

ชื่อ-นามสกุล ธนากร เอี่ยมอาจ ..... รหัสนิสิต 6421600093 ..... หมู่ 700 .....

## Lab 5 : Class and Object (2)

ให้นิสิตเขียนโปรแกรมภาษา Java แก้ปัญหาโจทย์ต่อไปนี้

### 1. สร้างไฟล์ให้พร้อมใช้งาน

1.1 ให้นิสิตสร้างไฟล์เดอร์ใหม่ชื่อ “Lab5” ซึ่งในที่นี่จะกำหนดให้เป็นไฟล์เดอร์หลัก และให้นิสิต

เปิดไฟล์เดอร์นั้นด้วย VS code

1.2 ให้นิสิตสร้างไฟล์เดอร์ชื่อ “Lib” ขึ้นในไฟล์เดอร์หลัก จากนั้นสร้างไฟล์ Number.java และ

Point.java ขึ้นมาพร้อมกับกำหนด package และ class ให้กับทั้งสองไฟล์

1.3 สร้างไฟล์ Main.java ในไฟล์เดอร์หลัก จากนั้นสร้างเมธอด main ขึ้นให้สามารถพิมพ์คำว่า

“Welcome to My Program.” ออกรทางจอภาพ

### 2. สร้างคลาส Point

คลาส Point คือจุดบนกราฟที่มีสองแกน โดยที่คลาสนี้จะมีแอ็ตทริบิวต์สองอย่างคือ x และ y ซึ่งเป็นตัวเลขศูนย์ เมื่อสร้างออบเจ็คขึ้นจะกำหนดให้มีค่าเริ่มต้นตามที่ผู้ใช้งานกำหนดไว้ แต่ถ้าผู้ใช้งานไม่ได้กำหนดค่าไว้จะเริ่มต้นที่จุด 0,0 แทน จากนั้นให้เขียนเมธอดให้คลาஸสามารถ set/get แอ็ตทริบิวต์ได้ ให้เมธอดสามารถหาระยะทางจากตัวเองไปสู่ Point อื่นได้ และหาระยะทางจากตัวเองไปสู่จุด x,y อื่นได้ และสุดท้ายหาระยะทางจากตัวเองไปสู่จุด 0,0 ได้ พร้อมกับเขียน UML Diagram

## UML Diagram

UML Diagram
- x : double
- y : double
+ distance : Point
+ distance : (double, double)
+ toString : String
+ getXY : double[]
+ getter/ setter

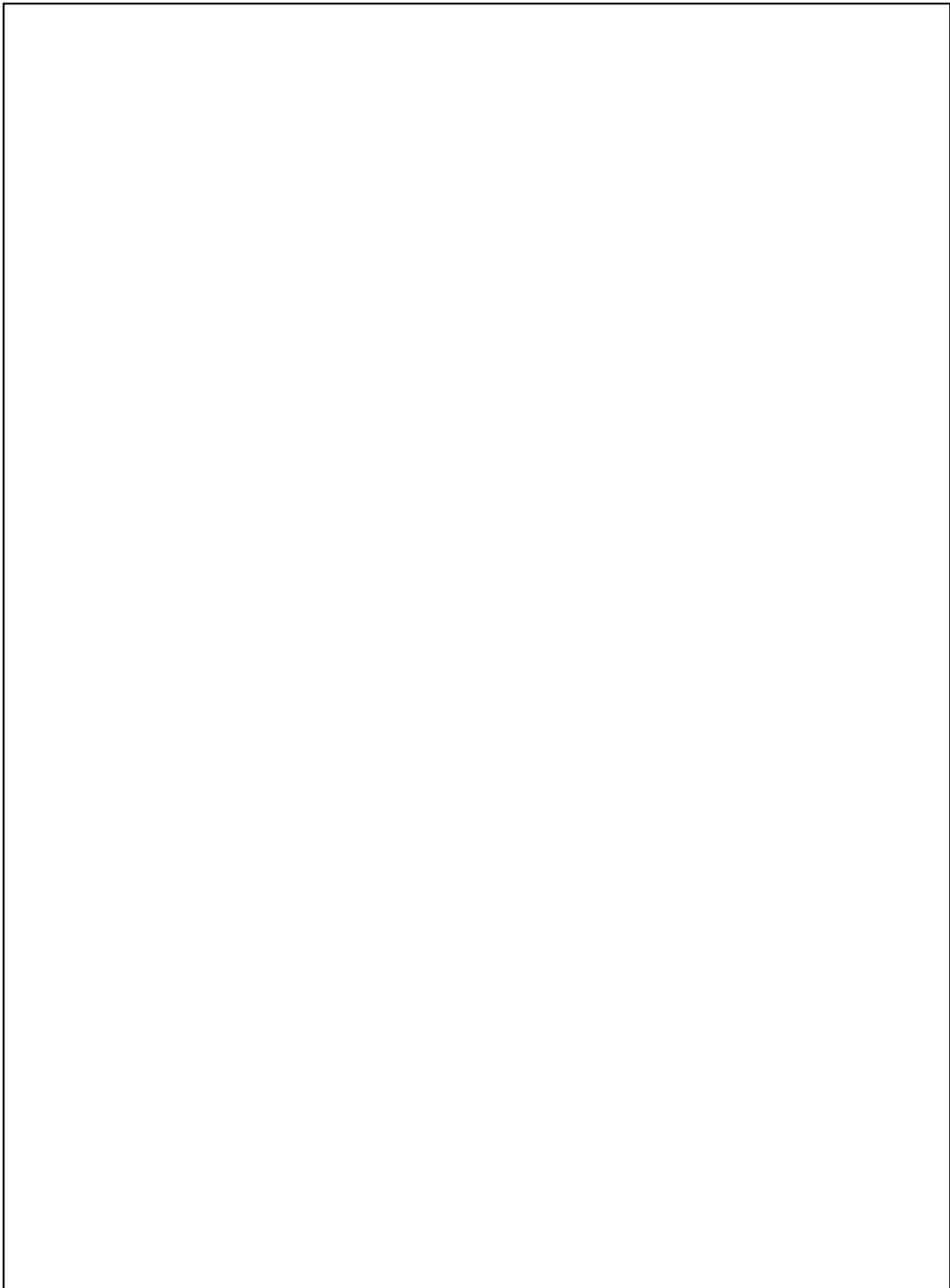
## โปรแกรมคำตอบ

```

package Lib;
public class Point {
    private double x;
    private double y;

    public Point() {
        new Point(0, 0);
    }
    public Point(double x, double y) {
        this.x = x;
        this.y = y;
    }
    public double Distance(Point p) {
        double value = Math.sqrt(Math.pow((p.getX() - this.x), 2)
            + (Math.pow((p.getY() - this.y), 2)));
        return value;
    }
    public double Distance(double x, double y) {
        return Distance(new Point(x, y));
    }
    public double Distance(){
        return Distance(new Point(0, 0));
    }
    public double[] getXY() {
        double[] xy = {this.x, this.y};
        return xy;
    }
    public String toString() {
        String ans = "";
        ans += "(" + this.x + ", " + this.y + ")";
        return ans;
    }
    public void setXY(double x, double y) {
        this.x = x;
        this.y = y;
    }
    public double getX() {
        return this.x;
    }
    public void setX(double x) {
        this.x = x;
    }
    public double getY() {
        return this.y;
    }
    public void setY(double y) {
        this.y = y;
    }
}

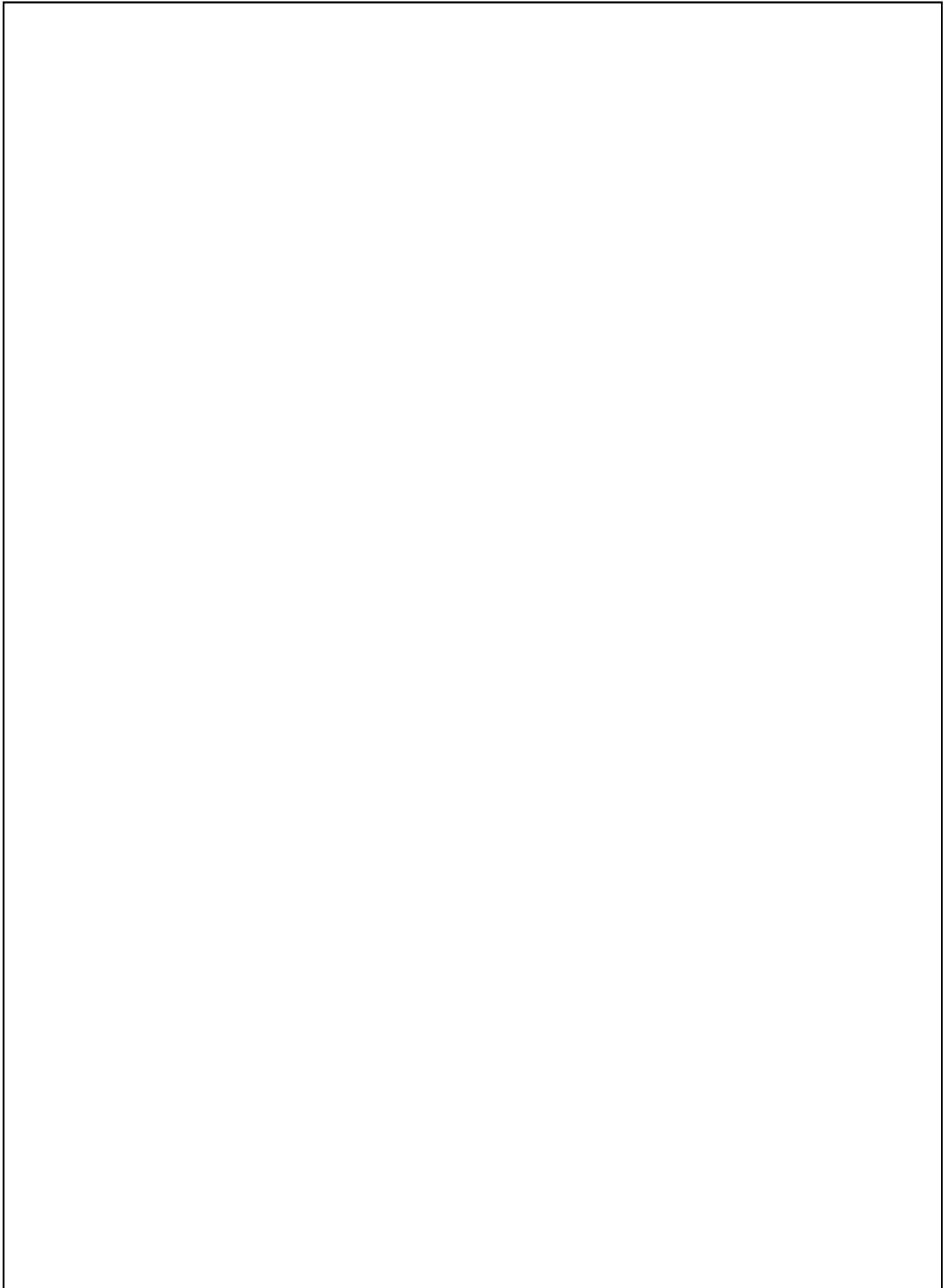
```



### 3. สร้างคลาส Number

คลาส Number จะเป็นคลาสที่มีแอตทริบิวต์เป็นตัวเลขจำนวนจริงเพียงหนึ่งตัวคือ  $n$  ซึ่งในตอนสร้างนั้นจะเป็นค่าตามที่ผู้ใช้กำหนด ถ้าผู้ใช้ไม่กำหนดจะเซ็ตค่าเริ่มต้นให้เป็น 1 และในส่วนของเมธอดนั้นจะมี

1. เช็คว่า  $n$  เป็น 0 หรือไม่
2. เช็คว่า  $n$  เป็นบวกหรือไม่
3. เช็คว่า  $n$  เป็นลบหรือไม่
4. เช็คว่า  $n$  เป็นเลขคู่หรือไม่
5. เช็คว่า  $n$  เป็นเลขคี่หรือไม่
6. เช็คว่า  $n$  เป็นจำนวนเฉพาะหรือไม่
7. หาค่า Factorial ของ  $n$
8. หาค่า  $n$  ยกกำลัง 2
9. หาค่าสแควร์รูท  $n$  (ใช้คลาส Math ได้)
10. หาผลรวมของตัวเลขในแต่ละหลัก
11. เมธอดที่กลับตัวเลข  $n$  จากหน้าไปหลัง
12. เมธอดที่หาตัวเลขทั้งหมดที่นำໄປหาร  $n$  ลงตัว
13. เมธอดแปลงเลข  $n$  เป็นเลขฐานสอง



```
package Lib;

public class Number {
    private int n;
    public Number(int n) {
        this.n = n;
    }
    public Number() {
        new Number(1);
    }
    public boolean isZero(){
        return this.n == 0;
    }
    public boolean isPositive() {
        return this.n > 0;
    }
    public boolean isNegative() {
        return this.n < 0;
    }
    public boolean isEven() {
        return this.n % 2 == 0;
    }
    public boolean isOdd() {
        return this.n % 2 == 1;
    }
    public boolean isPrime() {
        if (this.n == 1) {
            return false;
        }
        for (int i = 2; i < this.n; i++) {
            if (this.n % i == 0) {
                return false;
            }
        }
        return true;
    }
}
```

```
public double Factorial() {
    double ans = 1;
    for (int i = 1; i <= this.n; i++) {
        ans *= i;
    }
    return ans;
}
public double powTwo() {
    return Math.pow(this.n, 2);
}
public double sqrt() {
    return Math.sqrt(this.n);
}
public double sumOfDigits() {
    double ans = 0;
    int n_catch = this.n;
    n_catch = Math.abs(n_catch);
    while (n_catch > 0) {
        ans += n_catch % 10;
        n_catch /= 10;
    }
    return ans;
}
public String Reverse() {
    String ans = "";
    int n_catch = this.n;

    while (n_catch > 0) {
        // 987 % 10 = 7
        ans += n_catch % 10;
        n_catch /= 10;
    }
    return ans;
}
public String divisibleBy(){
    String ans = "";
    int n_catch = this.n;
    for (int i = 1; i <= n_catch; i++) {
        if (n_catch % i == 0) {
            ans += i + " ";
        }
    }
    return ans;
}
public String toBinary() {
    String ans = "";
    int n_catch = this.n;
    while (n_catch > 0) {
        ans = (n_catch % 2) + ans;
        n_catch /= 2;
    }
    return ans;
}
public int getN() {
    return this.n;
}
public void setN(int n) {
    this.n = n;
}
```