

วัตถุประสงค์

A. ฝึกการวิเคราะห์ปัญหา

B. ฝึกการใช้ Collections

C. ฝึกการใช้ Comparator

การแข่งขัน mini league ประกอบด้วย matches (เก็บ team1 vs team2) และ results เก็บ score1 : score2)

```
public class LabTable {
    // phy chem bio math stat com kdai
    static String [] names = {"phy", "chem", "bio", "math", "stat", "com", "kdai"};
    static ArrayList<String> name_lis = new ArrayList<>(Arrays.asList(names));
    static String [] matches = new String[21];
    //      1   2   3   4   5   6   7   8   9   10  11  12  13  14  15  16  17  18  19  20  21
    static String results_str = "1:2 ,2:0 ,0:0 ,0:1 ,1:2 ,2:2 ,3:2 ,0:1 ,3:3 ,3:0 ,2:0 ,1:0 ,1:0 ,2:3 ,0:0 ,3:1 ,0:0 ,1:2 ,0:0 ,1:0 ,1:0";
    static String [] results = results_str.split(regex:",");
    static final int WIN = 3;
    static final int LOSE = 0;
    static final int DRAW = 1;
    static {
        StringBuilder sb = new StringBuilder();
        for (int team_i = 0; team_i < names.length - 1; team_i++)
            for (int team_j = team_i + 1; team_j < names.length; team_j++)
                sb.append(names[team_i] + " vs " + names[team_j] + ";");
        String a_String = sb.toString();
        String [] tmp = a_String.split(regex:",");
        for (int i = 0; i < matches.length; i++) {
            matches[i] = tmp[i].trim();
        }
    }
}
```

เพื่อเก็บข้อมูลสำหรับตารางคะแนน ได้มีการออกแบบ class Team ไว้ดังนี้

```
class Team {
    String dept;
    int num_games;
    int goal_for;
    int goal_against;
    int points;
    public Team(String d) {
        dept = d;
    }
    public String getName() { return dept; }
    public int getGoalsFor() { return goal_for; }
    public int getPoints() { return points; }
    public int getGoalsDiff() {
        return goal_for - goal_against;
    }
    void accumulate_match_stat(int gf, int ga, int p) {
        num_games++;
        goal_for += gf;
        goal_against += ga;
        points += p;
    }
    public String toString() {
        return dept + "\t" + num_games + "\t" + goal_for + "\t" + goal_against + "\t" + points;
    }
}
```

accumulate\_match\_stat(int gf, int ga, int p) คือการ update สถิติจากผลการแข่งขันแต่ละนัด

เราสามารถใช้ .indexOf() เพื่อดึง team1 และ team2

เขียน byList() โดยส่วนของ engine นั้นเรียงตามคะแนน หากคะแนนเท่ากัน เรียงตามผลต่างประตูได้-เสีย และ ตามประตูได้ ตามลำดับ ตัวอย่าง output (ตัวอย่าง ผลการแข่งขันของ com ไม่ได้แสดงใน code ที่ให้)

```
static void byList() {
    ArrayList<Team> lis = new ArrayList<>();
    for (int i = 0; i < names.length; i++) {
        lis.add(new Team(names[i]));
    }
    int match_num = 0;
    int score1, score2;
    Team team1, team2;
    /* your code */
    Comparator<Team> engine = new Comparator<Team>() {
        public int compare(Team t1, Team t2) {
            return 0; /* your code */
        }
    };
    Collections.sort(lis, engine);
    Collections.reverse(lis);
    for (Team t : lis) {
        System.out.println(t);
    }
}
```

```
phy vs. com    1:2
chem vs. com   3:0
bio vs. com    2:3
math vs. com   0:0
stat vs. com   0:0
com vs. kdai   1:0
```

chem	6	13	7	11
com	6	6	6	11
math	6	5	4	8
stat	6	6	7	7
bio	6	6	8	7
kdai	6	4	7	7
phy	6	6	7	6

แนะนำให้ใช้ .trim() ให้เป็นนิสัย สำหรับการประมวลผลสตริง

ตัวอย่าง page ศึกษา multiple criteria comparator <https://www.java67.com/2019/06/top-5-sorting-examples-of-comparator-and-comparable-in-java.html>

Q1. Code สำหรับ Retrieve team1 team2 จาก [] matches

หมายเหตุ

- เนื่องจาก TreeMap<K,V> ไม่สามารถเรียงด้วย V ได้ตรงๆ หาก implement ด้วย Map<String, Team> ยังต้องแปลงเป็น lis.addAll(hm.values()) นักศึกษาสามารถทดลองได้
- ระยะหลัง java จัดการการ reverse comparator ใน library ที่สูงขึ้น ตัวอย่างเช่นทำใน priority queue ตรงๆไม่ได้

กำหนดส่ง TBA