# ใบงานการทดลองที่ 10 เรื่อง การควบคุมเวอร์ชันการทำงานผ่านโปรแกรม Eclipse

## จุดประสงค์ทั่วไป

### รู้และเข้าใจการติดต่อกับผู้ใช้งาน และการหลายงานพร้อมกัน

### รู้และเข้าใจการติดต่อระหว่างงาน

## เครื่องมือและอุปกรณ์

เครื่องคอมพิวเตอร์ 1 เครื่อง ที่ติดตั้งโปรแกรม Eclipse

## ทฤษฎีการทดลอง

### Version Control System (VCS) คืออะไร? มีประโยชน์อย่างไร?

ระบบที่จัดเก็บการเปลี่ยนแปลงที่เกิดขึ้นกับไฟล์หนึ่งหรือหลายไฟล์เพื่อที่คุณสามารถเรียกเวอร์ชั่นใดเวอร์ชั่นหนึ่งกลับมาดูเมื่อไรก็ได้ หนังสือเล่มนี้จะยกตัวอย่างจากไฟล์ที่เป็นซอร์สโค้ดของซอฟต์แวร์ แต่ขอให้เข้าใจว่าจริง ๆ แล้วคุณสามารถใช้ version control กับไฟล์ชนิดใดก็ได้

### Git ต่างกับ Github อย่างไร?

Git คือ Version Control System ส่วน Github บริษัทที่พัฒนาเกี่ยวกับ Git

### Repository คืออะไร? อธิบายพร้อมยกตัวอย่างประกอบ

คือ ที่เก็บไฟล์ในลักษณะคลาว แต่ก่อนที่จะนำเข้าไปเก็บได้ต้องได้รับอนุญาตก่อน

### Clone คืออะไร? อธิบายพร้อมยกตัวอย่างประกอบ

คือการก็อป Repository จาก Remote มาลงเครื่องเรา

### Commit คืออะไร? อธิบายพร้อมยกตัวอย่างประกอบ

คือ การเก็บข้อมูลที่ถูกแก้ไขไว้ใน VCS หรือการ Backup

### Staged และ Unstaged คืออะไร? อธิบายพร้อมยกตัวอย่างประกอบ

Unstaged คือไฟล์ที่เราไม่ได้นำขึ้นระบบ ส่วน Staged ก็คือไฟล์ที่เรานำขึ้นไปไว้ในระบบ

### Push คืออะไร? อธิบายพร้อมยกตัวอย่างประกอบ

คือการพลักเอา Commit ไปไว้ใน Remote ในระบบ

### Pull คืออะไร? อธิบายพร้อมยกตัวอย่างประกอบ

คือการ ดึง Commit ที่เราอัพขึ้นระบบลงมาแก้ไข

### Fetch คืออะไร? อธิบายพร้อมยกตัวอย่างประกอบ

คือการเช็คข้อมูลก่อนว่าข้อมูลที่เราจะอัพขึ้นไปมีการแก้ไขหรือไม่

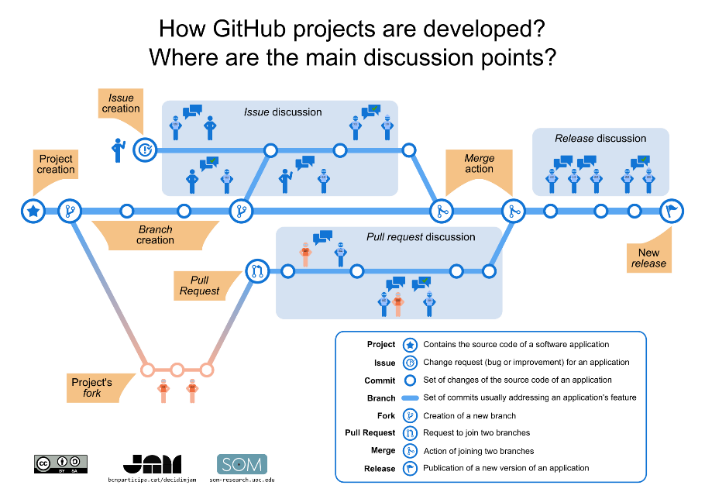
### Conflict ใน VSC คืออะไร? อธิบายพร้อมยกตัวอย่างประกอบ

คือโค้ดที่ทับซ้อนกัน หรือความขัดแย้งของโค้ด

### Merge Commit คืออะไร? อธิบายพร้อมยกตัวอย่างประกอบ

คือการที่เราจะอัพไฟล์ขึ้นไปในระบบแต่มีไฟล์ที่เพื่อนทำไว้อยู่แล้ว เราจึงต้องทำการ Pull ลงมาก่อนเพื่ออัพเดตข้อมูลเก่า

### ขั้นตอนที่อยู่ในระหว่าง Development Process ภายใน VSC มีอะไรบ้าง?



### จงบอกและอธิบายขั้นตอนการติดตั้งส่วนขยายใน Eclipse เพื่อให้ใช้งาน Git

## ลำดับขั้นการปฏิบัติการ

### ลงทะเบียน Github และตกแต่ง Profile ของตนเองให้เรียบร้อย

### สร้าง Repository ใน Github

### ทำการติดตั้งส่วนเสริมของ Git ลงใน Eclipse เพื่อเตรียมใช้งาน Version Control System ของ Github

### การสร้างผลงานโค้ดโปรแกรมใน Github

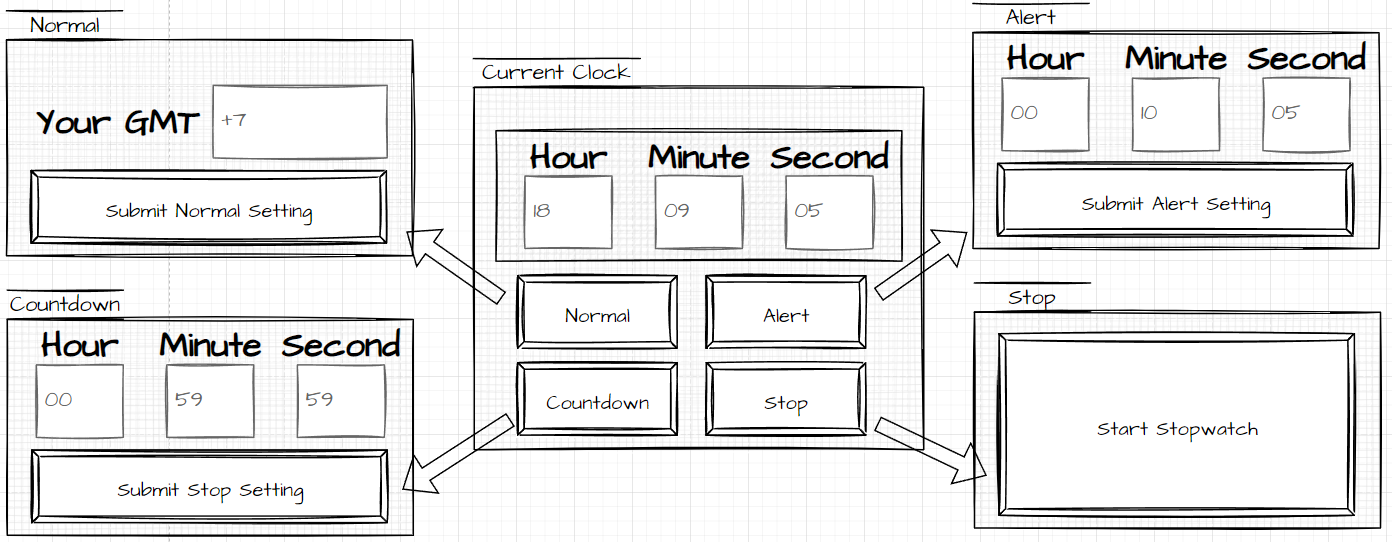
#### เชื่อมต่อ Eclipse ของคุณเข้ากับ Github

#### ทำการ Push โค้ดโปรแกรมตั้งแต่การทดลองที่ 1 ถึง 8 ขึ้นสู่ Remote ใน Github ผ่านโปรแกรม Eclipse

|  |
| --- |
| ลิงค์ Github ที่เก็บไฟล์ข้อมูลของการทดลองที่ 1 ถึง 8 ของคุณ |
| ลิงค์การทดลองที่ 1 ->[OOP\_class/lab1 at master · PawitRMUTL/OOP\_class (github.com)](https://github.com/PawitRMUTL/OOP_class/tree/master/lab1)  ลิงค์การทดลองที่ 2 ->[OOP\_class/lap2 at master · PawitRMUTL/OOP\_class (github.com)](https://github.com/PawitRMUTL/OOP_class/tree/master/lap2)  ลิงค์การทดลองที่ 3 ->[OOP\_class/lap3 at master · PawitRMUTL/OOP\_class (github.com)](https://github.com/PawitRMUTL/OOP_class/tree/master/lap3)  ลิงค์การทดลองที่ 4 ->[OOP\_class/lab4 at master · PawitRMUTL/OOP\_class (github.com)](https://github.com/PawitRMUTL/OOP_class/tree/master/lab4)  ลิงค์การทดลองที่ 5 ->[OOP\_class/lab5 at master · PawitRMUTL/OOP\_class (github.com)](https://github.com/PawitRMUTL/OOP_class/tree/master/lab5)  ลิงค์การทดลองที่ 6 ->[OOP\_class/lab6 at master · PawitRMUTL/OOP\_class (github.com)](https://github.com/PawitRMUTL/OOP_class/tree/master/lab6)  ลิงค์การทดลองที่ 7 ->[OOP\_class/lab7 at master · PawitRMUTL/OOP\_class (github.com)](https://github.com/PawitRMUTL/OOP_class/tree/master/lab7)  ลิงค์การทดลองที่ 8 -> |

### ทำการ Push โค้ดโปรแกรมตั้งแต่การทดลองที่ 1 ถึง 8 ขึ้นสู่ Remote โดยใช้โปรแกรม Eclipse

### สร้างโปรเจคใหม่ใน Eclipse ที่เชื่อมต่อกับ Github ให้เรียบร้อย พร้อมทั้งหาสมาชิกร่วมกลุ่มจำนวน 3-4 คน เพื่อสร้างโปรแกรม “นาฬิกาสารพัดประโยชน์” ที่มีส่วนประกอบของฟีจเจอร์ต่างๆ ดังนี้



#### หน้าต่าง Current Clock เพื่อแสดงนาฬิกาที่จะทำงานตามโหมดต่างๆ ที่ผู้ใช้สั่งตามปุ่มต่างๆ

#### หน้าต่าง Normal จะปรากฏหน้าต่างนี้เมื่อคลิกปุ่ม Normal ที่อยู่ในหน้า Current Clock ซึ่งจะ แสดงส่วนการตั้งค่า GMT ให้กับนาฬิกาหลักหลังจากกดปุ่ม Submit Normal Setting เรียบร้อยแล้ว

#### หน้าต่าง Countdown จะปรากฏหน้าต่างนี้เมื่อคลิกปุ่ม Countdown ที่อยู่ในหน้า Current Clock ซึ่งจะ แสดงส่วนการตั้งค่าการนับเวลาถอยหลัง สามารถปรับค่าได้ในระดับชั่วโมง นาที และวินาที หลังจากกดปุ่ม Submit เรียบร้อย หน้าต่างการตั้งค่าจะหายไป และส่วนการแสดงนาฬิกาใน Current Clock ก็จะทำการเริ่มต้นนับถอยหลังไปเรื่อยๆ จนถึงเลข 0 นาฬิกา 0 นาที 0 วินาที

#### หน้าต่าง Alert จะปรากฏหน้าต่างนี้เมื่อคลิกปุ่ม Alert ที่อยู่ในหน้า Current Clock ซึ่งจะ แสดงส่วนการตั้งค่าเวลาปลุกเมื่อเวลาปัจจุบันเดินทางมาถึงเวลาที่กำหนดไว้ สามารถปรับค่าได้ในระดับชั่วโมง นาที และวินาที หลังจากกดปุ่ม Submit เรียบร้อย หน้าต่างการตั้งค่าจะหายไป และส่วนการแสดงนาฬิกาใน Current Clock ก็จะแสดงเวลาตามปกติ แต่เมื่อถึงเวลาที่ตั้งปลุกเอาไว้ ระบบก็จะปรากฏหน้าต่างแจ้งเตือน

#### (**หากมีสมาชิกในกลุ่มไม่ถึง 4 คน ไม่ต้องทำฟีจเจอร์นี้**) 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 และจำนวนวินาทีจะเริ่มต้นเพิ่มขึ้นไปเรื่อยๆ จนกว่าผู้ใช้งานจะกดปุ่ม Stop อีกครั้ง เพื่อเป็นการหยุดการทำงานของนาฬิกาจับเวลา

### จากฟีจเจอร์การทำงานของนาฬิกาข้างต้น ให้นักศึกษาแบ่งหน้าที่ในการกับเพื่อนร่วมงานในกลุ่มเพื่อสร้าง Repository และทำงานร่วมกันภายใน Remote นี้

#### ผู้รับผิดชอบทั้งหมด สร้างและพัฒนาส่วนของ Current Clock

#### ผู้รับผิดชอบคนที่ 1 สร้างและพัฒนาส่วนของ Normal

#### ผู้รับผิดชอบคนที่ 2 สร้างและพัฒนาส่วนของ Countdown

#### ผู้รับผิดชอบคนที่ 3 สร้างและพัฒนาส่วนของ Alert

#### ผู้รับผิดชอบคนที่ 4 (**ถ้ามี**) สร้างและพัฒนาส่วนของ Stop

### นักศึกษาจะต้องทำงานร่วมกัน เพื่อให้เห็นภาพรวมการใช้งาน Eclipse ร่วมกับ Github ให้มองเห็นการทำงานเพื่อการแยก Branch, การ Merge Branch, การจัดการโค้ดโปรแกรมเมื่อเกิด Conflict

|  |
| --- |
| รายชื่อสมาชิกภายในกลุ่มของคุณ และหน้าที่รับผิดชอบภายในกลุ่ม |
|  |

|  |
| --- |
| ลิงค์งานกลุ่มของคุณที่อยู่ใน Github |
|  |
| ผลลัพธ์การทำงานของโปรแกรม |
|  |

|  |
| --- |
| โค้ดโปรแกรมภายในหน้าต่าง Current Clock |
| **import** org.eclipse.swt.widgets.Display;  **import** org.eclipse.swt.widgets.Shell;  **import** org.eclipse.swt.widgets.Button;  **import** java.time.LocalDateTime;  **import** java.time.format.DateTimeFormatter;  **import** org.eclipse.swt.SWT;  **import** org.eclipse.swt.widgets.Label;  **import** org.eclipse.swt.widgets.Text;  **import** org.eclipse.swt.events.SelectionAdapter;  **import** org.eclipse.swt.events.SelectionEvent;  **import** org.eclipse.wb.swt.SWTResourceManager;  **public** **class** main {  **protected** Shell shell;  **private** Text tbShowHour;  **private** Text tbShowMinute;  **private** Text tbShowSecond;  /\*  \* Launch the application.  \* @param args  \*/  **public** **static** **void** main(String[] args) {  **try** {  main window = **new** main();  window.open();  } **catch** (Exception e) {  e.printStackTrace();  }  }  /\*  \* Open the window.  \*/  **public** **void** open() {  Display display = Display.*getDefault*();  createContents();  shell.open();  shell.layout();  **while** (!shell.isDisposed()) {  **if** (!display.readAndDispatch()) {  display.sleep();  }  }  }  **protected** **void** createContents() {  shell = **new** Shell();  shell.setSize(423, 325);  shell.setText("SWT Application");  Button btnNormal = **new** Button(shell, SWT.***NONE***);  btnNormal.addSelectionListener(**new** SelectionAdapter() {  @Override  **public** **void** widgetSelected(SelectionEvent e) {    }  });  btnNormal.setBounds(61, 152, 136, 44);  btnNormal.setText("Normal");  Button btnCountdown = **new** Button(shell, SWT.***NONE***);  btnCountdown.setBounds(116, 202, 176, 44);  btnCountdown.setText("Countdown");  Button btnAlert = **new** Button(shell, SWT.***NONE***);  btnAlert.addSelectionListener(**new** SelectionAdapter() {  @Override  **public** **void** widgetSelected(SelectionEvent e) {  }  });  btnAlert.setBounds(203, 152, 136, 44);  btnAlert.setText("Alert");  Label lblShowTitleHour = **new** Label(shell, SWT.***NONE***);  lblShowTitleHour.setBounds(61, 42, 55, 15);  lblShowTitleHour.setText("Hour");  Label lblShowTitleMinute = **new** Label(shell, SWT.***NONE***);  lblShowTitleMinute.setBounds(184, 42, 55, 15);  lblShowTitleMinute.setText("Minute");  Label lblShowTitleSecond = **new** Label(shell, SWT.***NONE***);  lblShowTitleSecond.setBounds(300, 42, 55, 15);  lblShowTitleSecond.setText("Second");  tbShowHour = **new** Text(shell, SWT.***BORDER***);  tbShowHour.setBackground(SWTResourceManager.*getColor*(255, 255, 255));  tbShowHour.setBounds(42, 63, 76, 60);  tbShowMinute = **new** Text(shell, SWT.***BORDER***);  tbShowMinute.setBounds(167, 63, 76, 60);  tbShowSecond = **new** Text(shell, SWT.***BORDER***);  tbShowSecond.setBounds(279, 63, 76, 60);  setTime();  }  **public** **void** setTime() {  LocalDateTime localDate = LocalDateTime.*now*();  DateTimeFormatter hh = DateTimeFormatter.*ofPattern*("hh");  tbShowHour.setText(hh.format(localDate));  LocalDateTime localDate\_mm = LocalDateTime.*now*();  DateTimeFormatter mm = DateTimeFormatter.*ofPattern*("mm");  tbShowMinute.setText(mm.format(localDate\_mm));  LocalDateTime localDate\_ss = LocalDateTime.*now*();  DateTimeFormatter ss = DateTimeFormatter.*ofPattern*("ss");  tbShowSecond.setText(ss.format(localDate\_ss));  **try** {  Thread.*sleep*(1000);  } **catch** (InterruptedException e) {  // **TODO** Auto-generated catch block  e.printStackTrace();  }  } |

|  |
| --- |
| โค้ดโปรแกรมภายในหน้าต่าง Normal |
| import org.eclipse.swt.widgets.Display;  import org.eclipse.swt.widgets.Shell;  import org.eclipse.swt.widgets.Text;  import org.eclipse.swt.SWT;  import org.eclipse.swt.widgets.Label;  import org.eclipse.wb.swt.SWTResourceManager;  import org.eclipse.swt.widgets.Button;  import org.eclipse.swt.events.SelectionAdapter;  import org.eclipse.swt.events.SelectionEvent;  public class currenttime {  protected Shell shell;  private Text txtGmt;  public String txt = "";  /\*\*  \* Launch the application.  \* @param args  \*/  public static void main(String[] args) {  try {  currenttime window = new currenttime();  window.open();  } catch (Exception e) {  e.printStackTrace();  }  }  /\*\*  \* Open the window.  \*/  public void open() {  Display display = Display.getDefault();  createContents();  shell.open();  shell.layout();  while (!shell.isDisposed()) {  if (!display.readAndDispatch()) {  display.sleep();  }  }  }  /\*\*  \* Create contents of the window.  \*/  protected void createContents() {  Text gmt ;    shell = new Shell();  shell.setSize(423, 325);  shell.setText("Normal Setting");    txtGmt = new Text(shell, SWT.BORDER);  txtGmt.setText("GMT+01:00");  txtGmt.setBounds(210, 56, 131, 91);    Label lblYourGmt = new Label(shell, SWT.NONE);  lblYourGmt.setFont(SWTResourceManager.getFont("Segoe UI Historic", 21, SWT.NORMAL));  lblYourGmt.setBounds(52, 82, 151, 46);  lblYourGmt.setText("Your GMT ");    Button btnNewButton = new Button(shell, SWT.NONE);  btnNewButton.addSelectionListener(new SelectionAdapter() {  @Override  public void widgetSelected(SelectionEvent e) {  main form1 = new main();  form1.open();  }  });  btnNewButton.setBounds(55, 171, 301, 60);  btnNewButton.setText("Submit Normal Setting");    Label lblExexmpleGmt = new Label(shell, SWT.NONE);  lblExexmpleGmt.setBounds(210, 22, 120, 28);  lblExexmpleGmt.setText("Exemple : GMT+02:30");  }  } |

|  |
| --- |
| โค้ดโปรแกรมภายในหน้าต่าง Countdown |
| **import** org.eclipse.swt.widgets.Display;  **import** org.eclipse.swt.widgets.Shell;  **import** org.eclipse.swt.widgets.Text;  **import** org.eclipse.swt.SWT;  **import** org.eclipse.wb.swt.SWTResourceManager;  **import** org.eclipse.swt.widgets.Button;  **import** org.eclipse.swt.events.SelectionAdapter;  **import** org.eclipse.swt.events.SelectionEvent;  **import** org.eclipse.swt.widgets.Label;  **public** **class** countdowntime {  **protected** Shell shlCountdownTimer;  **private** Text Hr;  **private** Text Mi;  **private** Text Se;  **private** Label Hr\_lb\_1;  **private** Label Hr\_lb\_2;  **private** Label Hr\_lb\_3;  **private** Label Hr\_lb\_4;  **private** Label CD\_s;  **private** Label Hr\_lb\_5;  **private** Label CD\_m;  **private** Label CD\_h;  **private** Label Hr\_lb\_6;  **private** Label Hr\_lb\_7;  **private** Label Hr\_lb\_8;  **private** Label Hr\_lb\_9;  /\*\*  \* Launch the application.  \* **@param** args  \*/  **public** **static** **void** main(String[] args) {  **try** {  countdowntime window = **new** countdowntime();  window.open();  } **catch** (Exception e) {  e.printStackTrace();  }  }  /\*\*  \* Open the window.  \*/  **public** **void** open() {  Display display = Display.*getDefault*();  createContents();  shlCountdownTimer.open();  shlCountdownTimer.layout();  **while** (!shlCountdownTimer.isDisposed()) {  **if** (!display.readAndDispatch()) {  display.sleep();  }  }  }  /\*\*  \* Create contents of the window.  \*/  **protected** **void** createContents() {  shlCountdownTimer = **new** Shell();  shlCountdownTimer.setSize(741, 498);  shlCountdownTimer.setText("countdown timer");    Hr\_lb\_8 = **new** Label(shlCountdownTimer, SWT.***NONE***);  Hr\_lb\_8.setText("Countdown");  Hr\_lb\_8.setForeground(SWTResourceManager.*getColor*(SWT.***COLOR\_LINK\_FOREGROUND***));  Hr\_lb\_8.setFont(SWTResourceManager.*getFont*("TH Baijam", 55, SWT.***BOLD***));  Hr\_lb\_8.setBounds(374, -18, 349, 101);    Hr = **new** Text(shlCountdownTimer, SWT.***BORDER*** | SWT.***CENTER***);  Hr.setText("00");  Hr.setFont(SWTResourceManager.*getFont*("TH Baijam", 30, SWT.***BOLD***));  Hr.setBounds(30, 81, 83, 62);    Mi = **new** Text(shlCountdownTimer, SWT.***BORDER*** | SWT.***CENTER***);  Mi.setText("00");  Mi.setFont(SWTResourceManager.*getFont*("TH Baijam", 30, SWT.***BOLD***));  Mi.setBounds(140, 81, 83, 62);        Se = **new** Text(shlCountdownTimer, SWT.***BORDER*** | SWT.***CENTER***);  Se.setText("00");  Se.setFont(SWTResourceManager.*getFont*("TH Baijam", 30, SWT.***BOLD***));  Se.setBounds(250, 81, 83, 62);    /////////////////////////////////////////////////////////////////////////////////// button    Button Submit = **new** Button(shlCountdownTimer, SWT.***NONE***);  Submit.setFont(SWTResourceManager.*getFont*("TH Baijam", 30, SWT.***BOLD***));  Submit.addSelectionListener(**new** SelectionAdapter() {  @Override  **public** **void** widgetSelected(SelectionEvent e) {    String sec = Se.getText();  String min = Mi.getText();  String hou = Hr.getText();    **int** secnum = Integer.*parseInt*(sec);  **int** minnum = Integer.*parseInt*(min);  **int** hounum = Integer.*parseInt*(hou);      CD\_h.setText(Integer.*toString*(hounum));  CD\_m.setText(Integer.*toString*(minnum));  CD\_s.setText(Integer.*toString*(secnum));      **if**(secnum > 60 || minnum > 60 || hounum > 24) //check time  { System.***out***.println("ERROR");    } **else** {    **do** {    **if**(secnum != 0) {  **for** (**int** i = secnum; i > 0; i--) {  **try** {  Thread.*sleep*(1000);  } **catch** (InterruptedException e1) {  // **TODO** Auto-generated catch block  e1.printStackTrace();  }  System.***out***.print(i + " ");  CD\_s.setText(Integer.*toString*(i));  }  secnum = 0;  }    **for**(**int** j = minnum; j > 0 ; j--) {  **if**(secnum == 0) {  minnum = minnum - 1;  secnum = secnum + 60;  CD\_m.setText(Integer.*toString*(minnum));  CD\_s.setText(Integer.*toString*(secnum));  **for** (**int** i = secnum; i > 0; i--) {  **try** {  Thread.*sleep*(1000);  } **catch** (InterruptedException e1) {  // **TODO** Auto-generated catch block  e1.printStackTrace();  }  System.***out***.print(i + " ");  CD\_s.setText(Integer.*toString*(i));  }  secnum = 0;  }  }    **for**(**int** k = hounum; k > 0 ; k--) {  **if**(minnum == 0 && secnum == 0) {  hounum = hounum - 1;  minnum = minnum + 60;  CD\_h.setText(Integer.*toString*(hounum));  CD\_m.setText(Integer.*toString*(minnum));  CD\_s.setText(Integer.*toString*(secnum));  **for**(**int** j = minnum; j > 0 ; j--) {  **if**(secnum == 0) {  minnum = minnum - 1;  secnum = secnum + 60;  CD\_m.setText(Integer.*toString*(minnum));  CD\_s.setText(Integer.*toString*(secnum));  **for** (**int** i = secnum; i > 0; i--) {  **try** {  Thread.*sleep*(1000);  } **catch** (InterruptedException e1) {  // **TODO** Auto-generated catch block  e1.printStackTrace();  }  System.***out***.print(i + " ");  CD\_s.setText(Integer.*toString*(i));  }  secnum = 0;  }  }  }  }    }**while**(hounum !=0 && minnum != 0 && secnum != 0);    }  }  });  Submit.setBounds(140, 372, 445, 69);  Submit.setText("Submit Stop Setting");    /////////////////////////////////////////////////////////////////////////////////// end button    Label Hr\_lb = **new** Label(shlCountdownTimer, SWT.***CENTER***);  Hr\_lb.setFont(SWTResourceManager.*getFont*("TH Baijam", 20, SWT.***BOLD***));  Hr\_lb.setBounds(30, 41, 83, 34);  Hr\_lb.setText("Hour");    Hr\_lb\_1 = **new** Label(shlCountdownTimer, SWT.***CENTER***);  Hr\_lb\_1.setText("Minute");  Hr\_lb\_1.setFont(SWTResourceManager.*getFont*("TH Baijam", 20, SWT.***BOLD***));  Hr\_lb\_1.setBounds(140, 41, 83, 34);    Hr\_lb\_2 = **new** Label(shlCountdownTimer, SWT.***CENTER***);  Hr\_lb\_2.setText("Second");  Hr\_lb\_2.setFont(SWTResourceManager.*getFont*("TH Baijam", 20, SWT.***BOLD***));  Hr\_lb\_2.setBounds(250, 41, 83, 34);    Hr\_lb\_3 = **new** Label(shlCountdownTimer, SWT.***NONE***);  Hr\_lb\_3.setText(":");  Hr\_lb\_3.setFont(SWTResourceManager.*getFont*("TH Baijam", 40, SWT.***BOLD***));  Hr\_lb\_3.setBounds(119, 70, 15, 59);    Hr\_lb\_4 = **new** Label(shlCountdownTimer, SWT.***NONE***);  Hr\_lb\_4.setText(":");  Hr\_lb\_4.setFont(SWTResourceManager.*getFont*("TH Baijam", 40, SWT.***BOLD***));  Hr\_lb\_4.setBounds(229, 70, 15, 59);    CD\_s = **new** Label(shlCountdownTimer, SWT.***NONE***);  CD\_s.setAlignment(SWT.***CENTER***);  CD\_s.setText("00");  CD\_s.setBackground(SWTResourceManager.*getColor*(SWT.***COLOR\_WIDGET\_BACKGROUND***));  CD\_s.setFont(SWTResourceManager.*getFont*("TH Baijam", 99, SWT.***BOLD***));  CD\_s.setBounds(489, 174, 152, 181);    Hr\_lb\_5 = **new** Label(shlCountdownTimer, SWT.***NONE***);  Hr\_lb\_5.setForeground(SWTResourceManager.*getColor*(SWT.***COLOR\_LINK\_FOREGROUND***));  Hr\_lb\_5.setText("Enter Time");  Hr\_lb\_5.setFont(SWTResourceManager.*getFont*("TH Baijam", 20, SWT.***BOLD***));  Hr\_lb\_5.setBounds(10, 10, 214, 34);    Label label = **new** Label(shlCountdownTimer, SWT.***NONE***);  label.setBackground(SWTResourceManager.*getColor*(SWT.***COLOR\_WIDGET\_FOREGROUND***));  label.setBounds(0, 157, 364, 11);    Label label\_1 = **new** Label(shlCountdownTimer, SWT.***NONE***);  label\_1.setBackground(SWTResourceManager.*getColor*(SWT.***COLOR\_WIDGET\_FOREGROUND***));  label\_1.setBounds(353, 0, 11, 158);    CD\_m = **new** Label(shlCountdownTimer, SWT.***NONE***);  CD\_m.setAlignment(SWT.***CENTER***);  CD\_m.setText("00");  CD\_m.setFont(SWTResourceManager.*getFont*("TH Baijam", 99, SWT.***BOLD***));  CD\_m.setBackground(SWTResourceManager.*getColor*(SWT.***COLOR\_WIDGET\_BACKGROUND***));  CD\_m.setBounds(293, 174, 152, 187);    CD\_h = **new** Label(shlCountdownTimer, SWT.***NONE***);  CD\_h.setAlignment(SWT.***CENTER***);  CD\_h.setText("00");  CD\_h.setFont(SWTResourceManager.*getFont*("TH Baijam", 99, SWT.***BOLD***));  CD\_h.setBackground(SWTResourceManager.*getColor*(SWT.***COLOR\_WIDGET\_BACKGROUND***));  CD\_h.setBounds(94, 174, 150, 193);    Hr\_lb\_6 = **new** Label(shlCountdownTimer, SWT.***NONE***);  Hr\_lb\_6.setText(":");  Hr\_lb\_6.setFont(SWTResourceManager.*getFont*("TH Baijam", 90, SWT.***BOLD***));  Hr\_lb\_6.setBounds(252, 174, 36, 144);    Hr\_lb\_9 = **new** Label(shlCountdownTimer, SWT.***CENTER***);  Hr\_lb\_9.setText("Time");  Hr\_lb\_9.setForeground(SWTResourceManager.*getColor*(SWT.***COLOR\_LINK\_FOREGROUND***));  Hr\_lb\_9.setFont(SWTResourceManager.*getFont*("TH Baijam", 55, SWT.***BOLD***));  Hr\_lb\_9.setBounds(370, 59, 349, 84);    Hr\_lb\_7 = **new** Label(shlCountdownTimer, SWT.***NONE***);  Hr\_lb\_7.setText(":");  Hr\_lb\_7.setFont(SWTResourceManager.*getFont*("TH Baijam", 90, SWT.***BOLD***));  Hr\_lb\_7.setBounds(451, 174, 36, 144);  }  } |

|  |
| --- |
| โค้ดโปรแกรมภายในหน้าต่าง Alert |
| import java.awt.\*;  import java.awt.event.\*;  import java.text.\*;  import java.util.Date;  import javax.swing.\*;  /\*  /\*  To change this template, choose Tools | Templates  and open the template in the editor.  \*/  /\*  AlarmTrigger.java  Created on Mar 28, 2015, 10:40:19 AM  \*/  /\*\*  @author Sujit Reddy  \*/  public class AlarmTrigger extends javax.swing.JFrame  {  int ss, mm, hh;  private boolean verify = false;  /\*\* Creates new form AlarmTrigger \*/  public AlarmTrigger() {  initComponents();  setTitle("....Alarm System....");  getContentPane().setBackground(Color.WHITE);  final DateFormat timeFormat = new SimpleDateFormat("HH:mm:ss");  ActionListener timerListener = new ActionListener()  {  public void actionPerformed(ActionEvent e)  {  if (!verify)  {  jLabel6.setText("Alarm Not Set");  }  if (verify)  {  jLabel6.setText("Alarm Set to " + hh + ":" + mm + ":" + ss);  }  Date date = new Date();  String time = timeFormat.format(date);  jLabel2.setText(time);  int sc = date.getSeconds();  int mn = date.getMinutes();  int hr = date.getHours();  if (sc == ss && mn == mm && hr == hh)  { System.out.print("Matched ");  verify = false;  // Code to Do actions  Component Jframe = null;  JOptionPane.showMessageDialog(Jframe, "Hello World", "Alarm Ringing", JOptionPane.PLAIN\_MESSAGE);  }  }  };  Timer timer = new Timer(1000, timerListener);  // to make sure it doesn't wait one second at the start  timer.setInitialDelay(0);  timer.start();  }  /\*\* This method is called from within the constructor to  initialize the form.  WARNING: Do NOT modify this code. The content of this method is  always regenerated by the Form Editor.  \*/  @SuppressWarnings("unchecked")  // <editor-fold defaultstate="collapsed" desc="Generated Code">//GEN-BEGIN:initComponents  private void initComponents() {  jDialog1 = new javax.swing.JDialog();  jPanel2 = new javax.swing.JPanel();  jLabel1 = new javax.swing.JLabel();  jLabel5 = new javax.swing.JLabel();  jLabel2 = new javax.swing.JLabel();  jPanel3 = new javax.swing.JPanel();  jLabel4 = new javax.swing.JLabel();  jLabel6 = new javax.swing.JLabel();  jPanel4 = new javax.swing.JPanel();  jLabel7 = new javax.swing.JLabel();  jTextField1 = new javax.swing.JTextField();  jTextField2 = new javax.swing.JTextField();  jTextField3 = new javax.swing.JTextField();  jLabel8 = new javax.swing.JLabel();  jLabel9 = new javax.swing.JLabel();  jLabel10 = new javax.swing.JLabel();  jButton1 = new javax.swing.JButton();  javax.swing.GroupLayout jDialog1Layout = new javax.swing.GroupLayout(jDialog1.getContentPane());  jDialog1.getContentPane().setLayout(jDialog1Layout);  jDialog1Layout.setHorizontalGroup(  jDialog1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  .addGap(0, 400, Short.MAX\_VALUE)  );  jDialog1Layout.setVerticalGroup(  jDialog1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  .addGap(0, 300, Short.MAX\_VALUE)  );  jPanel2.setBackground(new java.awt.Color(255, 255, 255));    jLabel1.setFont(new java.awt.Font("Tahoma", 0, 18)); // NOI18N  jLabel1.setText("Time:");    jLabel2.setFont(new java.awt.Font("Tahoma", 1, 36)); // NOI18N  jLabel2.setText("Time");  javax.swing.GroupLayout jPanel2Layout = new javax.swing.GroupLayout(jPanel2);  jPanel2.setLayout(jPanel2Layout);  jPanel2Layout.setHorizontalGroup(  jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  .addGroup(jPanel2Layout.createSequentialGroup()  .addGap(69, 69, 69)  .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED\_SIZE, 148, javax.swing.GroupLayout.PREFERRED\_SIZE)  .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)  .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  .addComponent(jLabel5, javax.swing.GroupLayout.PREFERRED\_SIZE, 150, javax.swing.GroupLayout.PREFERRED\_SIZE)  .addComponent(jLabel2, javax.swing.GroupLayout.PREFERRED\_SIZE, 187, javax.swing.GroupLayout.PREFERRED\_SIZE))  .addContainerGap(50, Short.MAX\_VALUE))  );  jPanel2Layout.setVerticalGroup(  jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel2Layout.createSequentialGroup()  .addGap(20, 20, 20)  .addComponent(jLabel5)  .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)  .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)  .addComponent(jLabel2)  .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED\_SIZE, 36, javax.swing.GroupLayout.PREFERRED\_SIZE))  .addContainerGap())  );  jPanel3.setBackground(new java.awt.Color(255, 255, 255));  jLabel4.setText("Allarm Status:");    jLabel6.setBackground(new java.awt.Color(204, 255, 204));  jLabel6.setForeground(new java.awt.Color(255, 0, 0));  jLabel6.setText("status");  javax.swing.GroupLayout jPanel3Layout = new javax.swing.GroupLayout(jPanel3);  jPanel3.setLayout(jPanel3Layout);  jPanel3Layout.setHorizontalGroup(  jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  .addGroup(jPanel3Layout.createSequentialGroup()  .addContainerGap()  .addComponent(jLabel4, javax.swing.GroupLayout.PREFERRED\_SIZE, 87, javax.swing.GroupLayout.PREFERRED\_SIZE)  .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)  .addComponent(jLabel6, javax.swing.GroupLayout.PREFERRED\_SIZE, 123, javax.swing.GroupLayout.PREFERRED\_SIZE)  .addContainerGap(32, Short.MAX\_VALUE))  );  jPanel3Layout.setVerticalGroup(  jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  .addGroup(jPanel3Layout.createSequentialGroup()  .addContainerGap()  .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)  .addComponent(jLabel4)  .addComponent(jLabel6))  .addContainerGap(javax.swing.GroupLayout.DEFAULT\_SIZE, Short.MAX\_VALUE))  );  jPanel4.setBackground(new java.awt.Color(255, 255, 255));  jLabel7.setText("Set Alaram:");  jLabel8.setText("HH:");  jLabel9.setText("MM:");  jLabel10.setText("SS:");  jButton1.setText("Set Alaram");  jButton1.addActionListener(new java.awt.event.ActionListener() {  public void actionPerformed(java.awt.event.ActionEvent evt) {  jButton1ActionPerformed(evt);  }  });  javax.swing.GroupLayout jPanel4Layout = new javax.swing.GroupLayout(jPanel4);  jPanel4.setLayout(jPanel4Layout);  jPanel4Layout.setHorizontalGroup(  jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  .addGroup(jPanel4Layout.createSequentialGroup()  .addGap(40, 40, 40)  .addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)  .addComponent(jButton1, javax.swing.GroupLayout.PREFERRED\_SIZE, 119, javax.swing.GroupLayout.PREFERRED\_SIZE)  .addGroup(jPanel4Layout.createSequentialGroup()  .addComponent(jLabel7, javax.swing.GroupLayout.PREFERRED\_SIZE, 67, javax.swing.GroupLayout.PREFERRED\_SIZE)  .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)  .addComponent(jLabel8)  .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)  .addComponent(jTextField1, javax.swing.GroupLayout.PREFERRED\_SIZE, 38, javax.swing.GroupLayout.PREFERRED\_SIZE)  .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)  .addComponent(jLabel9)  .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)  .addComponent(jTextField2, javax.swing.GroupLayout.PREFERRED\_SIZE, 41, javax.swing.GroupLayout.PREFERRED\_SIZE)  .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)  .addComponent(jLabel10)))  .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)  .addComponent(jTextField3, javax.swing.GroupLayout.PREFERRED\_SIZE, 39, javax.swing.GroupLayout.PREFERRED\_SIZE)  .addContainerGap(37, Short.MAX\_VALUE))  );  jPanel4Layout.setVerticalGroup(  jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  .addGroup(jPanel4Layout.createSequentialGroup()  .addGap(24, 24, 24)  .addGroup(jPanel4Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)  .addComponent(jLabel7)  .addComponent(jTextField2, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)  .addComponent(jTextField3, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)  .addComponent(jLabel8)  .addComponent(jTextField1, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)  .addComponent(jLabel9)  .addComponent(jLabel10))  .addGap(18, 18, 18)  .addComponent(jButton1, javax.swing.GroupLayout.DEFAULT\_SIZE, 33, Short.MAX\_VALUE)  .addContainerGap())  );  javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());  getContentPane().setLayout(layout);  layout.setHorizontalGroup(  layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  .addGroup(layout.createSequentialGroup()  .addContainerGap(71, Short.MAX\_VALUE)  .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()  //.addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)  .addGap(50, 50, 50))  .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()  .addComponent(jPanel3, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)  .addGap(169, 169, 169))  .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()  .addComponent(jPanel4, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)  .addGap(138, 138, 138))  .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()  .addComponent(jPanel2, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)  .addGap(78, 78, 78))))  );  layout.setVerticalGroup(  layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  .addGroup(layout.createSequentialGroup()  .addGap(18, 18, 18)  .addComponent(jPanel2, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)  .addGap(35, 35, 35)  .addComponent(jPanel3, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)  .addGap(18, 18, 18)  .addComponent(jPanel4, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)  .addContainerGap(61, Short.MAX\_VALUE))  );  pack();  }// </editor-fold>//GEN-END:initComponents  private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event\_jButton1ActionPerformed  // TODO add your handling code here:  hh = Integer.parseInt(jTextField1.getText());  mm = Integer.parseInt(jTextField2.getText());  ss = Integer.parseInt(jTextField3.getText());  verify = true;  jTextField1.setText("");  jTextField2.setText("");  jTextField3.setText("");  }  /\*\*  @param args the command line arguments  \*/  public static void main(String args[]) {  try {  for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {  if ("Nimbus".equals(info.getName())) {  javax.swing.UIManager.setLookAndFeel(info.getClassName());  break;  }  }  } catch (ClassNotFoundException ex) {  java.util.logging.Logger.getLogger(AlarmTrigger.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);  } catch (InstantiationException ex) {  java.util.logging.Logger.getLogger(AlarmTrigger.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);  } catch (IllegalAccessException ex) {  java.util.logging.Logger.getLogger(AlarmTrigger.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);  } catch (javax.swing.UnsupportedLookAndFeelException ex) {  java.util.logging.Logger.getLogger(AlarmTrigger.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);  }  //</editor-fold>  /\* Create and display the form \*/  java.awt.EventQueue.invokeLater(new Runnable() {  public void run() {  new AlarmTrigger().setVisible(true);  }  });  }  // Variables declaration - do not modify//GEN-BEGIN:variables  private javax.swing.JButton jButton1;  private javax.swing.JDialog jDialog1;  private javax.swing.JLabel jLabel1;  private javax.swing.JLabel jLabel10;  private javax.swing.JLabel jLabel2;  private javax.swing.JLabel jLabel4;  private javax.swing.JLabel jLabel5;  private javax.swing.JLabel jLabel6;  private javax.swing.JLabel jLabel7;  private javax.swing.JLabel jLabel8;  private javax.swing.JLabel jLabel9;  private javax.swing.JPanel jPanel2;  private javax.swing.JPanel jPanel3;  private javax.swing.JPanel jPanel4;  private javax.swing.JTextField jTextField1;  private javax.swing.JTextField jTextField2;  private javax.swing.JTextField jTextField3;  // End of variables declaration//GEN-END:variables  } |

|  |
| --- |
| โค้ดโปรแกรมภายในหน้าต่าง Stop |
|  |

## สรุปผลการปฏิบัติการ

การนำไฟล์งานมาเชื่อมต่อกับ Git ป็นเรื่องที่งงมาก ตอนนี้ยังไม่ได้เลย

## คำถามท้ายการทดลอง

### ควร Commit อย่างไร เพื่อหลีกเลี่ยงการเกิด Conflict ให้เหมาะสมที่สุด

คือการคุยกันกับเพื่อนในกลุ่มให้มากที่สุดเพื่อหลีกเลี่ยง Conflict

### ควรมีหลักเกณฑ์ในการ Push ขึ้นไปบน Remote เมื่อใดจึงจะเหมาะสมที่สุด

เมื่อเราทำโค้ดเสร็จแล้วบางส่วนแล้วค่อยทยอยอัพขึ้นเรื่อยๆจนเร็จ

### เมื่อใดจึงควรใช้คำสั่ง Fetch

ใช้เมื่อเราจะอัพไฟล์ขึ้นไปในระบบ

### เราควรจะแยก Branch เมื่อใด? และควรจะ Merge Branch เมื่อใด?

เมื่อเวลาที่เราจะนำงานทั้งหมดมารวมกันเพื่อทำการส่งลูกค้า