

ใบงานการทดลองที่ 10

เรื่อง การควบคุมเวอร์ชันการทำงานผ่านโปรแกรม Eclipse

1. จุดประสงค์ทั่วไป

- 1.1. รู้และเข้าใจการติดต่อกับผู้ใช้งาน และการทลายงานพร้อมกัน
- 1.2. รู้และเข้าใจการติดต่อระหว่างงาน

2. เครื่องมือและอุปกรณ์

เครื่องคอมพิวเตอร์ 1 เครื่อง ที่ติดตั้งโปรแกรม Eclipse

3. ทฤษฎีการทดลอง

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

คือคือระบบซอฟแวร์ที่จะคอยบันทึกเวอร์ชันการเปลี่ยนแปลงของโค้ดหรือเอกสารต่างๆ โดยจะทำการบันทึกไว้ด้วยการเปลี่ยนแปลงแทล์คัล์ชันท่าเพื่อจะไม่หลงหายได้โดยจะสามารถอ้อนฟีลส์บางไฟลส์หรือแม้กระทั่งไฟล์ทั้งไฟล์ที่เปลี่ยนไปเป็นเวอร์ชันเก่าได้

- 3.2. Git ต่างกับ Github อย่างไร?

Git เป็นระบบที่ช่วยจัดการการแก้ไขใน ๓ Repository ส่วน GitHub เป็นบริการจัดเก็บ Repository ออนไลน์พร้อมกับ API เจ้าของรายความหลากหลายต่าง ๆ ที่ให้เราไปทำงานร่วมกันบนอินเทอร์เน็ต

- 3.3. Repository คืออะไร? อธิบายพร้อมยกตัวอย่างประกอบ

คือการเก็บรวบรวมชื่อมูลและการเปลี่ยนแปลงของ Source Code ทำให้สามารถอ้อนกลับไปที่เวอร์ชันเดิม ค่อนหน้าและดูรายละเอียดการเปลี่ยนแปลงของแต่ละเวอร์ชันได้ นอกเหนือนั้นสามารถดูได้ว่าใครเป็นคนแก้ไข

- 3.4. Clone คืออะไร? อธิบายพร้อมยกตัวอย่างประกอบ

เวลาที่ผู้อำนวยRepository อยู่ใน Remote ซึ่งแห่งอยู่แล้ว และต้องการ Sync มาลงเครื่องของเรา เราจะต้องทำสิ่งที่เรียกว่าClone Repository หรือก็คือการกู้ไป Repository จาก Remote คือเวลาที่ผู้อำนวยRepository อยู่บน Remote ซึ่งแห่งอยู่แล้ว และต้องการ Sync มาลงเครื่องของเรา เราจะต้องทำสิ่งที่เรียกว่า Clone Repository หรือก็คือการ กู้ไป Repository จาก Remote

- 3.5. Commit คืออะไร? อธิบายพร้อมยกตัวอย่างประกอบ

เวลาที่มีข้อมูลที่แก้ไขเสร็จแล้ว (สำคัญที่เปลี่ยนค่าส่วนของบางส่วน) แล้วอยากจะทำการ Backup เก็บไว้ใน VCS จะเรียกว่า Commit

3.6. Staged และ Unstaged คืออะไร? อธิบายพร้อมยกตัวอย่างประกอบ

Staged และ Unstaged เป็นสถานะของไฟล์ที่ถูกติดตามการเปลี่ยนแปลงใน Git repository ระหว่างการทำงานกับ Git ซึ่งเป็นส่วนหนึ่งของ Git staging area หรือ index ซึ่งเป็นชั้นรากกลางที่ใช้ในการเตรียมไฟล์ที่จะถูก commit เข้า repository โดยที่ Git จะบันทึกเพียงสถานะของไฟล์ใน working directory กับ staging area เพื่อตรวจสอบว่าไฟล์ได้ถูกเปลี่ยนแปลงหรือไม่ โดยมีความหมายดังนี้

3.7. Push คืออะไร? อธิบายพร้อมยกตัวอย่างประกอบ

คำว่า "Push" เป็นคำสั่งใน Git ที่ใช้สำหรับการอัปโหลด (upload) การเปลี่ยนแปลงของโค้ดจากเครื่อง local repository ขึ้นไปยัง remote repository บนเว็บไซต์เช่น GitHub, GitLab, Bitbucket เป็นต้น เพื่อให้คนอื่นสามารถเข้าถึงและดาวน์โหลด (download) ได้ต่อไปนี้ได้

3.8. Pull คืออะไร? อธิบายพร้อมยกตัวอย่างประกอบ

Pull เป็นคำสั่งที่ใช้ในการดึง (fetch) ข้อมูลจาก remote repository มาลง local repository โดยรวมถึงการทำ merge ข้อมูลใน remote repository กับ local repository ด้วย

3.9. Fetch คืออะไร? อธิบายพร้อมยกตัวอย่างประกอบ

คำสั่ง Fetch ใน Git คือการดึงข้อมูล (fetch) จาก Repository บน Server มาลง Local Repository โดยไม่ทำการส่งการเปลี่ยนแปลงที่เกิดขึ้นใน Local Repository กลับไปยัง Repository บน Server ด้วย ซึ่งจะมีประโยชน์ในกรณีที่ต้องการดูส่วน哪 ฉบับของ Repository บน Server หรือต้องการตรวจสอบว่ามีการแก้ไขหรือเพิ่มไฟล์ใน Repository บน Server หรือไม่

3.10. Conflict ใน VSC คืออะไร? อธิบายพร้อมยกตัวอย่างประกอบ

Conflict เกิดขึ้นใน Version Control System (VCS) เมื่อมีการแก้ไขไฟล์เดียวกันในบรรทัดเดียวกันโดยพิនฐานแล้ว Conflict จะเกิดขึ้นเมื่อ 2 ไฟล์ที่มีการแก้ไขไม่เหมือนกัน และไม่สามารถรวมการแก้ไขที่โดยอัตโนมัติจะเข้ากันได้ แต่ต้องแก้ไขด้วยมือ หรือใช้เครื่องมือในการรวมการแก้ไข (merge tool) เพื่อรวมการแก้ไขทั้ง 2 และแก้ไข Conflict ให้ถูกต้องตามวิธีการที่เหมาะสม

3.11. Merge Commit คืออะไร? อธิบายพร้อมยกตัวอย่างประกอบ

Merge commit เป็นการรวมการเปลี่ยนแปลงของ branch ที่ต่างกันไปใน 1 แต่อนสุดท้ายของการ merge โดยเรียก Merge commit เป็นอย่างไรว่า "commit" ในเมืองไทย มักจะเรียกว่า "commit" ไม่ใช่ "merge" แม้พหูพจน์การ merge โดยเฉพาะ ลักษณะนี้ การ merge commit จึงเรียกว่า "merge commit" หรือ "merge commit" สำหรับในงานที่เกี่ยวกับ Version Control System (VCS) เช่น Git

3.12. ขั้นตอนที่อยู่ในระหว่าง Development Process ภายใน VSC มีอะไรบ้าง?

สร้าง branch ใหม่: สร้าง branch ที่ใช้สำหรับพัฒนาไฟล์ เจอร์ชีฟ์ ไบ ญหาแบบ isolated จาก branch หลัก และทำการ switch ไปยัง branch ใหม่ ผ่านกระบวนการ commit หรือการแก้ไขไฟล์ แล้ว commit ให้เสร็จสิ้น ทำการพัฒนาและทดสอบไฟล์ เจอร์ชีฟ์ ไบ ญหาบน branch ใหม่ โดยมักจะใช้วิธีการ commit เพื่อเก็บ snapshot ของการเปลี่ยนแปลงที่ทำใน branch ใหม่โดยอัตโนมัติ

3.13. จบออกและอธิบายขั้นตอนการติดตั้งส่วนขยายใน Eclipse เพื่อให้ใช้งาน Git

ปิด Eclipse และคลิกเมนู Help จากนั้นเลือก Eclipse Marketplace

ค้นหา Git ในช่องค้นหา และคลิกที่ "Install"

เลือก Git Team Provider จากผลลัพธ์ที่แสดงออกมานะ และคลิกที่ "Install"

ตั้งค่าการติดตั้งโดยกด Next และยอมรับข้อกำหนดและเงื่อนไข กดตกลง Next

4. ลำดับขั้นการปฏิบัติการ

- 4.1. ลงทะเบียน Github และตกแต่ง Profile ของตนให้เรียบร้อย
- 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. เข้ามายัง Eclipse ของคุณเข้ากับ Github
 - 4.4.2. ทำการ Push โค้ดโปรแกรมทั้งหมดที่ทำการทดลองที่ 1 ถึง 8 ขึ้นสู่ Remote ใน Github ผ่านโปรแกรม Eclipse

ลิงค์ Github ที่เก็บไฟล์ข้อมูลของการทดลองที่ 1 ถึง 8 ของคุณ

ลิงค์การทดลองที่ 1 -> <https://github.com/Anunchukarn/ENGCE174-Object-Oriented-Programming/blob/main/Lab-01-Eclipse.pdf>

ลิงค์การทดลองที่ 2 -> <https://github.com/Anunchukarn/ENGCE174-Object-Oriented-Programming/blob/main/Lab-02-Java-and-OOP.pdf>

ลิงค์การทดลองที่ 3 -> https://github.com/Anunchukarn/ENGCE174-Object-Oriented-Programming/blob/main/Lab-03-Array-String-Function_d.pdf

ลิงค์การทดลองที่ 4 -> <https://github.com/Anunchukarn/ENGCE174-Object-Oriented-Programming/blob/main/Lab-04-Instant-Object-Inherit-Encap.pdf>

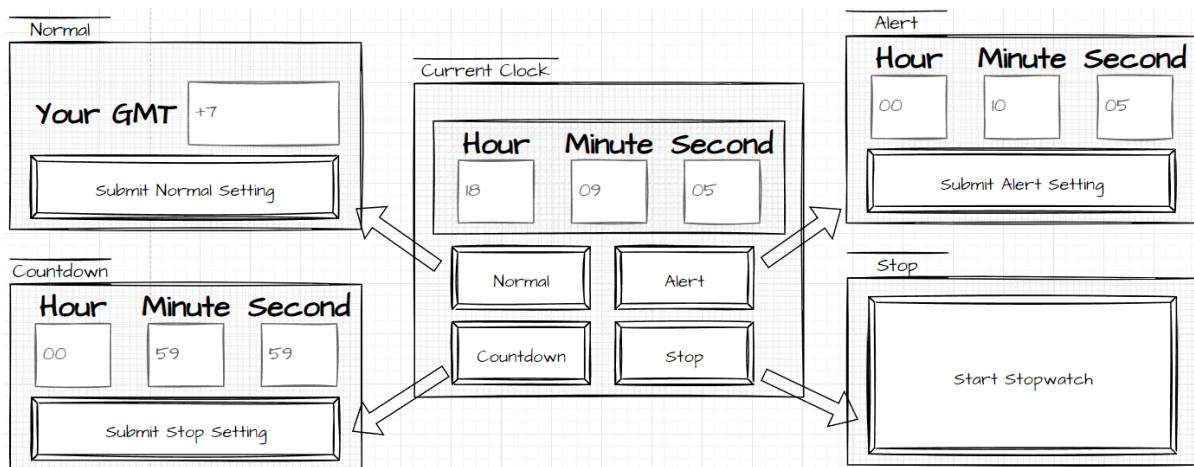
ลิงค์การทดลองที่ 5 -> <https://github.com/Anunchukarn/ENGCE174-Object-Oriented-Programming/blob/main/Lab-05-WindowApp.pdf>

ลิงค์การทดลองที่ 6 -> <https://github.com/Anunchukarn/ENGCE174-Object-Oriented-Programming/blob/main/Lab-06-OOP-and-Math-class.pdf>

ลิงค์การทดลองที่ 7 -> <https://github.com/Anunchukarn/ENGCE174-Object-Oriented-Programming/blob/main/Lab-07-Overloading-and-Overriding.pdf>

ลิงค์การทดลองที่ 8 -> <https://github.com/Anunchukarn/ENGCE174-Object-Oriented-Programming/blob/main/Lab-08-WrapperClass-and-MathClass.pdf>

- 4.5. ทำการ Push โค้ดโปรแกรมทั้งหมดที่ทำการทดลองที่ 1 ถึง 8 ขึ้นสู่ Remote โดยใช้โปรแกรม Eclipse
- 4.6. สร้างโปรเจกใหม่ใน Eclipse ที่เพิ่อมต่อ กับ Github ให้เรียบร้อย พ้อ้มทั้งหมดมาชิกร่วมกันจำนวน 3-4 คน เพื่อสร้าง โปรแกรมหา “นาฬิกาการพัสดุประจำปี” ที่มีส่วนประกอบของไฟล์เอกสารต่างๆ ดังนี้

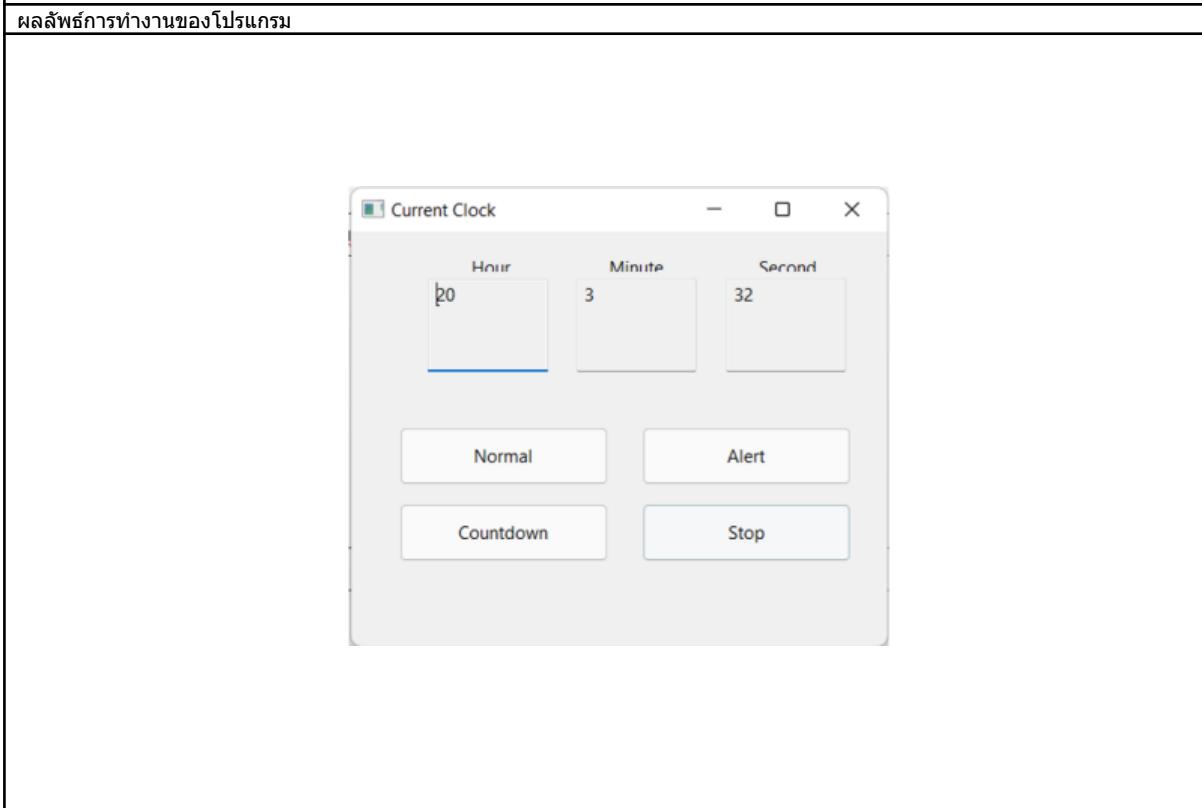


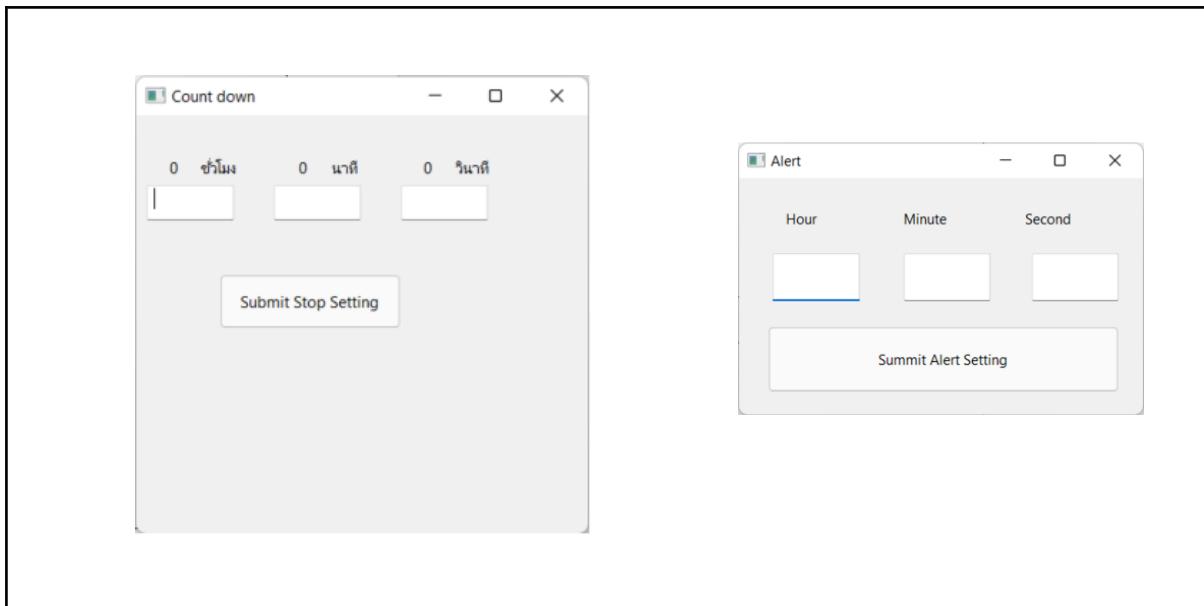
- 4.6.1. หน้าต่าง Current Clock เพื่อแสดงนาฬิกาที่จะทำงานตามโมdem ต่างๆ ที่ผู้ใช้สั่งตามปุ่มต่างๆ
- 4.6.2. หน้าต่าง Normal จะปรากฏหน้าต่างนี้เมื่อกลิกปุ่ม Normal ที่อยู่ในหน้า Current Clock ซึ่งจะ แสดงส่วนการตั้งค่า GMT ให้กับนาฬิกาหลักหลังจากกดปุ่ม Submit Normal Setting เรียบร้อยแล้ว
- 4.6.3. หน้าต่าง Countdown จะปรากฏหน้าต่างนี้เมื่อกลิกปุ่ม Countdown ที่อยู่ในหน้า Current Clock ซึ่งจะ แสดงส่วนการตั้งค่าการนับเวลาอยหลัง สามารถปรับค่าได้ในระดับชั่วโมง นาที และวินาที หลังจากกดปุ่ม Submit เรียบร้อย หน้าต่างการตั้งค่าจะหายไป และส่วนการแสดงนาฬิกาใน Current Clock ก็จะแสดงเวลาตามปกติ แต่เมื่อถูกเวลา ที่ตั้งปลุกเอ้าไว้ ระบบจะปรากฏหน้าต่างแจ้งเตือน
- 4.6.4. หน้าต่าง Alert จะปรากฏหน้าต่างนี้เมื่อกลิกปุ่ม Alert ที่อยู่ในหน้า Current Clock ซึ่งจะ แสดงส่วนการตั้งค่าเวลาปลุกเมื่อเวลาปัจจุบันเดินทางมาถึงเวลาที่กำหนดไว้ สามารถปรับค่าได้ในระดับชั่วโมง นาที และวินาที หลังจากกดปุ่ม Submit เรียบร้อย หน้าต่างการตั้งค่าจะหายไป และส่วนการแสดงนาฬิกาใน Current Clock ก็จะแสดงเวลาตามปกติ แต่เมื่อถูกเวลา ที่ตั้งปลุกเอ้าไว้ ระบบจะปรากฏหน้าต่างแจ้งเตือน
- 4.6.5. (หากมีสมาชิกในกลุ่มไม่ถึง 4 คน ไม่ต้องทำ斐เจอร์นี้) หน้าต่าง Stop จะปรากฏหน้าต่างนี้เมื่อกลิกปุ่ม 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 เพื่อเป็นการหยุดการทำงานของนาฬิกา
จับเวลา
- 4.7. จากไฟล์เจอร์การทำงานของนาฬิกาข้างต้น ให้นักศึกษาแบ่งหน้าที่ในการกับเพื่อนร่วมงานในกลุ่มเพื่อสร้าง Repository และทำ
งานร่วมกันภายใน 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 เพื่อให้เที่ยวพร้อมการใช้งาน Eclipse ร่วมกับ Github ให้มองเห็นการทำงานเพื่อการแยก
Branch, การ Merge Branch, การจัดการโอดีตโปรแกรมเมื่อเกิด Conflict

รายชื่อสมาชิกภายในกลุ่มของคุณ และหน้าที่รับผิดชอบภายในกลุ่ม	
คุณที่ 1 ชื่อ-นามสกุล	รหัสนักศึกษา
..... หน้าที่รับผิดชอบ ... Current.Clock
คุณที่ 2 ชื่อ-นามสกุล	รหัสนักศึกษา
..... หน้าที่รับผิดชอบ... normal
คุณที่ 3 ชื่อ-นามสกุล	รหัสนักศึกษา
..... หน้าที่รับผิดชอบ ... Countdown
คุณที่ 4 ชื่อ-นามสกุล	รหัสนักศึกษา
(ถ้ามี) หน้าที่รับผิดชอบ ... Alert

ลิสต์งานกลุ่มของคุณที่อยู่ใน Github
.....





โค้ดโปรแกรมภายในหน้าต่าง Current Clock

```

10 import java.awt.Button;]
15
16 public class Main1 {
17
18     protected Shell shell;
19
20     SimpleDateFormat timeFormat;
21
22     private String JH;
23     private String JM;
24     private String JS;
25     private Text Hour;
26     private int sec ;
27     private int minute ;
28     private int hour ;
29     public int Gmt = 0;
30     public int r = 0;
31     public int ah = 0;
32     public int am = 0;
33     public int as = 0;
34     Normal Nm = new Normal();
35     Alert Al = new Alert();
36     Alert2 Al2 = new Alert2();
37     Countdown Cd = new Countdown();
38     private Text Min;
39     private Text Sec;
40
41
42     /**
43      * Launch the application.
44      * @param args
45      */
46     public static void main(String[] args) {
47
48
49         try {
50             Main1 window = new Main1();
51             window.open();
52
53
54         } catch (Exception e) {
55             e.printStackTrace();
56         }
57     }
58
59 }
60
61
62     /**
63      * Open the window.
64      */
65     public void open() {
66         Display display = Display.getDefault();
67         setTime();
68         createContents();
69         shell.open();
70         shell.layout();
71
72         while (!shell.isDisposed()) {
73
74             if (!display.readAndDispatch()) {
75
76                 display.sleep();
77
78             }
79         }
80
81     }
82

```

```

87     /*Create contents of the window
88
89● protected void createContents() {
90     shell = new Shell();
91     shell.setSize(473, 327);
92     shell.setText("Current Clock");
93
94     //setTitle()
95
96     Composite composite = new Composite(shell, SWT.NONE);
97     composite.setLayout(new GridLayout(3, 1));
98     composite.setBounds(30, 10, 415, 160);
99
100    Label lb1 = new Label(composite, SWT.NONE);
101    lb1.setBounds(70, 10, 95, 10);
102    lb1.setText("Current");
103
104    Label lb2 = new Label(composite, SWT.NONE);
105    lb2.setBounds(110, 10, 50, 10);
106    lb2.setText("Hour");
107
108    Label lb3 = new Label(composite, SWT.NONE);
109    lb3.setFont(SWTResourceManager.getFont("AppleSystemUIFont", 11, SWT.NORMAL));
110    lb3.setBounds(110, 10, 50, 10);
111    lb3.setText("Second");
112
113    Hour = new Text(composite, SWT.BORDER);
114    Hour.setFont(SWTResourceManager.getFont("AppleSystemUIFont", 10, SWT.NORMAL));
115    Hour.setEditable(true);
116    Hour.setBounds(110, 10, 100, 70);
117
118    Min = new Text(composite, SWT.BORDER);
119    Min.setFont(SWTResourceManager.getFont("AppleSystemUIFont", 10, SWT.NORMAL));
120    Min.setEditable(true);
121    Min.setBounds(110, 10, 100, 70);
122
123    Sec = new Text(composite, SWT.BORDER);
124    Sec.setFont(SWTResourceManager.getFont("AppleSystemUIFont", 10, SWT.NORMAL));
125    Sec.setEditable(false);
126    Sec.setBounds(230, 10, 100, 70);
127    //for window adapt(text, true, true);
128
129    //test Fetch
130
131
132
133    Button btnNewButton = new Button(shell, SWT.NONE);
134    btnNewButton.addListener(new SelectionAdapter() {
135
136●        public void widgetSelected(SelectionEvent e) {
137            Nm.open();
138
139            Get = Nm.getGMT();
140            if (Get >=24) {
141                Get = Get - 24;
142            }
143
144        }
145
146
147
148    });
149    btnNewButton.setBounds(41, 166, 177, 49);
150    btnNewButton.setText("New");
151
152
153    Button btnNewButton_1 = new Button(shell, SWT.NONE);
154●    btnNewButton_1.addListener(new SelectionAdapter() {
155●        @Override
156            public void widgetSelected(SelectionEvent e) {
157
158                Al.open();
159                setAlert();
160
161            }
162        });
163    btnNewButton_1.setBounds(247, 166, 177, 49);
164    btnNewButton_1.setText("Alert");
165
166
167    Button btnNewButton_2 = new Button(shell, SWT.NONE);
168●    btnNewButton_2.addListener(new SelectionAdapter() {
169●        @Override
170            public void widgetSelected(SelectionEvent e) {
171
172                Cd.open();
173
174            }
175        });
176    btnNewButton_2.setBounds(41, 231, 177, 49);
177    btnNewButton_2.setText("Countdown");
178
179    Button btnNewButton_3 = new Button(shell, SWT.NONE);
180●    btnNewButton_3.addListener(new SelectionAdapter() {
181●        @Override
182            public void widgetSelected(SelectionEvent e) {
183
184        });
185    btnNewButton_3.setBounds(247, 231, 177, 49);
186    btnNewButton_3.setText("Stop");
187
188
189
190    }
191
192●    public void setTime() {
193
194        new Thread(new Runnable() {
195●            public void run() {
196●                while (true) {
197                    try { Thread.sleep(1000); } catch (Exception e) { }
198                    Display.getDefault().asyncExec(new Runnable() {
199                        public void run() {
200
201                            Calendar cal = new GregorianCalendar();
202                            minute = cal.get(Calendar.MINUTE);
203                            hour = cal.get(Calendar.HOUR_OF_DAY);
204                            sec = cal.get(Calendar.SECOND);
205
206
207                            hour = hour+gmt;
208
209
210                            if(hour >=24) {
211                                hour = hour- 24;
212
213                            }
214
215                            }
216
217                            if(hour == ah && minute == am && sec == as) {
218
219                                Al2.open();
220
221                            }
222
223
224                            JH =" "+hour;
225                            JM =" "+minute;
226                            JS =" "+sec;
227                            Hour.setText(JH);
228                            Min.setText(JM);
229                            Sec.setText(JS);
230
231
232                        });
233
234                    });
235                }).start();
236
237
238    }
239
240●    public void setAlert() {
241        ah = Al.h ;
242        am = Al.m ;
243        as = Al.s ;
244
245
246    }
247
248
249    }
250

```

โค้ดโปรแกรมภาษา Java หน้าต่าง Normal

```
10 import org.eclipse.swt.widgets.Display;
11
12 public class Normal {
13
14     protected Shell shell;
15     private Text text;
16     public int GMT = 0;
17
18     /**
19      * Launch the application.
20      * @param args
21      */
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     protected void createContents() {
47         shell = new Shell();
48         shell.setSize(450, 300);
49         shell.setText("Normal");
50         Main1 window1 = new Main1();
51
52         Label lblYourGmt = new Label(shell, SWT.NONE);
53         lblYourGmt.setFont(SWTResourceManager.getFont("AppleSystemUIFont", 44, SWT.NORMAL));
54         lblYourGmt.setBounds(10, 76, 200, 84);
55         lblYourGmt.setText("Your GMT");
56
57         text = new Text(shell, SWT.BORDER);
58         text.setBounds(206, 68, 220, 77);
59
60         Button btnNewButton = new Button(shell, SWT.NONE);
61         btnNewButton.addSelectionListener(new SelectionAdapter() {
62             @Override
63             public void widgetSelected(SelectionEvent e) {
64
65                 GMT = Integer.parseInt(text.getText());
66                 window1.r = GMT;
67                 shell.close();
68             }
69         });
70         btnNewButton.setBounds(20, 161, 406, 77);
71         btnNewButton.setText("Summit Normal Setting");
72
73     }
74
75
76     });
77
78     btnNewButton.setBounds(20, 161, 406, 77);
79     btnNewButton.setText("Summit Normal Setting");
80
81 }
82
83
84     public int getGMT() {
85
86         return this.GMT ;
87     }//end method
88 }
```

โค้ดโปรแกรมภาษาในหน้าต่าง Countdown

```

18 import org.eclipse.swt.widgets.Display;
19
20 public class Countdown {
21     protected Shell shell;
22     private Text text1;
23     private Text text2;
24     private Text text3;
25     private Label la3;
26     private Label la2;
27     private Label la1;
28     private Label lblNewLabel;
29     private Label lblNewLabel_1;
30     private Label lblNewLabel_2;
31
32     /**
33      * Launch the application.
34      */
35     @param args
36     */
37     public static void main(String[] args) {
38         try {
39             Countdown window = new Countdown();
40             window.open();
41         } catch (Exception e) {
42             e.printStackTrace();
43         }
44     }
45     /**
46      * Open the window.
47      */
48     public void open() {
49         Display display = Display.getDefault();
50
51         // Create window
52         shell.open();
53         shell.setLayout();
54         while (!shell.isDisposed()) {
55             if (!display.readAndDispatch())
56                 display.sleep();
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(112, 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                 usese1 = Integer.parseInt(hour1);
84
85                 min1 = text2.getText();
86                 la2.setText(min1);
87                 usese2 = Integer.parseInt(min1);
88
89                 sec1 = text3.getText();
90                 la3.setText(sec1);
91                 usese3 = Integer.parseInt(sec1);
92
93                 new Thread(new Runnable() {
94                     public void run() {
95                         while (true) {
96                             try { Thread.sleep(1000); } catch (Exception e) {}}
97                             Display.getDefault().asyncExec(new Runnable() {
98                                 public void run() {
99                                     if (usese1 > 0) {
100                                         usese3--;
101                                         String usetime3 = Integer.toString(usese3);
102                                         la3.setText(usetime3);
103                                         if (usetime3 == 0) {
104                                             usese2--;
105                                             if (usese2 == 0) {
106                                                 usese2 = 60;
107                                                 usese1--;
108                                                 String usetime1 = Integer.toString(usese1);
109                                                 la1.setText(usetime1);
110
111                                             }
112                                         }
113                                         }
114                                     }
115                                 }
116                             });
117                         ); //if(usese1 == 0) {
118                         // break;
119                         //}
120                     }
121
122                 }
123             }
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("\u0E27\u0E34\u0E19\u0E32\u0E17\u0E35");
149
150     lblNewLabel_1 = new Label(shell, SWT.NONE);
151     lblNewLabel_1.setFont(SWTResourceManager.getFont("Segoe UI", 12, SWT.NORMAL));
152     lblNewLabel_1.setText("\u0E0A\u0E31\u0E48\u0E27\u0E42\u0E21\u0E07");
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("\u0E0A\u0E31\u0E48\u0E27\u0E42\u0E21\u0E07");
158     lblNewLabel_2.setBounds(62, 40, 44, 21);
159
160     }

```

โค้ดโปรแกรมภาษา Java ที่แสดง Alert

```
10 import org.eclipse.swt.widgets.Display;
11 public class Alert {
12
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●    protected void createContents() {
49        shell = new Shell();
50        shell.setSize(450, 300);
51        shell.setText("Alert");
52        Main1 M1 = new Main1();
53
54        Label lblNewLabel = new Label(shell, SWT.NONE);
55        lblNewLabel.setFont(SWTResourceManager.getFont("AppleSystemUIFont", 30, SWT.NORMAL));
56        lblNewLabel.setBounds(50, 33, 73, 30);
57        lblNewLabel.setText("Hour");
58
59        H = new Text(shell, SWT.BORDER);
60        H.setBounds(36, 80, 93, 51);
61
62        Button btnNewButton = new Button(shell, SWT.NONE);
63        btnNewButton.addSelectionListener(new SelectionAdapter() {
64            @Override
65            public void widgetSelected(SelectionEvent e) {
66
67                h = Integer.parseInt(H.getText());
68                m = Integer.parseInt(M.getText());
69                s = Integer.parseInt(S.getText());
70
71                M1.hh = h;
72                M1.mm = m;
73                M1.ss = s;
74
75                shell.close();
76            }
77        });
78        btnNewButton.setFont(SWTResourceManager.getFont("AppleSystemUIFont", 21, SWT.NORMAL));
79        btnNewButton.setBounds(31, 158, 375, 70);
80        btnNewButton.setText("Summit Alert Setting");
81
82        M = new Text(shell, SWT.BORDER);
83        M.setBounds(176, 80, 93, 51);
84
85        S = new Text(shell, SWT.BORDER);
86        S.setBounds(313, 80, 93, 51);
87
88        Label lblMinute = new Label(shell, SWT.NONE);
89        lblMinute.setText("Minute");
90        lblMinute.setFont(SWTResourceManager.getFont("AppleSystemUIFont", 30, SWT.NORMAL));
91        lblMinute.setBounds(176, 33, 93, 30);
92
93        Label lblNewLabel_3_1 = new Label(shell, SWT.NONE);
94        lblNewLabel_3_1.setText("Second");
95        lblNewLabel_3_1.setFont(SWTResourceManager.getFont("AppleSystemUIFont", 30, SWT.NORMAL));
96        lblNewLabel_3_1.setBounds(306, 33, 116, 30);
97
98    }
99
100 }
```

โคดโปรแกรมภายในหน้าต่าง Stop

5. สรุปผลการปฏิบัติการ

6. คำถามท้ายการทดลอง

6.1. ควร Commit อย่างไร เพื่อหลีกเลี่ยงการเกิด Conflict ให้เหมาะสมที่สุด

1. ดึงข้อมูลมาสู่จาก repository ก่อนทำการแก้ไขได้

2. เก็บไว้ได้ด้วยดูบัน branch ของคุณโดยเลือกที่จะเก็บเพียงไฟล์เดียวในแต่ละครั้งเท่านั้น

3. ทำการ commit เป็นบางช่วงของการทำงานเป็นระยะๆ และควรจะ commit เมื่อคุณได้ทำงานนั้นได้ที่สมบูรณ์แล้ว และไม่ควร commit ให้ต่อไปเป็นช่วงๆ หรือที่ซึ่งไม่ได้ทดสอบให้เรียบร้อย

6.2. ควรเมื่อหลักเกณฑ์ในการ Push ขึ้นไปบน Remote เนื่องจากจังหวะสมควร

1. คุณควร push ให้ด้วยก่อนเมื่อคุณได้ทำการ commit ให้ต่อไปสมบูรณ์และไม่มีข้อผิดพลาด 2. คุณควร push ให้ด้วยก่อนเมื่อคุณทำการแก้ไขใน local repository

และต้องการให้ผู้อื่นเห็นการเปลี่ยนแปลงโดยดูของคุณ 3. คุณควร push ให้ด้วยก่อนหลังจากที่คุณได้ทำการ pull ให้ด้วยก่อนใน remote repository

เพื่อหลีกเลี่ยง conflict ในการ push 4. คุณควร push ให้ด้วยก่อนเมื่อคุณต้องการให้ผู้อื่นสามารถทดสอบและตรวจสอบให้ด้วยก่อนการนำไปใช้งาน

5. คุณควร push ให้ด้วยก่อนหลังจากที่คุณได้ทำการทดสอบและตรวจสอบให้ด้วยก่อนให้เรียบร้อยแล้ว

6.3. เมื่อใดจึงควรใช้คำสั่ง Fetch

1. เมื่อคุณต้องการดูว่ามีการเปลี่ยนแปลงอะไรบ้างใน remote repository โดยไม่ต้องทำการอัปเดต local repository ของคุณ เช่น คุณอาจต้องการตรวจ

ดูว่ามีการ push ให้ด้วยก่อนสามารถซึ่งในที่มีของคุณหรือไม่ 2. เมื่อคุณต้องการดูประวัติการเปลี่ยนแปลงของ remote repository โดยไม่ต้องทำการ

อัปเดต local repository ของคุณ ดังนั้นคุณสามารถใช้คำสั่ง fetch เพื่อดูประวัติการ commit และการเปลี่ยนแปลงของให้ด้วย remote repository ได้

เมื่อคุณต้องการดึง branch ใหม่จาก remote repository มาใช้ local repository โดยไม่ต้องทำการอัปเดตให้ด้วยคุณ โดยคุณสามารถใช้คำสั่ง fetch

โดยระบุชื่อบranch ที่ต้องการดึงมาในคำสั่งได้

6.4. เราควรจะแยก Branch เมื่อใด? และควรจะ Merge Branch เมื่อใด?

Branch เมื่อ คุณต้องการพัฒนาฟีเจอร์ใหม่: การพัฒนาฟีเจอร์ใหม่โดยตรงบน branch ใหม่จะช่วยให้คุณสามารถแยกกิจกรรมของการพัฒนาจากฟีเจอร์อื่นๆ ที่อยู่บน branch อื่นๆ ที่มีอยู่ ทำให้ง่ายต่อการจัดการ ให้ด้วยและการรวมให้ด้วย

merge branch ควรจะดำเนินเมื่อ ฟีเจอร์ที่พัฒนาบน branch ใหม่เสร็จสมบูรณ์และดำเนินการทดสอบ: เมื่อฟีเจอร์ที่ต้องการรวมเข้ากับ branch

หลักสิ่งที่สำคัญที่สุดคือการทดสอบและดำเนินการทดสอบแล้ว คุณควรจะ merge branch ใหม่เข้ากับ branch หลักเพื่อให้ฟีเจอร์นั้นสามารถใช้งานได้