

Lab Hours

Extra Labs

Lab 2-3

- 修改Lab 2的程式碼，輸出團體人數最多的人數。
- Modify your solution to Lab 2 so that the program outputs the max number of group members.

Lab 2-4

- 最初，我們假設friendship array的每一個元素都不會重複。
- 但是真實世界中，有些人就是人緣超好！
- 現在開放大家隨意填寫自己最要好的朋友 (也就是說可能會出現重複的數字)。 如何產生可能重複的數字？
- 修改Lab 2的程式碼，使得程式輸出人緣最好的人。 可能會有一個以上的人都是人緣最好，該怎麼辦？

Lab 2-4

- Recall that we assume a friendship array with distinct integers.
- However, some people are more popular among us.
- Now consider that one could write down his/her most intimate friend (that is, there exist duplicates of numbers).
How to generate random integers which could be duplicate?
- Modify your solution to Lab 2 so that the program outputs the most popular person among us. If there are two (or more) of the most popular people, how to output all of them?

Memory & You

- 根據最後一頁投影片的程式碼，畫出該程式碼在記憶體模型中的配置。
- 你需要標示出記憶體中儲存在stack下的區域變數與儲存在heap中的物件。
- 對於所有的reference，你需要用箭頭符號連結該變數與物件。

Memory & You

- Plot the scheme of memory allocation for the program shown in the next page.
- You should indicate the local variables in the stack and the objects in the heap.
- For all reference variables, you need to draw the arrow from the reference to the object.

```
1 public class CircleDemo {
2
3     public static void main(String[] args) {
4
5         Point p = new Point(3, 4);
6         Circle c = new Circle(p, 10);
7
8     }
9 }
10
11 class Point {
12
13     private double x, y;
14
15     Point(double x, double y) {
16         this.x = x;
17         this.y = y;
18     }
19
20 }
21
22 class Circle {
23
24     private Point center;
25     private double radius;
26
27     Circle(Point center, double radius) {
28         this.center = center;
29         this.radius = radius;
30     }
31
32 }
```