

## ใบงานการทดลองที่ 6

เรื่อง การเขียนโปรแกรมเชิงวัตถุร่วมกับคลาสทางคณิตศาสตร์

3.1. ก่อนที่จะส่งข้อมูลจากฟอร์ม 1 ไปยังฟอร์ม 2 ควรมีการเตรียมตัวอย่างไร ?

การเก็บข้อมูลไว้ในตัวแปรและส่งไปยังฟอร์มสอง ต้องผ่านฟังก์ชัน

การส่งข้อมูล static ไปยังฟอร์ม2ต้องผ่าน class

การส่งข้อมูล even ไปยังฟอร์มสองทำได้โดยการสร้างevenและadd evenlist

3.2. ฟังก์ชันเรียกตัวเองคืออะไร? อธิบายพร้อมยกตัวอย่างประกอบ

คือฟังก์ชันที่เรียกใช้ตัวเองในขณะที่ทำงาน โดยจะทำงานเป็นส่วนแต่ละส่วนจะเรียกใช้ตัวเองเพื่อแก้ปัญหา

#### 4. ลำดับขั้นการปฏิบัติการ

4.1. จงสร้าง Window Builder ในโปรแกรม Eclipse เพื่อสร้างโปรแกรมจำลองการทำงานเพื่อหาค่าของ Factorial ผ่านแบบ จำลองแบบ Recursion บนโครงสร้างข้อมูลแบบ Stack โดยโปรแกรมจะการทำงานอยู่ 2 ฟอรัม และมลักษณะการทำงาน ดังต่อไปนี้

4.1.1. ฟอรัม 1 โดยจะมีปุ่มเพื่อให้ผู้ใช้กด และเรียกหน้าต่าง ฟอรัม 2 ขึ้นมา

4.1.2. ฟอรัม 2 เป็นหน้าต่างใหม่ที่เตรียมให้ผู้ใช้กรอกเลขที่ต้องการหาค่า Factorial ลงไปในช่อง Textbox โดยที่ผู้ใช้จะถูกจำกัดให้ กรอกได้เฉพาะเลข 1 ถึง 5 เท่านั้น

4.1.3. เมื่อกรอกข้อมูลในฟอรัม 2 เสร็จแล้ว และกดปุ่ม Enter โปรแกรมจะนำเลข 5 ที่ได้จากช่อง Textbox ในฟอรัม 2 ส่งค่ากลับ ไปยังฟอรัม 1 อีกครั้ง และแสดงตัวเลขนั้นในช่องก่อนนำข้อมูล Push เข้าไปใน Stack เมื่อกดปุ่ม Run ทางด้านซ้ายล่าง ให้ โปรแกรมทำการ Push ข้อมูล 5! เข้าไปใน Stack

4.1.4. หลังจากกดปุ่ม Run เลข 5! จะเข้าไปอยู่ภายใน Stack และจะมีเลข 4! ที่รี อยู่ตำแหน่งรอง Push เข้าไปใน Stack ดังนั้น หากด้านบนสุดของ Stack ยังไม่ใช่เลข 1! เมื่อกดปุ่ม Run ระบบก็จะค่อยๆ นำข้อมูลเข้าไปใน Stack เรื่อยๆ

4.1.5. เช่นเดียวกันกับกรณีเมื่อครั้ง หลังกดปุ่ม Run เลข 4! ก็จะถูก Push เข้าไปใน Stack ในตำแหน่งด้านบนสุด

4.1.6. เช่นเดียวกันกับกรณีเมื่อครั้ง หลังกดปุ่ม Run เลข 3! ก็จะถูก Push เข้าไปใน Stack ในตำแหน่งด้านบนสุด

4.1.7. เช่นเดียวกันกับกรณีเมื่อครั้ง หลังกดปุ่ม Run เลข 2! ก็จะถูก Push เข้าไปใน Stack ในตำแหน่งด้านบนสุด

4.1.8. ทีนี้หลังจาก Push เลข 1! เข้าไปในระบบ จากกฎที่ว่า  $1! = 1$  ดังนั้นทำให้เราสามารถหาคำตอบของ 1! ได้และเมื่อได้คำตอบ ให้ทำการ Pop เลข 1! ออกมา และใส่ไว้ในตัวแปร Fac พร้อมทั้งแสดงออกมาผ่านทาง Label เพื่อให้ผู้ใช้เห็นผลการคูณของ ชุดตัวเลข

4.1.9. เมื่อกดปุ่ม Run อีกรอบ ระบบก็จะ Pop ตัวเลขบนสุดของ Stack ออกมา แล้วนำไปคูณค่า Fac ให้ผู้ใช้เห็นดังรูป

4.1.10. เมื่อกดปุ่ม Run อีกรอบ ระบบก็จะ Pop ตัวเลขบนสุดของ Stack ออกมา แล้วนำไปคูณค่า Fac ให้ผู้ใช้เห็นดังรูป

4.1.11. เมื่อกดปุ่ม Run อีกรอบ ระบบก็จะ Pop ตัวเลขบนสุดของ Stack ออกมา แล้วนำไปคูณค่า Fac ให้ผู้ใช้เห็นดังรูป

#### 4.2. จงเขียนโค้ดโปรแกรมที่อยู่ภายในปุ่ม Run

```
Button btnRun = new Button(shell, SWT.NONE);
btnRun.setBounds(36, 174, 102, 36);
btnRun.setText("Run");
btnRun.addSelectionListener(new SelectionAdapter() {
    int now_number = number;
    String number_stack = "";
    int countDown = number;
    int a[]=new int[number + 1];
    int result = 1;
    String number_result = "";
    String stack_text_number = "";
    public void widgetSelected(SelectionEvent e) {
        String text_number = String.valueOf(now_number);
        if(now_number != 0) {
            number_stack += text_number + "|";
            a[now_number] = now_number;
            now_number--;
            text_number = String.valueOf(now_number);
            txtNewText.setText(number_stack);
            label_number.setText(text_number);
            if(now_number == 0) {
                label_number.setText("0");
            }
        } else {
            if(countDown > 0) {
                int i_number = a[countDown];
                text_number = String.valueOf(i_number);
                number_result += text_number;
                result *= i_number;
                //System.out.print(number_result + "|");
                label_result.setText(number_result + "=" + result);
                number_result += "x";
                countDown--;
                for(int i = countDown; i > 0; i--) {
                    text_number = String.valueOf(a[i]);
                    stack_text_number += text_number + "|";
                }
                txtNewText.setText(stack_text_number);
                stack_text_number = "";
            } else {
                label_number.setText("Finish");
                txtNewText.setText("Finish");
            }
        }
    }
});
Label lblStack = formToolkit.createLabel(shell, "Stack", SWT.NONE);
lblStack.setBounds(244, 10, 55, 20);
}
```

## 5. สรุปผลการปฏิบัติการ

การทดลองฟังก์ชันมีการแบ่งปัญหาเป็นส่วนทำให้ง่ายต่อการแก้ปัญหา สามารถใช้แก้ปัญหาที่ซับซ้อนได้

## 6. คำถามท้ายการทดลอง

### 6.1. ฟังก์ชันการทำงานใน Stack ควรมีอะไรบ้าง?

Push

Pop

Peel

IsEmpty

size

### 6.2. การคำนวณ Factorial มีสูตรว่าอย่างไร ?

$$N! = n(n-1)(n-2)\dots(1)$$

### 6.3. หลักการสร้าง Recursion คืออะไร?

กำหนดเงื่อนไขสุดท้าย

แบ่งปัญหาเป็นส่วนๆ

ทำการเรียกใช้ฟังก์ชันและเรียกค่าตัวแปรให้ถูกต้อง

### 6.4. ข้อควรระวังในการส่งข้อมูลข้ามฟอร์มคืออะไร ?

การจัดการข้อมูล

การเข้ารหัสข้อมูล

การควบคุม