

# ใบงานการทดลองที่ 10

## เรื่อง การควบคุมเวอร์ชันการทำงานผ่านโปรแกรม Eclipse

### 1. จุดประสงค์ทั่วไป

- 1.1. รู้และเข้าใจการติดต่อกับผู้ใช้งาน และการหลายงานพร้อมกัน
- 1.2. รู้และเข้าใจการติดต่อระหว่างงาน

### 2. เครื่องมือและอุปกรณ์

เครื่องคอมพิวเตอร์ 1 เครื่อง ที่ติดตั้งโปรแกรม Eclipse

### 3. ทฤษฎีการทดลอง

- 3.1. Version Control System (VCS) คืออะไร? มีประโยชน์อย่างไร?

คือ คือระบบซอฟต์แวร์ที่จะคอยบันทึกเวอร์ชันการเปลี่ยนแปลงของโค้ดหรือเอกสารต่างๆ โดยจะทำการบันทึกไว้ด้วยการเปลี่ยนแปลงแต่ละครั้งนั้นทำเพื่ออะไร และทำโดยใครประโยชน์ ช่วยให้สามารถย้อนไฟล์บางไฟล์หรือแม้กระทั่งทั้งโปรเจกต์กลับไปเป็นเวอร์ชันเก่าได้นอกจากนั้นระบบ VCS ยังจะช่วยให้เปรียบเทียบการแก้ไขที่เกิดขึ้นในอดีต

- 3.2. Git ต่างกับ Github อย่างไร?

Git เป็นระบบที่ช่วยจัดการการแก้ไขใน Repository ส่วน Github เป็นบริการจัดเก็บ Repository ออนไลน์พร้อมกันที่เจอร้านขายความสะดวกต่าง ๆ ที่ให้เราไปทำงานร่วมกันคนอื่นได้

- 3.3. Repository คืออะไร? อธิบายพร้อมยกตัวอย่างประกอบ

คือการเก็บสำรองข้อมูลและการเปลี่ยนแปลงของ Source Code ทำให้สามารถย้อนกลับไปทีเวอร์ชันใดๆ ก่อนหน้า และดูรายละเอียดการเปลี่ยนแปลงของแต่ละเวอร์ชันได้ นอกจากนั้นยังสามารถดูได้ว่าใครเป็นคนแก้ไข

- 3.4. Clone คืออะไร? อธิบายพร้อมยกตัวอย่างประกอบ

เวลาที่ผู้อ่านมี Repository อยู่บน Remote ชักแห่งอยู่แล้ว และต้องการ Sync มาลงเครื่องของเรา เราจะต้องทำสิ่งที่เรียกว่า Clone Repository หรือก็คือการก๊อปปี้ Repository จาก Remote คือ เวลาที่ผู้อ่านมี Repository อยู่บน Remote ชักแห่งอยู่แล้ว และต้องการ Sync มาลงเครื่องของเรา เราจะต้องทำสิ่งที่เรียกว่า Clone Repository หรือก็คือการก๊อปปี้ Repository จาก Remote

- 3.5. Commit คืออะไร? อธิบายพร้อมยกตัวอย่างประกอบ

เวลาที่มีข้อมูลแก้ไขเสร็จแล้ว (โค้ดที่เขียนคำสั่งบางอย่างเสร็จแล้ว) แล้วอยากจะทำ Backup เก็บไว้ใน VCS จะเรียกกันว่า Commit

3.6. Staged และ Unstaged คืออะไร? อธิบายพร้อมยกตัวอย่างประกอบ

เวลาเราแก้ไขโค้ดหรือแก้ไขข้อมูล ไฟล์ที่ถูกแก้ไขจะอยู่ในสถานะ Unstaged และเวลาที่เราทำอะไรเสร็จเรียบร้อยแล้ว แล้วอยากจะ Commit เก็บไว้ จะต้องเลือกไฟล์ที่ต้องการเพื่อย้ายเข้าสู่สถานะ Staged ก่อนถึงจะทำการ Commit ได้

3.7. Push คืออะไร? อธิบายพร้อมยกตัวอย่างประกอบ

Push คือการนำโค้ดหรือไฟล์เข้าตัวระบบ Git Repository

3.8. Pull คืออะไร? อธิบายพร้อมยกตัวอย่างประกอบ

คือ เวลา Sync จาก Remote เพื่อดึงข้อมูล Commit ใหม่ ๆ ลงมาเก็บไว้ในเครื่องจะเรียกขั้นตอนนี้ว่า Pull

3.9. Fetch คืออะไร? อธิบายพร้อมยกตัวอย่างประกอบ

อยากเช็คสถานะของ Remote เลยว่าใคร Push ข้อมูลใหม่ขึ้นไปที่ Remote หรือป่าว เราเรียกวิธีนี้ว่า Fetch

3.10. Conflict ใน VSC คืออะไร? อธิบายพร้อมยกตัวอย่างประกอบ

คือ การเกิดปัญหาการชนของข้อมูลในไฟล์งานที่ทำร่วมกันกับเพื่อนเรา ซึ่งในช่วงที่เราพัฒนาโปรแกรมหรือเขียนโค้ดกับเพื่อนร่วมงาน อยู่ที่เราไม่สามารถรู้ได้เลยว่าเพื่อนเราจะเขียนโค้ดไปในรูปแบบไหน

3.11. Merge Commit คืออะไร? อธิบายพร้อมยกตัวอย่างประกอบ

คือการที่มีการแตก branch ออกไป develop แยกกัน โดยที่มีการแก้ไขไฟล์เดียวกันซึ่งโค้ดนั้นอาจมีการทับซ้อน หรือข้อบกพร่องเดียวกัน เมื่อใครคนใดคนหนึ่งนำโค้ดมา Merge รวมกันนั้นจะเกิดสิ่งที่เรียกว่า Conflict คือโค้ดของทั้งสองคนมีความขัดแย้งกัน

3.12. ขั้นตอนที่อยู่ในระหว่าง Development Process ภายใน VSC มีอะไรบ้าง?

3.13. จงบอกและอธิบายขั้นตอนการติดตั้งส่วนขยายใน Eclipse เพื่อให้ใช้งาน Git

1. Install Plugin ทำการ Click ไปที่ Help และ Install new software

2. จากนั้นก็พิมพ์ <http://download.eclipse.org/egit/updates> ลงในช่อง URL แล้วคลิกที่ Egit

#### 4. ลำดับขั้นการปฏิบัติการ

- 4.1. ลงทะเบียน Github และตกแต่ง Profile ของตนเองให้เรียบร้อย
- 4.2. สร้าง Repository ใน Github
- 4.3. ทำการติดตั้งส่วนเสริมของ Git ลงใน Eclipse เพื่อเตรียมใช้งาน Version Control System ของ Github
- 4.4. การสร้างผลงานโค้ดโปรแกรมใน Github
  - 4.4.1. เชื่อมต่อ Eclipse ของคุณเข้ากับ Github
  - 4.4.2. ทำการ Push โค้ดโปรแกรมตั้งแต่การทดลองที่ 1 ถึง 8 ขึ้นสู่ Remote ใน Github ผ่านโปรแกรม Eclipse

##### ลิงค์ Github ที่เก็บไฟล์ข้อมูลของการทดลองที่ 1 ถึง 8 ของคุณ

ลิงค์การทดลองที่ 1 -> <https://github.com/tula05/Lab-01-Eclipse>

ลิงค์การทดลองที่ 2 -> <https://github.com/tula05/lab2>

ลิงค์การทดลองที่ 3 -> [https://github.com/tula05/lab\\_3](https://github.com/tula05/lab_3)

ลิงค์การทดลองที่ 4 -> <https://github.com/tula05/Lab4>

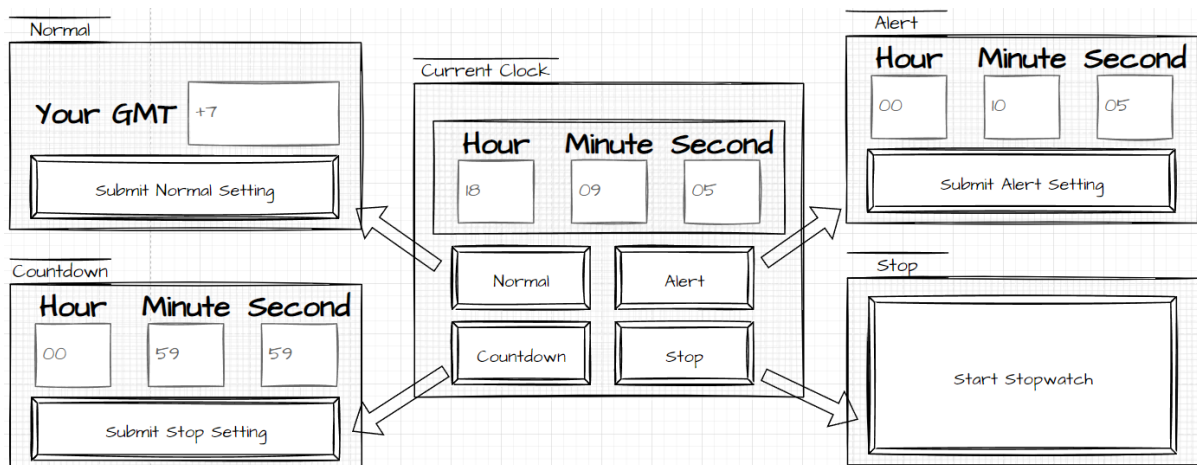
ลิงค์การทดลองที่ 5 -> <https://github.com/tula05/Lab5>

ลิงค์การทดลองที่ 6 -> <https://github.com/tula05/lab6>

ลิงค์การทดลองที่ 7 ->

ลิงค์การทดลองที่ 8 ->

- 4.5. ทำการ Push โค้ดโปรแกรมตั้งแต่การทดลองที่ 1 ถึง 8 ขึ้นสู่ Remote โดยใช้โปรแกรม Eclipse
- 4.6. สร้างโปรเจกใหม่ใน Eclipse ที่เชื่อมต่อกับ Github ให้เรียบร้อย พร้อมทั้งหาสมาชิกในกลุ่มจำนวน 3-4 คน เพื่อสร้างโปรแกรม “นาฬิกาสารพัดประโยชน์” ที่มีส่วนประกอบของฟิจเจอร์ต่างๆ ดังนี้



- 4.6.1. หน้าต่าง Current Clock เพื่อแสดงนาฬิกาที่จะทำงานตามโหมดต่างๆ ที่ผู้ใช้สั่งตามปุ่มต่างๆ
- 4.6.2. หน้าต่าง Normal จะปรากฏหน้าต่างนี้เมื่อคลิกปุ่ม Normal ที่อยู่ในหน้า Current Clock ซึ่งจะ แสดงส่วนการตั้งค่า GMT ให้กับนาฬิกาหลักหลังจากกดปุ่ม Submit Normal Setting เรียบร้อยแล้ว
- 4.6.3. หน้าต่าง Countdown จะปรากฏหน้าต่างนี้เมื่อคลิกปุ่ม Countdown ที่อยู่ในหน้า Current Clock ซึ่งจะ แสดงส่วนการตั้งค่าการนับเวลาถอยหลัง สามารถปรับค่าได้ในระดับชั่วโมง นาที และวินาที หลังจากกดปุ่ม Submit เรียบร้อย หน้าต่างการตั้งค่าจะหายไป และส่วนการแสดงนาฬิกาใน Current Clock ก็จะทำให้การเริ่มต้นนับถอยหลังไปเรื่อยๆ จนถึงเลข 0 นาฬิกา 0 นาที 0 วินาที
- 4.6.4. หน้าต่าง Alert จะปรากฏหน้าต่างนี้เมื่อคลิกปุ่ม Alert ที่อยู่ในหน้า Current Clock ซึ่งจะ แสดงส่วนการตั้งค่าเวลาปลุกเมื่อเวลาปัจจุบันเดินทางมาถึงเวลาที่กำหนดไว้ สามารถปรับค่าได้ในระดับชั่วโมง นาที และวินาที หลังจากกดปุ่ม Submit เรียบร้อย หน้าต่างการตั้งค่าจะหายไป และส่วนการแสดงนาฬิกาใน Current Clock ก็จะแสดงเวลาตามปกติ แต่เมื่อถึงเวลาที่ตั้งปลุกเอาไว้ ระบบก็จะปรากฏหน้าต่างแจ้งเตือน
- 4.6.5. (หากมีสมาชิกในกลุ่มไม่ถึง 4 คน ไม่ต้องทำฟิจเจอร์นี้) หน้าต่าง Stop จะปรากฏหน้าต่างนี้เมื่อคลิกปุ่ม Stop ที่อยู่ในหน้า Current Clock ซึ่งจะ แสดงส่วนการตั้งค่าการจับเวลา หลังจากกดปุ่ม Start Stopwatch เรียบร้อย หน้าต่างการตั้งค่าจะหายไป และส่วนการแสดงนาฬิกาใน Current Clock ก็จะเริ่มต้นจับเวลา โดยเริ่มตั้งแต่ 0 นาฬิกา 0 นาที 0 วินาที และ

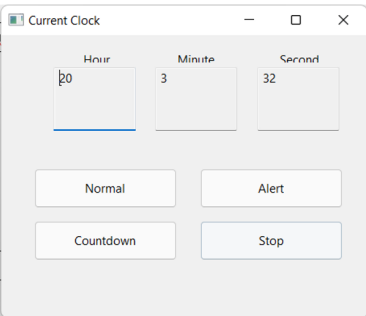
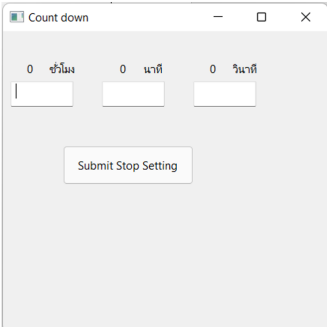
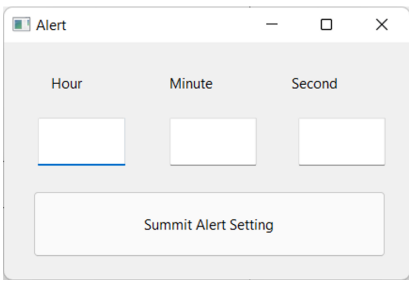
จำนวนวินาทีจะเริ่มต้นเพิ่มขึ้นไปเรื่อยๆ จนกว่าผู้ใช้งานจะกดปุ่ม Stop อีกครั้ง เพื่อเป็นการหยุดการทำงานของนาฬิกาจับเวลา

- 4.7. จากฟีดแบ็กการทำงานของนาฬิกาข้างต้น ให้นักศึกษาแบ่งหน้าที่ในการกับเพื่อนร่วมงานในกลุ่มเพื่อสร้าง Repository และทำงานร่วมกันภายใน Remote นี้
- 4.7.1. ผู้รับผิดชอบทั้งหมด สร้างและพัฒนาส่วนของ Current Clock
- 4.7.2. ผู้รับผิดชอบคนที่ 1 สร้างและพัฒนาส่วนของ Normal
- 4.7.3. ผู้รับผิดชอบคนที่ 2 สร้างและพัฒนาส่วนของ Countdown
- 4.7.4. ผู้รับผิดชอบคนที่ 3 สร้างและพัฒนาส่วนของ Alert
- 4.7.5. ผู้รับผิดชอบคนที่ 4 (ถ้ามี) สร้างและพัฒนาส่วนของ Stop
- 4.8. นักศึกษาจะต้องทำงานร่วมกัน เพื่อให้เห็นภาพรวมการใช้งาน Eclipse ร่วมกับ Github ให้มองเห็นการทำงานเพื่อการแยก Branch, การ Merge Branch, การจัดการโค้ดโปรแกรมเมื่อเกิด Conflict

รายชื่อสมาชิกภายในกลุ่มของคุณ และหน้าที่รับผิดชอบภายในกลุ่ม			
คนที่ 1	ชื่อ-นามสกุล .....	นางสาวศิริพร สีนประเสริฐ.....	รหัสนักศึกษา ..62543502009-8.....
	หน้าที่รับผิดชอบ .....	Current Clock.....	
คนที่ 2	ชื่อ-นามสกุล .....	นางสาวศิริพร สีนประเสริฐ.....	รหัสนักศึกษา ..62543502009-8.....
	หน้าที่รับผิดชอบ .....	normal.....	
คนที่ 3	ชื่อ-นามสกุล .....	นางสาวศิริพร สีนประเสริฐ.....	รหัสนักศึกษา ..62543502009-8.....
	หน้าที่รับผิดชอบ .....	Countdown.....	
คนที่ 4	ชื่อ-นามสกุล .....	นางสาวศิริพร สีนประเสริฐ.....	รหัสนักศึกษา ..62543502009-8.....
(ถ้ามี)	หน้าที่รับผิดชอบ .....	Alert.....	

ลิงค์งานกลุ่มของคุณที่อยู่ใน Github

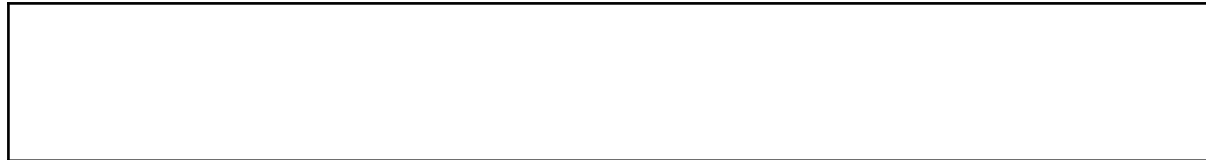
ผลลัพธ์การทำงานของโปรแกรม

#### โค้ดโปรแกรมภายในหน้าต่าง Current Clock

```
1 import java.awt.Button;
15
16 public class Main1 {
17     protected Shell shell;
18
19     SimpleDateFormat timeFormat;
20
21     private String JH;
22     private String JM;
23     private String JS;
24     private Text Hour;
25     private int sec ;
26     private int minute ;
27     private int hour ;
28     public int Gmt = 0;
29     public int r = 0;
30     public int ah = 0;
31     public int am = 0;
32     public int as = 0;
33     Normal Nm = new Normal();
34     Alert Al = new Alert();
35     Alert2 Al2 = new Alert2();
36     Countdown Cd = new Countdown();
37     private Text Min;
38     private Text Sec;
39
40     /**
41      * Launch the application.
42      * @param args
43      */
44     public static void main(String[] args) {
45
46         try {
47
48             Main1 window = new Main1();
```

```
52             window.open();
53
54             } catch (Exception e) {
55                 e.printStackTrace();
56             }
57         }
58
59     /**
60      * Open the window.
61      */
62     public void open() {
63         Display display = Display.getDefault();
64         setTime();
65         createContents();
66         shell.open();
67         shell.layout();
68
69         while (!shell.isDisposed()) {
70             if (!display.readAndDispatch()) {
71                 display.sleep();
72             }
73         }
74     }
75
76     /**
77      * Create contents of the window.
```



```
87  * Create contents of the window.
88  */
89
90  protected void createContents() {
91      shell = new Shell();
92      shell.setSize(473, 327);
93      shell.setText("Current Clock");
94
95
96      //setTime();
97      Composite composite = new Composite(shell, SWT.NONE);
98      composite.setBounds(30, 10, 415, 136);
99
100     Label lblH = new Label(composite, SWT.NONE);
101     lblH.setBounds(72, 10, 59, 14);
102     lblH.setText("Hour");
103
104     Label lblM = new Label(composite, SWT.NONE);
105     lblM.setBounds(189, 10, 59, 14);
106     lblM.setText("Minute");
107
108     Label lblS = new Label(composite, SWT.NONE);
109     lblS.setFont(SWTResourceManager.getFont(".AppleSystemUIFont", 11, SWT.NORMAL));
110     lblS.setBounds(316, 10, 59, 14);
111     lblS.setText("Second");
112
113     Hour = new Text(composite, SWT.BORDER);
114     Hour.setFont(SWTResourceManager.getFont(".AppleSystemUIFont", 30, SWT.NORMAL));
115     Hour.setEditable(false);
116     Hour.setBounds(35, 30, 102, 79);
117
118     Min = new Text(composite, SWT.BORDER);
119     Min.setFont(SWTResourceManager.getFont(".AppleSystemUIFont", 30, SWT.NORMAL));
120     Min.setEditable(false);
121     Min.setBounds(161, 30, 102, 79);
122
123     Sec = new Text(composite, SWT.BORDER);
124     Sec.setFont(SWTResourceManager.getFont(".AppleSystemUIFont", 30, SWT.NORMAL));
125     Sec.setEditable(false);
126     Sec.setBounds(288, 30, 102, 79);
127     //formToolkit.adapt(text, true, true);
128
129     //test Fetch
130
131
132
133     Button btnNewButton = new Button(shell, SWT.NONE);
134     btnNewButton.addSelectionListener(new SelectionAdapter() {
135
136         public void widgetSelected(SelectionEvent e) {
137
138             Nm.open();
139
140             Gmt = Nm.getGMT();
141             if(Gmt >=24) {
142                 Gmt = Gmt - 24;
143             }
144
145         }
146
147     });
148     btnNewButton.setBounds(41, 166, 177, 49);
149     btnNewButton.setText("Normal");
150
151
152     Button btnNewButton_1 = new Button(shell, SWT.NONE);
153     btnNewButton_1.addSelectionListener(new SelectionAdapter() {
154         @Override
155         public void widgetSelected(SelectionEvent e) {
156
157             Al.open();
158             setAlert();
159
160         }
161     });
162     btnNewButton_1.setBounds(247, 166, 177, 49);
163     btnNewButton_1.setText("Alert");
164
165     Button btnNewButton_2 = new Button(shell, SWT.NONE);
166     btnNewButton_2.addSelectionListener(new SelectionAdapter() {
167         @Override
168         public void widgetSelected(SelectionEvent e) {
169             Cd.open();
170
171         }
172     });
173     btnNewButton_2.setBounds(41, 231, 177, 49);
174     btnNewButton_2.setText("Countdown");
175
176     Button btnNewButton_3 = new Button(shell, SWT.NONE);
177     btnNewButton_3.addSelectionListener(new SelectionAdapter() {
178         @Override
179         public void widgetSelected(SelectionEvent e) {
180
181         }
182     });
183     btnNewButton_3.setBounds(247, 231, 177, 49);
184     btnNewButton_3.setText("Stop");
185
186
187
188
189
190 }
```

```
193  public void setTime() {
194
195      new Thread(new Runnable() {
196          public void run() {
197              while (true) {
198                  try { Thread.sleep(1000); } catch (Exception e) { }
199                  Display.getDefault().asyncExec(new Runnable() {
200                      public void run() {
201
202                          Calendar cal = new GregorianCalendar();
203                          minute = cal.get(Calendar.MINUTE);
204                          hour = cal.get(Calendar.HOUR_OF_DAY);
205                          sec = cal.get(Calendar.SECOND);
206
207
208                          hour = hour+Gmt;
209
210
211                          if(hour >=24) {
212                              hour = hour- 24;
213                          }
214
215                          if(hour == ah && minute == am && sec == as) {
216                              Al2.open();
217                          }
218
219                          JH = ""+hour;
220                          JM = ""+minute;
221                          JS = ""+sec;
222                          Hour.setText(JH);
223                          Min.setText(JM);
224                          Sec.setText(JS);
225
226                      }
227                  }
228              }
229          }
230      });
231
232      });
233
234      }
235      }).start();
236
237
238  }
239
240  public void setAlert() {
241
242      ah = Al.h ;
243      am = Al.m ;
244      as = Al.s;
245
246
247  }
248
249  }
250 }
```

# โค้ดโปรแกรมภายในหน้าต่าง Normal

```

10 import org.eclipse.swt.widgets.Display;
13
14 public class Normal {
15
16     protected Shell shell;
17     private Text text;
18     public int GMT = 0;
19     /**
20      * Launch the application.
21      * @param args
22      */
23     public static void main(String[] args) {
24         try {
25             Normal window = new Normal();
26             window.open();
27         } catch (Exception e) {
28             e.printStackTrace();
29         }
30     }
31
32     /**
33      * Open the window.
34      */
35     public void open() {
36         Display display = Display.getDefault();
37         createContents();
38         shell.open();
39         shell.layout();
40         while (!shell.isDisposed()) {
41             if (!display.readAndDispatch()) {
42                 display.sleep();
43             }
44         }
45     }
46
47     /**
48      * Create contents of the window.
49      */

```

```

50     protected void createContents() {
51         shell = new Shell();
52         shell.setSize(450, 300);
53         shell.setText("Normal");
54         Main1 window1 = new Main1();
55
56         Label lblYourGmt = new Label(shell, SWT.NONE);
57         lblYourGmt.setFont(SWTResourceManager.getFont(".AppleSystemUIFont", 44, SWT.NORMAL));
58         lblYourGmt.setBounds(10, 76, 200, 84);
59         lblYourGmt.setText("Your GMT");
60
61         text = new Text(shell, SWT.BORDER);
62         text.setBounds(206, 68, 220, 77);
63
64         Button btnNewButton = new Button(shell, SWT.NONE);
65         btnNewButton.addSelectionListener(new SelectionAdapter() {
66             @Override
67             public void widgetSelected(SelectionEvent e) {
68
69                 GMT = Integer.parseInt(text.getText());
70                 window1.r = GMT;
71                 shell.close();
72             }
73         });
74         btnNewButton.setBounds(20, 161, 406, 77);
75         btnNewButton.setText("Summit Normal Setting");
76     }
77
78     public int getGMT() {
79         return this.GMT;
80     }
81 }
82
83
84
85
86
87
88

```

## โค้ดโปรแกรมภายในหน้าต่าง Countdown

```

10 import org.eclipse.swt.widgets.Display;
11 public class Countdown {
12
13     protected Shell shell;
14     private Text text1;
15     private Text text2;
16     private Text text3;
17
18     Timer timer;
19     int sec,min,hour;
20     int usesec1,usesec2,usesec3;
21     String sec1="";
22     String min1="";
23     String hour1="";
24     private Label la3;
25     private Label la2;
26     private Label la1;
27     private Label lblNewLabel;
28     private Label lblNewLabel_1;
29     private Label lblNewLabel_2;
30
31     /**
32      * Launch the application.
33      * @param args
34      */
35     public static void main(String[] args) {
36         try {
37             Countdown window = new Countdown();
38             window.open();
39         } catch (Exception e) {
40             e.printStackTrace();
41         }
42     }
43
44     /**
45      * Open the window.
46      */
47     public void open() {
48         Display display = Display.getDefault();
49
50         createContents();
51         shell.open();
52         shell.layout();
53         while (!shell.isDisposed()) {
54             if (!display.readAndDispatch()) {
55                 display.sleep();
56             }
57         }
58     }
59
60     /**
61      * Create contents of the window.
62      */
63     protected void createContents() {
64         shell = new Shell();
65         shell.setSize(364, 275);
66         shell.setText("Count down");
67
68         text1 = new Text(shell, SWT.BORDER);
69         text1.setBounds(10, 67, 84, 34);
70
71         text2 = new Text(shell, SWT.BORDER);
72         text2.setBounds(132, 67, 84, 34);
73
74         text3 = new Text(shell, SWT.BORDER);
75         text3.setBounds(254, 67, 84, 34);
76
77         Button button1 = new Button(shell, SWT.NONE);
78         button1.addSelectionListener(new SelectionAdapter() {
79             @Override
80             public void widgetSelected(SelectionEvent e) {
81                 hour1 = text1.getText();
82                 la1.setText(hour1);
83                 usesec1 = Integer.parseInt(hour1);
84
85                 min1 = text2.getText();
86                 la2.setText(min1);
87                 usesec2 = Integer.parseInt(min1);
88
89                 sec1 = text3.getText();
90                 la3.setText(sec1);
91                 usesec3 = Integer.parseInt(sec1);
92
93                 new Thread(new Runnable() {
94                     @Override
95                     public void run() {
96                         try {
97                             Thread.sleep(1000);
98                         } catch (InterruptedException e) {
99                             e.printStackTrace();
100                         }
101                         usesec1--;
102                         String usesec1Str = Integer.toString(usesec1);
103                         la1.setText(usesec1Str);
104                         usesec2--;
105                         String usesec2Str = Integer.toString(usesec2);
106                         la2.setText(usesec2Str);
107                         usesec3--;
108                         String usesec3Str = Integer.toString(usesec3);
109                         la3.setText(usesec3Str);
110                     }
111                 }, "Countdown Timer")
112                 .start();
113             }
114         });
115         button1.setText("Submit Stop Setting");
116     }
117 }

```

```

124         }).start();
125     }
126
127     button1.setBounds(80, 153, 174, 52);
128     button1.setText("Submit Stop Setting");
129
130     la3 = new Label(shell, SWT.NONE);
131     la3.setFont(SWTResourceManager.getFont("Segoe UI", 12, SWT.NORMAL));
132     la3.setBounds(276, 40, 24, 21);
133     la3.setText("0");
134
135     la2 = new Label(shell, SWT.NONE);
136     la2.setFont(SWTResourceManager.getFont("Segoe UI", 12, SWT.NORMAL));
137     la2.setBounds(156, 40, 24, 21);
138     la2.setText("0");
139
140     la1 = new Label(shell, SWT.NONE);
141     la1.setFont(SWTResourceManager.getFont("Segoe UI", 12, SWT.NORMAL));
142     la1.setText("0");
143     la1.setBounds(32, 40, 24, 21);
144
145     lblNewLabel = new Label(shell, SWT.NONE);
146     lblNewLabel.setFont(SWTResourceManager.getFont("Segoe UI", 12, SWT.NORMAL));
147     lblNewLabel.setBounds(307, 40, 44, 21);
148     lblNewLabel.setText("\u00E2\u00E3\u00E9\u00E2\u00E3\u00E3");
149
150     lblNewLabel_1 = new Label(shell, SWT.NONE);
151     lblNewLabel_1.setFont(SWTResourceManager.getFont("Segoe UI", 12, SWT.NORMAL));
152     lblNewLabel_1.setText("\u00E9\u00E3\u00E2\u00E3\u00E3");
153     lblNewLabel_1.setBounds(188, 40, 39, 21);
154
155     lblNewLabel_2 = new Label(shell, SWT.NONE);
156     lblNewLabel_2.setFont(SWTResourceManager.getFont("Segoe UI", 12, SWT.NORMAL));
157     lblNewLabel_2.setText("\u00E0\u00E3\u00E4\u00E2\u00E2\u00E0\u00E7");
158     lblNewLabel_2.setBounds(62, 40, 44, 21);
159 }
160 }

```



## โค้ดโปรแกรมภายในหน้าต่าง Alert

```
10 import org.eclipse.swt.widgets.Display;
11
12 public class Alert {
13     protected Shell shell;
14     private Text H;
15     private Text M;
16     private Text S;
17     public int h = 0;
18     public int m = 0;
19     public int s = 0;
20
21     /**
22      * Launch the application.
23      * @param args
24      */
25     public static void main(String[] args) {
26         try {
27             Alert window = new Alert();
28             window.open();
29         } catch (Exception e) {
30             e.printStackTrace();
31         }
32     }
33
34     /**
35      * Open the window.
36      */
37     public void open() {
38         Display display = Display.getDefault();
39         createContents();
40         shell.open();
41         shell.layout();
42         while (!shell.isDisposed()) {
43             if (!display.readAndDispatch()) {
44                 display.sleep();
45             }
46         }
47     }
48
49     /**
50      * Create contents of the window.
51      */
52     protected void createContents() {
53         shell = new Shell();
54         shell.setSize(450, 300);
55         shell.setText("Alert");
56         Main1 M1 = new Main1();
57
58         Label lblNewLabel = new Label(shell, SWT.NONE);
59         lblNewLabel.setFont(SWTResourceManager.getFont("AppleSystemUIFont", 30, SWT.NORMAL));
60         lblNewLabel.setBounds(59, 33, 73, 30);
61         lblNewLabel.setText("Hour");
62
63         H = new Text(shell, SWT.BORDER);
64         H.setBounds(36, 80, 93, 51);
65
66         Button btnNewButton = new Button(shell, SWT.NONE);
67         btnNewButton.addSelectionListener(new SelectionAdapter() {
68             @Override
69             public void widgetSelected(SelectionEvent e) {
70
71                 h = Integer.parseInt(H.getText());
72                 m = Integer.parseInt(M.getText());
73                 s = Integer.parseInt(S.getText());
74
75                 M1.ah = h;
76                 M1.am = m;
77                 M1.as = s;
78
79                 shell.close();
80
81             }
82         });
83         btnNewButton.setFont(SWTResourceManager.getFont("AppleSystemUIFont", 21, SWT.NORMAL));
84         btnNewButton.setBounds(31, 158, 375, 70);
85         btnNewButton.setText("Summit Alert Setting");
86
87         M = new Text(shell, SWT.BORDER);
88         M.setBounds(176, 80, 93, 51);
89
90         S = new Text(shell, SWT.BORDER);
91         S.setBounds(313, 80, 93, 51);
92
93         Label lblMinute = new Label(shell, SWT.NONE);
94         lblMinute.setText("Minute");
95         lblMinute.setFont(SWTResourceManager.getFont("AppleSystemUIFont", 30, SWT.NORMAL));
96         lblMinute.setBounds(176, 33, 93, 30);
97
98         Label lblNewLabel_3_1 = new Label(shell, SWT.NONE);
99         lblNewLabel_3_1.setText("Second");
100         lblNewLabel_3_1.setFont(SWTResourceManager.getFont("AppleSystemUIFont", 30, SWT.NORMAL));
101         lblNewLabel_3_1.setBounds(306, 33, 110, 30);
102
103     }
104
105 }
```

5. สรุปผลการปฏิบัติการ

---

---

---

---

---

---

## 6. คำถามท้ายการทดลอง

6.1. ควร Commit อย่างไร เพื่อหลีกเลี่ยงการเกิด Conflict ให้เหมาะสมที่สุด

---

---

---

---

6.2. ควรมีหลักเกณฑ์ในการ Push ขึ้นไปบน Remote เมื่อใดจึงจะเหมาะสมที่สุด

---

---

---

---

6.3. เมื่อใดจึงควรใช้คำสั่ง Fetch

---

---

---

---

6.4. เราควรแยก Branch เมื่อใด? และควร Merge Branch เมื่อใด?

---

---

---

---