

ใบงานการทดลองที่ 8

เรื่อง Wrapper Class และ Math Class

1. จุดประสงค์ทั่วไป

- 1.1. รู้และเข้าใจการติดต่อระหว่างงาน และการเรียกใช้งานคลาสสำคัญทางคณิตศาสตร์
- 1.2. รู้และเข้าใจการใช้งานคลาส Wrapper

2. เครื่องมือและอุปกรณ์

เครื่องคอมพิวเตอร์ 1 เครื่อง ที่ติดตั้งโปรแกรม Eclipse

3. ทฤษฎีการทดลอง

- 3.1. Primitive Data Type คืออะไร ?

Primitive data type หรือชนิดข้อมูลพื้นฐานเป็นชนิดข้อมูลในโปรแกรมที่เก็บค่าเป็นค่าพื้นฐานแบบตรรกะ (boolean), ตัวเลขจำนวนเต็ม (integer) หรือทศนิยม (floating point) และอักขระ (character) โดยที่ไม่ได้ถูกสร้างขึ้นมาจากการรวมข้อมูลชนิดอื่นๆ

- 3.2. Wrapper Class คืออะไร? มีอะไรบ้าง? อธิบายพร้อมยกตัวอย่างประกอบ

Wrapper class เป็นคลาสที่ใช้สำหรับแปลงหรือแพ็คเกจข้อมูลพื้นฐาน (primitive data type) เช่น `int`, `double`, `char` เป็นอ็อบเจกต์ เพื่อให้สามารถนำไปใช้กับคลาสที่ต้องการรับพารามิเตอร์เป็น Object ได้ โดยสามารถใช้ method และ property ต่าง ๆ ของ Wrapper class ได้ตามปกติ

- 3.3. ยกตัวอย่าง Method ที่สามารถเรียกใช้งานได้ใน Wrapper Class

```
String numStr = "123";  
int num = Integer.parseInt(numStr);
```

- 3.4. คำสั่ง `Math.max(x, y)` ; ใช้สำหรับทำอะไร? อธิบายพร้อมยกตัวอย่างประกอบ

คำสั่ง `Math.max(x, y)` ใช้สำหรับหาค่าสูงสุด (maximum) ระหว่าง `x` และ `y` โดยจะคืนค่าเป็นตัวเลขน้อยที่สุดระหว่าง `x` และ `y` ออกมา

- 3.5. คำสั่ง `Math.min(x, y)` ; ใช้สำหรับทำอะไร? อธิบายพร้อมยกตัวอย่างประกอบ

คำสั่ง `Math.min(x, y)` ใช้สำหรับหาค่าต่ำสุด (minimum) ระหว่าง `x` และ `y` โดยจะคืนค่าเป็นตัวเลขน้อยที่สุดระหว่าง `x` และ `y` ออกมา

3.6. คำสั่ง `Math.sqrt(x)` ; ใช้สำหรับทำอะไร? อธิบายพร้อมยกตัวอย่างประกอบ

คำสั่ง `Math.sqrt(x)` ใช้สำหรับหาค่า square root (รากที่สอง) ของตัวเลข x โดยจะคืนค่าเป็นตัวเลขที่เป็น double ออกมา ตัวอย่างการใช้งาน `Math.sqrt(x)` เพื่อหาค่า square root ของตัวเลข 16 คือ 4

3.7. คำสั่ง `Math.abs(x)` ; ใช้สำหรับทำอะไร? อธิบายพร้อมยกตัวอย่างประกอบ

คำสั่ง `Math.abs(x)` ใช้สำหรับหาค่า absolute (ค่าสัมบูรณ์) ของตัวเลข x โดยไม่สนใจเครื่องหมาย (sign) ของ x ว่าเป็นบวกหรือลบ และจะคืนค่าเป็นตัวเลขที่เป็นจำนวนเต็มหรือจำนวนจริง (integer หรือ double) ออกมา

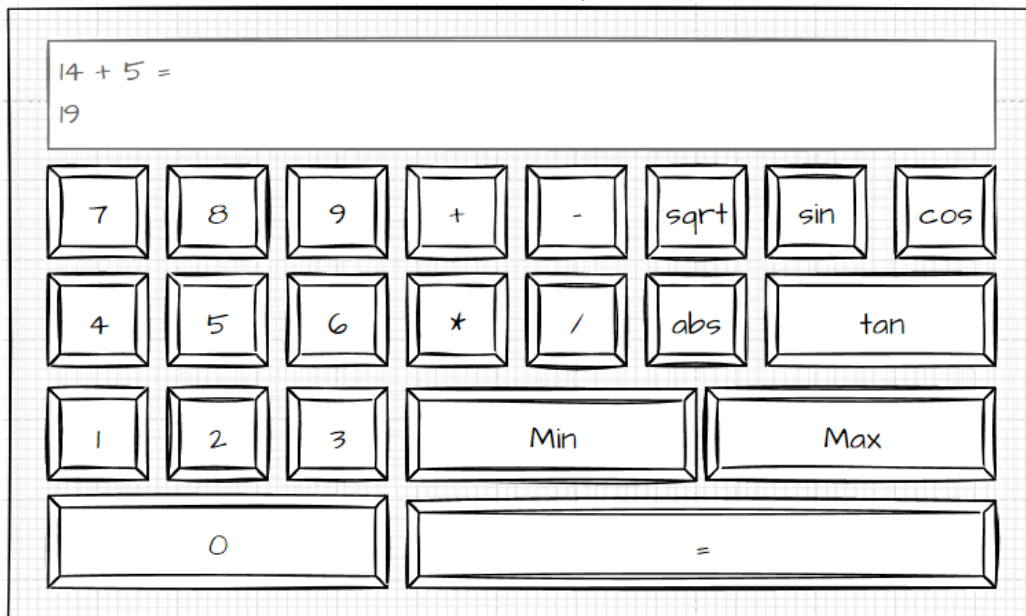
3.8. คำสั่ง `Math.random()` ; ใช้สำหรับทำอะไร? อธิบายพร้อมยกตัวอย่างประกอบ

คำสั่ง `Math.random()` ใช้สำหรับสุ่มตัวเลขที่เป็นจำนวนจริง (double) ที่มีค่าอยู่ในช่วง 0.0 ถึง 1.0 (ไม่รวม 1.0) โดยตัวเลขที่ได้จะมีการสุ่มแบบสุ่มที่ไม่ซ้ำกันและเป็นลำดับอนุกรมที่ไม่สม่ำเสมอจนโดยสามารถนำผลลัพธ์ที่ได้มาใช้ในการสุ่มตัวเลขหรือสิ่งอื่นๆ ได้

4. ลำดับขั้นการปฏิบัติการ

4.1. จงพัฒนาและปรับปรุงโปรแกรมเครื่องคิดเลขจากใบงานทดลองที่ 5 โดยการเพิ่มปุ่มให้กับเครื่องคิดเลขที่มีความสามารถเพิ่มเติมดังต่อไปนี้

- 4.1.1. ปุ่ม `sqrt` เพื่อใช้ในการหาค่ารากที่ 2 ของตัวเลขที่ผู้ใช้กรอก
- 4.1.2. ปุ่ม `abs` เพื่อใช้ในการหาค่าสัมบูรณ์ของตัวเลขที่ผู้ใช้กรอก
- 4.1.3. ปุ่ม `sin` เพื่อใช้ในการหา Sin ของตัวเลขที่ผู้ใช้กรอก
- 4.1.4. ปุ่ม `cos` เพื่อใช้ในการหา Cos ของตัวเลขที่ผู้ใช้กรอก
- 4.1.5. ปุ่ม `tan` เพื่อใช้ในการหา Tan ของตัวเลขที่ผู้ใช้กรอก
- 4.1.6. ปุ่ม `Min` เพื่อใช้ในการหาค่าที่น้อยที่สุดของตัวเลขที่ผู้ใช้กรอกมา 2 ค่า
- 4.1.7. ปุ่ม `Max` เพื่อใช้ในการหาค่าที่มากที่สุดของตัวเลขที่ผู้ใช้กรอกมา 2 ค่า



โค้ดโปรแกรมภายในปุ่มเครื่องหมายเท่ากับ (เฉพาะส่วนที่เพิ่มเข้ามาในใบงานการทดลองนี้เท่านั้น)

```

Button btnTotal = new Button(panelMain, SWT.CENTER);
btnTotal.setFont(SWTResourceManager.getFont("Timespace", 78, SWT.BOLD));
btnTotal.addSelectionListener(new SelectionAdapter() {
    @Override
    public void widgetSelected(SelectionEvent e) {
        if (answer == null) {
            if (operation == "+") {
                result = Double.parseDouble(lblText1.getText()) + Double.parseDouble(lblText2.getText());
                answer = String.format("%.2f", result);
                lblShowText.setText(lblText1.getText() + " + " + lblText2.getText() + " = " + "\n" + answer);
            } else if (operation == "-") {
                result = Double.parseDouble(lblText1.getText()) - Double.parseDouble(lblText2.getText());
                answer = String.format("%.2f", result);
                lblShowText.setText(lblText1.getText() + " - " + lblText2.getText() + " = " + "\n" + answer);
            } else if (operation == "*") {
                result = Double.parseDouble(lblText1.getText()) * Double.parseDouble(lblText2.getText());
                answer = String.format("%.2f", result);
                lblShowText.setText(lblText1.getText() + " * " + lblText2.getText() + " = " + "\n" + answer);
            } else if (operation == "/") {
                result = Double.parseDouble(lblText1.getText()) / Double.parseDouble(lblText2.getText());
                answer = String.format("%.2f", result);
                lblShowText.setText(lblText1.getText() + " / " + lblText2.getText() + " = " + "\n" + answer);
            } else if (operation == "Min") {
                result = Math.min(Double.parseDouble(lblText1.getText()), Double.parseDouble(lblText2.getText()));
                answer = String.format("%.2f", result);
                lblShowText.setText(lblText1.getText() + " Min " + lblText2.getText() + " = " + "\n" + answer);
            } else if (operation == "Max") {
                result = Math.max(Double.parseDouble(lblText1.getText()), Double.parseDouble(lblText2.getText()));
                answer = String.format("%.2f", result);
                lblShowText.setText(lblText1.getText() + " Max " + lblText2.getText() + " = " + "\n" + answer);
            } else if (operation == "sqrt") {
                result = Math.sqrt(Double.parseDouble(lblText1.getText()));
                answer = String.format("%.2f", result);
                lblShowText.setText(answer);
                tempStr = "0";
            } else if (operation == "abs") {
                result = Math.abs(Double.parseDouble(lblText1.getText()));
                answer = String.format("%.2f", result);
                lblShowText.setText(tempStr2 + " = " + "\n" + answer);
                tempStr = "0";
            } else if (operation == "sin") {
                result = Math.sin(Double.parseDouble(lblText1.getText()));
                answer = String.format("%.4f", result);
                lblShowText.setText(tempStr2 + " = " + "\n" + answer);
                tempStr = "0";
            } else if (operation == "cos") {
                result = Math.cos(Double.parseDouble(lblText1.getText()));
                answer = String.format("%.4f", result);
                lblShowText.setText(tempStr2 + " = " + "\n" + answer);
                tempStr = "0";
            } else if (operation == "tan") {
                result = Math.tan(Double.parseDouble(lblText1.getText()));
                answer = String.format("%.4f", result);
                lblShowText.setText(tempStr2 + " = " + "\n" + answer);
                tempStr = "0";
            }
        }
    }
});

```

5. สรุปผลการปฏิบัติการ

6. คำถามท้ายการทดลอง

6.1. Wrapper Class ควรใช้เมื่อใด? เพราะอะไร?

int เป็น keyword ที่หมายถึง data type แบบ Integer หรือจำนวนเต็ม

ส่วน Integer คือ class Integer จะใช้ในการทำ wrapper class เพื่อให้ค่าต่างๆสามารถใช้คำสั่งหรือ method ของ class Integer ได้ครับ เช่น parseInt() เป็นต้น

6.2. ข้อควรระวังในการใช้ Math Class คืออะไร?

การใช้งาน method ที่อยู่ภายใน Math class จะเป็น Method ที่เกี่ยวข้องกับการคำนวณทางคณิตศาสตร์ ซึ่ง method เหล่านี้จัดอยู่ในประเภท “Method ของ Class (Class Method)”