

ใบงานการทดลองที่ 6

เรื่อง การเขียนโปรแกรมเชิงวัตถุร่วมกับคลาสทางคณิตศาสตร์

1. จุดประสงค์ทั่วไป

- 1.1. รู้และเข้าใจในการติดต่อกับผู้ใช้ และการติดต่อระหว่างงาน
- 1.2. รู้และเข้าใจในการสร้างโปรแกรมเชิงวัตถุโดยใช้ภาษาโปรแกรมเชิงวัตถุใหม่ๆ

2. เครื่องมือและอุปกรณ์

เครื่องคอมพิวเตอร์ 1 เครื่อง ที่ติดตั้งโปรแกรม Eclipse

3. ทฤษฎีการทดลอง

- 3.1. ก่อนที่จะส่งข้อมูลจากฟอร์ม 1 ไปยังฟอร์ม 2 ควรมีการเตรียมตัวอย่างไร ?

1. ทำให้งานแต่ละส่วนรับกับ ส่งข้อมูลไปยังฟอร์ม 2 โดยไม่ต้องสร้าง method function

2. ทำให้ static รับกับ ส่งข้อมูล และ ส่งข้อมูลไปยังฟอร์ม 2 โดยไม่ต้องสร้าง static method class

3. มี Event listener รับกับ ส่งข้อมูล จากฟอร์ม 1 ไปยังฟอร์ม 2 โดยใช้ method Event listener และ add Event listener

- 3.2. ฟังก์ชันเรียกตัวเองคืออะไร? อธิบายพร้อมยกตัวอย่างประกอบ

ฟังก์ชันเรียกตัวเอง (Recursion) คือฟังก์ชันที่เรียกตัวเองซ้ำๆ กัน โดยจะวนซ้ำเรื่อยๆ จนถึงจุดที่เรียกตัวเองซ้ำๆ กันจนกว่าจะจบการทำงาน

Ex. public static double average (int[] arr, int n) {

if (n == 0) {

return 0;

return (average(arr, n-1) + arr[n-1]) / n;

4. ลำดับขั้นการปฏิบัติการ

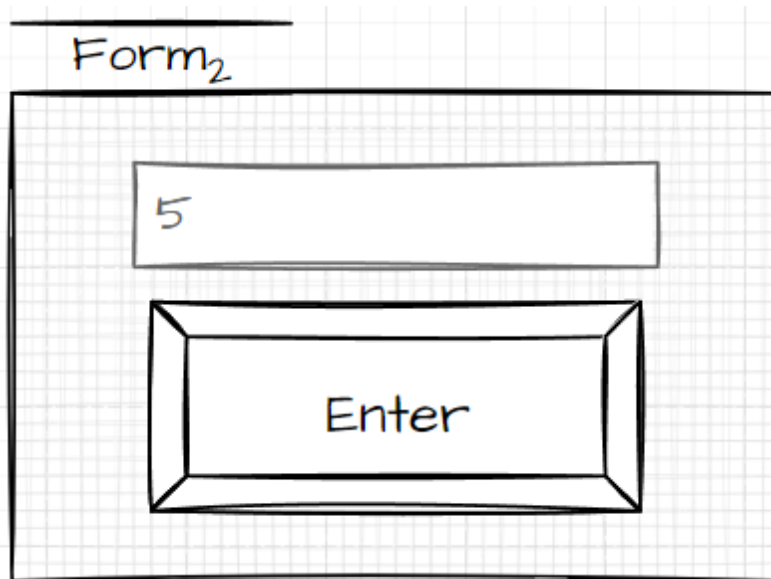
- 4.1. จงสร้าง Window Builder ในโปรแกรม Eclipse เพื่อสร้างโปรแกรมจำลองการทำงานเพื่อหาค่าของ Factorial ผ่านแบบจำลองแบบ Recursion บนโครงสร้างข้อมูลแบบ Stack โดยโปรแกรมจะมีการทำงานอยู่ 2 ฟอร์ม และมีลักษณะการทำงานดังต่อไปนี้

- 4.1.1. ฟอร์ม 1 โดยจะมีปุ่มเพื่อให้ผู้ใช้กด และเรียกหน้าต่าง ฟอร์ม 2 ขึ้นมา

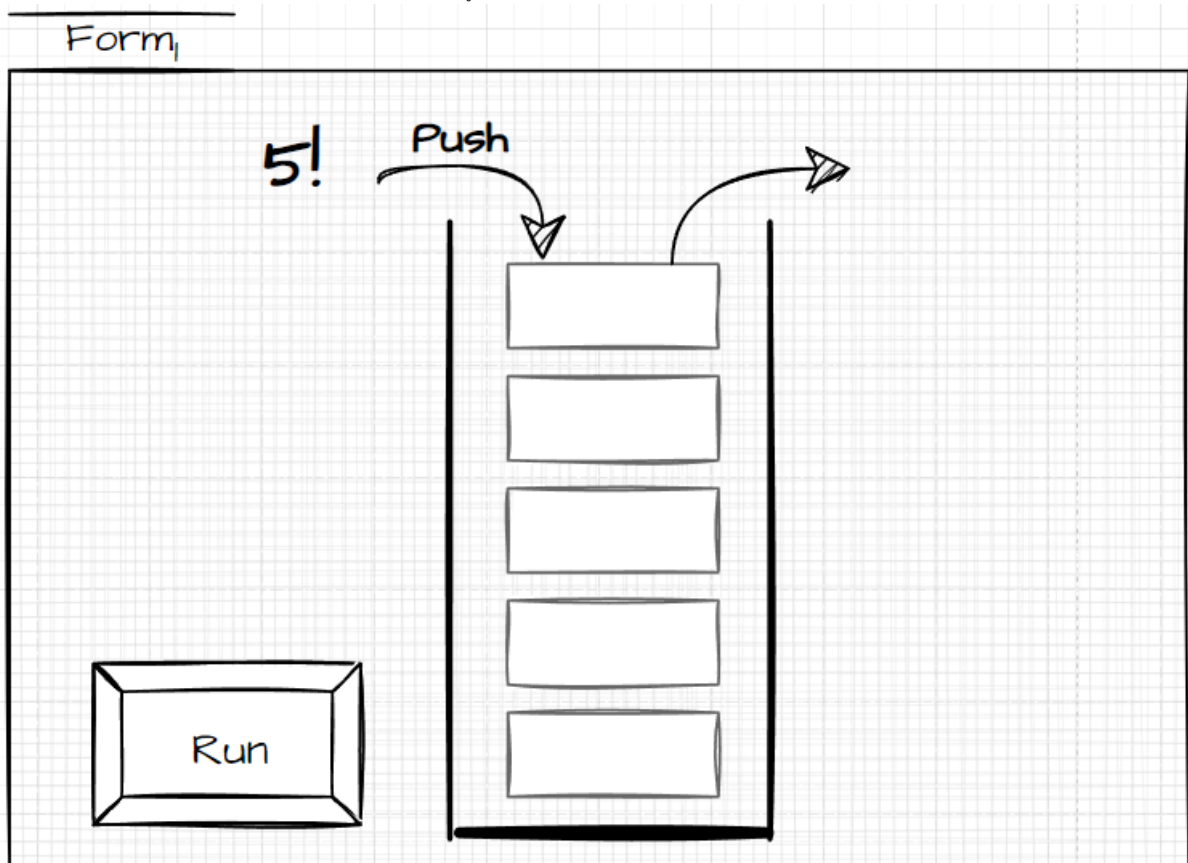
Form1

Enter Factorial Number

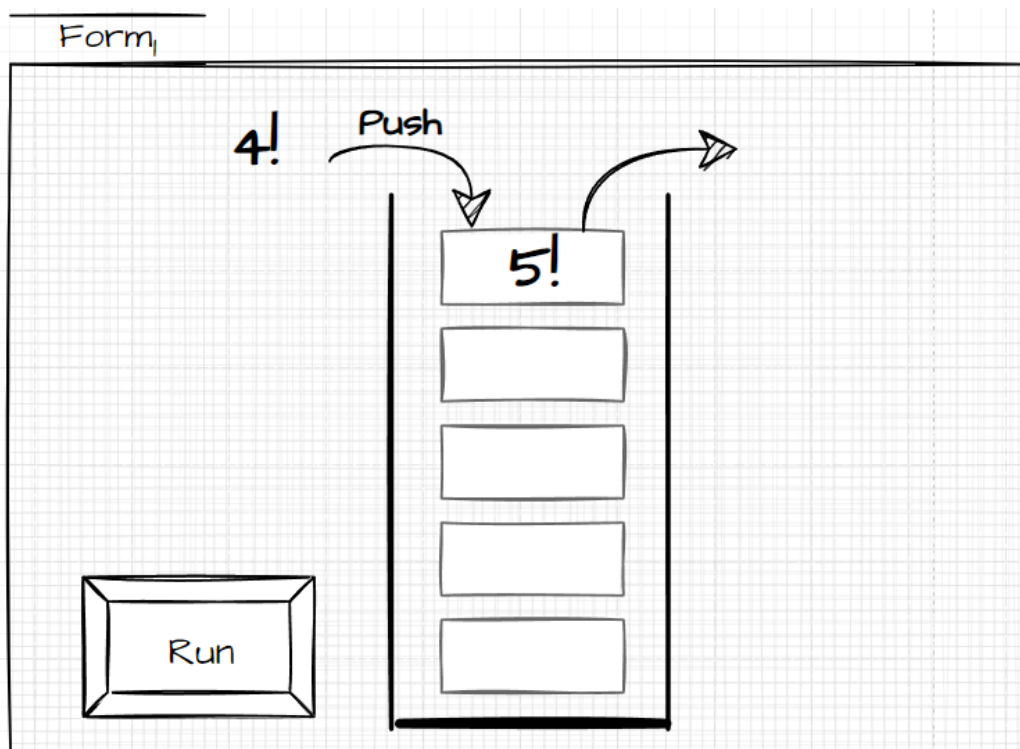
- 4.1.2. ฟอร์ม 2 เป็นหน้าต่างใหม่ที่เตรียมให้ผู้ใช้กรอกเลขที่ต้องการหาค่า Factorial ลงไปในช่อง Textbox โดยที่ผู้ใช้จะถูกจำกัดให้กรอกได้เฉพาะเลข 1 ถึง 5 เท่านั้น



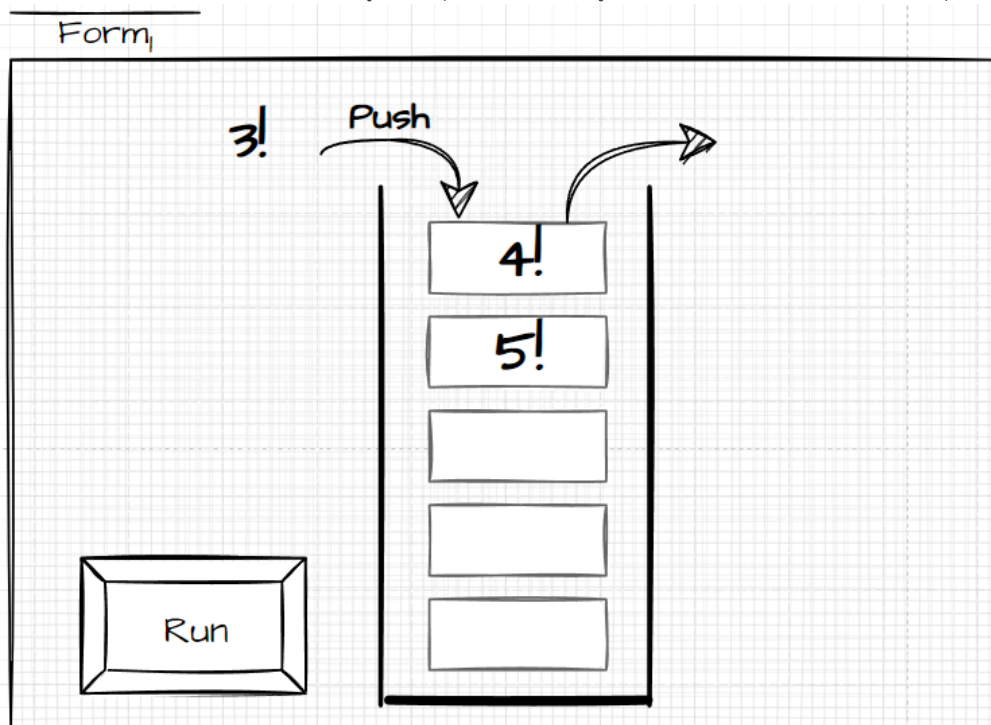
- 4.1.3. เมื่อกรอกข้อมูลในฟอร์ม 2 เสร็จแล้ว และกดปุ่ม Enter โปรแกรมจะนำเลข 5 ที่ได้จากช่อง Textbox ในฟอร์ม 2 ส่งค่ากลับไปยังฟอร์ม 1 อีกครั้ง และแสดงตัวเลขนั้นในช่องก่อนนำข้อมูล Push เข้าไปใน Stack เมื่อกดปุ่ม Run ทางด้านซ้ายล่าง ให้โปรแกรมทำการ Push ข้อมูล 5! เข้าไปใน Stack



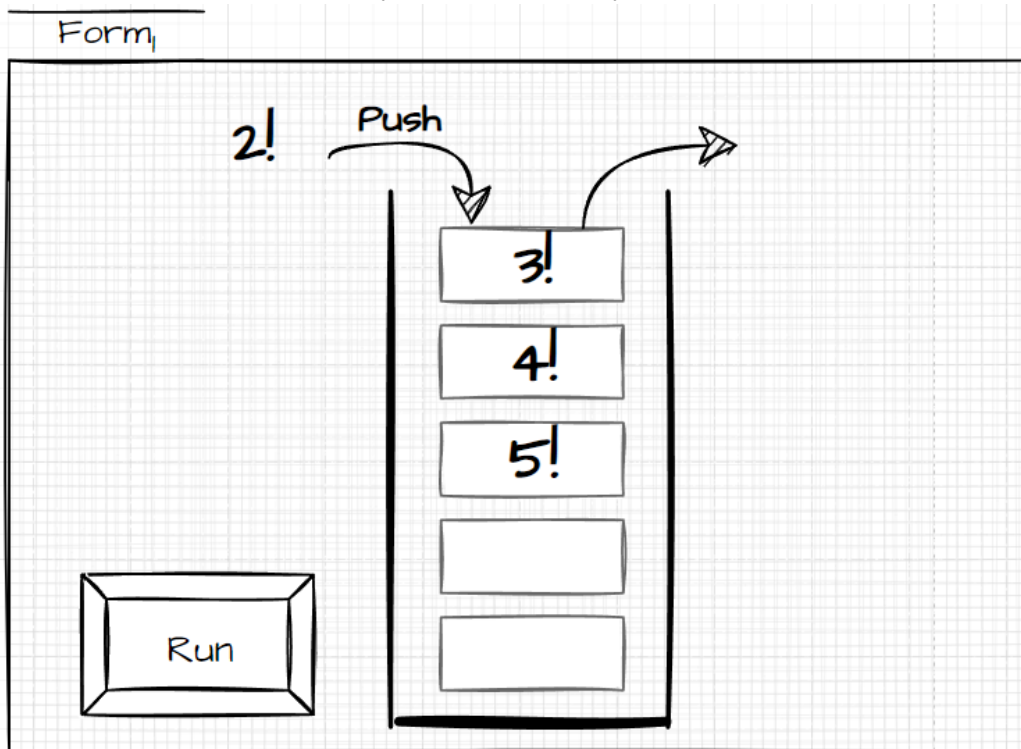
- 4.1.4. หลังจากกดปุ่ม Run เลข 5! จะเข้าไปอยู่ภายใน Stack และจะมีเลข 4! ที่รออยู่ในตำแหน่งรอ Push เข้าไปใน Stack ดังนั้นหากด้านบนสุดของ Stack ยังไม่ใช่เลข 1! เมื่อกดปุ่ม Run ระบบก็จะค่อยๆ นำข้อมูลเข้าไปใน Stack เรื่อยๆ



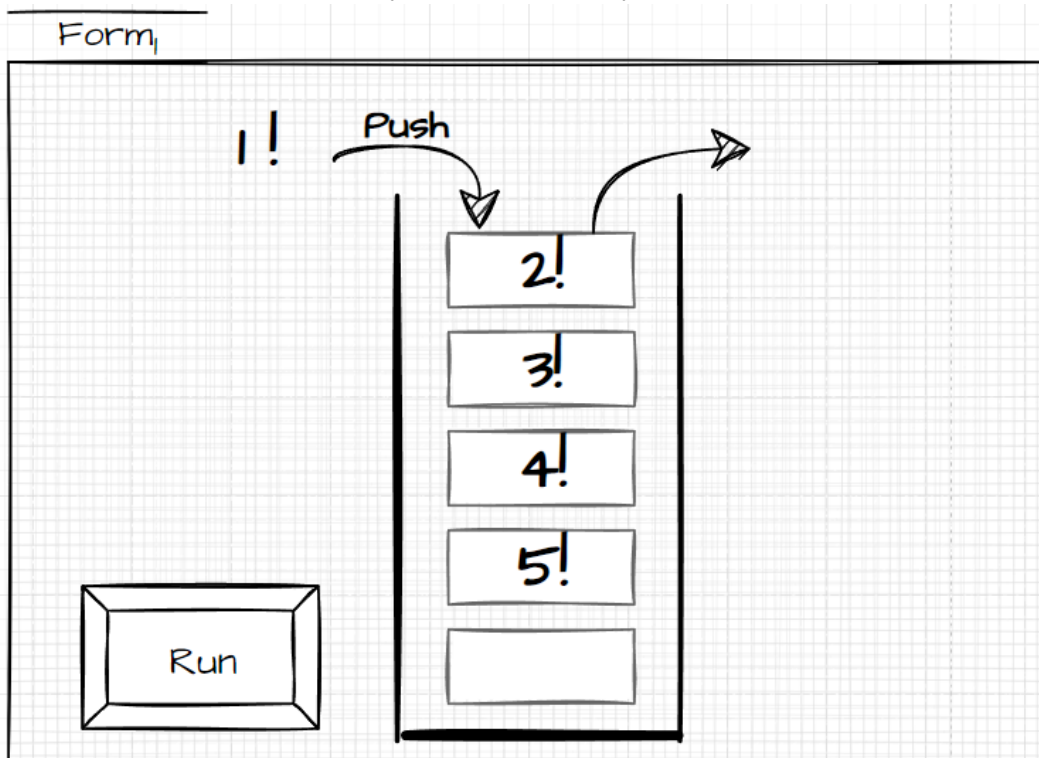
4.1.5. เช่นเดียวกันกับกรณีเมื่อครู่ หลังกดปุ่ม Run เลข 4! ก็จะถูก Push เข้าไปใน Stack ในตำแหน่งด้านบนสุด



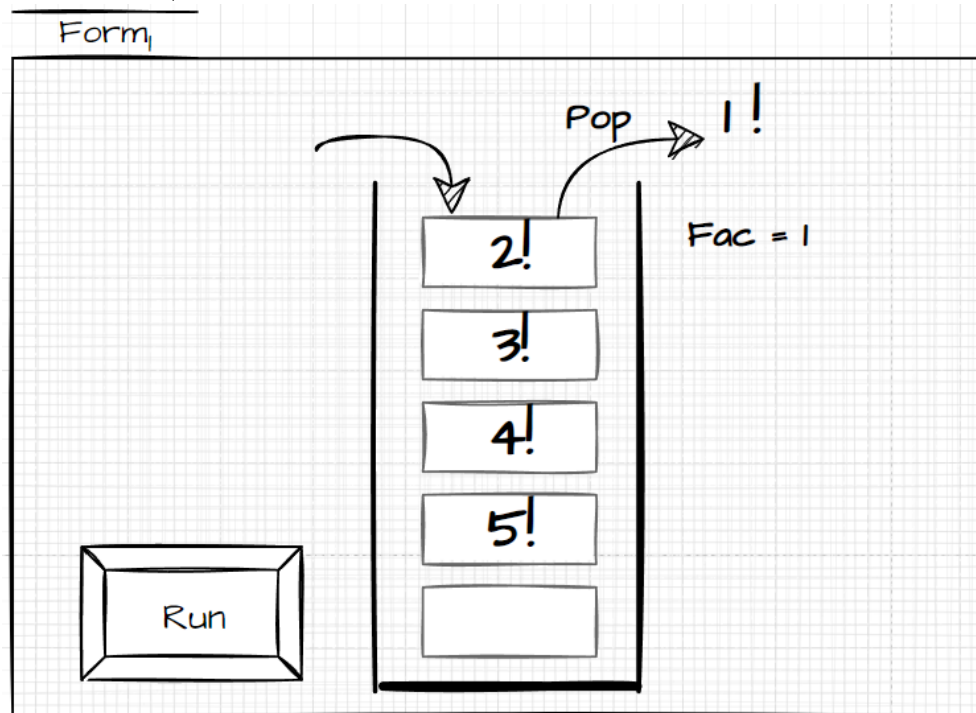
4.1.6. เช่นเดียวกันกับกรณีเมื่อครู! หลังกดปุ่ม Run เลข 3! ก็จะถูก Push เข้าไปใน Stack ในตำแหน่งด้านบนสุด



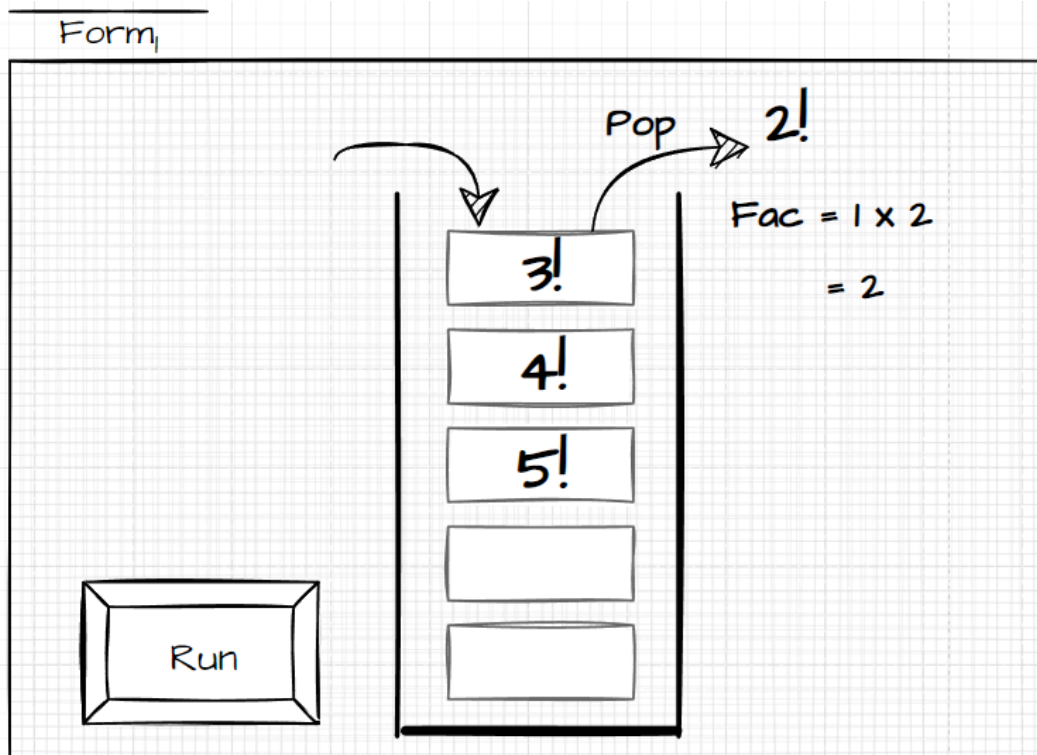
4.1.7. เช่นเดียวกันกับกรณีเมื่อครู! หลังกดปุ่ม Run เลข 2! ก็จะถูก Push เข้าไปใน Stack ในตำแหน่งด้านบนสุด



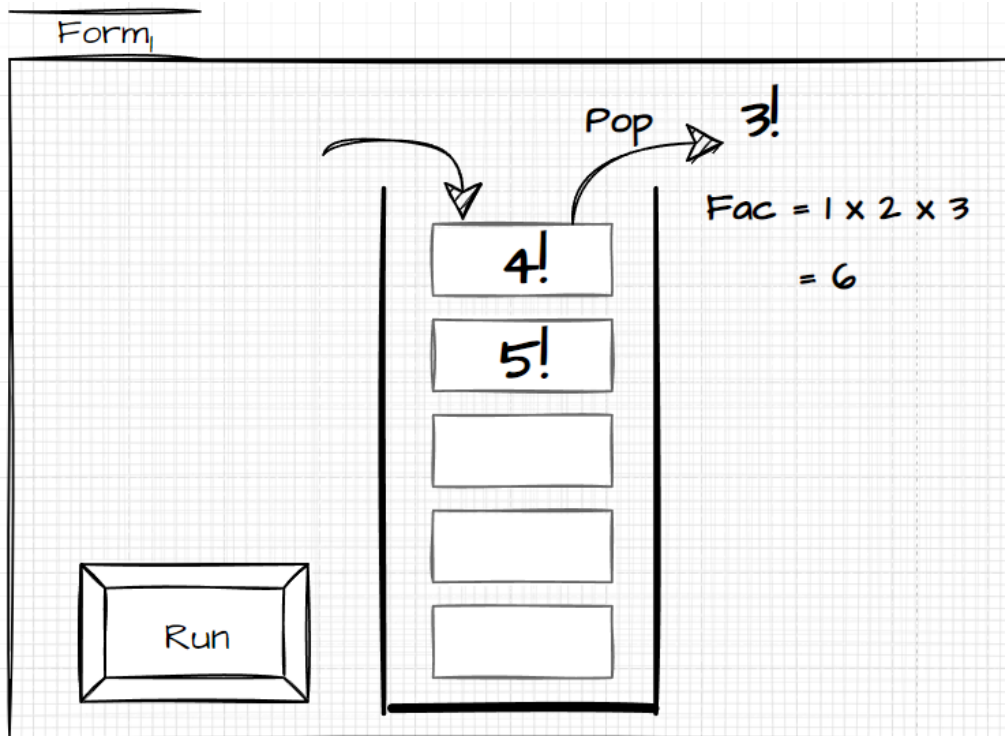
- 4.1.8. ที่นี้หลังจาก Push เลข 1! เข้าไปในระบบ จากกฎที่ว่า $1! = 1$ ดังนั้นทำให้เราสามารถหาคำตอบของ 1! ได้ และเมื่อได้คำตอบให้ทำการ Pop เลข 1! ออกมา และใส่ไว้ในตัวแปร Fac พร้อมทั้งแสดงออกมาผ่านทาง Label เพื่อให้ผู้ใช้เห็นผลการคูณของชุดตัวเลข



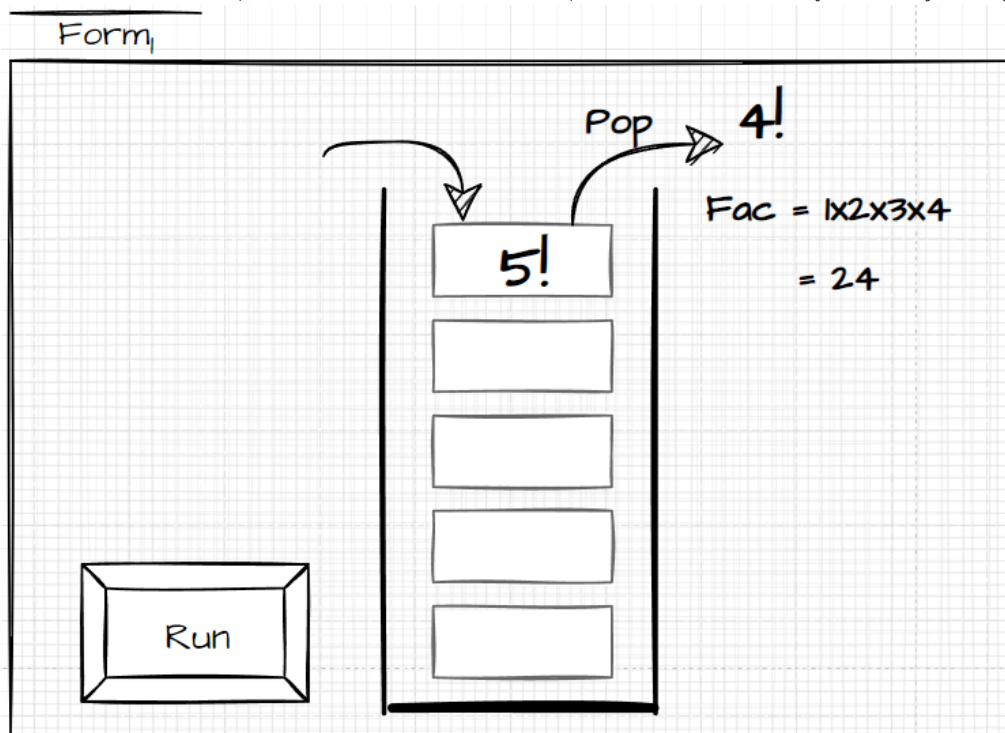
- 4.1.9. เมื่อกดปุ่ม Run อีกรอบ ระบบก็จะ Pop ตัวเลขบนสุดของ Stack ออกมา แล้วนำไปคูณค่า Fac ให้ผู้ใช้เห็นดังรูป



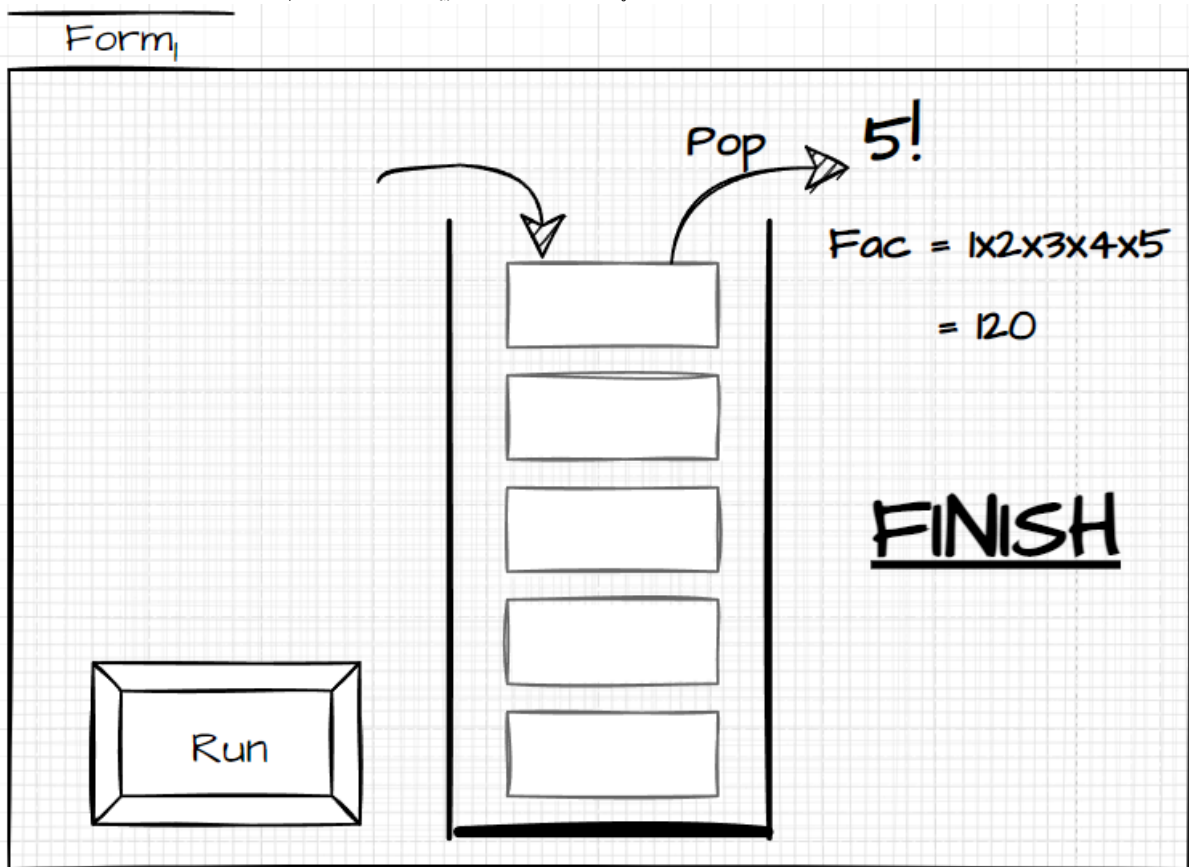
4.1.10. เมื่อกดปุ่ม Run อีกรอบ ระบบก็จะ Pop ตัวเลขบนสุดของ Stack ออกมา แล้วนำไปคูณค่า Fac ให้ผู้ใช้เห็นดังรูป



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



- 4.1.12. เมื่อกดปุ่ม Run อีกรอบ ระบบก็จะ Pop ตัวเลขบนสุดของ Stack ออกมา แล้วนำไปคูณค่า Fac ให้ผู้ใช้เห็นดังรูป และเมื่อถึงค่าสุดท้าย จะต้องปรากฏคำว่า "Finish" ขึ้นดังรูปด้วยเช่นกัน



- 4.2. จงเขียนโค้ดโปรแกรมที่อยู่ภายในปุ่ม Run

โค้ดโปรแกรมภายในปุ่ม Run

```
95 Button btnRun = new Button(shell, SWT.NONE);
96 btnRun.setBounds(36, 174, 102, 36);
97 btnRun.setText("Run");
98 btnRun.addSelectionListener(new SelectionAdapter() {
99     int now_number = number;
100     String number_stack = "";
101     int countDown = number;
102     int a[]=new int[number + 1];
103     int result = 1;
104     String number_result = "";
105     String stack_text_number = "";
106     public void widgetSelected(SelectionEvent e) {
107         String text_number = String.valueOf(now_number);
108         if(now_number != 0) {
109             number_stack += text_number + "|";
110             a[now_number] = now_number;
111             now_number--;
112             text_number = String.valueOf(now_number);
113             txtNewText.setText(number_stack);
114             label_number.setText(text_number);
115             if(now_number == 0) {
116                 label_number.setText("0");
117             }
118         }
119     }
120 }
```

```

118     } else {
119         if(countDown > 0) {
120             int i_number = a[countDown];
121             text_number = String.valueOf(i_number);
122             number_result += text_number;
123             result += i_number;
124             //System.out.print(number_result + "|");
125             label_result.setText(number_result + "=" + result);
126             number_result += "x";
127             countDown--;
128             for(int i = countDown; i > 0; i--) {
129                 text_number = String.valueOf(a[i]);
130                 stack_text_number += text_number + "|";
131             }
132             txtNewText.setText(stack_text_number);
133             stack_text_number = "";
134         } else {
135             label_number.setText("Finish");
136             txtNewText.setText("Finish");
137         }
138     }
139 }
140 }
141 );

```

```

142     Label lblStack = formToolkit.createLabel(shell, "Stack", SWT.NONE);
143     lblStack.setBounds(244, 10, 55, 20);
144 }
145 }
146 }
147

```

5. สรุปผลการปฏิบัติการ

สิ่งที่พบเมื่อรันโปรแกรมได้ผลตามที่ออกแบบไว้คือได้ผลลัพธ์ที่ถูกต้องตามที่กำหนดไว้
 ข้อสังเกตในการรันโปรแกรมคือ
 การรันโปรแกรมพบว่าโปรแกรมสามารถทำงานได้ตามที่ออกแบบไว้
 ข้อสังเกตในการรันโปรแกรมคือโปรแกรมสามารถทำงานได้ตามที่ออกแบบไว้
 ข้อสังเกตในการรันโปรแกรมคือโปรแกรมสามารถทำงานได้ตามที่ออกแบบไว้
 ข้อสังเกตในการรันโปรแกรมคือโปรแกรมสามารถทำงานได้ตามที่ออกแบบไว้

6. คำถามท้ายการทดลอง

6.1. ฟังก์ชันการทำงานใน Stack ควรมีอะไรบ้าง?

1. push() 5. size()

2. pop()

3. peek()

4. isEmpty()

6.2. การคำนวณ Factorial มีสูตรอย่างไร ?

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

6.3. หลักการสร้าง Recursion คืออะไร?

1. กำหนดเงื่อนไขสุดขอบ (base case)

2. เรียกตัวเองซ้ำ

3. กำหนดเงื่อนไขการจบการทำงาน โดยปกติมักใช้ n < 0 หรือ n < 1
เรียกในจุดนั้น

6.4. ข้อควรระวังในการส่งข้อมูลข้ามฟอร์มคืออะไร ?

1. การส่งข้อมูล : ถ้าในฟังก์ชันส่งค่ากลับแล้วแต่ไม่ส่งค่ากลับ

2. การรับข้อมูล : ถ้าไม่รับข้อมูลแล้วแต่ยังรับค่าในฟังก์ชันโดยไม่ได้

3. การลบข้อมูล : ถ้าลบข้อมูลแล้วแต่ไม่ลบ