

ใบงานการทดลองที่ 6

เรื่อง การเขียนโปรแกรมเชิงวัตถุร่วมกับคลาสทางคณิตศาสตร์

1. จุดประสงค์ทั่วไป

- 1.1. รู้และเข้าใจในการติดต่อกับผู้ใช้และ การติดต่อระหว่างงาน
- 1.2. รู้และเข้าใจในการสร้างโปรแกรมเชิงวัตถุโดยใช้ภาษาโปรแกรมเชิงวัตถุใหม่ๆ

2. เครื่องมือและอุปกรณ์

เครื่องคอมพิวเตอร์ เครื่อง ที่ติดตั้งโปรแกรม Eclipse

3. ทฤษฎีการทดลอง

- 3.1. ก่อนที่จะส่งข้อมูลจากฟอร์ม 1 ไปยังฟอร์ม 2 ควรมีการเตรียมตัวอย่างไร ?

สร้างฟอร์มทั้งสองก่อนจากนั้นให้สร้างฟังก์ชัน เพื่อเชื่อมต่อไปยังฟอร์มที่ต้องการส่งข้อมูล

- 3.2. ฟังก์ชันเรียกตัวเองคืออะไร? อธิบายพร้อมยกตัวอย่างประกอบ

ฟังก์ชันที่เรียกใช้ตัวเอง ไปเรื่อยจนเงื่อนไขหยุดการทำงานของฟังก์ชัน คือ **break case** เป็นจริง การทำงานของฟังก์ชันที่

ทำงานเป็นวนลูปกี่จะหยุดทำงาน

```
in fucloop(n){  
    if( n == 10 )    return 0;  
    }else{    Fucloop( n+1 ); }
```

4. ลำดับขั้นตอนการปฏิบัติการ

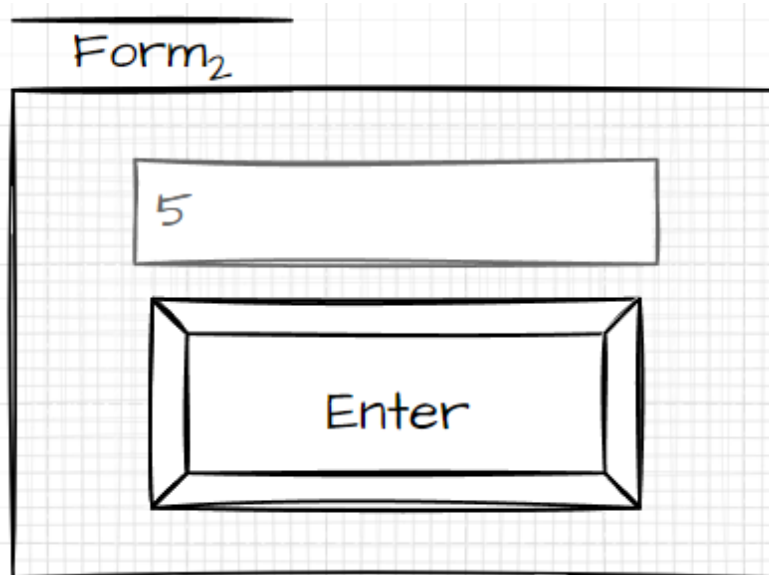
- 4.1. จงสร้าง Window Builder ในโปรแกรม Eclipse เพื่อสร้างโปรแกรมจำลองการทำงานเพื่อหาค่าของ Factorial ผ่านแบบจำลองแบบ Recursion บนโครงสร้างข้อมูลแบบ Stack โดยโปรแกรมจะมีการทำงานอยู่ 2 ฟอร์ม และมีลักษณะการทำงานดังต่อไปนี้

- 4.1.1. ฟอร์ม 1 โดยจะมีปุ่มเพื่อให้ผู้ใช้กด และเรียกหน้าต่าง ฟอร์ม 2 ขึ้นมา

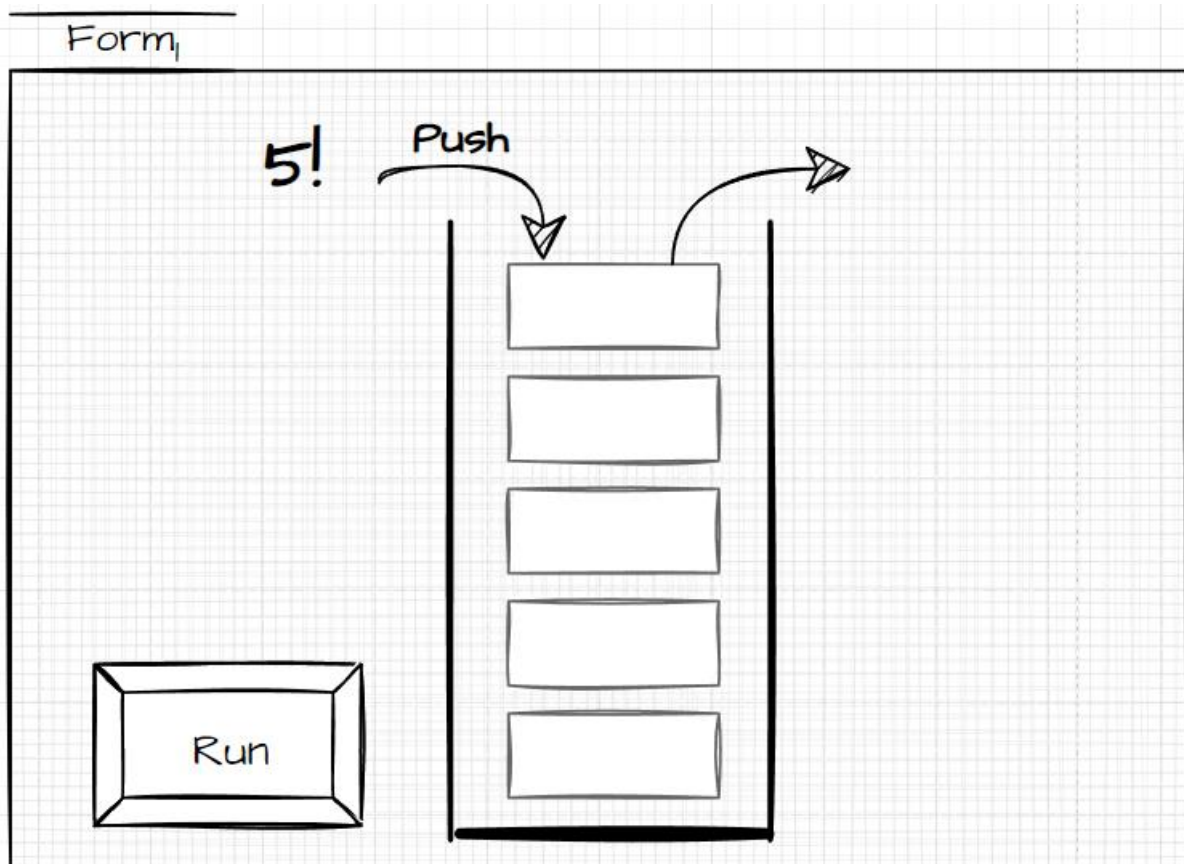
Form₁

Enter Factorial Number

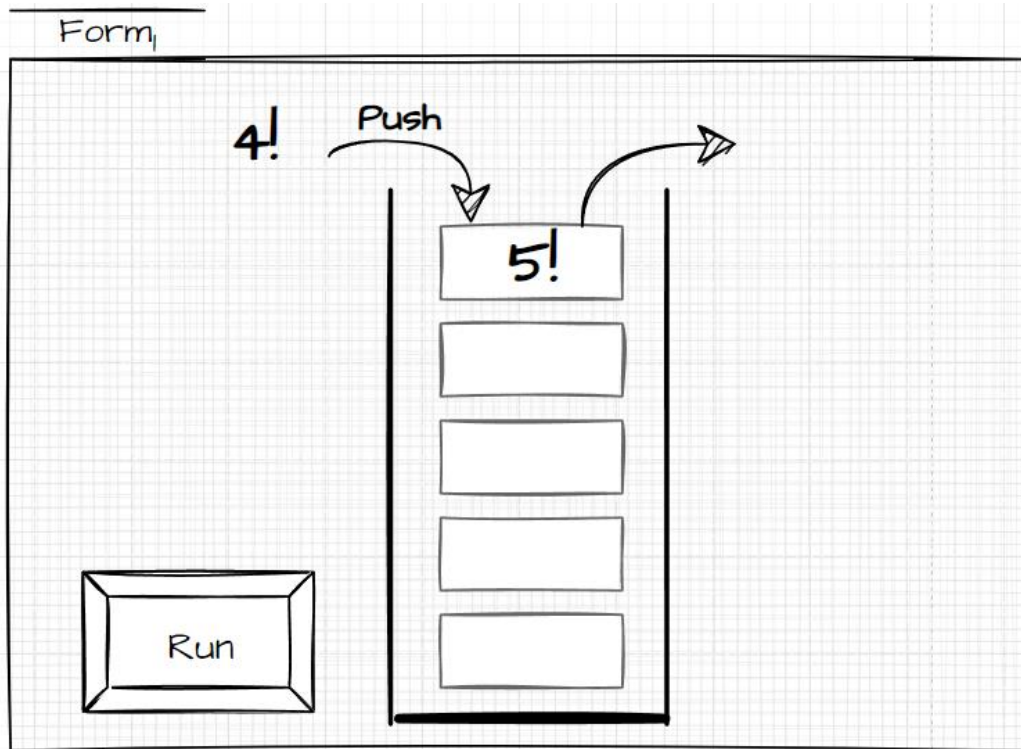
4.1.2. ฟอर्म 2 เป็นหน้าต่างใหม่ที่เตรียมให้ผู้ใช้ออกเลขที่ต้องการหาค่า Factorial ลงไปในช่อง Textbox โดยที่ผู้ใช้จะถูกจำกัดให้กรอกได้เฉพาะเลข 1 ถึง 5 เท่านั้น



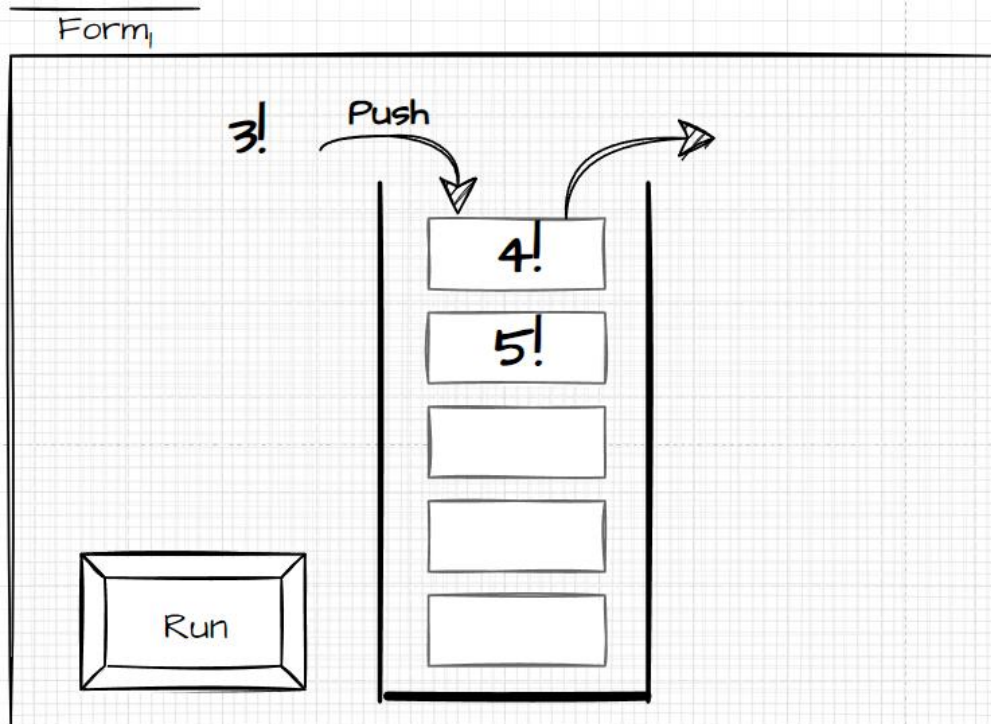
4.1.3. เมื่อกรอกข้อมูลในฟอर्म 2 เสร็จแล้ว และกดปุ่ม Enter โปรแกรมจะนำเลข 5 ที่ได้จากช่อง Textbox ในฟอर्म 2 ส่งค่ากลับไปยังฟอर्म 1 อีกครั้ง และแสดงตัวเลขนั้นในช่องก่อนนำข้อมูล Push เข้าไปใน Stack เมื่อกดปุ่ม Run ทางด้านซ้ายล่าง ให้โปรแกรมทำการ Push ข้อมูล 5! เข้าไปใน Stack



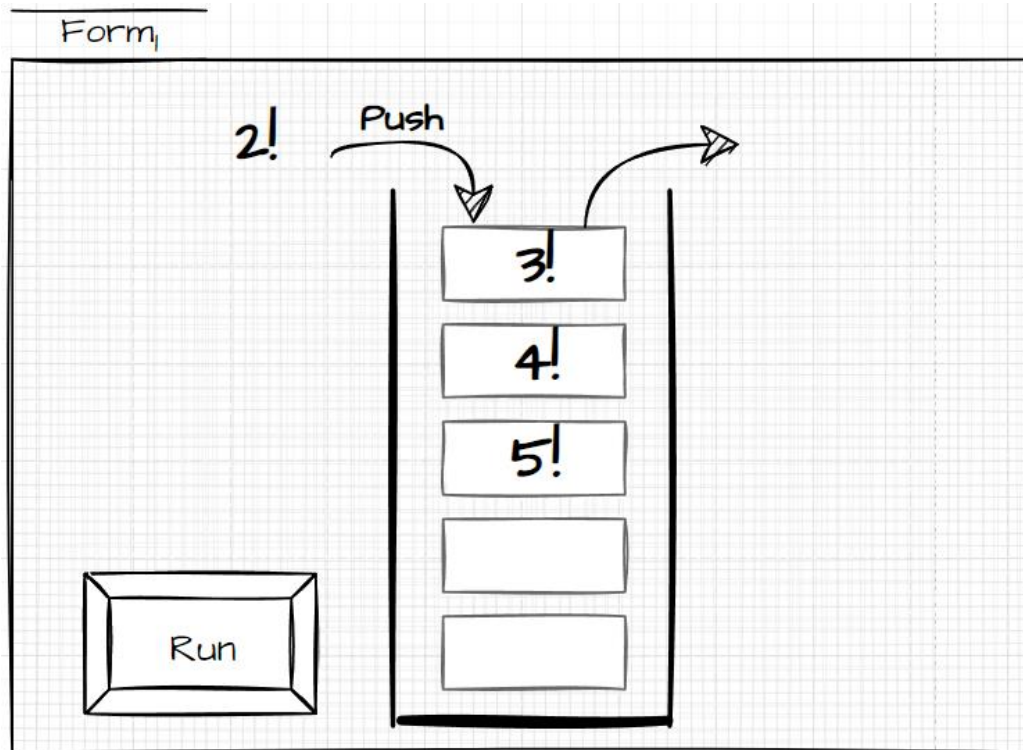
4.1.4. หลังจากกดปุ่ม Run เลข 5! จะเข้าไปอยู่ภายใน Stack และจะมีเลข 4! ที่อยู่ในตำแหน่งรอ Push เข้าไปใน Stack ดังนั้น หากด้านบนสุดของ Stack ยังไม่ใช่เลข 1! เมื่อกดปุ่ม Run ระบบก็จะค่อยๆ นำข้อมูลเข้าไปใน Stack เรื่อยๆ



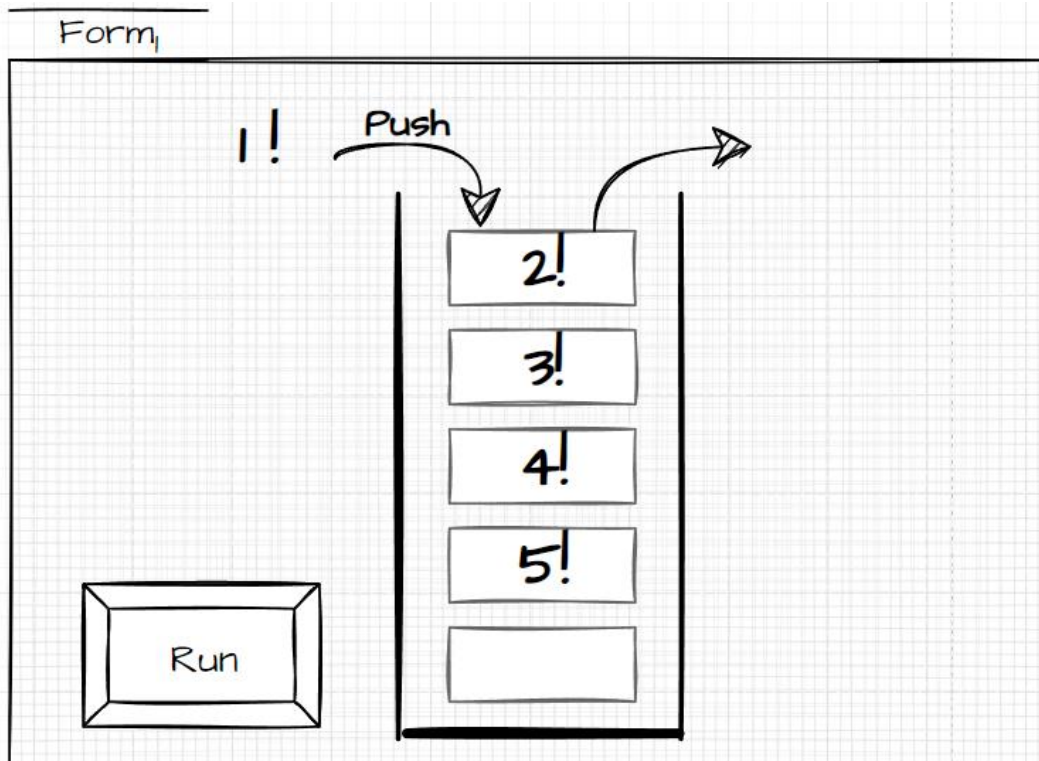
4.1.5. เช่นเดียวกันกับกรณีเมื่อครั้ง หลังกดปุ่ม Run เลข 4! ก็จะถูก Push เข้าไปใน Stack ในตำแหน่งด้านบนสุด



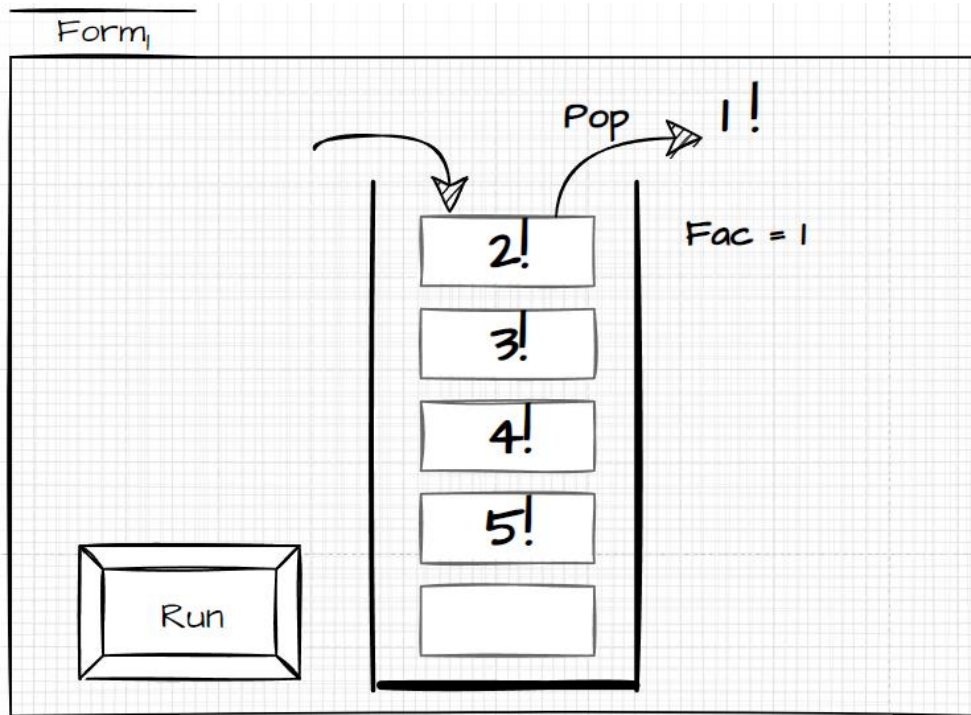
4.1.6. เช่นเดียวกันกับกรณีเมื่อครู หลังกดปุ่ม Run เลข 3! ก็จะถูก Push เข้าไปใน Stack ในตำแหน่งด้านบนสุด



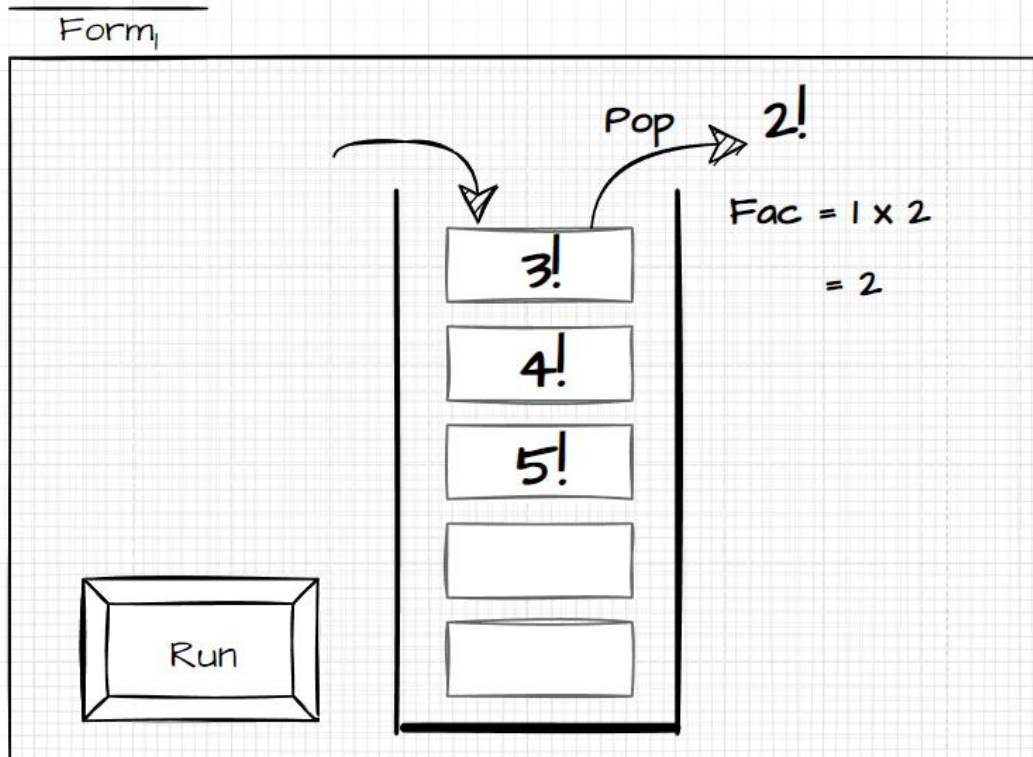
4.1.7. เช่นเดียวกันกับกรณีเมื่อครู หลังกดปุ่ม Run เลข 2! ก็จะถูก Push เข้าไปใน Stack ในตำแหน่งด้านบนสุด



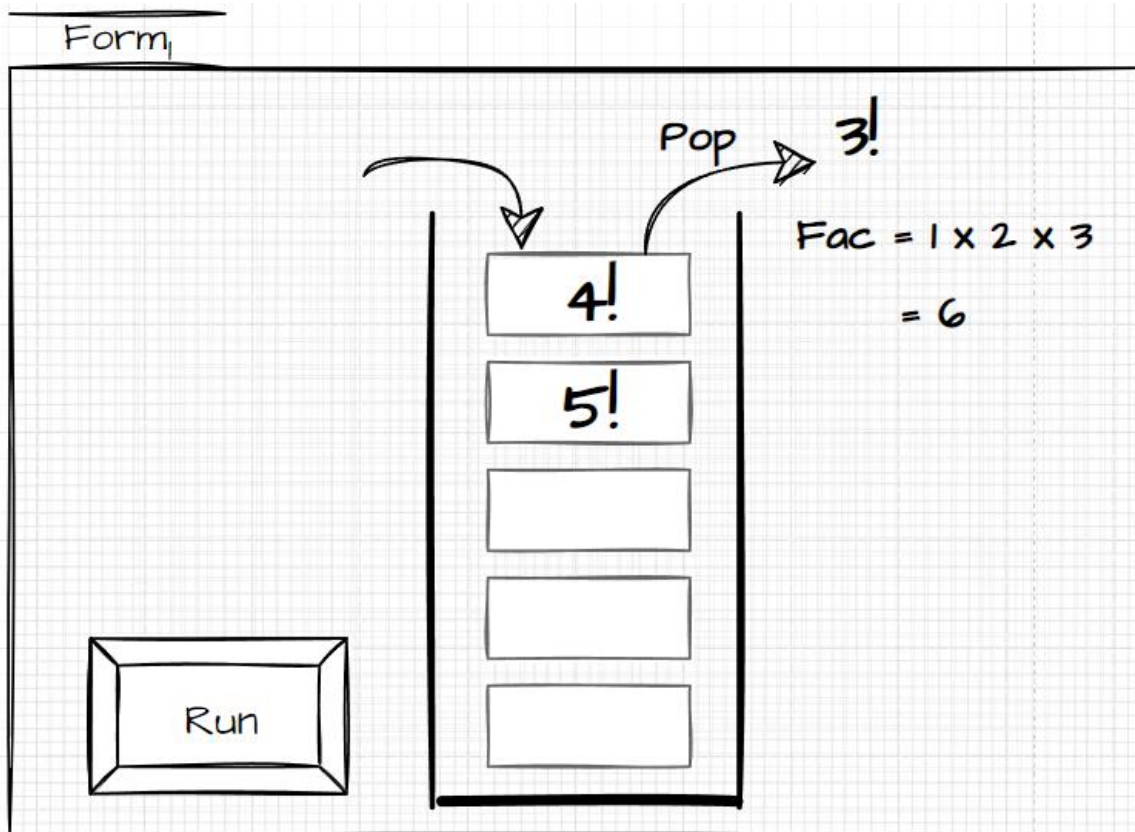
4.1.8. ทีนี้หลังจาก Push เลข 1! เข้าไปในระบบ จากกฎที่ว่า $1! = 1$ ดังนั้นทำให้เราสามารถหาคำตอบของ 1! ได้และเมื่อได้คำตอบ ให้ทำการ Pop เลข 1! ออกมา และใส่ไว้ในตัวแปร Fac พร้อมทั้งแสดงออกมาผ่านทาง Label เพื่อให้ผู้ใช้เห็นผลการคูณของ ชุดตัวเลข



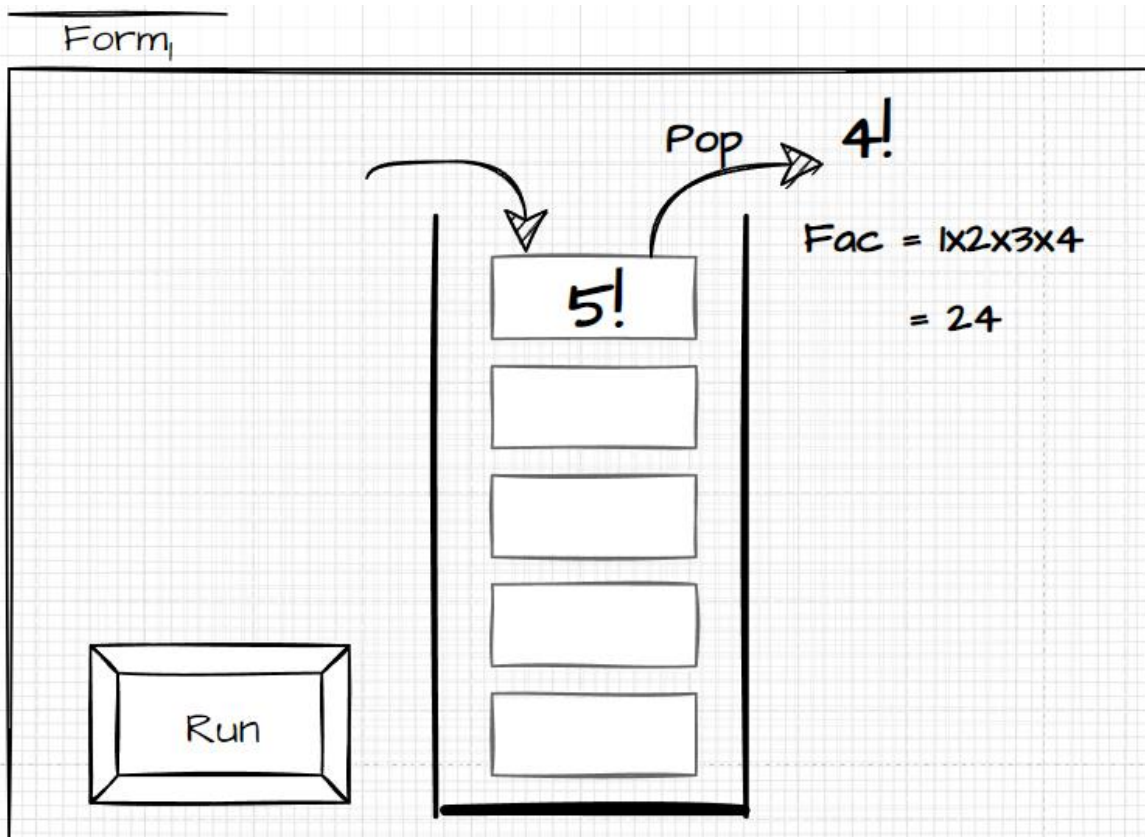
4.1.9. เมื่อกดปุ่ม Run อีกรอบ ระบบก็จะ Pop ตัวเลขบนสุดของ Stack ออกมา แล้วนำไปคูณค่า Fac ให้ผู้ใช้เห็นดังรูป



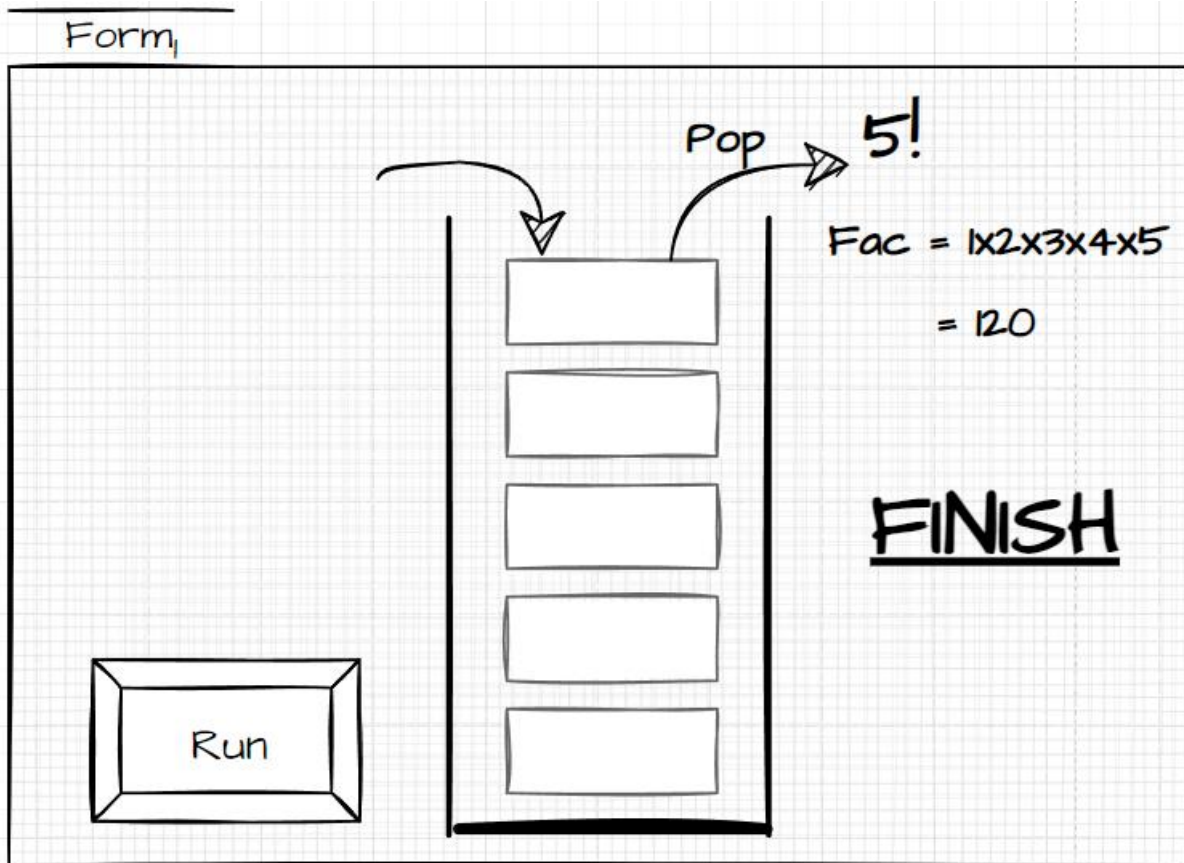
4.1.10. เมื่อกดปุ่ม Run อีกรอบ ระบบก็จะ Pop ตัวเลขบนสุดของ Stack ออกมา แล้วนำไปคูณค่า Fac ให้ผู้ใช้เห็นดังรูป



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



4.1.12. เมื่อคอมพิวเตอร์ Run อีกรอบ ระบบก็จะ Pop ตัวเลขบนสุดของ Stack ออกมา แล้วนำไปคูณค่า Fac ให้ผู้ใช้เห็นดังรูป และเมื่อถึง ค่าสุดท้าย จะต้องปรากฏคำว่า “Finish” ขึ้นดังรูปด้วยเช่นกัน



4.2. จงเขียนโค้ด โปรแกรมที่อยู่ภายในปุ่ม Run

โค้ดโปรแกรมภายในปุ่ม Run

```
Button btnRun = new Button(shell, SWT.NONE);
    btnRun.addSelectionListener(new SelectionAdapter() {
        int now_number = number;
        int Fac_number = number;
        String number_stack = "";
        int countDown = 1;
        int a[] = new int[number + 1];
        int result = 1;
        String number_result = "";
        String stack_text_number = "";
        @Override
        public void widgetSelected(SelectionEvent e) {
            {
                String text_number = String.valueOf(now_number);

                if(now_number != 0) {
                    number_stack = text_number + "!" + number_stack;
                    a[now_number] = now_number;
                    now_number--;
                    text_number = String.valueOf(now_number);
                    text.setText(number_stack);
                    label_number.setText(text_number + "-> ");
                    if(now_number == 0) {
                        label_number.setText("0");
                    }
                } else {
                    if(countDown <= Fac_number) {
                        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("-> " + countDown + "! \n" + "Fac = " + number_result + "\n=" + result);
                        number_result += "x ";
                        countDown++;
                        for(int i = countDown; i <= Fac_number; i++) {
                            text_number = String.valueOf(a[i]);
                            stack_text_number += text_number + "\n";
                        }
                        text.setText(stack_text_number);
                        stack_text_number = "";
                    } else {
                        label_number.setText("Finish");
                        text.setText("");
                    }
                }
            }
        }
    });
    btnRun.setBounds(20, 360, 149, 75);
```



```
btnRun.setText("Run");
```

5. สรุปผลการปฏิบัติการ

สามารถส่งข้อมูลรหัสผ่านฟอร์มได้ถ้าเข้าใจหลักการทำงาน

6. คำถามท้ายการทดลอง

6.1. ฟังก์ชันการทำงานใน Stack ควรมีอะไรบ้าง?

1.Push() 2.pop() 3.peek() 4.isEmpty() 5.size()

6.2. การคำนวณ Factorial มีสูตรว่าอย่างไร ?

$n! = n(n-1)!$

6.3. หลักการสร้าง Recursion คืออะไร?

- ทำความเข้าใจโจทย์
- หาจุดวนกลับ (Initial condition หรือบางคนเรียก Base case)
- หาขั้นตอนที่ต้องเรียกซ้ำ

6.4. ข้อควรระวังในการส่งข้อมูลข้ามฟอร์มคืออะไร ?

ควรใช้ชื่อแมตทอร์ที่เหมือนกัน