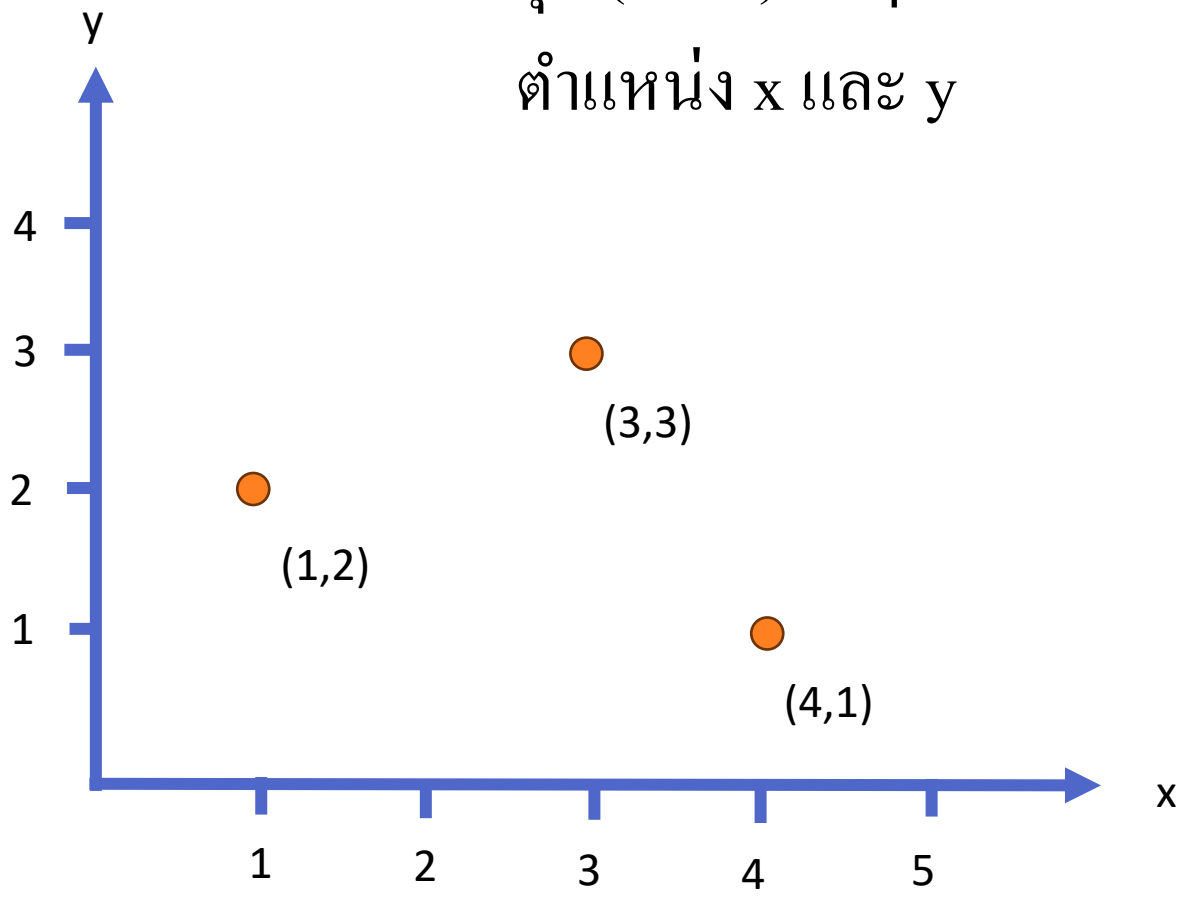
Lab 2 Multi-Class

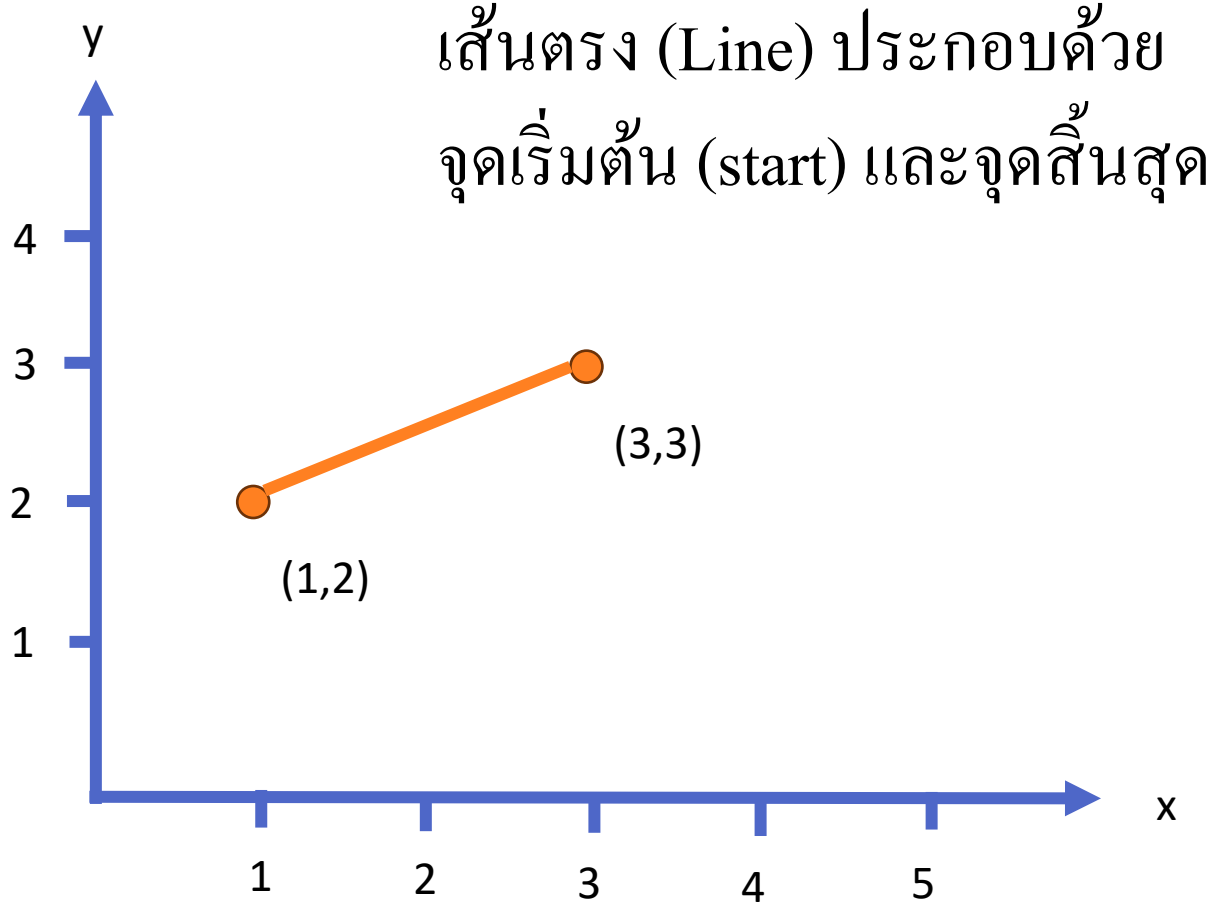
ข้อ 1 Point & Line

จุด (Point) ใดๆ ประกอบด้วย
ตำแหน่ง x และ y



ข้อ 1 Point & Line

เส้นตรง (Line) ประกอบด้วย
จุดเริ่มต้น (start) และจุดสิ้นสุด (end)



แผนภาพคลาสจุด (Point) และคลาสเส้น (Line)

Point
x : int y : int
Point() Point(x:int, y:int) setX(x : int) : void setY(y : int) : void setLocation(x:int, y:int):void getX() : int getY() : int toString() : String

Line
start : Point end : Point
Line() Line(start:Point, end:Point) Line(xStart:int, yStart:int, xEnd:int, yEnd:int) setStartPoint (x:int, y:int):void setEndPoint (x:int, y:int):void getStartPoint():Point getEndPoint():Point distance():double toString() : String

รายละเอียดคลาสจุด (Point)

- Constructor
  - Point() กำหนดค่าพิกัด x และ y มีค่าเป็น 0
  - Point(x:int, y:int) กำหนดค่าพิกัด x และ y ตามค่า parameter
- setLocation(x:int, y:int) กำหนดค่า x และ y ตามค่า parameter
- toString() ให้คืนค่า String ในรูปแบบ Point (x = 0, y = 0)

รายละเอียดคลาสเส้น (Line)

- Constructor
  - Line() สร้างเส้นตรงใหม่โดยที่มีจุดเริ่มต้นเป็น (0,0)
  - Line(start:Point, end:Point)
  - Line(xStart:int, yStart:int, xEnd:int, yEnd:int)
- setStartPoint และ setEndPoint สำหรับกำหนดค่าจุดเริ่มต้นและจุดสิ้นสุด
- getStartPoint() และ getEndPoint() สำหรับคืนค่าจุดเริ่มต้นและจุดสิ้นสุด
- distance หาความยาวเส้นตรง หาได้จากสูตร $\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$
- toString ให้คืนค่า String ในรูปแบบ Line [start=Point (x = 1, y = 1), end=Point (x = 2, y = 2)]

สร้างคลาส Tester สำหรับทดสอบ

ลำดับการทำงานของโปรแกรม

- สร้างจุด 2 จุด ชื่อ p1, p2
  - p1 อยู่ที่ตำแหน่ง (1,1)
  - p2 อยู่ที่ตำแหน่ง (5,5)
- แสดงข้อมูลของ p1 และ p2 โดยใช้เมธอด toString
- สร้างเส้นตรง lineA จากจุด p1 และ p2
- แสดงข้อมูล lineA โดยใช้เมธอด toString และแสดงความยาวเส้นตรงด้วยเมธอด distance
- เปลี่ยนตำแหน่งของ p1 ไปที่ (2,2)
- แสดงข้อมูล lineA โดยใช้เมธอด toString และแสดงความยาวเส้นตรงด้วยเมธอด distance

ตัวอย่างผลลัพธ์

```
Point (x = 1, y = 1)
Point (x = 5, y = 5)
Line A : Line [start=Point (x = 1, y = 1), end=Point (x = 5, y = 5)]
Distance : 5.66
Line A : Line [start=Point (x = 2, y = 2), end=Point (x = 5, y = 5)]
Distance : 4.24
```

สร้างคลาส Tester สำหรับทดสอบ

ลำดับการทำงานของโปรแกรม

- สร้างเส้นตรง lineB โดยกำหนดจุดเริ่มต้น (2,3) และจุดสิ้นสุด (2,5)
- แสดงข้อมูล lineB โดยใช้เมธอด toString และแสดงความยาวเส้นตรงด้วยเมธอด distance
- รับข้อมูลตำแหน่งจุดเริ่มต้น (x, y) และจุดสิ้นสุด (x, y) นำไปสร้างเส้นตรงชื่อว่า lineC
- แสดงข้อมูล lineC โดยใช้เมธอด toString และแสดงความยาวเส้นตรงด้วยเมธอด distance
- หาว่าเส้นตรง 3 เส้นนี้ เส้นใดยาวที่สุด

ตัวอย่างผลลัพธ์

```
Line B : Line [start=Point (x = 2, y = 3), end=Point (x = 2, y = 5)]
Distance : 2.00
Input start point x and y : 3 3
Input end point x and y : 6 7
Line C : Line [start=Point (x = 3, y = 3), end=Point (x = 6, y = 7)]
Distance : 5.00
Longest line is LineC
```

ข้อ 2 บุคคล (Person) มีกระเป๋าสตางค์ (Wallet)

แผนภาพคลาสบุคคล (Person) คลาสกระเป๋าสตางค์ (Wallet)

Person
firstname : String lastname : String myWallet : Wallet
Person(firstname : String, lastname : String) checkMoney() : int useMoney(money : int) : void saveMoney(money : int) : void changeNewWallet(wallet: Wallet): void printInfo():void

Wallet
balance : int
getBalance() : int putMoney(int money) : void getMoney(int money) : void

คลาสกระเป๋าสตางค์ (Wallet)

- กระเป๋าสตางค์มีข้อมูลดังนี้
  - จำนวนเงิน (balance)
- มีเมธอด
  - ตรวจสอบเงินในกระเป๋าสตางค์ getBalance() สำหรับคืนค่าจำนวนเงิน
  - ใส่เงิน putMoney(int money) สำหรับเพิ่มเงินลงในกระเป๋าสตางค์
  - หยิบเงิน getMoney(int money) สำหรับหยิบเงินออกจากกระเป๋าสตางค์

คลาสบุคคล

- Person(firstname : String, lastname : String)
- checkMoney() เช็คเงินในกระเป๋าสตางค์ว่ามีเท่าใด
- useMoney(money : int) หยิบเงินออกจากกระเป๋าสตางค์
- saveMoney(money : int) เพิ่มเงินใส่กระเป๋าสตางค์
- changeNewWallet(wallet: Wallet) เปลี่ยนกระเป๋าสตางค์ใบใหม่ (โดยปกติแล้ว เรามักเอาเงินจากกระเป๋าสตางค์ใบเดิมมาใส่กระเป๋าสตางค์ใหม่ด้วย)
- printInfo() แสดงข้อมูลของบุคคล (ชื่อ สกุล จำนวนเงินที่มีในกระเป๋าสตางค์)

สร้างคลาส Tester ทดสอบ

```java
public static void main(String[] args) {
    Person jame = new Person("Jame", "Smith");
    jame.printInfo();
    System.out.println();

    System.out.println("save money 1000");
    jame.saveMoney(1000);
    jame.printInfo();
    System.out.println();

    System.out.println("use money 200");
    jame.useMoney(200);
    jame.printInfo();
    System.out.println();

    System.out.println("change wallet");
    Wallet chap = new Wallet();
    jame.changeNewWallet(chap);
    jame.printInfo();
    System.out.println();
}
```

ตัวอย่างผลลัพธ์

```
Name : Jame Smith
Wallet : 0

save money 1000
Name : Jame Smith
Wallet : 1000

use money 200
Name : Jame Smith
Wallet : 800

change wallet
Name : Jame Smith
Wallet : 800
```

Sending