

ใบงานการทดลองที่ 6

เรื่อง การเขียนโปรแกรมเชิงวัตถุร่วมกับคลาสทางคณิตศาสตร์

1. จุดประสงค์ทั่วไป

- 1.1. รู้และเข้าใจในการติดต่อกับผู้ใช้ และ การติดต่อระหว่างงาน
- 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

การที่มันเรียกใช้ตัวมันเองในฟังก์ชัน เช่น (ต่อ) }

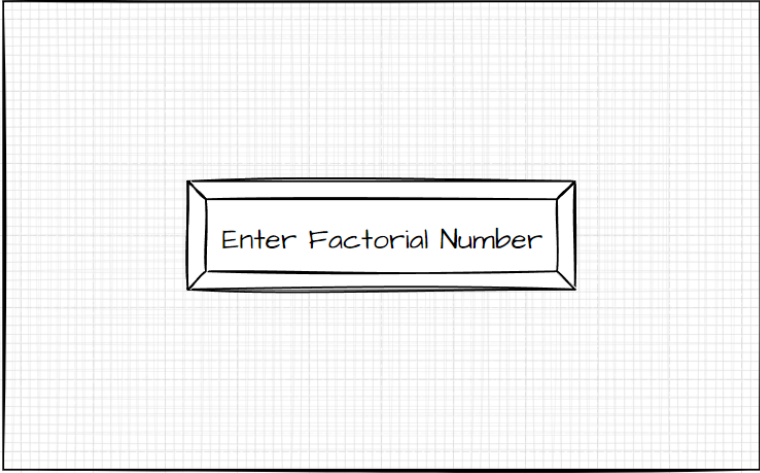
```
int factorial(int N) else{
    int x, y; x = N-1;
    if(N==0) { printf("%2d! = %2d*2d!\n", N, N, y, y*N);
        printf("\tRoll back point\n"); return(N*y);
    }
    return(1); }
```

4. ลำดับขั้นตอนการปฏิบัติการ

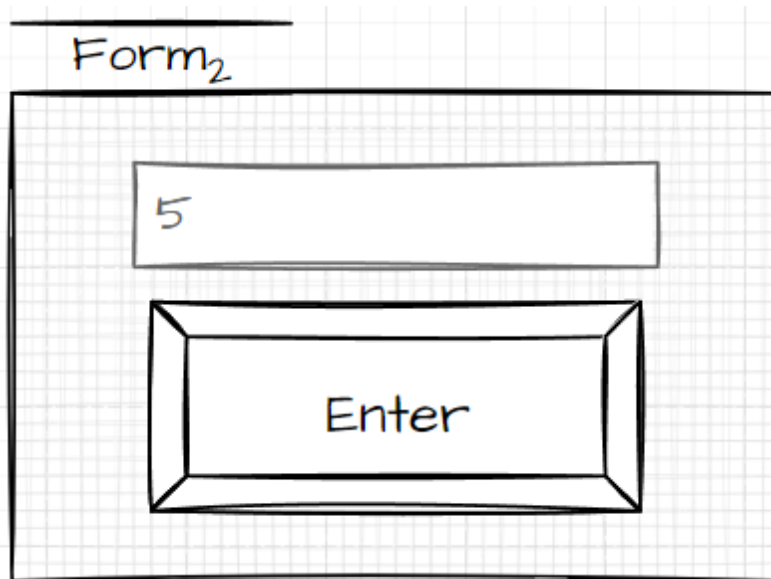
- 4.1. จงสร้าง Window Builder ในโปรแกรม Eclipse เพื่อสร้างโปรแกรมจำลองการทำงานเพื่อหาค่าของ Factorial ผ่านแบบจำลองแบบ Recursion บนโครงสร้างข้อมูลแบบ Stack โดยโปรแกรมจะมีการทำงานอยู่ 2 ฟอร์ม และมีลักษณะการทำงานดังต่อไปนี้

- 4.1.1. ฟอร์ม 1 โดยจะมีปุ่มเพื่อให้ผู้ใช้กด และเรียกหน้าต่าง ฟอร์ม 2 ขึ้นมา

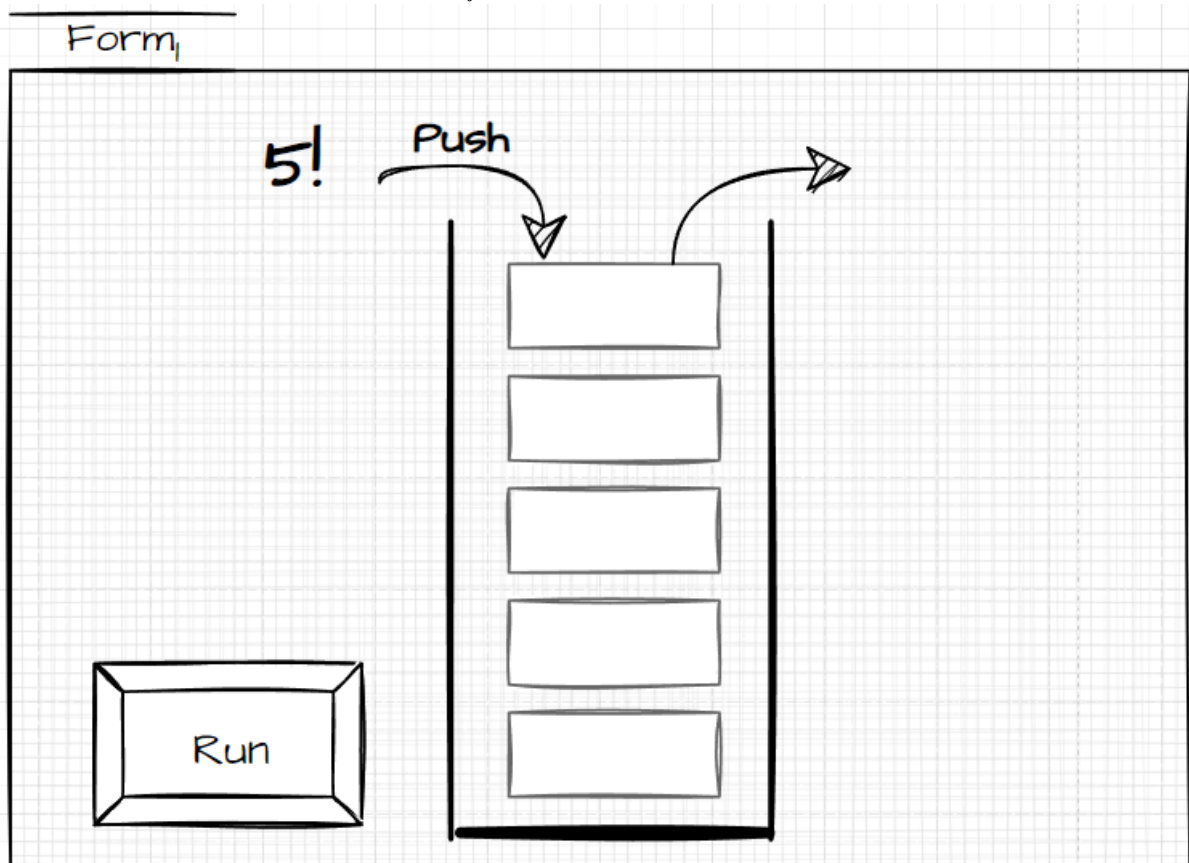
Form₁



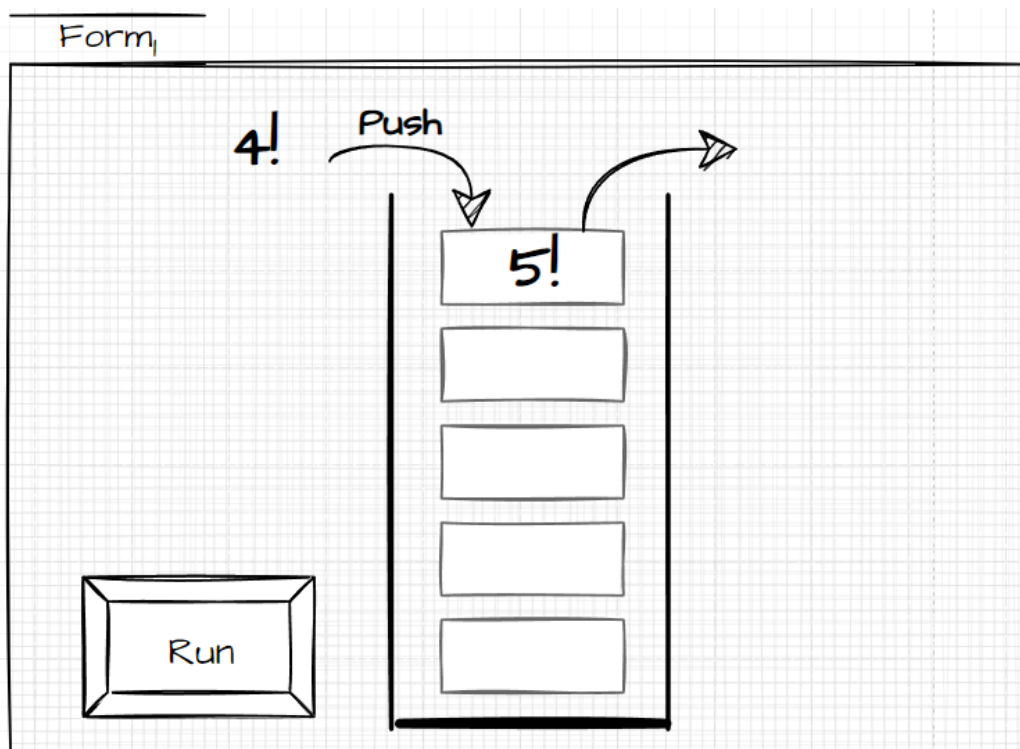
- 4.1.2. ฟอร์ม 2 เป็นหน้าต่างใหม่ที่เตรียมให้ผู้ใช้กรอกเลขที่ต้องการหาค่า Factorial ลงไปในช่อง Textbox โดยที่ผู้ใช้จะถูกจำกัดให้กรอกได้เฉพาะเลข 1 ถึง 5 เท่านั้น



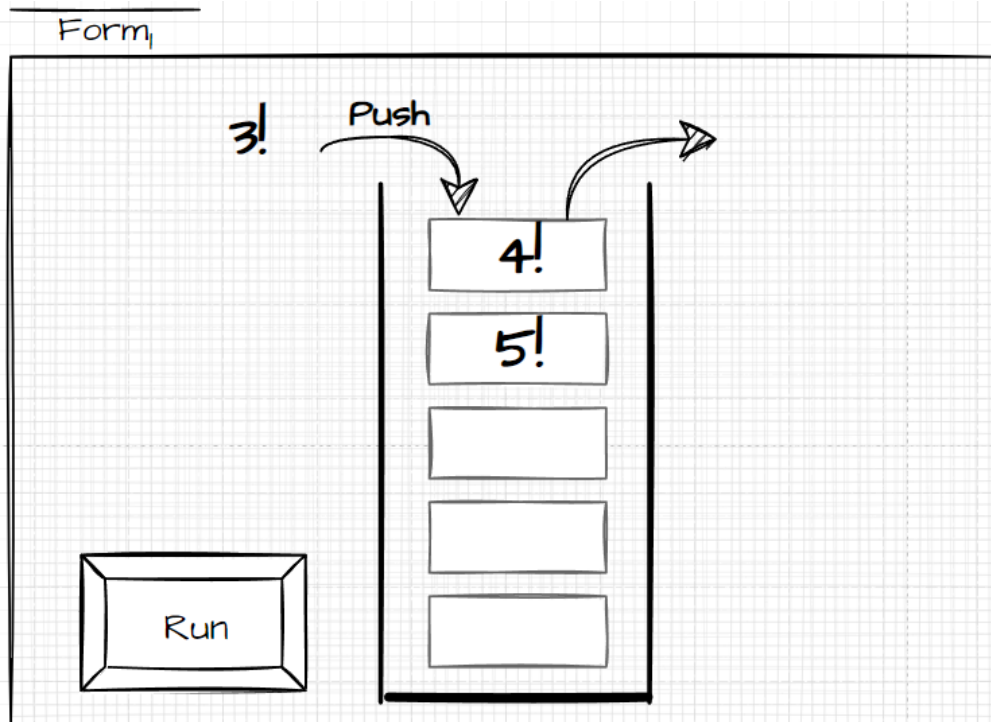
- 4.1.3. เมื่อกรอกข้อมูลในฟอร์ม 2 เสร็จแล้ว และกดปุ่ม Enter โปรแกรมจะนำเลข 5 ที่ได้จากช่อง Textbox ในฟอร์ม 2 ส่งค่ากลับไปยังฟอร์ม 1 อีกครั้ง และแสดงตัวเลขนั้นในช่องก่อนนำข้อมูล Push เข้าไปใน Stack เมื่อกดปุ่ม Run ทางด้านซ้ายล่าง ให้โปรแกรมทำการ Push ข้อมูล 5! เข้าไปใน Stack



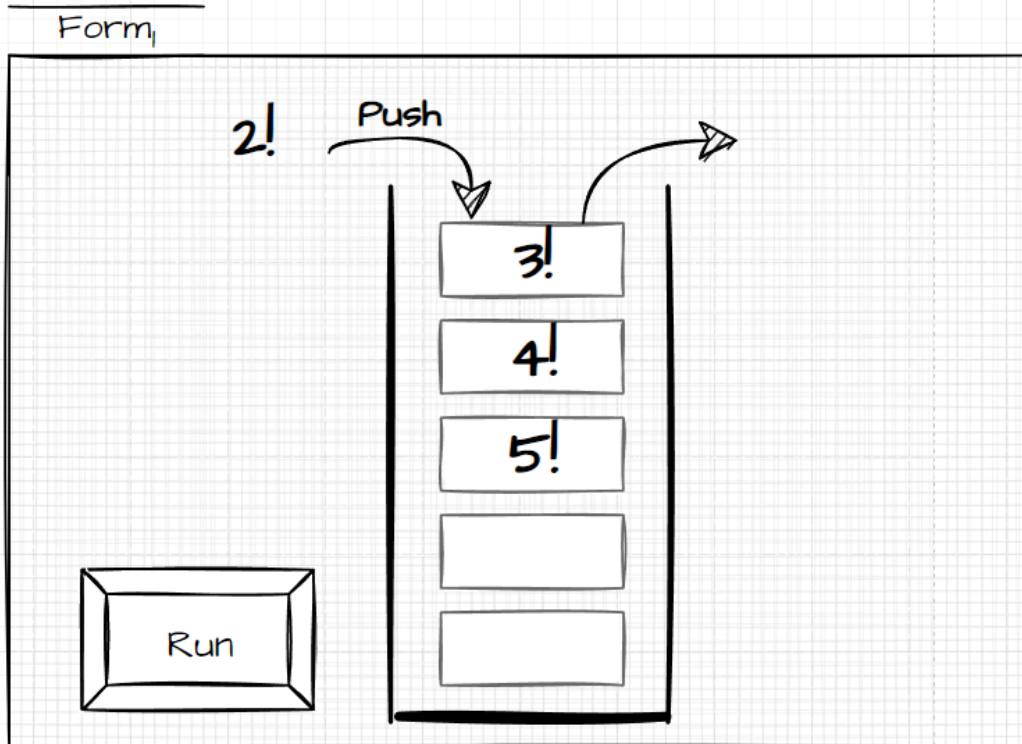
- 4.1.4. หลังจากกดปุ่ม Run เลข 5! จะเข้าไปอยู่ภายใน Stack และจะมีเลข 4! ที่รออยู่ในตำแหน่งรอ Push เข้าไปใน Stack ดังนั้นหากด้านบนสุดของ Stack ยังไม่ใช่เลข 1! เมื่อกดปุ่ม Run ระบบก็จะค่อยๆ นำข้อมูลเข้าไปใน Stack เรื่อยๆ



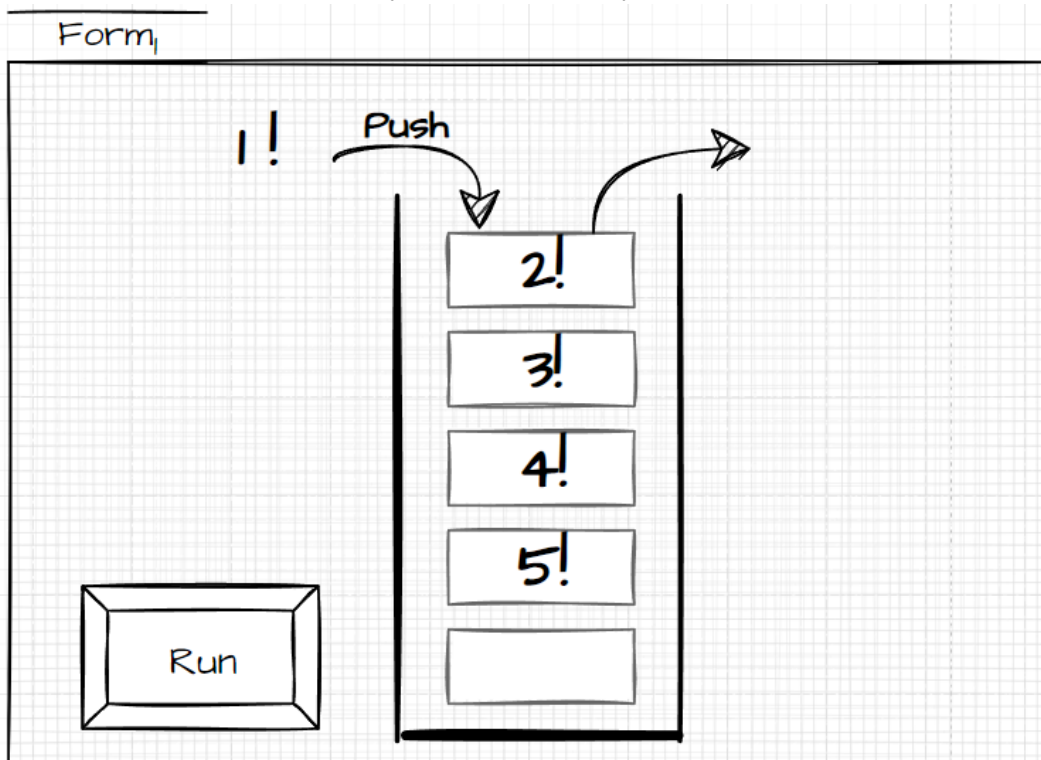
4.1.5. เช่นเดียวกันกับกรณีเมื่อครู่ หลังกดปุ่ม Run เลข 4! ก็จะถูก Push เข้าไปใน Stack ในตำแหน่งด้านบนสุด



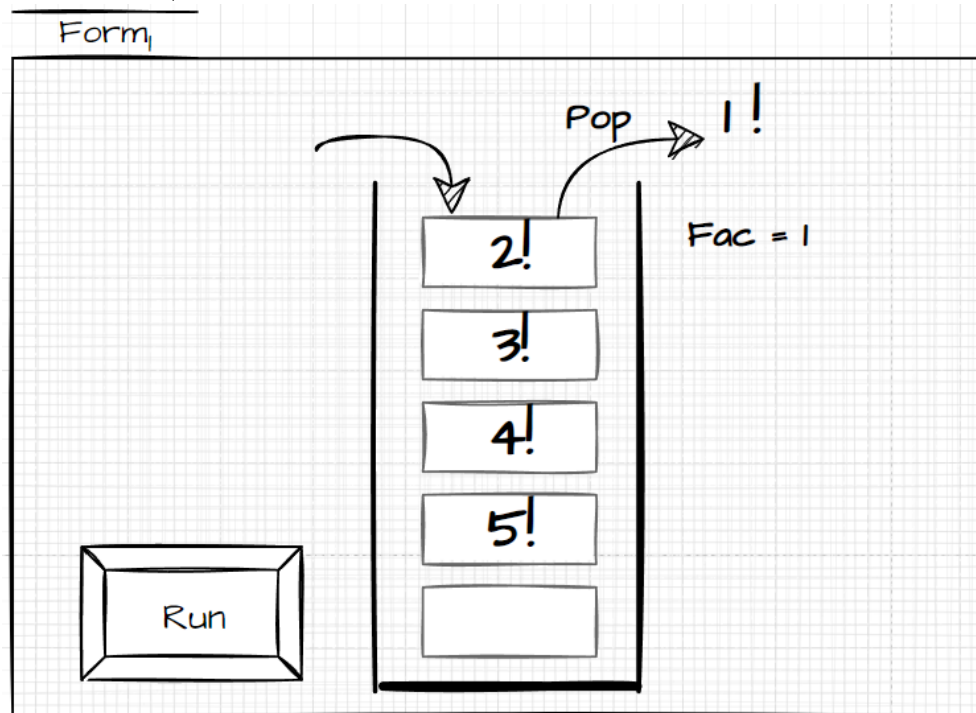
4.1.6. เช่นเดียวกันกับกรณีเมื่อครู! หลังกดปุ่ม Run เลข 3! ก็จะถูก Push เข้าไปใน Stack ในตำแหน่งด้านบนสุด



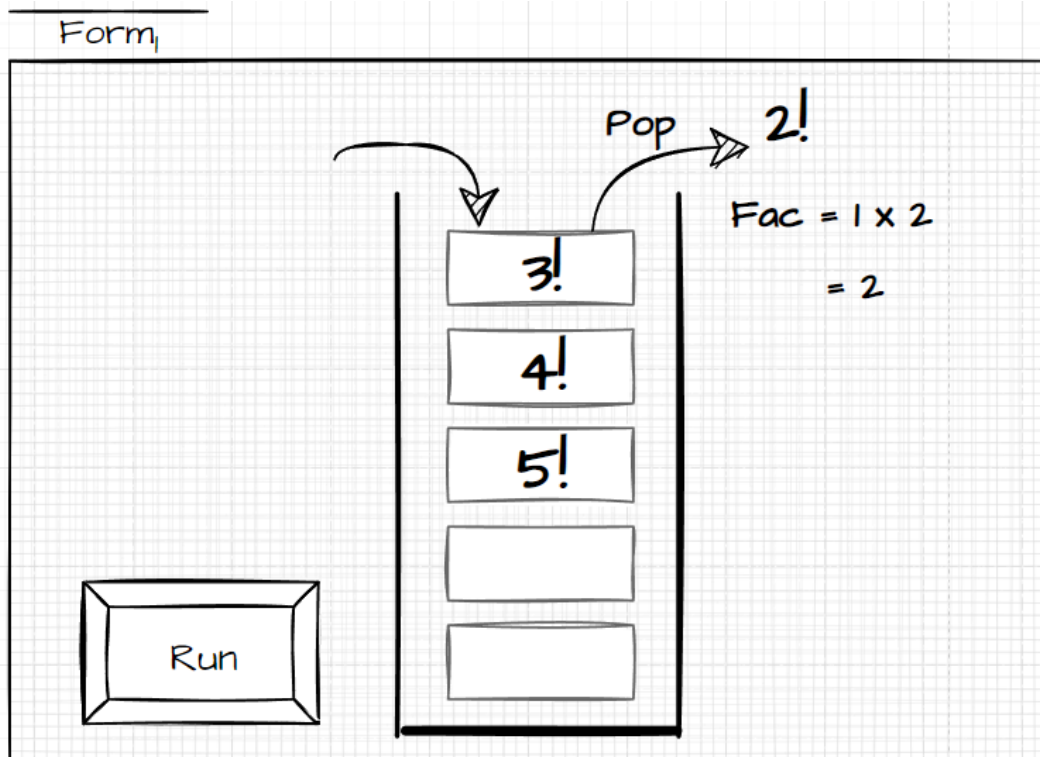
4.1.7. เช่นเดียวกันกับกรณีเมื่อครู! หลังกดปุ่ม Run เลข 2! ก็จะถูก Push เข้าไปใน Stack ในตำแหน่งด้านบนสุด



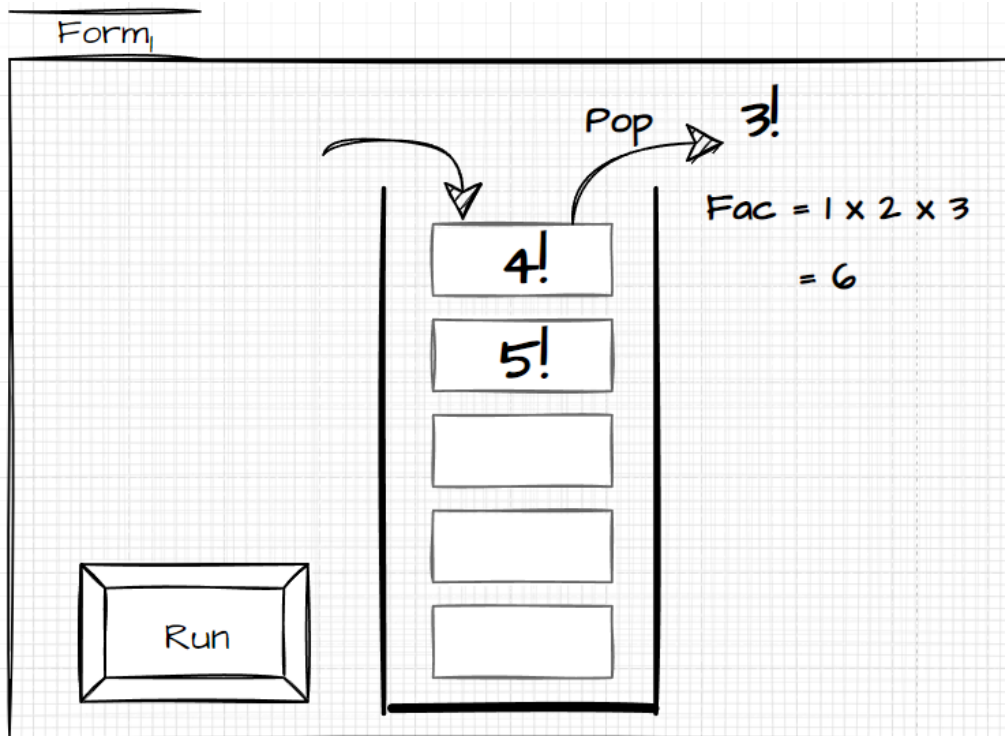
- 4.1.8. ที่นี้หลังจาก Push เลข 1! เข้าไปในระบบ จากกฎที่ว่า $1! = 1$ ดังนั้นทำให้เราสามารถหาคำตอบของ 1! ได้ และเมื่อได้คำตอบให้ทำการ Pop เลข 1! ออกมา และใส่ไว้ในตัวแปร Fac พร้อมทั้งแสดงออกมาผ่านทาง Label เพื่อให้ผู้ใช้เห็นผลการคูณของชุดตัวเลข



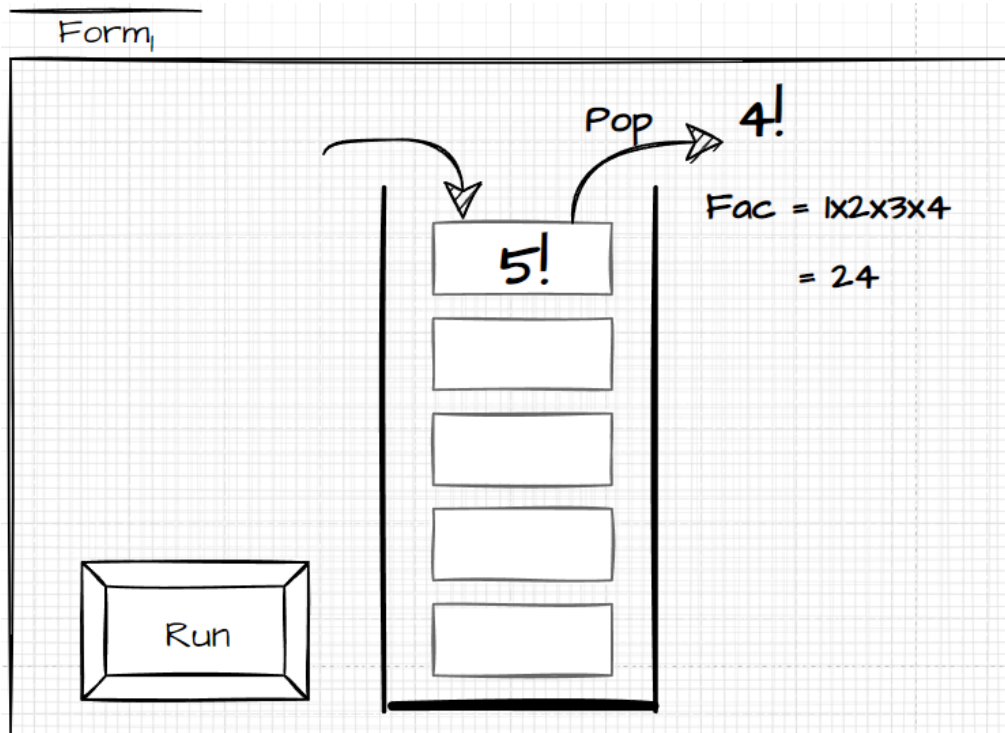
- 4.1.9. เมื่อกดปุ่ม Run อีกรอบ ระบบก็จะ Pop ตัวเลขบนสุดของ Stack ออกมา แล้วนำไปคูณค่า Fac ให้ผู้ใช้เห็นดังรูป



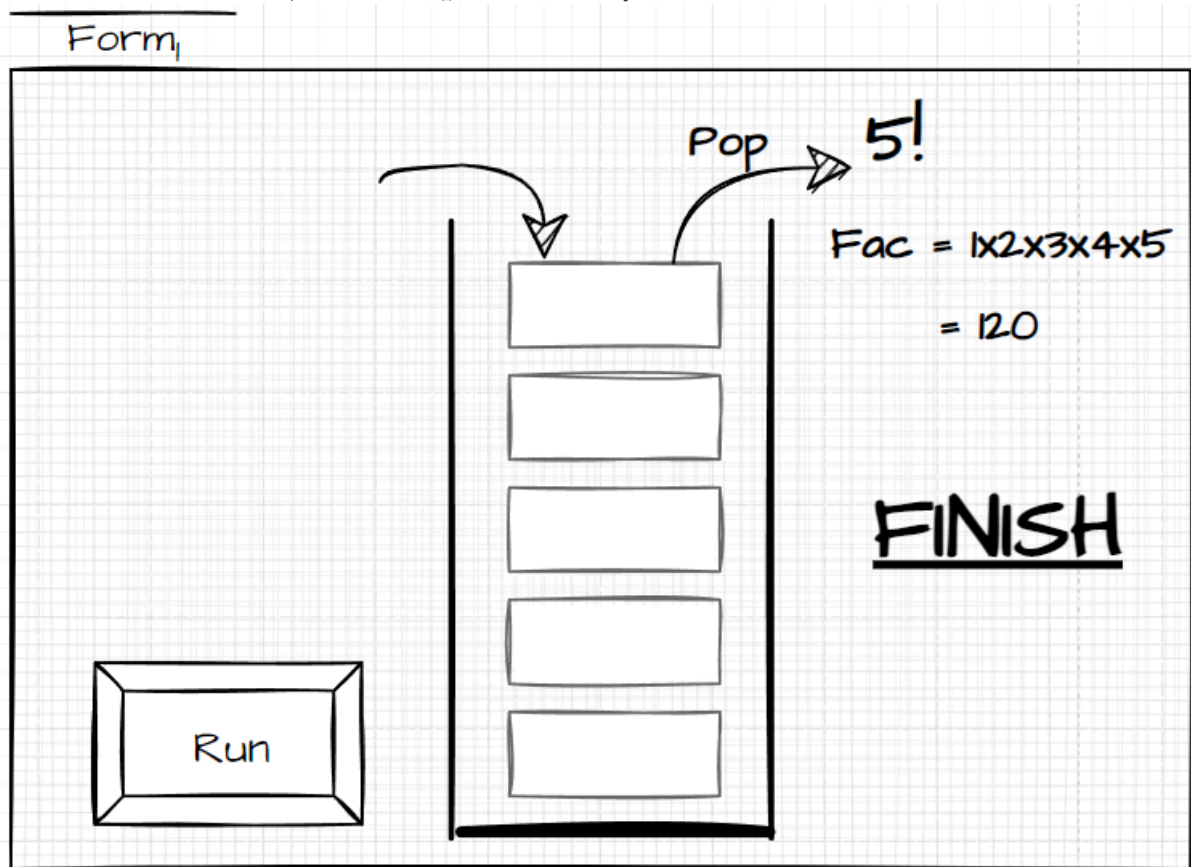
4.1.10. เมื่อกดปุ่ม Run อีกรอบ ระบบก็จะ Pop ตัวเลขบนสุดของ Stack ออกมา แล้วนำไปคูณค่า Fac ให้ผู้ใช้เห็นดังรูป



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

```

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                 switch(i) {
145                     case 1:
146                         lbStack5.setText("1");
147                         lbStack4.setText("");
148                         lbStack3.setText("");
149                         lbStack2.setText("");
150                         lbStack1.setText("");
151                         break;
152                     case 2:
153                         lbStack5.setText("");
154                         lbStack4.setText("");
155                         lbStack3.setText("");
156                         lbStack2.setText("");
157                         lbStack1.setText("");
158                         break;
159                     case 3:
160                         fm1.close();
161                         break;
162                 }
163                 break;
164             case 2:
165                 switch(i) {
166                     case 1:
167                         lbStack5.setText("1");
168                         lbStack4.setText("2");
169                         lbStack3.setText("");
170                         lbStack2.setText("");
171                         lbStack1.setText("");
172                         break;
173                     case 2:
174                         lbStack5.setText("2");
175                         lbStack4.setText("");
176                         lbStack3.setText("");
177                         lbStack2.setText("");
178                         lbStack1.setText("");
179                         break;
180                 }
181             }
182         }
183     }
184 }

```

```

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

```


<pre> 251 252 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 </pre>	<pre> case 4: lbStack5.setText("4"); lbStack4.setText(""); lbStack3.setText(""); lbStack2.setText(""); lbStack1.setText(""); break; case 5: lbStack5.setText(""); lbStack4.setText(""); lbStack3.setText(""); lbStack2.setText(""); lbStack1.setText(""); break; case 6: fm1.close(); break; } break; case 5: switch(i) { case 1: lbStack5.setText("1"); lbStack4.setText("2"); lbStack3.setText("3"); lbStack2.setText("4"); lbStack1.setText("5"); break; case 2: lbStack5.setText("2"); lbStack4.setText("3"); lbStack3.setText("4"); lbStack2.setText("5"); lbStack1.setText(""); break; case 3: </pre>	<pre> 287 288 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325 326 327 328 329 330 331 </pre>	<pre> lbStack5.setText("3"); lbStack4.setText("4"); lbStack3.setText("5"); lbStack2.setText(""); lbStack1.setText(""); break; case 4: lbStack5.setText("4"); lbStack4.setText("5"); lbStack3.setText(""); lbStack2.setText(""); lbStack1.setText(""); break; case 5: lbStack5.setText("5"); lbStack4.setText(""); lbStack3.setText(""); lbStack2.setText(""); lbStack1.setText(""); break; case 6: lbStack5.setText(""); lbStack4.setText(""); lbStack3.setText(""); lbStack2.setText(""); lbStack1.setText(""); break; case 7: fm1.close(); break; } break; } if(i == 1) { backward = !backward; } if(backward) { i--; } else { i++; } }); btnRun.setBounds(54, 213, 75, 25); btnRun.setText("Run"); </pre>
--	---	--	--

5. สรุปผลการปฏิบัติการ

ส่วนใหญ่โปรแกรมจะใช้คำสั่งสวิตช์เพื่อให้ทำตามสิ่งที่ได้ มีการติดปัญหาเรื่องของการที่กดปุ่มแล้วทำการรันทีเดียวนัดเลขไม่สามารถกดมันให้ออกมาทีละเลขได้

6. คำถามท้ายการทดลอง

6.1. ฟังก์ชันการทำงานใน Stack ควรมีอะไรบ้าง?

1. ตัวชี้ stack หรือ stack pointer

2. ส่วนสมาชิกของ stack

6.2. การคำนวณ Factorial มีสูตรอย่างไร ?

$n = n * (n-1)$

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

6.3. หลักการสร้าง Recursion คืออะไร?

การเขียนฟังก์ชันมาตัวหนึ่ง ถ้ายังหาคำตอบไม่ได้ก็ให้เรียกตัวเองซ้ำไปเรื่อยๆ จนกว่าจะเจอคำตอบ

6.4. ข้อควรระวังในการส่งข้อมูลข้ามฟอร์มคืออะไร ?

ต้องเช็คว่าสร้างตัวแปรไว้พร้อมทั้ง 2 ฝ่ายยัง แล้วก็ควรเช็คการทำงานในปุ่มที่ทำการเคลื่อนย้ายข้อมูลด้วย