและพัฒนาโปรแกรม โดยใช้ Data Structure เรื่อง Graph มาสร้างเป็นโปรแกรม “ฉันคนนี้ เคยชนะใครมานะ” เพื่อตรวจสอบว่าคนที่กรอกชื่อซึ่งเป็นผู้เข้าแข่งขันใน Janken Tournament 2020 นั้นมีผลการแข่งขันเป็นอย่างไร

ทางคณะผู้จัดทำคาดหวังเป็นอย่างยิ่งว่าการจัดทำเอกสารฉบับนี้จะเป็นประโยชน์ต่อผู้อ่านและผู้สนใจศึกษาโปรแกรมอย่างดี

คณะผู้จัดทำ

สารบัญ

หัวข้อ	หน้า
ฉันทน์นี้ เคยชนะใครมานะ	1
ความสำคัญและที่มา	1
การประยุกต์ใช้โครงสร้างข้อมูล.....	2
การดำเนินการด้วยภาษาจาวา	5
ส่วนของ Graph Node.....	6
GraphNode.java	6
GraphNodeADT.java	8
ส่วนของการสร้าง Graph.....	9
GraphTournament.java	9
GraphTournamentADT.java	14
ส่วนของ Data.....	15
Member.java.....	15
ส่วนการทำงานหลัก (Driver Class)	17
MemWinCheck.java	17
MemWinCheck.form	22
การทำงานของโปรแกรม	25
1. ภาพรวมของหน้าต่าง	25
2. วิธีการใช้โปรแกรม.....	26
3. กรณีย้อนข้อมูลผิดพลาด	27
สรุปและข้อเสนอแนะ	28
สรุปผล	28
ข้อเสนอแนะ	28

การแบ่งหน้าที่ในโครงการ.....	29
64102010077 นายธนีสร ธนาวิชญ์.....	29
64102010294 นางสาวบรรรัตน์ หลิมเจริญ.....	29
64102010297 นางสาวภัทรานิษฐ์ ชัยกันทะ.....	29
64102010299 นายวิทวัส มาตรคำมี.....	29
บรรณานุกรม.....	31

บัญชีภาพประกอบ

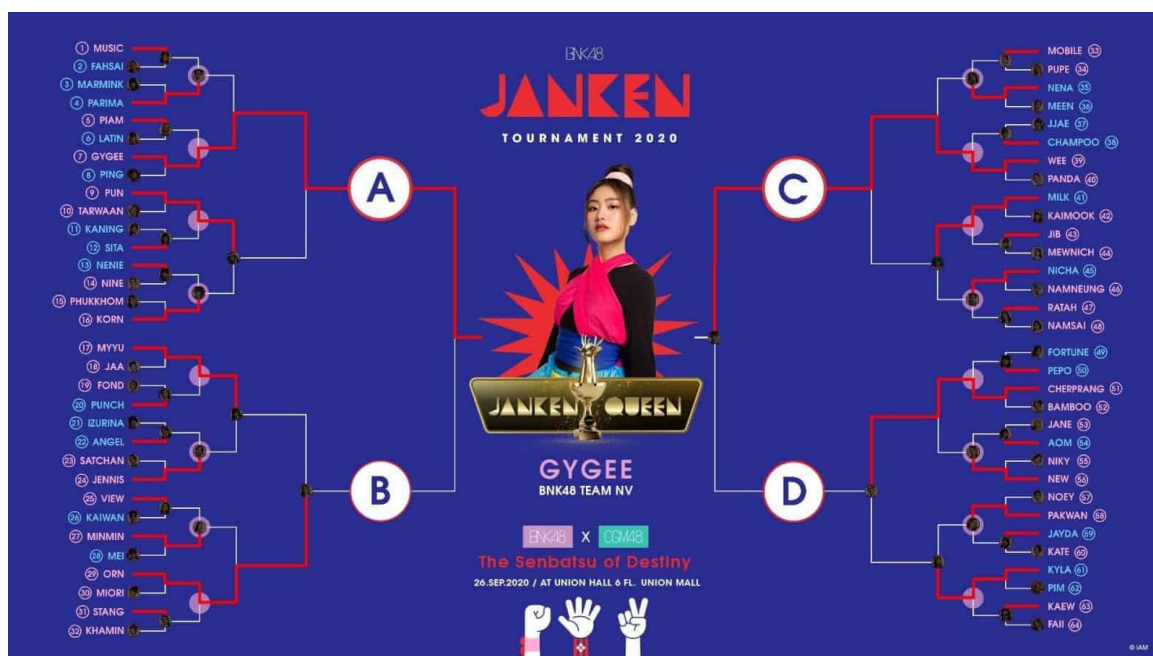
ภาพประกอบ	หน้า
ภาพประกอบที่ 1 แผนผังสรุปผลการแข่งขัน BNK48 Janken Tournament 2020	2
ภาพประกอบที่ 2 Janken Tournament ของ 16 อันดับแรก เมื่อเขียนแบบ Tournament Tree	3
ภาพประกอบที่ 3 Janken Tournament ของ 16 อันดับแรก เมื่อเขียนแบบ Directed Graph	3
ภาพประกอบที่ 4 Class Diagram ของโครงงาน	5
ภาพประกอบที่ 5 หน้าแรกของโปรแกรม	25
ภาพประกอบที่ 6 การกรอกชื่อสมาชิกลงในโปรแกรม	26
ภาพประกอบที่ 7 การแสดงผลของโปรแกรม	26
ภาพประกอบที่ 8 การแสดงผลเมื่อชื่อสมาชิกที่กรอกไม่สอดคล้องกับชื่อสมาชิกที่แข่งขัน	27

ฉันทน์ เคยชนะไครมานะ

ความสำคัญและที่มา

BNK48 เป็นกลุ่มไอดอลที่มีชื่อเสียงและรู้จักกันดีของประเทศไทย ถือเป็นไอดอลที่มีต้นฉบับมาจากญี่ปุ่นซึ่งใช้ระบบวงแบบเดียวกับวงไอดอลชื่อดังอย่าง AKB48 ทำให้มีการทำงานและทำกิจกรรมตามในแบบฉบับญี่ปุ่นเป็นหลัก แน่นอนว่าหนึ่งในกิจกรรมหลักของวง BNK48 ก็คือการร้องเพลง ในหลายปีที่ผ่านมา วง BNK48 ได้ปล่อยทั้งซิงเกิลและอัลบั้มมากมายให้แฟนคลับได้ฟังกัน แต่ในเบื้องหลัง แต่ละซิงเกิลและอัลบั้ม ด้วยจำนวนสมาชิกของวงที่มีมาก ทำให้ไม่สามารถนำสมาชิกทุกคนมาร้องหรือขึ้นเวทีพร้อมกันหมดได้ จึงมีสิ่งที่เรียกว่าเซมบัตสึ ซึ่งหมายถึงสมาชิกที่ได้รับเลือกให้มีส่วนร่วมในซิงเกิลนั้น ๆ โดยแต่ละซิงเกิลก็จะมีจำนวนสมาชิกเซมบัตสึต่างกันไปแล้วแต่เพลง การจะได้รับเซมบัตสึนี้ก็สามารถได้มาจากหลายทาง ทั้งจากการเลือกตั้ง หรือโพรดิวเซอร์ หรือกับต้นทีมเป็นคนเลือกเองตามความเหมาะสมของแต่ละซิงเกิล (moomanowww, 2560) โดยวิธีการเลือกเซมบัตสึที่ทางคณะผู้จัดทำรู้สึกก็คือ การเลือกเซมบัตสึในซิงเกิลอัลบั้มชุดที่ 3 (Warota People – หัวเราะเซ)

งาน JANKEN (จังกั้น) แปลเป็นไทยคือ งานเป่ายิ้งฉุบ เป็นงานใหญ่ประจำปีของวง AKB48 มาตั้งแต่ปี 2010 เพื่อเปิดโอกาสให้สมาชิก (เมมเบอร์) ทุกคนมีโอกาสเท่าเทียมกันที่จะติดเซมบัตสึในซิงเกิลด้วยการใช้ทักษะและดวงมาเป่ายิ้งฉุบกัน (Janken Tournament, 2022) ต่อมางานนี้ค่อย ๆ ปรับเปลี่ยนรูปแบบไปหลากหลาย อาทิ การแบ่งยูนิตของเมมเบอร์เพื่อมาปะทะกันแล้วออกซิงเกิลในนามยูนิต ซึ่งทาง BNK48 ก็ได้นำรูปแบบงาน JANKEN ของ AKB48 มาใช้ในการเลือกเซมบัตสึในซิงเกิลอัลบั้มชุดที่ 3 โดยตั้งชื่องานว่า BNK48 JANKEN TOURNAMENT 2020 - SENBATSU OF DESTINY ในงานนี้นอกจากจะมี BNK48 แล้ว ยังมี CGM48 เข้าร่วมการแข่งขัน โดยแบ่งสายการแข่งขันออกเป็น 4 กลุ่ม โดยผู้ที่สามารถเอาชนะติดต่อกัน 2 ครั้ง จะได้เข้าเป็นหนึ่งในเซมบัตสึ 16 คนทันที และผู้ชนะอันดับที่ 1 จะได้รับตำแหน่งเซเนเตอร์ในซิงเกิลอัลบั้มชุดที่ 3 ของ BNK48 ต่อจากอัลบั้ม (reviewaraidee, 2563; ‘BNK48 JANKEN TOURNAMENT 2020’ ส่องเวียนด์ที่ต้องมีดวง, 2563) เมื่อแข่งขันกันเสร็จแล้ว แฟนคลับหลายคนติดตาม BNK48 ก็คงจะอยากรู้อยากเห็นถึงผลแพ้ชนะของสมาชิกแต่ละคนว่าโดยรวมแล้วเป็นอย่างไรบ้างเนื่องมาจากแผนผังสรุปผลการแข่งขันที่อาจจะทำให้ทำความเข้าใจได้ยาก โดยแผนผังสรุปผลการแข่งขันมีลักษณะต้นฉบับ ดังรูป



ภาพประกอบที่ 1 แผนผังสรุปผลการแข่งขัน BNK48 Janken Tournament 2020

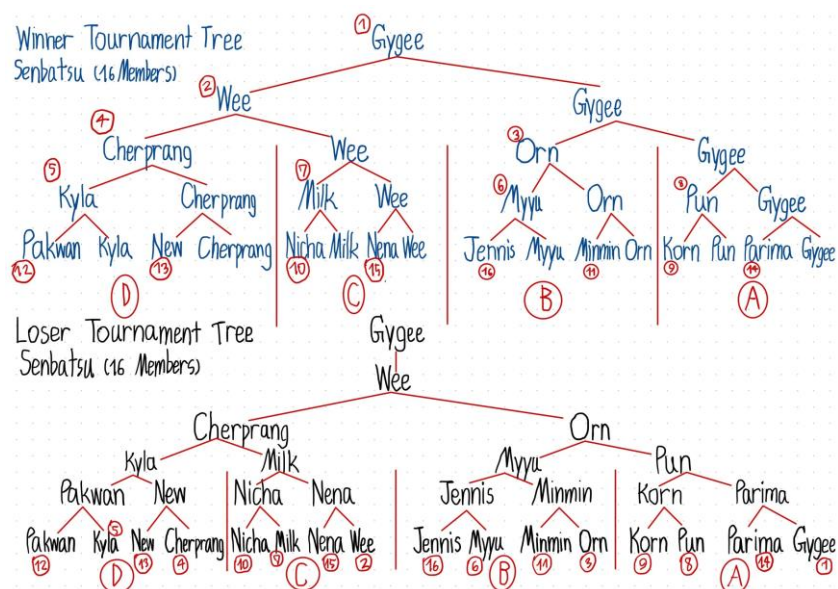
ที่มา: <https://www.facebook.com/bnk48official/photos/a.891767180950541/3208157602644809>

โครงงานนี้จัดทำขึ้นเพื่อทำความเข้าใจในเรื่องของ Data Structure และฝึกใช้ทำเป็นโครงงานที่ทางคณะผู้จัดทำได้ศึกษามา นำมาสร้างเป็น “ฉันทน์นี้ เคยชนะใครมานะ” เพื่อให้ผู้ที่ใช้สามารถตรวจสอบข้อมูลว่าคนที่รอกันชนะใครมาบ้าง ใน Tournament นั้น

คณะผู้จัดทำได้ศึกษาการการสร้างและพัฒนาโปรแกรมโดยใช้ภาษา Java และใช้โปรแกรม IntelliJ IDEA ในการพัฒนาโปรแกรม โดยใช้เนื้อหา Data Structure ในบทเรียนในส่วนของเรื่อง Graph เป็นหลัก

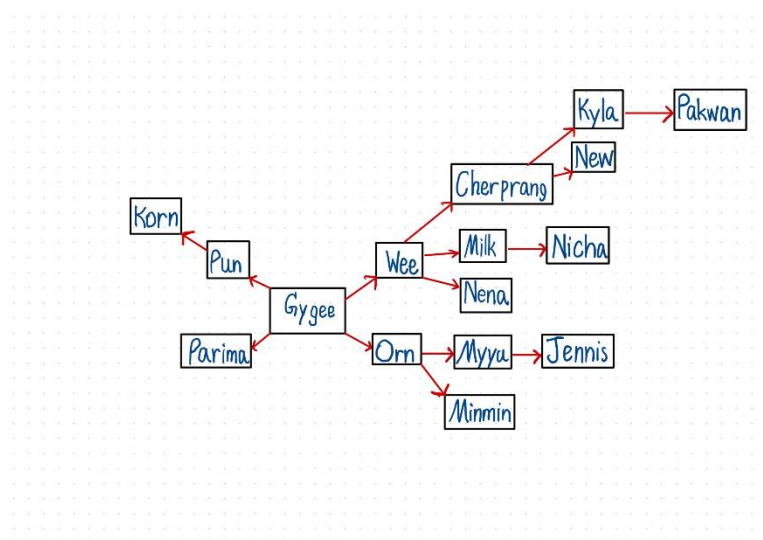
การประยุกต์ใช้โครงสร้างข้อมูล

- แรกเริ่ม ออกแบบจาก Tournament Tree ซึ่งการแข่งขันที่นำมาศึกษา (Janken Tournament 2020) โดยนำ 16 อันดับแรกของการแข่งขันมาศึกษาเท่านั้น) สามารถเขียนเป็น Tournament Tree ได้ดังนี้ (Chakraborty, 2020; Tournament Tree (Winner Tree) and Binary Heap, 2022; Chaturvedi, 2022; Trakunphutthirak, Tree Data Structure, 2022)



ภาพประกอบที่ 2 Janken Tournament ของ 16 อันดับแรก เมื่อเขียนแบบ Tournament Tree

- ต่อมา เมื่อได้ลอง Implement แล้ว ก็พบอุปสรรคในการใช้ Tournament Tree ในการทำงานตามวัตถุประสงค์ จึงได้ศึกษาและปรึกษาเพิ่มเติม เพื่อหาโครงสร้างข้อมูลที่ดีกว่า จึงได้มาเป็น Directed Graph
- โดยที่ไม่ได้ใช้ Tree เป็นเพราะต้องการแสดงให้เห็นถึงทิศทางว่า หัวลูกศรชี้ไปที่ใคร โดยนั้นจะหมายความว่า คนที่ชี้ไป ชนะ คนชี้ถูกชี้
- ซึ่งการแข่งขันที่นำมาศึกษา (Janken Tournament 2020 โดยนำ 16 อันดับแรกของการแข่งขันมาศึกษาเท่านั้น) สามารถเขียนเป็น Directed Graph ได้ดังนี้



ภาพประกอบที่ 3 Janken Tournament ของ 16 อันดับแรก เมื่อเขียนแบบ Directed Graph

- Node ของแต่ละ Graph จะเก็บข้อมูล ดังนี้ (Trakunphutthirak, Graph, 2022; Trakunphutthirak, Tree Data Structure, 2022)
 - Incoming เก็บเป็น Array List โดยเป็น Array ของ Node ที่ชี้มาหาตัวเอง (Agarwal, 2020; Class ArrayList<E>, n.d.)
 - Data เก็บเป็นข้อมูลของ Node ตัวเอง โดยในงานนี้ เก็บเป็นข้อมูล Member
 - Outcoming เก็บเป็น Array List โดยเป็น Array ของ Node ที่ชี้ไปหา Node ตัวอื่น
 - เหตุผลที่ใช้เป็น Array List เพราะงานนี้ เน้นการเข้าถึงข้อมูล ไม่เน้นการเพิ่ม-ลบ-แก้ไขข้อมูล จึงใช้ List ในแบบ Array List ที่มีประสิทธิภาพการเข้าถึงข้อมูลเป็น $O(1)$ แทนแบบ Linked List ที่มีประสิทธิภาพการเข้าถึงข้อมูลเป็น $O(n)$

ส่วนของ Graph Node

GraphNode.java

```
package cp241project;
//เป็นการ Import เพื่อใช้ไลบรารี util
import java.util.*;

public class GraphNode implements GraphNodeADT{
    //ประกาศ Data Member ของ Class GraphNode
    //ประกาศตัวแปรชนิด Member เป็น dataMem
    private Member dataMem;
    //ประกาศตัวแปรที่เก็บค่าว่าชนะใครมาบ้าง โดยใช้ list ที่มีชื่อว่า GraphNode
    private List<GraphNode> memOutGoing;
    //ประกาศตัวแปรที่เก็บค่าว่าแพ้ใครมาบ้าง โดยใช้ list ที่มีชื่อว่า GraphNode
    private List<GraphNode> memInGoing;
    //สร้าง Constructor ของ GraphNode โดยมี Data เป็น Member
    public GraphNode(Member memIn){
        //กำหนดให้ตัวแปร dataMem เก็บข้อมูลจาก memIn
        dataMem = memIn;
        //สร้าง Array List ชื่อว่า memOutGoing เพื่อเก็บข้อมูลว่า คนนี้ชนะใครมาบ้าง
        memOutGoing = new ArrayList<>();
        //สร้าง Array List ชื่อว่า memInGoing เพื่อเก็บข้อมูลว่า คนนี้แพ้ใครมาบ้าง
        memInGoing = new ArrayList<>();
    }
    //ประกาศ Method getDataMem ของ Class GraphNode
    public Member getDataMem(){
        //ส่งค่า dataMem ของโหนดนี้ คืนให้ตัวแปรที่เรียกใช้
        return dataMem;
    }
    //ประกาศ Method setDataMem ให้ตั้งค่า Member memIn
    public void setDataMem(Member memIn){
        //กำหนดให้ตัวแปร this.dataMem เก็บข้อมูลจาก memIn
        this.dataMem = memIn;
    }
}
```

```

}

@Override
//ประกาศ Method addInGoing ที่รับ Parameter เป็น GraphNode ชื่อ gNode
public void addInGoing(GraphNode gNode) {
    /*นำ gNode ที่รับเข้ามา เพิ่มไปยัง Array List ชื่อ memInGoing เพื่อเก็บข้อมูลว่า คนนี้แพ้
    ใครมาบ้าง*/
    memInGoing.add(gNode);
}

@Override
//ประกาศ Method addOutGoing ที่รับ Parameter เป็น GraphNode ชื่อ gNode
public void addOutGoing(GraphNode gNode) {
    /*นำ gNode ที่รับเข้ามา เพิ่มไปยัง Array List ชื่อ memOutGoing เพื่อเก็บข้อมูลว่า คนนี้
    ชนะใครมาบ้าง*/
    memOutGoing.add(gNode);
}

/*ประกาศ Method addMultiOutGoing โดยรับค่าข้อมูลเป็นชนิดข้อมูล
Iterator<GraphNode>*/
public void addMultiOutGoing(Iterator<GraphNode> MemOut) {
    //สร้างลูป while loop โดยดึง MemOut มาทีละตัว และทำซ้ำไปเรื่อย ๆ จนครบทุกตัว
    while (MemOut.hasNext()){
        /*ทำการเพิ่ม MemOut ที่ GraphNode ในส่วนที่เป็น Out Going Node (ตัวที่ชี้ไปหา Node
        อื่น) ด้วยการเรียก Method addOutGoing เพื่อเพิ่มข้อมูลเข้า ArrayList memOutGoing ที่
        ละตัว ตามแต่ละรอบลูป*/
        addOutGoing(MemOut.next());
    }
}

/*ประกาศ Method addMultiInGoing โดยรับค่าข้อมูลเป็นชนิดข้อมูล
Iterator<GraphNode>*/
public void addMultiInGoing(Iterator<GraphNode> MemIn) {
    //สร้างลูป while loop โดยดึง MemIn มาทีละตัว และทำซ้ำไปเรื่อย ๆ จนครบทุกตัว
    while (MemIn.hasNext()){
        /*ทำการเพิ่ม MemIn ที่ GraphNode ในส่วนที่เป็น In Going Node (ตัวที่ Node อื่น ชี้มาหา
        เรา) ด้วยการเรียก Method addInGoing เพื่อเพิ่มข้อมูลเข้า ArrayList memInGoing ที่ละตัว
        ตามแต่ละรอบลูป*/
        addInGoing(MemIn.next());
    }
}

```

```

    }

    /*ประกาศ Method getMemInGoing() โดยกำหนดให้คืนค่าเป็นชนิดข้อมูล
    List<GraphNode>*/
    public List<GraphNode> getMemInGoing(){
        //คืนค่าข้อมูล memInGoing ที่เก็บค่าคนนี้แพ้ใครมา
        return memInGoing;
    }

    /*ประกาศ Method getMemOutGoing() โดยกำหนดให้คืนค่าเป็นชนิดข้อมูล
    List<GraphNode>*/
    public List<GraphNode> getMemOutGoing(){
        //คืนค่าข้อมูล memOutGoing ที่เก็บค่าคนนี้ชนะใครมา
        return memOutGoing;
    }
}

```

GraphNodeADT.java

```

package cp241project;

//เป็นการ Import เพื่อใช้ไลบรารี util.Iterator
import java.util.Iterator;
public interface GraphNodeADT<E> {
    //ประกาศ Method addInGoing โดยเก็บข้อมูล GraphNode เป็น data
    public void addInGoing(GraphNode data);
    //ประกาศ Method addOutGoing โดยเก็บข้อมูล GraphNode เป็น data
    public void addOutGoing(GraphNode data);
}

```

ส่วนของการสร้าง Graph

GraphTournament.java

```
package cp241project;
```

```
//เป็นการ Import เพื่อใช้ไลบรารี util
```

```
import java.util.*;
```

```
public class GraphTournament implements GraphTournamentADT {
```

```
//ประกาศ + สร้างข้อมูล GraphNode ของกราฟนี้
```

```
private final GraphNode Gygee = new GraphNode(new  
Member("Gygee",Band.BNK48,1,Block.A));
```

```
private final GraphNode Wee = new GraphNode(new  
Member("Wee",Band.BNK48,2,Block.C));
```

```
private final GraphNode Orn = new GraphNode(new  
Member("Orn",Band.BNK48,3,Block.B));
```

```
private final GraphNode Cherprang = new GraphNode(new  
Member("Cherprang",Band.BNK48,4,Block.D));
```

```
private final GraphNode Kyla = new GraphNode(new  
Member("Kyla",Band.CGM48,5,Block.D));
```

```
private final GraphNode Myyu = new GraphNode(new  
Member("Myyu",Band.BNK48,6,Block.B));
```

```
private final GraphNode Milk = new GraphNode(new  
Member("Milk",Band.CGM48,7,Block.C));
```

```
private final GraphNode Pun = new GraphNode(new  
Member("Pun",Band.BNK48,8,Block.A));
```

```
private final GraphNode Korn = new GraphNode(new  
Member("Korn",Band.BNK48,9,Block.A));
```

```
private final GraphNode Nicha = new GraphNode(new  
Member("Nicha",Band.CGM48,10,Block.C));
```

```
private final GraphNode Minmin = new GraphNode(new  
Member("Minmin",Band.BNK48,11,Block.B));
```

```
private final GraphNode Pakwan = new GraphNode(new  
Member("Pakwan",Band.BNK48,12,Block.D));
```

```
private final GraphNode New = new GraphNode(new  
Member("New",Band.BNK48,13,Block.D));
```

```
private final GraphNode Parima = new GraphNode(new  
Member("Parima",Band.CGM48,14,Block.A));
```

```
private final GraphNode Nena = new GraphNode(new
Member("Nena",Band.CGM48,15,Block.C));
```

```
private final GraphNode Jennis = new GraphNode(new
Member("Jennis",Band.BNK48,16,Block.B));
```

```
//ประกาศตัวแปร GraphNode เพื่อเตรียมเอาไปใช้ในการชี้ตัวที่เราสนใจ
```

```
private GraphNode currentMem;
```

```
//สร้าง Array ของ GraphNode เพื่อใช้ในการเลือกสมาชิกของ currentMem
```

```
private final GraphNode[] memberList = {Gygee,Wee,Orn, Cherprang, Kyla,Myyu,
Milk, Pun, Korn,Nicha, Minmin,Pakwan, New, Parima, Nena, Jennis};
```

```
//สร้าง Constructor ของ GraphTournament
```

```
public GraphTournament(){
```

```
    //เรียก Method setGraph มา
```

```
    setGraph();
```

```
}
```

```
/*ประกาศ Method checkMember โดยรับค่าข้อมูลเป็นชนิด String จาก nameln โดย
throws NullPointerException เพื่อดักจับข้อมูลที่มีค่าเป็น null หรือข้อมูล String ที่อยู่
นอกเหนือจาก Array ของ memberList */
```

```
public void checkMember(String nameln) throws NullPointerException{
```

```
    //กำหนดให้ currentMem มีค่าเป็น null ก่อน
```

```
    currentMem = null;
```

```
    //ดึงข้อมูลใน Array memberList ทีละตัว มาใส่ใน member เพื่อกำหนด currentMem
```

```
    for (GraphNode member : memberList) {
```

```
        /*ทำการเทียบว่า String จาก nameln เหมือนกับชื่อ Member ที่เก็บไว้ใน member ใหม่
        ผ่านการ .getDataMem() แล้วทำการ .getMemName() โดยใช้การ Sequential
        Search*/
```

```
        if (nameln.equalsIgnoreCase(member.getDataMem().getMemName())) {
```

```
            /*ถ้ามีที่เหมือนกัน ก็ทำการกำหนด currentMem ชี้ไปที่ Node ของ Member ที่
            เจอ แล้วหลุดจากลูปด้วยการ break*/
```

```
            currentMem = member;
```

```
            break;
```

```
        }
```

```
    }
```

```
    //ทำการตรวจสอบเงื่อนไขหลังจากออกจากลูปว่า currentMem ยังเป็น null อยู่หรือไม่
```

```
    if(currentMem == null){
```

```

/*ถ้ายังเป็น null (ที่พิมพ์ไป ไม่ตรงกับชื่อ Member ที่เก็บไว้ใน Array memberList)
ก็ให้โปรแกรม โยน NullPointerException มา*/

throw new NullPointerException( "Not found or not in top-16: Please
type again");

}

}

//ประกาศ Method setGraph

private void setGraph(){ //Set relation between winner and loser.

/*ประกาศ ArrayList เป็น GraphNode ที่กำหนดค่าเป็น GraphNode ของ Member ที่
แต่ละ Member ชนะมา*/

ArrayList<GraphNode> GygeeWin = new
ArrayList<>(Arrays.asList(Wee,Orn,Pun,Parima));

ArrayList<GraphNode> WeeWin = new
ArrayList<>(Arrays.asList(Cherprang,Milk,Nena));

ArrayList<GraphNode> CherprangWin = new ArrayList<>(Arrays.asList(Kyla,
Milk, New));

ArrayList<GraphNode> MilkWin = new ArrayList<>(Arrays.asList(Nicha, Nena));

ArrayList<GraphNode> OrnWin = new
ArrayList<>(Arrays.asList(Myyu,Minmin));

ArrayList<GraphNode> MyyuWin = new
ArrayList<>(Arrays.asList(Minmin,Jennis));

ArrayList<GraphNode> KylaWin = new ArrayList<>(List.of(Pakwan));

ArrayList<GraphNode> PunWin = new ArrayList<>(List.of(Korn));

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

Gygee.addMultiOutGoing(GygeeWin.iterator());
Wee.addMultiOutGoing(WeeWin.iterator());
Cherprang.addMultiOutGoing(CherprangWin.iterator());
Milk.addMultiOutGoing(MilkWin.iterator());
Orn.addMultiOutGoing(OrnWin.iterator());
Myyu.addMultiOutGoing(MyyuWin.iterator());
Kyla.addMultiOutGoing(KylaWin.iterator());
Pun.addMultiOutGoing(PunWin.iterator());

//เรียกใช้ Method setGraphIngoing เพื่อกำหนดคนที่แพ้ของแต่ละโหนด
setGraphIngoing();

}

```


//กำหนดตัวชี้ไปที่โหนดชื่อของคนที่แพ้แต่ละคน

```
public void setGraphIngoing() {
    Wee.addIngoing(Gygee);
    Cherprang.addIngoing(Wee);
    Milk.addIngoing(Wee);
    Orn.addIngoing(Gygee);
    Myyu.addIngoing(Orn);
    Kyla.addIngoing(Cherprang);
    Pun.addIngoing(Gygee);
    Pakwan.addIngoing(Kyla);
    New.addIngoing(Cherprang);
    Nena.addIngoing(Wee);
    Nicha.addIngoing(Milk);
    Jennis.addIngoing(Myyu);
    Minmin.addIngoing(Orn);
    Korn.addIngoing(Pun);
    Parima.addIngoing(Gygee);
}
```

//ประกาศ Method toString() โดยกำหนดให้คืนค่าข้อมูลเป็นชนิด String

```
public String toString(){
```

//สร้าง StringBuilder ที่ชื่อว่า prepareReturn

```
    StringBuilder prepareReturn = new StringBuilder();
```

/*ให้ตัวแปร prepareReturn เพิ่มข้อมูล ชื่อสมาชิก ชื่อวง สายการแข่งขัน และลำดับ ที่ขึ้นอยู่กับปัจจุบัน*/

```
    prepareReturn.append(String.format("\t%s\tfrom%6s\tBlock%2s\tis  
rank%3d\n", currentMem.getDataMem().getMemName(),  
currentMem.getDataMem().getBandOrigin(),  
currentMem.getDataMem().getBlock(), currentMem.getDataMem().getRank()));
```

//ให้ตัวแปร prepareReturn เพิ่มข้อมูล "Has win with"

```
    prepareReturn.append("\tHas win with\n");
```

/*สร้างตัวแปร MemWin ที่มีชนิดข้อมูลเป็น List<GraphNode> ให้ดึงข้อมูลว่าคนที่ช้อยู่ในปัจจุบันชนะใครมาบ้าง*/

```
    List<GraphNode> MemWin = currentMem.getMemOutGoing();
```

```

//สร้างเงื่อนไข if ว่าถ้า MemWin ว่าง ก็จะเป็นจริง ให้ทำเงื่อนไขด้านใน if
if(MemWin.isEmpty()){
    //ให้ตัวแปร prepareReturn เพิ่มข้อมูลว่า ไม่ได้ชนะกับใครใน 16 อันดับแรก
    prepareReturn.append("\tNo One in the rank of top-16\n");
    /*กรณีที่เห็นเท็จ ให้ทำงานในส่วน else ที่เป็นการดึงข้อมูลสมาชิกแต่ละคน มา
    แสดงผล*/
} else{
    //สร้าง for loop โดยให้วนลูปโดยดึงค่าจาก MemWin ทีละตัว
    for(GraphNode x:MemWin){
        //ให้ตัวแปร prepareReturn เพิ่มข้อมูล ชื่อสมาชิก ชื่อวง สายการแข่งขัน และ
        ลำดับ ของ MemWin แต่ละตัว*/
        prepareReturn.append(String.format("\t-
        %s\tfrom%6s\tBlock%2s\trank%3d\n", x.getDataMem().getMemName(),
        x.getDataMem().getBandOrigin(), x.getDataMem().getBlock(),
        x.getDataMem().getRank()));
    }
}

/*สร้างตัวแปร Memlose ที่มีชนิดข้อมูลเป็น List<GraphNode> ให้ดึงข้อมูลว่าคนที่อยู่ใน
ปัจจุบันแพ้ใครมาบ้าง*/
List<GraphNode> Memlose = currentMem.getMemInGoing();
//สร้างเงื่อนไข if ว่าถ้า Memlose ว่าง ก็จะเป็นจริง ให้ทำเงื่อนไขด้านใน if
if(Memlose.isEmpty()){
    //ให้ตัวแปร prepareReturn เพิ่มข้อมูลว่า เป็น Winner ของ Tournament นี้
    prepareReturn.append("\n\twinner\n");
    /*กรณีที่เห็นเท็จ ให้ทำงานในส่วน else ที่เป็นการดึงข้อมูลสมาชิกแต่ละคน มา
    แสดงผล*/
} else{
    //ให้ตัวแปร prepareReturn เพิ่มข้อมูล "Has lose with"
    prepareReturn.append("\n\tHas lose with\n");
    //สร้าง for loop โดยให้วนลูปโดยดึงค่าจาก Memlose ทีละตัว
    for(GraphNode x:Memlose){
        //ให้ตัวแปร prepareReturn เพิ่มข้อมูล ชื่อสมาชิก ชื่อวง สายการแข่งขัน และ
        ลำดับ ของ Memlose แต่ละตัว*/
        prepareReturn.append(String.format ("\t-
        %s\tfrom%6s\tBlock%2s\trank%3d\n",

```

```

        x.getDataMem().getMemName(),
        x.getDataMem().getBandOrigin(), x.getDataMem().getBlock(),
        x.getDataMem().getRank()));
    }
}

```

```

//คืนค่าตัวแปร prepareReturn โดยแปลงให้เป็น String ก่อน
return prepareReturn.toString();
}
}

```

GraphTournamentADT.java

```

package cp241project;

public interface GraphTournamentADT {
    /*ประกาศ Method checkMember ตาม String ของ nameIn และ throw ชื่อว่า
    NullPointerException*/
    public void checkMember(String nameIn) throws NullPointerException;
    //ประกาศชื่อ Method ที่ Return เป็น String โดยใช้ชื่อ toString
    public String toString();
}

```

ส่วนของ Data

Member.java

```
package cp241project;

//ประกาศ Band เป็น Enum โดยใน Band จะมีตัวเลือก BNK48, CGM48
enum Band {BNK48, CGM48}

//ประกาศ Block เป็น Enum โดยใน Block จะมีตัวเลือก A, B, C, D
enum Block {A, B, C, D}

//ประกาศคลาส Member
public class Member {

    //ประกาศ Data Member เป็น final String ชื่อ MemName
    private final String MemName;

    //ประกาศ Data Member เป็น final Band ชื่อ BandOrigin
    private final Band BandOrigin;

    //ประกาศ Data Member เป็น final int ชื่อ rankNum
    private final int rankNum;

    //ประกาศ Data Member เป็น final Block ชื่อ BlockTeam
    private final Block BlockTeam;

    /*สร้าง Constructor ของ Member ให้รับ name, Band, rank, Block TeamIn แล้วส่งค่า
    นั้น ต่อให้ Data Member*/
    public Member(String name, Band bandIn, int rank, Block BlockTeamIn){
        MemName = name;
        rankNum = rank;
        BandOrigin = bandIn;
        BlockTeam = BlockTeamIn;
    }

    //ประกาศ Method getMemName ที่ return ค่าเป็น String
    public String getMemName(){
        //คืนค่า MemName ที่เก็บค่าเป็นชื่อของเมมเบอร์
        return MemName;
    }

    //ประกาศ Method getBandOrigin ที่ return ค่าเป็น Band
    public Band getBandOrigin(){
```

```

        //คืนค่า Band ที่เก็บค่าเป็นชื่อวง
        return BandOrigin;
    }

    //ประกาศ Method getRank ที่ return ค่าเป็น int
    public int getRank(){
        //คืนค่า rankNum ที่เก็บค่าเป็นอันดับของเมมเบอร์
        return rankNum;
    }

    //ประกาศ Method getBlock ที่ return ค่าเป็น Block
    public Block getBlock(){
        //คืนค่า Block ที่เก็บค่าเป็น Block ที่อยู่
        return BlockTeam;
    }

    //ประกาศ Method toString ที่ return ค่าเป็น Block
    public String toString(){
        /*คืนค่า getMemname ที่เก็บค่าเป็นชื่อสมาชิก, getBandOrigin ที่เก็บค่าเป็นชื่อวง,
        getBlock ที่เก็บค่าเป็นสายการแข่งขัน และ getRank ที่เก็บค่าเป็นลำดับ*/
        return getMemName()+" from "+getBandOrigin()+" Block "+getBlock()+" is rank
        "+getRank();
    }
}

```

ส่วนการทำงานหลัก (Driver Class)

MemWinCheck.java

```
package cp241project;
import com.intellij.uiDesigner.core.GridConstraints;
import com.intellij.uiDesigner.core.GridLayoutManager;
import com.intellij.uiDesigner.core.Spacer;
//เป็นการ Import เพื่อใช้ไลบรารี swing
import javax.swing.*;
//เป็นการ Import เพื่อใช้ไลบรารี awt
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class MemWinCheck extends JFrame implements ActionListener {
    //ประกาศตัวแปร mainPanel ให้มีชนิดข้อมูลเป็น JPanel
    private JPanel mainPanel;
    //ประกาศตัวแปร memCheckTextField ให้มีชนิดข้อมูลเป็น JTextField
    private JTextField memCheckTextField;
    //ประกาศตัวแปร checkButton ให้มีชนิดข้อมูลเป็น JButton
    private JButton checkButton;
    //ประกาศตัวแปร infoTextArea ให้มีชนิดข้อมูลเป็น JTextArea
    private JTextArea infoTextArea;
    //ประกาศตัวแปร memName ให้มีชนิดข้อมูลเป็น String
    private String memName;
    //สร้าง GraphTournament ให้มีชื่อว่า gtn
    GraphTournament gtn = new GraphTournament();

    //สร้าง Constructor ของ Class นี้
    public MemWinCheck() {

        //กำหนด GUI
        setContentPane(mainPanel);
    }
}
```

```

setTitle("Janken Tournament Winner Check");
setSize(800, 600);
setResizable(false);
setLocationRelativeTo(null);
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
checkButton.addActionListener(this);
memCheckTextField.addActionListener(this);
}

```

//ประกาศ Method main

```

public static void main(String[] args) {
    //สร้าง MemWinCheck ที่มีชื่อตัวแปรว่า mwc
    MemWinCheck mwc = new MemWinCheck();
    //กำหนดให้แสดง mwc
    mwc.setVisible(true);
}

```

@Override

//สร้างตัว Method actionPerformed เพื่อให้ ActionEvent ทำงานเมื่อมีการคลิก

```

public void actionPerformed(ActionEvent e) {
    //กำหนดให้ตัวแปร memName รับค่าเป็น text จาก memCheckTextField
    memName = memCheckTextField.getText();
    //ให้ช่อง text จาก memCheckTextField ว่างเอาไว้
    memCheckTextField.setText("");

    //สร้าง try-catch เพื่อดักจับข้อผิดพลาด
    try {
        //ตรวจสอบข้อมูลตรง memName
        gtn.checkMember(memName);
        infoTextArea.setText(gtn.toString());
    } catch (Exception ex) {
        //แสดงข้อความเมื่อรับค่า null บน infoTextArea
        infoTextArea.setText(ex.getMessage());
    }
}

```

```

    }
}

{
// GUI initializer generated by IntelliJ IDEA GUI Designer
// >>> IMPORTANT!! <<<
// DO NOT EDIT OR ADD ANY CODE HERE!
    $$$setupUI$$$();
}

//เป็น Auto Generate จาก IntelliJ IDEA GUI Designer ในการสร้าง GUI
/**
 * Method generated by IntelliJ IDEA GUI Designer
 * >>> IMPORTANT!! <<<
 * DO NOT edit this method OR call it in your code!
 *
 * @noinspection ALL
 */
private void $$$setupUI$$$() {
    mainPanel = new JPanel();
    mainPanel.setLayout(new GridLayoutManager(5, 5, new Insets(0, 0, 0, 0), -
1, -1));
    mainPanel.setMinimumSize(new Dimension(800, 600));
    final JLabel label1 = new JLabel();
    label1.setIcon(new
ImageIcon(getClass().getResource("/image/JankenBanner.png")));
    label1.setText("");
    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));
    memCheckTextField = new JTextField();
    memCheckTextField.setText("");
    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(150, 10), null, 0,
false));

final JLabel label2 = new JLabel();

label2.setText("Janken Tournament 2020: Please Enter the member name
(in English) you want to check. (Top 16 Only.)");

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, 10), null, 0, false));

infoTextArea = new JTextArea();

infoTextArea.setEditable(false);

infoTextArea.setMargin(new Insets(5, 5, 5, 5));

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(150, 50),
null, 0, false));

final Spacer spacer1 = new Spacer();

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));

final Spacer spacer2 = new Spacer();

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));

final Spacer spacer3 = new Spacer();

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));

checkButton = new JButton();

checkButton.setText("Check");

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, 30), null, 0, false));

final Spacer spacer4 = new Spacer();

```

```
        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));
    }

    /**
     * @noinspection ALL
     */
    public JComponent $$$getRootComponent$$$() {
        return mainPanel;
    }
}
```

MemWinCheck.form

<!--เป็น Auto Generate จาก IntelliJ IDEA GUI Designer ในการสร้าง GUI -->

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<form xmlns="http://www.intellij.com/uidesigner/form/" version="1" bind-to-
class="cp241project.MemWinCheck">
```

```
  <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">
```

```
    <margin top="0" left="0" bottom="0" right="0"/>
```

```
    <constraints>
```

```
      <xy x="20" y="20" width="800" height="600"/>
```

```
    </constraints>
```

```
    <properties>
```

```
      <minimumSize width="800" height="600"/>
```

```
    </properties>
```

```
    <border type="none"/>
```

```
    <children>
```

```
      <component id="41991" class="javax.swing.JLabel">
```

```
        <constraints>
```

```
          <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"/>
```

```
        </constraints>
```

```
        <properties>
```

```
          <icon value="image/JankenBanner.png"/>
```

```
          <text value=""/>
```

```
        </properties>
```

```
      </component>
```

```
      <component id="49492" class="javax.swing.JTextField"
binding="memCheckTextField">
```

```
        <constraints>
```

```
          <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">
```

```
            <preferred-size width="150" height="10"/>
```

```
          </grid>
```

```
        </constraints>
```

```

<properties>
  <text value=""/>
</properties>
</component>
<component id="6c36" class="javax.swing.JLabel">
  <constraints>
    <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">
      <preferred-size width="5" height="10"/>
    </grid>
  </constraints>
  <properties>
    <text value="Janken Tournament 2020: Please Enter the member name (in
English) you want to check. (Top 16 Only.)"/>
  </properties>
</component>
<component id="67b06" class="javax.swing.JTextArea" binding="infoTextArea">
  <constraints>
    <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">
      <preferred-size width="150" height="50"/>
    </grid>
  </constraints>
  <properties>
    <editable value="false"/>
    <margin top="5" left="5" bottom="5" right="5"/>
  </properties>
</component>
<vspacer id="33d02">
  <constraints>
    <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"/>
  </constraints>
</vspacer>

```

```

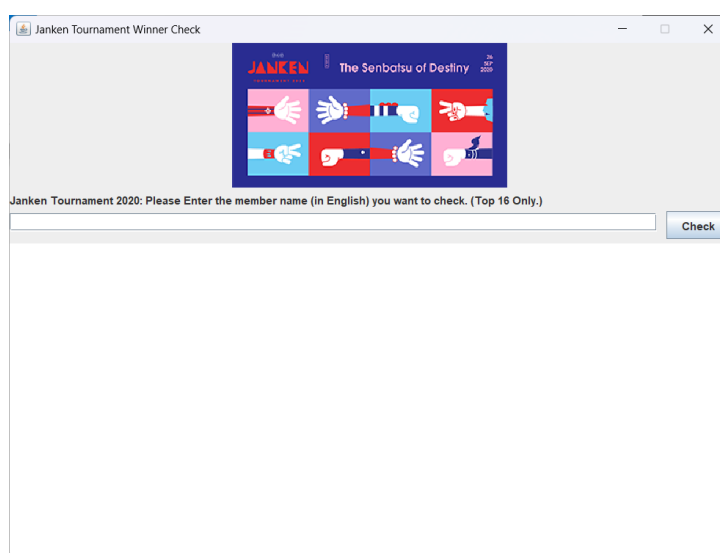
<vspacer id="2663">
  <constraints>
    <grid row="2" column="4" row-span="2" col-span="1" vsize-policy="6" hsize-
policy="1" anchor="0" fill="2" indent="0" use-parent-layout="false"/>
  </constraints>
</vspacer>
<hspacer id="6258e">
  <constraints>
    <grid row="3" column="1" row-span="1" col-span="1" vsize-policy="1" hsize-
policy="6" anchor="0" fill="1" indent="0" use-parent-layout="false"/>
  </constraints>
</hspacer>
<component id="68b1c" class="javax.swing.JButton" binding="checkButton">
  <constraints>
    <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">
      <preferred-size width="54" height="30"/>
    </grid>
  </constraints>
  <properties>
    <text value="Check"/>
  </properties>
</component>
<vspacer id="b3c99">
  <constraints>
    <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"/>
  </constraints>
</vspacer>
</children>
</grid>
</form>

```

การทำงานของโปรแกรม

1. ภาพรวมของหน้าต่าง

หน้าต่าง MemWinCheck เป็นหน้าต่างของโปรแกรมนี้อย่างไรบ้าง ส่วนบนจะเป็นรูปภาพของการแข่ง Janken Tournament 2020 ถัดมาจะมีช่องสำหรับใส่ชื่อสมาชิก โดยที่ทางด้านขวามีปุ่ม Check สำหรับกดดูผลการแข่งขัน และส่วนพื้นที่ว่างด้านล่างเป็นพื้นที่สำหรับหน้าจอแสดงผลการแข่งขัน



ภาพประกอบที่ 5 หน้าแรกของโปรแกรม

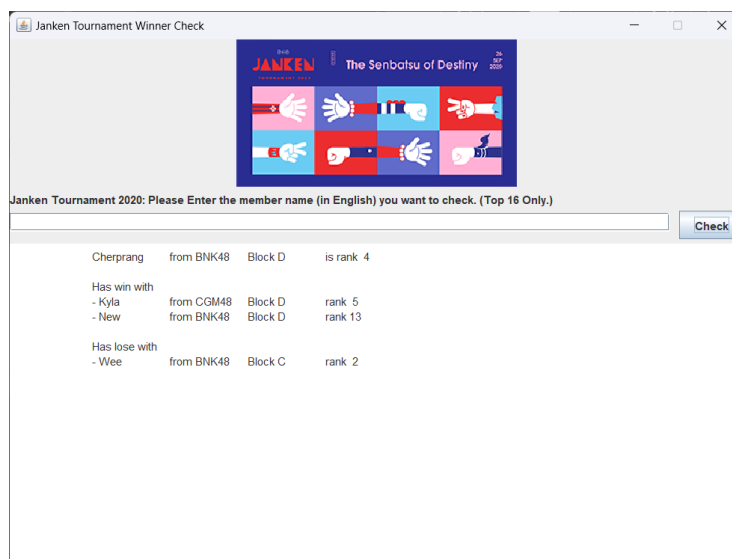
2. วิธีใช้โปรแกรม

ขั้นตอนที่ 1: ใส่ชื่อของสมาชิกที่อยากจะรู้ว่าผลการแข่งขันเป็นอย่างไร ชนะใครมาบ้าง หรือมีแพ้บ้างไหม



ภาพประกอบที่ 6 การกรอกชื่อสมาชิกลงในโปรแกรม

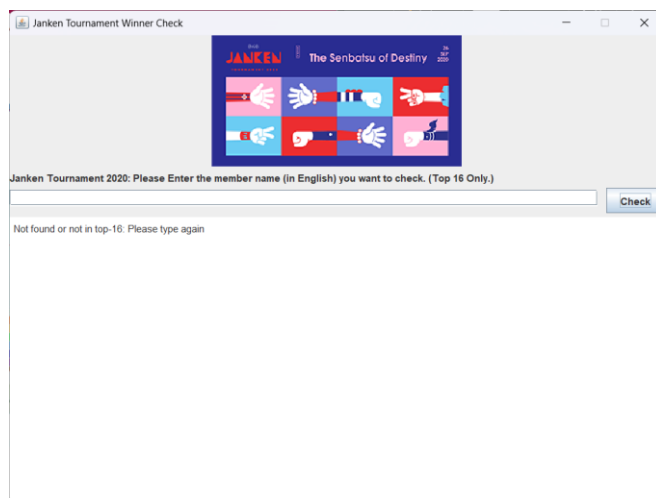
ขั้นตอนที่ 2: เมื่อใส่ชื่อเสร็จเรียบร้อยแล้วให้กดปุ่ม Check จากนั้นหน้าต่างตรงส่วนด้านล่างของช่องใส่ชื่อจะปรากฏข้อมูลของชื่อของคนที่เรากรอกไปว่าผลการแข่งขันเป็นอย่างไร



ภาพประกอบที่ 7 การแสดงผลของโปรแกรม

3. กรณีป้อนข้อมูลผิดพลาด

หากผู้ป้อนข้อมูลเป็นชื่อของสมาชิกที่ไม่ได้เข้าร่วมแข่งขันหรือไม่ติด 16 อันดับแรก หรือมีการสะกดผิด จะแสดงหน้าจอแจ้งในโปรแกรม



ภาพประกอบที่ 8 การแสดงผลเมื่อชื่อสมาชิกที่กรอกไม่สอดคล้องกับชื่อสมาชิกที่แข่งขัน

สรุปและข้อเสนอแนะ

สรุปผล

จากการทำโครงงาน ฉันคนนี้ เคยชนะใครมานะ สรุปได้ว่าสามารถนำความรู้ Data Structure ที่เกี่ยวกับ Graph มาประยุกต์ใช้สร้างโปรแกรม ฉันคนนี้ เคยชนะใครมานะ เพื่อให้ผู้ใช้สามารถตรวจสอบข้อมูลว่าคนที่กรอกนั้นมีผลการแข่งขัน Janken Tournament 2020 เป็นอย่างไรได้ โดยในการดำเนินการทางจาวาเพื่อเขียนโปรแกรมขึ้นมานั้นจะประกอบไปด้วย 4 ส่วนหลัก ๆ ได้แก่ ส่วนของ Graph Node สำหรับเก็บข้อมูลแพ้-ชนะของแต่ละคน, ส่วนของการสร้าง Graph สำหรับประกาศข้อมูลเพื่อชี้ว่าใครชนะใครและใครแพ้ใคร, ส่วนของ Data สำหรับเก็บข้อมูลผู้เข้าแข่งขันที่ติด 16 อันดับแรกทั้งหมด และส่วนการทำงานหลัก (Driver Class) ซึ่งจะเป็นตัวแสดงหน้าจอโปรแกรมขึ้นมา ซึ่งจากการเขียนโปรแกรมทำให้ได้เรียนรู้ว่าโครงสร้างข้อมูลแบบ Graph นั้นสามารถนำมาใช้งานได้อย่างไร และเข้าใจถึงวิธีการทำงาน โดยโปรแกรมก็จะเป็นประโยชน์ให้แก่ผู้ที่ติดตาม BNK48 ได้

ข้อเสนอแนะ

1. เพิ่มการตกแต่งตัวหน้าจอแสดงผลให้มีลูกเล่นหรือสีสันมากขึ้น
2. เพิ่มการแสดงผลชื่อสมาชิกแต่ละคนที่เข้าแข่งขันใน Janken Tournament 2020 เพื่อให้ผู้ใช้โปรแกรมนี้สามารถกรอกชื่อสมาชิกได้ถูกต้อง

การแบ่งหน้าที่ในโครงการ

64102010077 นายธนิศร ธนาวิชัย

ทำหน้าที่: พัฒนาโปรแกรม + ตรวจสอบความถูกต้องของรูปแบบรายงาน และเอกสารนำเสนอ

64102010294 นางสาวบรรณรัตน์ หลิมเจริญ

ทำหน้าที่: พัฒนาโปรแกรม + ตรวจสอบความถูกต้องของโปรแกรม

64102010297 นางสาวภัทรานิษฐ์ ชัยกันทะ

ทำหน้าที่: ตรวจสอบความถูกต้องของโปรแกรม + จัดทำรูปแบบรายงาน

64102010299 นายวิทวัส มาตรคำมี

ทำหน้าที่: ตรวจสอบความถูกต้องของโปรแกรม + จัดทำเอกสารนำเสนอ

บรรณานุกรม

บรรณานุกรม

- ‘BNK48 JANKEN TOURNAMENT 2020’ สัปดาห์เดียวที่ต้องมีดวง. (28 สิงหาคม 2563). เรียกใช้เมื่อ 30 พฤศจิกายน 2565 จาก กรุงเทพธุรกิจ:
<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=UhFdBdHCJM>
- 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
- moomanowww. (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*.