

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 ?

สร้างตัวแปรของทั้ง 2 ให้เรียบร้อยก่อนส่ง

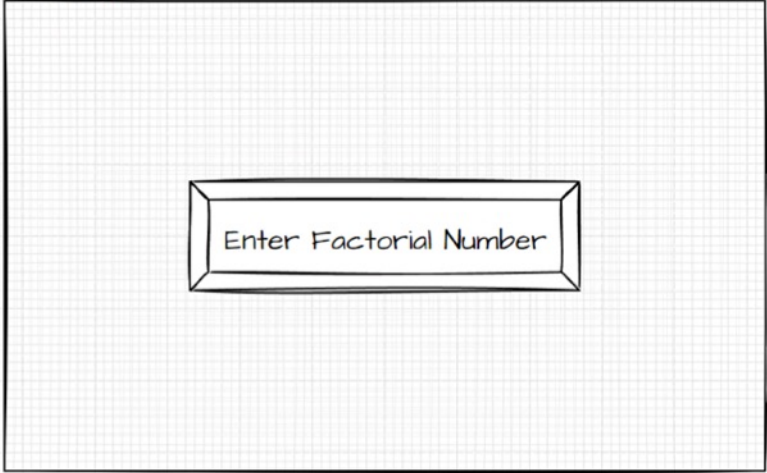
- 3.2. ฟังก์ชันเรียกตัวเองคืออะไร? อธิบายพร้อมยกตัวอย่างประกอบ
- การที่มันเรียกใช้ตัวมันเองในฟังก์ชันเช่น

```
int x,y;
if(N==0) {
    printf("\tRoll back point\n");
    return(1);
}
else{
    x=N-1;
    printf("%2d! = %2d*%2d!\n",N,N,x);
    y = factorial(x);
    printf("%2d! = %2d*%3d! =%5d\n",N,N,y,y*N);
    return(N*y);
}
```

4. ลำดับชั้นการปฏิบัติการ

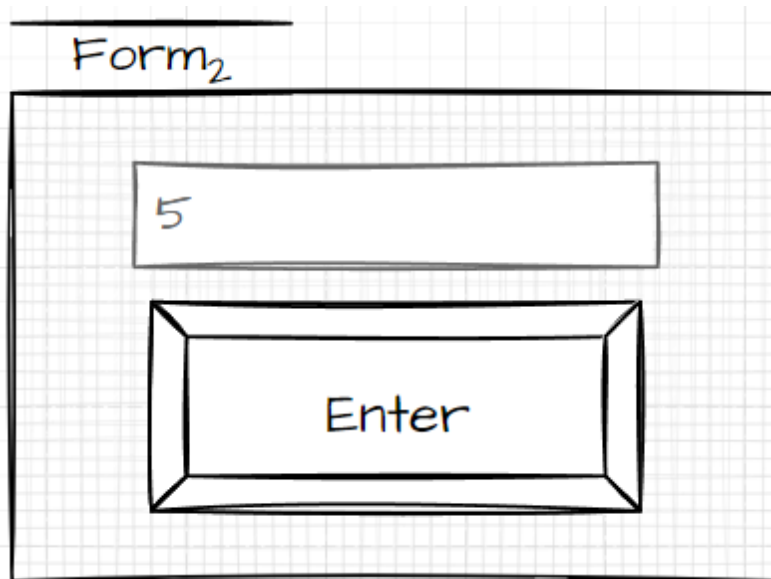
- 4.1. จงสร้าง Window Builder ในโปรแกรม Eclipse เพื่อสร้างโปรแกรมจำลองการทำงานเพื่อหาค่าของ Factorial ผ่านแบบจำลองแบบ Recursion บนโครงสร้างข้อมูลแบบ Stack โดยโปรแกรมจะมีการทำงานอยู่ 2 ฟอรัม และมีลักษณะการทำงานดังต่อไปนี้
- 4.1.1. ฟอรัม 1 โดยจะมีปุ่มเพื่อให้ผู้ใช้กด และเรียกหน้าต่าง ฟอรัม 2 ขึ้นมา

Form₁

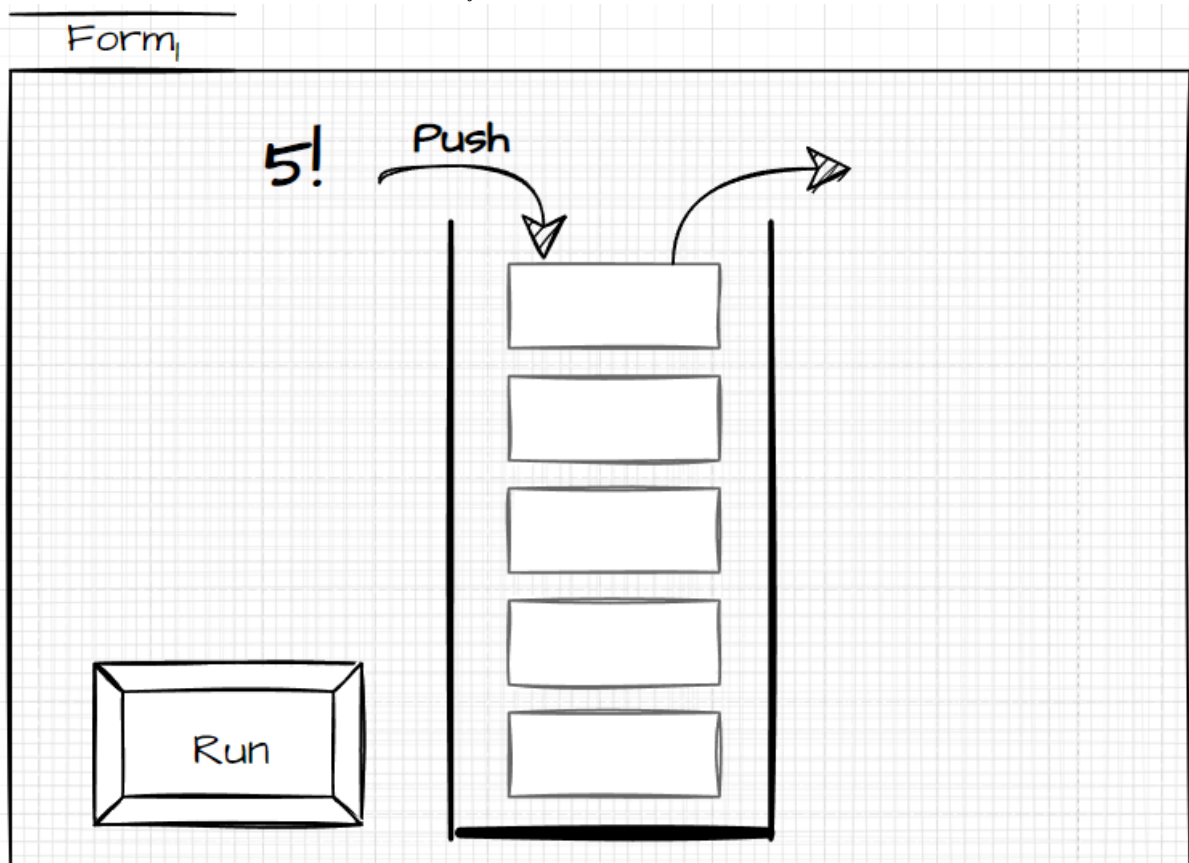


Enter Factorial Number

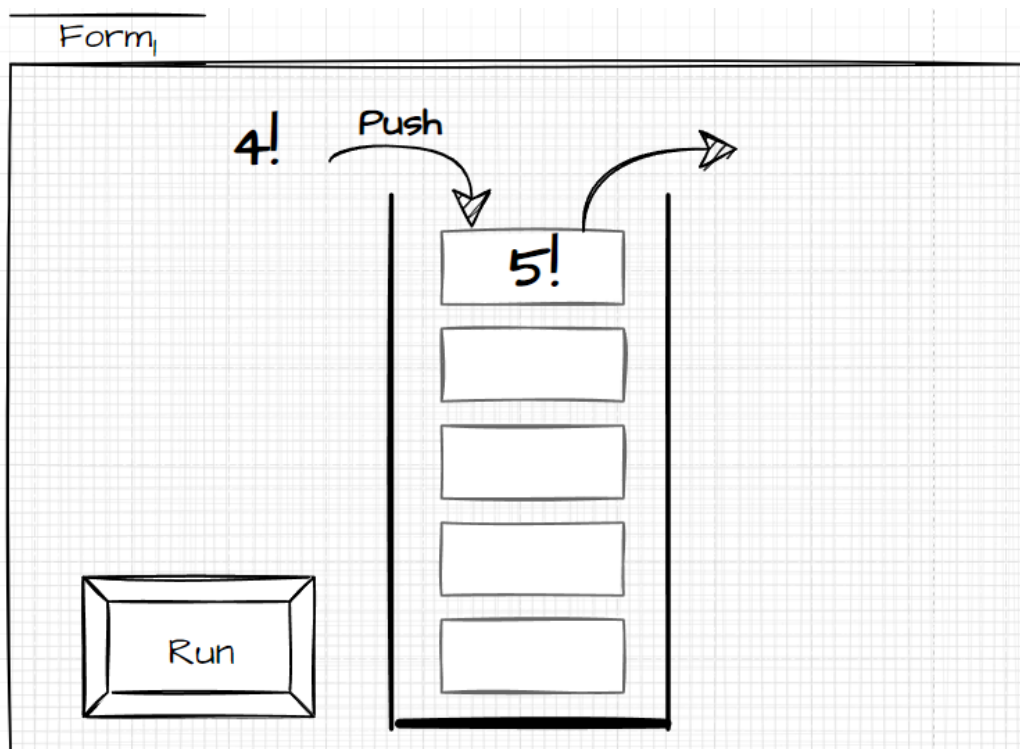
- 4.1.2. ฟอรัม 2 เป็นหน้าต่างใหม่ที่เตรียมให้ผู้ใช้ออกเลขที่ต้องการหาค่า Factorial ลงไปในช่อง Textbox โดยที่ผู้ใช้จะถูกจำกัดให้กรอกได้เฉพาะเลข 1 ถึง 5 เท่านั้น



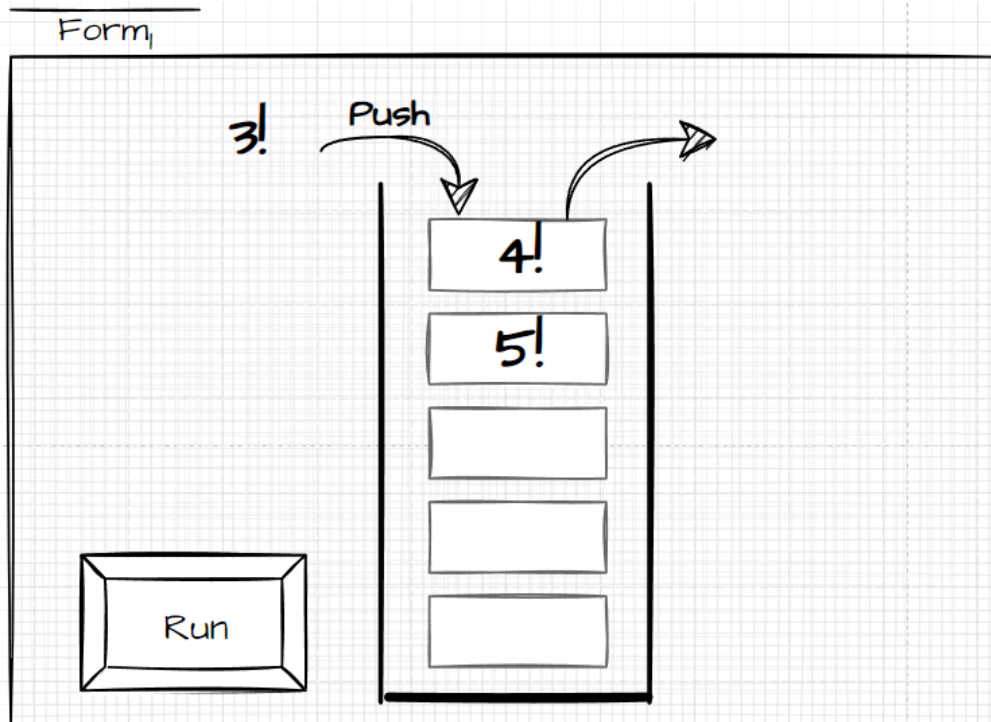
- 4.1.3. เมื่อกรอกข้อมูลในฟอร์ม 2 เสร็จแล้ว และกดปุ่ม Enter โปรแกรมจะนำเลข 5 ที่ได้จากช่อง Textbox ในฟอร์ม 2 ส่งค่ากลับไปยังฟอร์ม 1 อีกครั้ง และแสดงตัวเลขนั้นในช่องก่อนนำข้อมูล Push เข้าไปใน Stack เมื่อกดปุ่ม Run ทางด้านซ้ายล่าง ให้โปรแกรมทำการ Push ข้อมูล 5! เข้าไปใน Stack



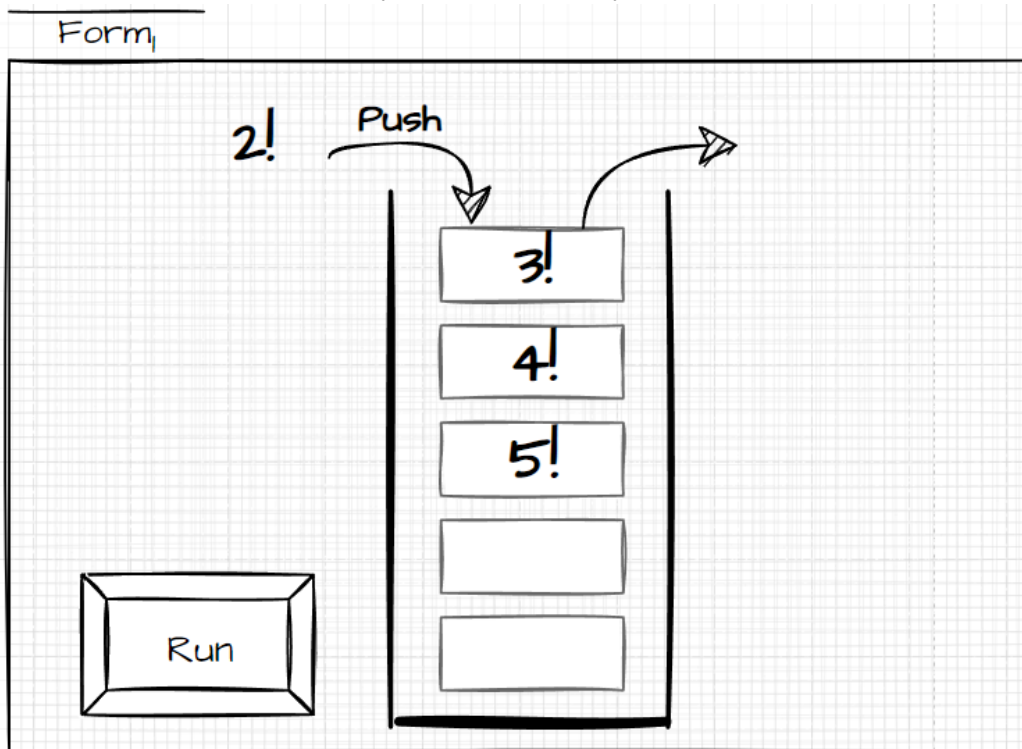
- 4.1.4. หลังจากกดปุ่ม Run เลข 5! จะเข้าไปอยู่ภายใน Stack และจะมีเลข 4! ที่รออยู่ในตำแหน่งรอ Push เข้าไปใน Stack ดังนั้นหากด้านบนสุดของ Stack ยังไม่ใช่เลข 1! เมื่อกดปุ่ม Run ระบบก็จะค่อยๆ นำข้อมูลเข้าไปใน Stack เรื่อยๆ



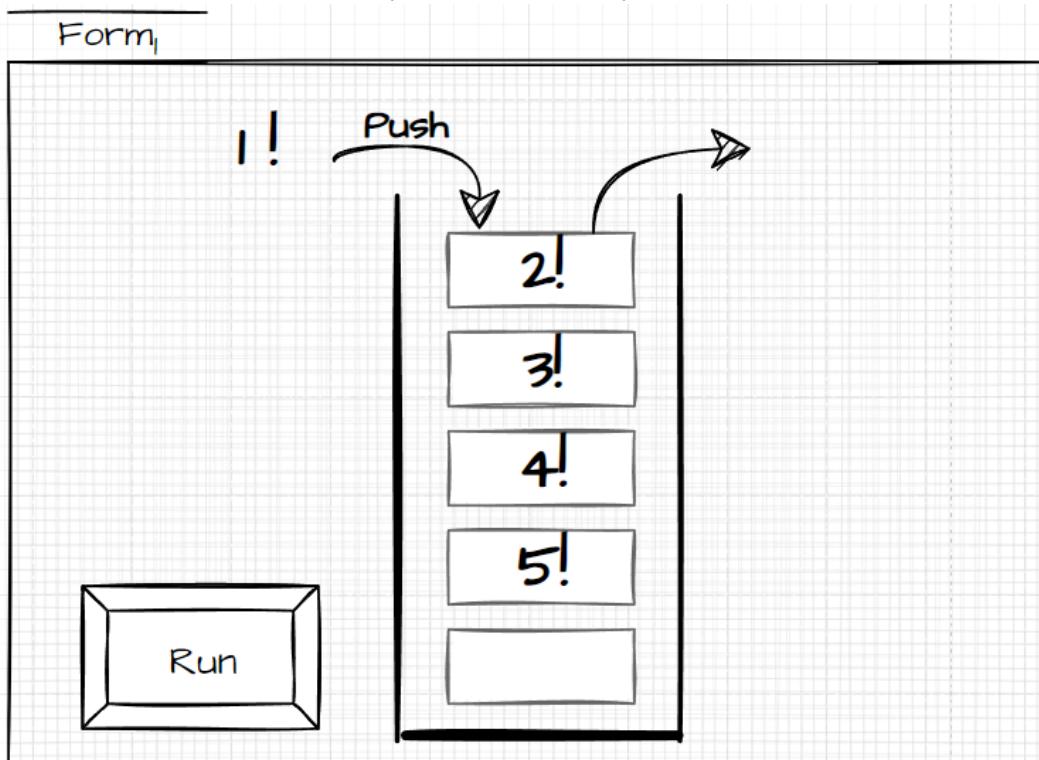
4.1.5. เช่นเดียวกันกับกรณีเมื่อครู่ หลังกดปุ่ม Run เลข 4! ก็จะถูก Push เข้าไปใน Stack ในตำแหน่งด้านบนสุด



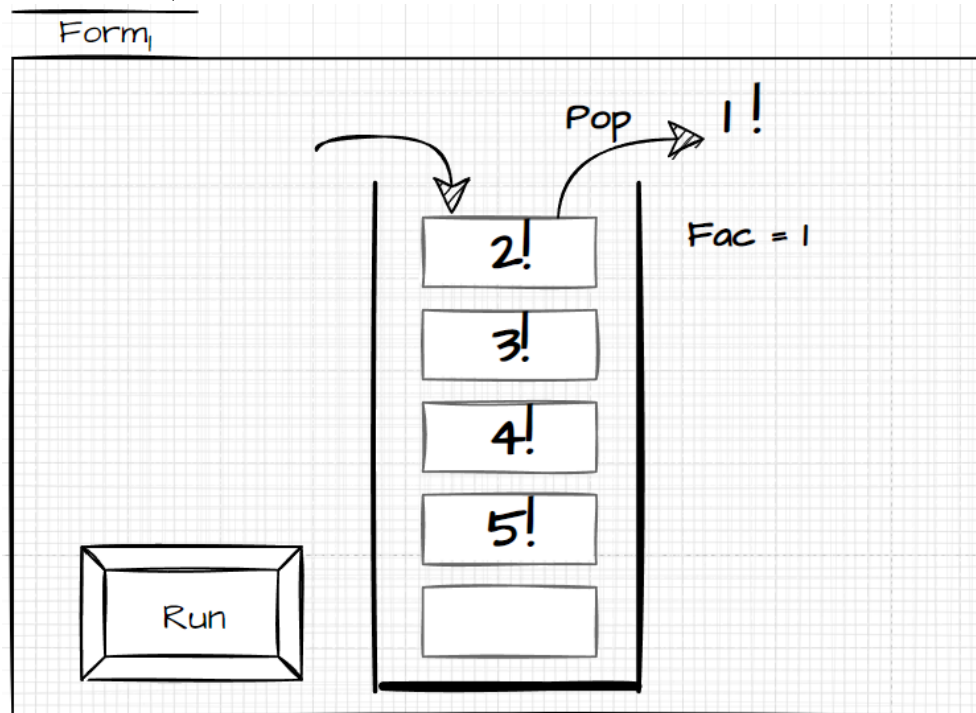
4.1.6. เช่นเดียวกันกับกรณีเมื่อครู! หลังกดปุ่ม Run เลข 3! ก็จะถูก Push เข้าไปใน Stack ในตำแหน่งด้านบนสุด



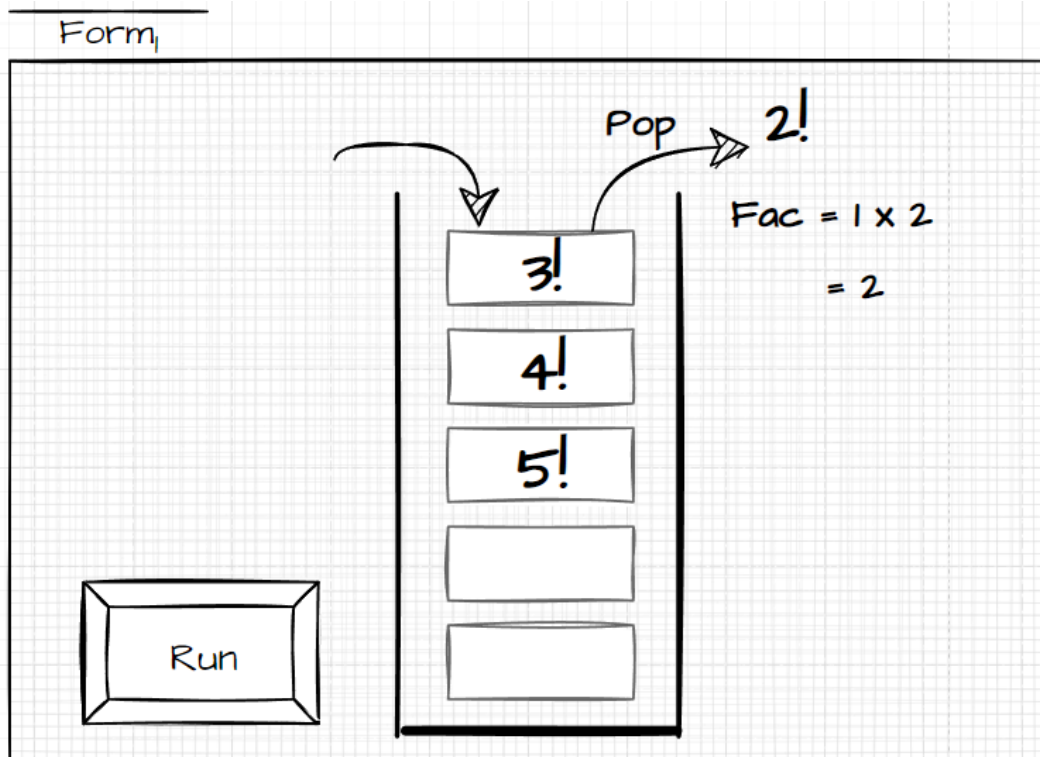
4.1.7. เช่นเดียวกันกับกรณีเมื่อครู! หลังกดปุ่ม Run เลข 2! ก็จะถูก Push เข้าไปใน Stack ในตำแหน่งด้านบนสุด



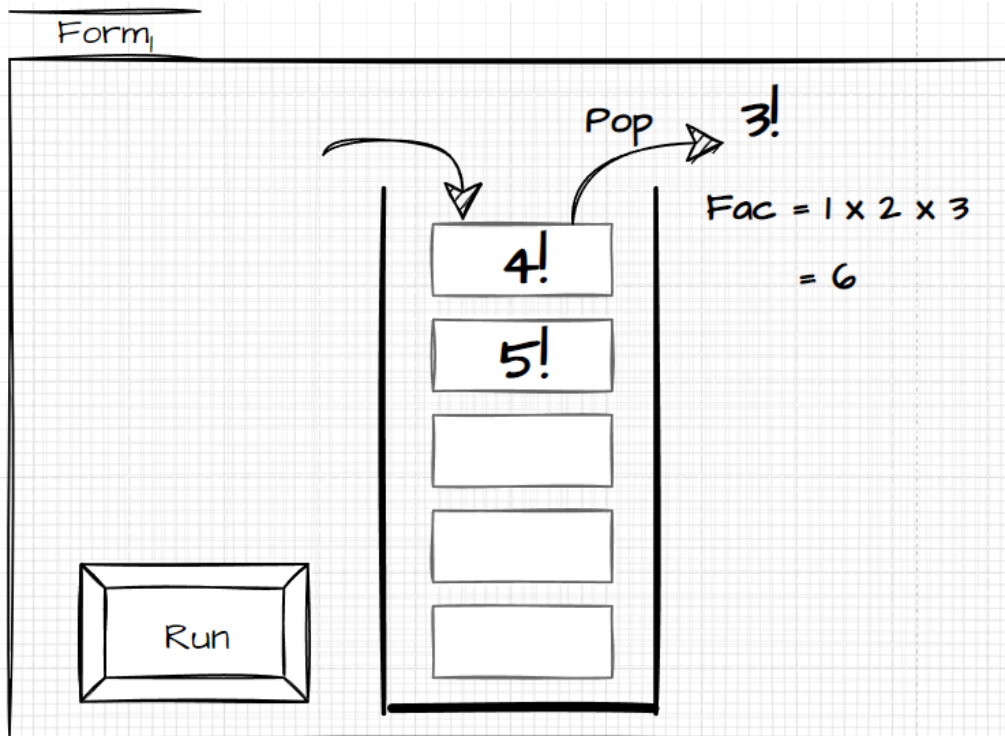
- 4.1.8. ที่นี้หลังจาก Push เลข 1! เข้าไปในระบบ จากกฎที่ว่า $1! = 1$ ดังนั้นทำให้เราสามารถหาคำตอบของ 1! ได้ และเมื่อได้คำตอบให้ทำการ Pop เลข 1! ออกมา และใส่ไว้ในตัวแปร Fac พร้อมทั้งแสดงออกมาผ่านทาง Label เพื่อให้ผู้ใช้เห็นผลการคูณของชุดตัวเลข



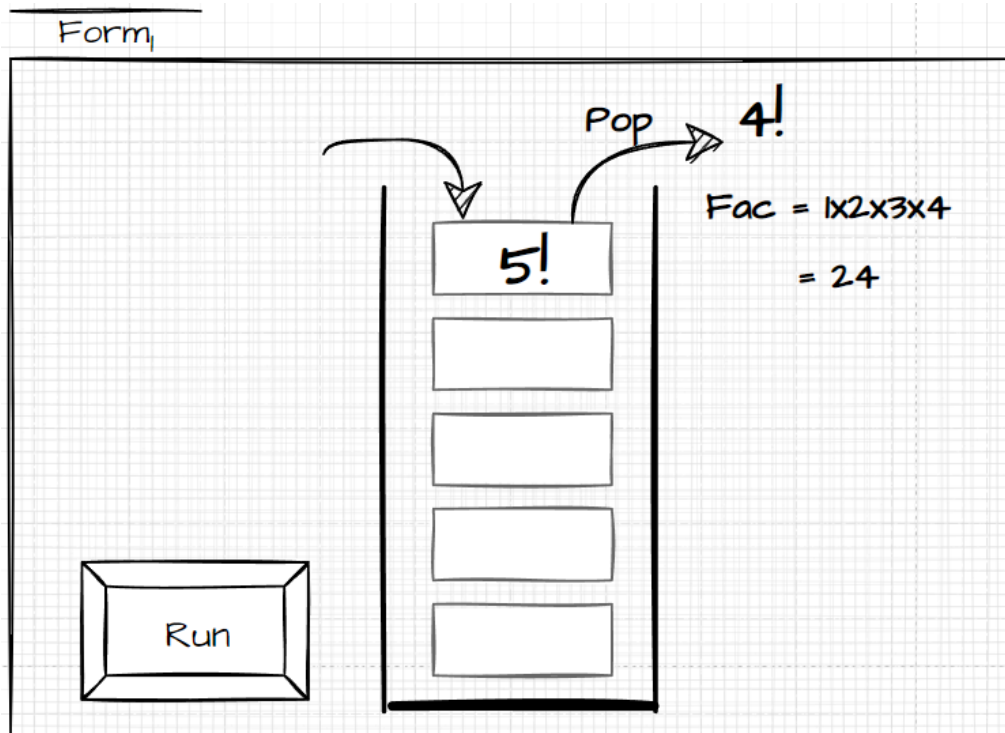
- 4.1.9. เมื่อกดปุ่ม Run อีกรอบ ระบบก็จะ Pop ตัวเลขบนสุดของ Stack ออกมา แล้วนำไปคูณค่า Fac ให้ผู้ใช้เห็นดังรูป



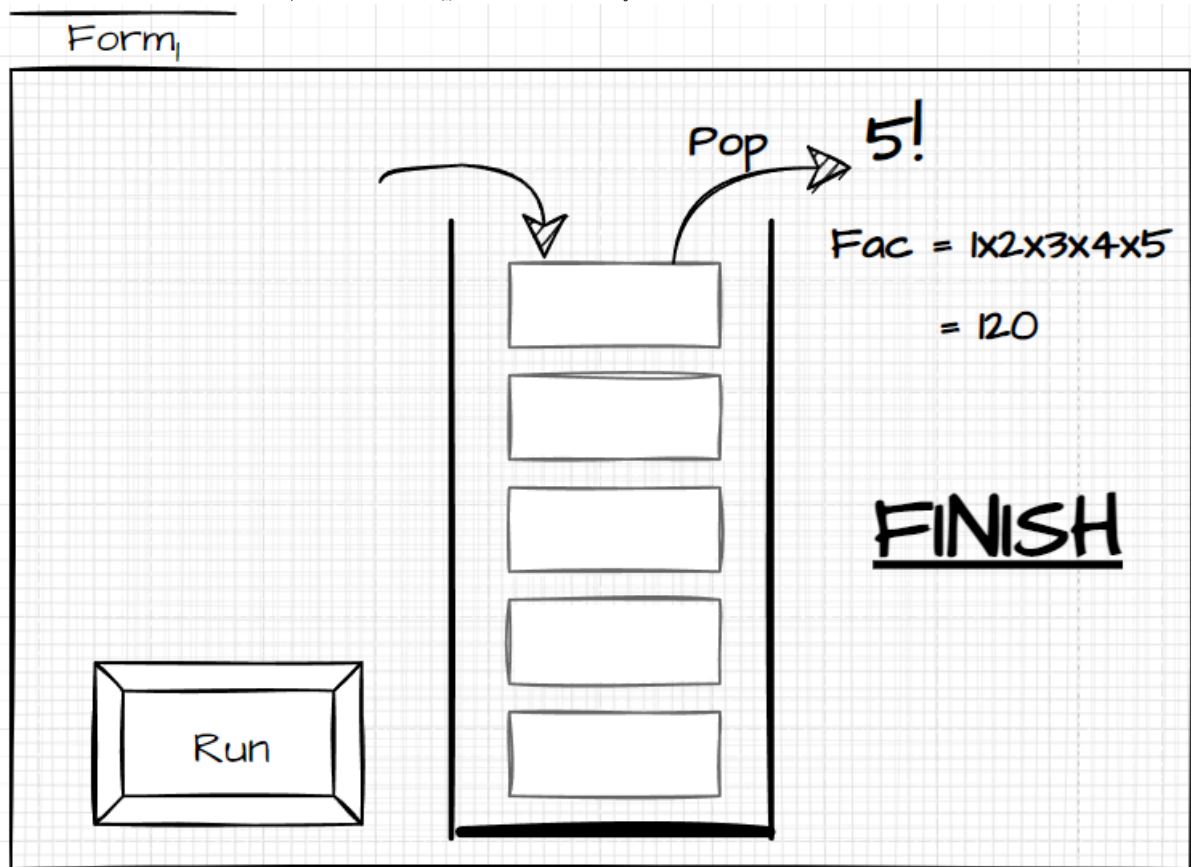
4.1.10. เมื่อกดปุ่ม Run อีกรอบ ระบบก็จะ Pop ตัวเลขบนสุดของ Stack ออกมา แล้วนำไปคูณค่า Fac ให้ผู้ใช้เห็นดังรูป



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

```
110 Button btnRun = new Button(fm1, SWT.NONE);
111 btnRun.addSelectionListener(new SelectionAdapter() {
112     @Override
113     public void widgetSelected(SelectionEvent e) {
114         if(backward) {
115             if(i > 1) {
116                 lbPush.setText((i-1) + "! Push");
117             } else {
118                 lbPush.setText("");
119             }
120             lbPop.setText("");
121         } else {
122             lbPush.setText("");
123             lbPop.setText("Pop " + (i-1) + "!");
124             switch(i-1) {
125                 case 1:
126                     facAns.setText("Fac\t= 1\n\t= " + (ans = fac(i-1)));
127                     break;
128                 case 2:
129                     facAns.setText("Fac\t= 1x2\n\t= " + (ans = fac(i-1)));
130                     break;
131                 case 3:
132                     facAns.setText("Fac\t= 1x2x3\n\t= " + (ans = fac(i-1)));
133                     break;
134                 case 4:
135                     facAns.setText("Fac\t= 1x2x3x4\n\t= " + (ans = fac(i-1)));
136                     break;
137                 case 5:
138                     facAns.setText("Fac\t= 1x2x3x4x5\n\t= " + (ans = fac(i-1)));
139                     break;
140             }
141         }
142         switch(num) {
143             case 1:
144
```



```

145         switch(i) {
146             case 1:
147                 lbStack5.setText("1");
148                 lbStack4.setText("");
149                 lbStack3.setText("");
150                 lbStack2.setText("");
151                 lbStack1.setText("");
152                 break;
153             case 2:
154                 lbStack5.setText("");
155                 lbStack4.setText("");
156                 lbStack3.setText("");
157                 lbStack2.setText("");
158                 lbStack1.setText("");
159                 break;
160             case 3:
161                 fm1.close();
162                 break;
163         }
164         break;
165     case 2:
166         switch(i) {
167             case 1:
168                 lbStack5.setText("1");
169                 lbStack4.setText("2");
170                 lbStack3.setText("");
171                 lbStack2.setText("");
172                 lbStack1.setText("");
173                 break;
174             case 2:
175                 lbStack5.setText("2");
176                 lbStack4.setText("");
177                 lbStack3.setText("");
178                 lbStack2.setText("");
179                 lbStack1.setText("");
180                 break;

```

```

181         case 3:
182             lbStack5.setText("");
183             lbStack4.setText("");
184             lbStack3.setText("");
185             lbStack2.setText("");
186             lbStack1.setText("");
187             break;
188         case 4:
189             fm1.close();
190             break;
191     }
192     break;
193 case 3:
194     switch(i) {
195         case 1:
196             lbStack5.setText("1");
197             lbStack4.setText("2");
198             lbStack3.setText("3");
199             lbStack2.setText("");
200             lbStack1.setText("");
201             break;
202         case 2:
203             lbStack5.setText("2");
204             lbStack4.setText("3");
205             lbStack3.setText("");
206             lbStack2.setText("");
207             lbStack1.setText("");
208             break;
209         case 3:
210             lbStack5.setText("3");
211             lbStack4.setText("");
212             lbStack3.setText("");
213             lbStack2.setText("");
214             lbStack1.setText("");
215             break;

```

```

216         case 4:
217             lbStack5.setText("");
218             lbStack4.setText("");
219             lbStack3.setText("");
220             lbStack2.setText("");
221             lbStack1.setText("");
222             break;
223         case 5:
224             fm1.close();
225             break;
226     }
227     break;
228 case 4:
229     switch(i) {
230     case 1:
231         lbStack5.setText("1");
232         lbStack4.setText("2");
233         lbStack3.setText("3");
234         lbStack2.setText("4");
235         lbStack1.setText("");
236         break;
237     case 2:
238         lbStack5.setText("2");
239         lbStack4.setText("3");
240         lbStack3.setText("4");
241         lbStack2.setText("");
242         lbStack1.setText("");
243         break;
244     case 3:
245         lbStack5.setText("3");
246         lbStack4.setText("4");
247         lbStack3.setText("");
248         lbStack2.setText("");
249         lbStack1.setText("");
250         break;

```

```

251         case 4:
252             lbStack5.setText("4");
253             lbStack4.setText("");
254             lbStack3.setText("");
255             lbStack2.setText("");
256             lbStack1.setText("");
257             break;
258         case 5:
259             lbStack5.setText("");
260             lbStack4.setText("");
261             lbStack3.setText("");
262             lbStack2.setText("");
263             lbStack1.setText("");
264             break;
265         case 6:
266             fm1.close();
267             break;
268     }
269     break;
270 case 5:
271     switch(i) {
272     case 1:
273         lbStack5.setText("1");
274         lbStack4.setText("2");
275         lbStack3.setText("3");
276         lbStack2.setText("4");
277         lbStack1.setText("5");
278         break;
279     case 2:
280         lbStack5.setText("2");
281         lbStack4.setText("3");
282         lbStack3.setText("4");
283         lbStack2.setText("5");
284         lbStack1.setText("");
285         break;
286     case 3:

```

```

287         lbStack5.setText("3");
288         lbStack4.setText("4");
289         lbStack3.setText("5");
290         lbStack2.setText("");
291         lbStack1.setText("");
292         break;
293     case 4:
294         lbStack5.setText("4");
295         lbStack4.setText("5");
296         lbStack3.setText("");
297         lbStack2.setText("");
298         lbStack1.setText("");
299         break;
300     case 5:
301         lbStack5.setText("5");
302         lbStack4.setText("");
303         lbStack3.setText("");
304         lbStack2.setText("");
305         lbStack1.setText("");
306         break;
307     case 6:
308         lbStack5.setText("");
309         lbStack4.setText("");
310         lbStack3.setText("");
311         lbStack2.setText("");
312         lbStack1.setText("");
313         break;
314     case 7:
315         fm1.close();
316         break;
317     }
318     break;
319 }
320 if(i == 1) {
321     backward = !backward;
322 }
323 if(backward) {
324     i--;
325 } else {
326     i++;
327 }
328 }
329 });
330 btnRun.setBounds(54, 213, 75, 25);
331 btnRun.setText("Run");

```

5. สรุปผลการปฏิบัติการ

ส่วนใหญโปรแกรมจะใช้คำสั่งสวิตช์เพื่อให้ทำตามคำสั่งได้มีการติดปัญหาเรื่องการกดปุ่มแล้วรันที่เดียวหมดเลยไม่สามารถกดให้มันออกมาที่เลขได้

6. คำถามท้ายการทดลอง

6.1. ฟังก์ชันการทำงานใน Stack ควรมีอะไรบ้าง?

1. ตัวชี้แทนหรือ stack pointer

2. ส่วนสมาชิกของสแตค

6.2. การคำนวณ Factorial มีสูตรว่าอย่างไร ?

$$n = n \cdot (n-1)$$

$$9! = 9 \cdot 8 \cdot 7 \cdot 6 \cdot 5 \cdot 4 \cdot 3 \cdot 2 \cdot 1 = 362,880$$

6.3. หลักการสร้าง Recursion คืออะไร?

การเขียนฟังก์ชันมาตัวหนึ่งถ้ายังหาคำตอบไม่ได้ก็ให้เรียกตัวเองซ้ำไปเรื่อยๆ จนกว่าจะเจอคำตอบ

6.4. ข้อควรระวังในการส่งข้อมูลข้ามฟอร์มคืออะไร ?

ต้องเช็คว่าสแตคก่อนว่าสร้างตัวแปรไว้พร้อมทั้ง 2 ฟังก์ชันแล้วก็ควรเช็คการทำงานในบัพ ที่ทำการเคลื่อนย้ายข้อมูลด้วย