



## โครงการ Mini Project

### Signy IDE

จัดทำโดย

นายฐิติโชติ	ใจเมือง	60070019
นายศิวาวัฒน์	ศรีสุข	60070096
นายศุภกิตต์	รอดทอง	60070099

รายงานนี้เป็นส่วนหนึ่งของรายวิชา

06016317 Object-Oriented Programming

คณะเทคโนโลยีสารสนเทศ สาขาวิชาเทคโนโลยีสารสนเทศ

ภาคการศึกษาที่ 1 ปีการศึกษาที่ 2561

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

## คำนำ

รายงานเล่มนี้เป็นส่วนหนึ่งของรายวิชา 06016317 การเขียนโปรแกรมเชิงวัตถุ (Object-Oriented Programming) สาขาวิชาเทคโนโลยีสารสนเทศ คณะเทคโนโลยีสารสนเทศ โดยรายงานเล่มนี้จะเป็นเอกสารประกอบการพัฒนาซอฟต์แวร์ที่มีชื่อว่า Signy IDE โดยเป็นซอฟต์แวร์สำหรับผู้ที่พัฒนาโปรแกรมภาษา Java ซึ่งเป็นภาษาง่ายแก่การพัฒนา ใช้ได้ในหลายๆแพลตฟอร์มทำให้ภาษานี้เป็นที่นิยมมากที่สุดภาษาหนึ่ง โดยซอฟต์แวร์นี้ผู้จัดทำได้ใช้ภาษา Java ในการเขียน

ภายในเล่มนี้จะมีเนื้อหาเกี่ยวกับรายละเอียดของการทำซอฟต์แวร์ ตั้งแต่วัตถุประสงค์และจุดมุ่งหมายของการทำซอฟต์แวร์ เทคโนโลยีที่ใช้ รูปแบบการทำงานโดยรวม รวมไปถึงฟังก์ชันการทำงานต่างๆ และมีคู่มือสำหรับส่วนผู้ใช้อย่างละเอียด เป็นต้น

คณะผู้จัดทำ

17 ธันวาคม 2561

# สารบัญ

เรื่อง

หน้า

<b>1. บทนำ.....</b>	<b>1</b>
1.1. ความเป็นมาของโครงการ.....	1
1.2. วัตถุประสงค์และเป้าหมาย.....	1
1.3. ประโยชน์ที่คาดว่าจะได้รับ และการนำไปใช้งาน.....	2
1.4. ตารางระบุความรับผิดชอบ และขั้นตอนการดำเนินโครงการ .....	2
<b>2. เทคโนโลยีที่เกี่ยวข้อง.....</b>	<b>3</b>
2.1. ภาษา JAVA.....	3
2.2. JAVAFX .....	5
2.3. ECLIPSE JDT .....	7
2.4. RICHTEXTFX .....	8
<b>3. รายละเอียดพัฒนาโปรแกรมในเชิงเทคนิค .....</b>	<b>9</b>
3.1. INPUT SPECIFICATION / OUTPUT SPECIFICATION.....	9
3.2. FUNCTIONAL SPECIFICATION .....	9
3.3. DESIGNอย่างคร่าวของโปรแกรม.....	12
3.4. ขอบเขต และข้อจำกัดของโปรแกรม .....	12
<b>4. คู่มือ / ฟังก์ชันการทำงาน.....</b>	<b>13</b>
4.1. การติดตั้ง JAVA PLATFORM STANDARD EDITION 10.....	13
4.2. ส่วนประกอบต่างๆ ของซอฟต์แวร์ SIGNY IDE (MOCK UP) .....	16
4.3. การใช้งาน และ การคอมไพล์เพื่อทดสอบโค้ด .....	17
4.4. เครื่องมือช่วยเหลือในการเขียนซอฟต์แวร์ .....	18
<b>5. เอกสารอ้างอิง.....</b>	<b>22</b>

# 1. บทนำ

## 1. ความเป็นมาของโครงการ

ปัจจุบันเทคโนโลยีมีบทบาทกับเราอย่างมาก มีโปรแกรม และ เครื่องมือจำนวนมากไม่น้อยที่สามารถเป็นทางเลือกใช้งานใหม่ๆให้กับตัวผู้ใช้ หรือแม้กระทั่งตัวผู้พัฒนาโปรแกรมเอง เมื่อเวลาผ่านไป ผู้พัฒนาโปรแกรมเครื่องมือทางเลือกเหล่านี้ต่างก็พัฒนาโปรแกรมของตนเอง ให้มีประสิทธิภาพที่ดียิ่งขึ้น ใช้งานง่ายและเร็วขึ้น ทำให้ออกเวอร์ชันใหม่ๆมามากมาย ไม่ว่าจะเป็น Nano หรือ VIM ที่เดิมทียังใช้งานยาก จึงเกิดการพัฒนาซอฟต์แวร์อย่าง Visual Studio Code หรือ Sublime Text เป็นต้น

ทางผู้พัฒนา ได้เล็งเห็นถึงจุดๆนี้ และได้มีความคิดที่จะทำชุดโปรแกรมพัฒนาซอฟต์แวร์ หรือ IDE (Integrated Development Environment) ซึ่งมีประโยชน์อย่างมากในการใช้ศึกษา และสามารถพัฒนาซอฟต์แวร์ได้ในขณะเดียวกัน โดยโปรแกรมที่คณะผู้จัดทำได้คิดริเริ่มนี้จะสามารถถูกนำไปต่อยอด หรือพัฒนาเป็นโครงการต่อไปได้ เป็นการฝึกทักษะการเขียนและการใช้โปรแกรม รู้ระบบการดำเนินงาน ทำให้รู้สิ่งที่ควรพัฒนาต่อยอดและจุดบกพร่องของซอฟต์แวร์ได้เป็นอย่างดี

โดยโปรแกรมพัฒนาซอฟต์แวร์ โดยคณะผู้จัดทำใช้ชื่อว่า Signy IDE จะเป็นโปรแกรมเขียนภาษา Java พัฒนามาจากโปรแกรม Eclipse โดยใช้ภาษา Java ในการเขียน ซึ่งโปรแกรมนี้อจะเป็นโปรแกรมต้นแบบเพื่อนำมาจัดการพัฒนาซอฟต์แวร์ของนักพัฒนา มีระบบที่สะดวก และตอบสนอง ต่อการใช้งานของนักเขียนโปรแกรมคอมพิวเตอร์ และผู้พัฒนาสามารถพัฒนาโปรแกรมได้แม้ว่าคอมพิวเตอร์จะมีประสิทธิภาพไม่สูง เป็นต้น

## 2. วัตถุประสงค์และเป้าหมาย

- มีความเข้าใจในการเขียนโปรแกรมเชิงวัตถุ หรือ Object Oriented Programming มากขึ้น
- เพื่อฝึกทักษะการใช้โปรแกรม Eclipse ในการเขียนภาษา Java และสามารถนำไปประยุกต์ใช้ได้
- เพื่อให้มีระบบสำหรับการพัฒนาซอฟต์แวร์ ที่ตอบสนองความต้องการของนักพัฒนาซอฟต์แวร์ และเพิ่มประสิทธิภาพในการพัฒนาซอฟต์แวร์ให้กับผู้พัฒนา
- เพื่อพัฒนาเสริมสร้างทักษะความคิดริเริ่มในการเขียนโปรแกรมต้นแบบ ซึ่งเป็นทักษะสำคัญที่จะใช้ในการพัฒนาซอฟต์แวร์ หรือ นำไปต่อยอดในอนาคต
- เพื่อเสริมสร้างความรู้ความสามารถให้นำไปใช้ประโยชน์ได้ต่อในอนาคต

### 3. ประโยชน์ที่คาดว่าจะได้รับ และการนำไปใช้งาน

- ทำให้ได้เรียนรู้เกี่ยวกับการทำงาน ประสานงาน พร้อมทั้งมีการวางแผนการทำงาน
- ทำให้เกิดการพัฒนาความคิด รู้จักการแก้ปัญหาต่าง ๆ โดยเฉพาะการแก้ปัญหาเฉพาะหน้า
- มีความเข้าใจในการเขียนโปรแกรมเชิงวัตถุ หรือ Object Oriented Programming มากขึ้น
- นำทักษะการใช้โปรแกรมไปประยุกต์ใช้ได้
- ซอฟต์แวร์ที่ถูกพัฒนาขึ้นสามารถนำไปใช้ได้จริง

### 4. ตารางระบุความรับผิดชอบ และขั้นตอนการดำเนินโครงการ

ขั้นตอนการดำเนินงาน	ระยะเวลา			ผู้รับผิดชอบ
	กันยายน	ตุลาคม	พฤศจิกายน	
หาข้อมูล ศึกษาความเป็นไปได้ในการจัดทำ				ฐิติโชติ ศิวาวัดน์ ศุภกิตต์
ออกแบบโปรแกรม และจัดวาง Layout				ศิวาวัดน์ ศุภกิตต์
พัฒนาโครงสร้างต้นแบบของโปรแกรม และ พัฒนา Text Editor				ศุภกิตต์
ออกแบบส่วน Compiler และ Console				ศิวาวัดน์
ออกแบบส่วนขยายและเครื่องมือเสริม				ศิวาวัดน์
ทำรูปเล่มรายงานและเอกสาร				ฐิติโชติ
ปรับแต่งระบบของ Text Editor เพิ่มเติม				ฐิติโชติ ศุภกิตต์
ตรวจสอบโค้ด และทดสอบระบบ				ศิวาวัดน์ ศุภกิตต์

## 2. เทคโนโลยีที่เกี่ยวข้อง

### 1. ภาษา Java



**ภาษาจาวา** (Java programming language) เป็นภาษาโปรแกรมเชิงวัตถุ (อังกฤษ: Object Oriented Programming) พัฒนาโดย James Gosling และวิศวกรคนอื่นๆ ที่ Sun Microsystems ภาษาจาวาถูกพัฒนาขึ้นในปี พ.ศ. 2534 (ค.ศ. 1991) โดยเป็นส่วนหนึ่งของ โครงการกรีน (the Green Project) และสำเร็จออกสู่สาธารณะในปี พ.ศ. 2538 (ค.ศ. 1995) ซึ่งภาษานี้มีจุดประสงค์เพื่อใช้แทนภาษา C++ โดยรูปแบบที่เพิ่มเติมขึ้นคล้ายกับภาษา Objective-C แต่เดิมภาษานี้เรียกว่า ภาษาโอ๊ก (Oak) ซึ่งตั้งชื่อตามต้นโอ๊ก ใกล้ที่ทำงานของ James Gosling แต่มีปัญหาทางลิขสิทธิ์ จึงเปลี่ยนไปใช้ชื่อ "จาวา" ซึ่งเป็นชื่อกาแฟแทน

และแม้ว่าจะมีชื่อคล้ายกัน แต่ภาษาจาวาไม่มีความเกี่ยวข้องใด ๆ กับภาษาจาวาสคริปต์ (JavaScript) ปัจจุบันมาตรฐานของภาษาจาวาดูแลโดย Java Community Process ซึ่งเป็นกระบวนการอย่างเป็นทางการ ที่อนุญาตให้ผู้สนใจเข้าร่วมกำหนดความสามารถในจาวาแพลตฟอร์มได้

#### ข้อดีของ ภาษา Java

- ภาษา Java เป็นภาษาที่สนับสนุนการเขียนโปรแกรมเชิงวัตถุแบบสมบูรณ์ ซึ่งเหมาะสำหรับพัฒนาระบบที่มีความซับซ้อน การพัฒนาโปรแกรมแบบวัตถุจะช่วยให้เราสามารถใช้คำหรือชื่อ ต่าง ๆ ที่มีอยู่ในระบบงานนั้นมาใช้ในการออกแบบโปรแกรมได้ ทำให้เข้าใจได้ง่ายขึ้น
- โปรแกรมที่เขียนขึ้นโดยใช้ภาษา Java จะมีความสามารถทำงานได้ในระบบปฏิบัติการที่แตกต่างกัน ไม่จำเป็นต้องดัดแปลงแก้ไขโปรแกรม เช่น หากเขียนโปรแกรมบนเครื่อง Sun โปรแกรมนั้นก็สามารถูก compile และ run บนเครื่องพีซีธรรมดาได้

- ภาษาจาวามีการตรวจสอบข้อผิดพลาดทั้งตอน compile time และ runtime ทำให้ลดข้อผิดพลาดที่อาจเกิดขึ้นในโปรแกรม และช่วยให้ debug โปรแกรมได้ง่าย

- ภาษาจาวามีความซับซ้อนน้อยกว่าภาษา C++ เมื่อเปรียบเทียบ code ของโปรแกรมที่เขียนขึ้นโดยภาษา Java กับ C++ พบว่า โปรแกรมที่เขียนโดยภาษา Java จะมีจำนวนcode น้อยกว่าโปรแกรมที่เขียนโดยภาษา C++ ทำให้ใช้งานได้ง่ายกว่าและลดความผิดพลาดได้มากขึ้น

- ภาษาจาวาถูกออกแบบมาให้มีความปลอดภัยสูงตั้งแต่แรก ทำให้โปรแกรมที่เขียนขึ้นด้วยจาวามีความปลอดภัยมากกว่าโปรแกรมที่เขียนขึ้น ด้วยภาษาอื่น เพราะ Java มี security ทั้ง low level และ high level ได้แก่ electronic signature, public and private key management, access control และ certificates

- มี IDE, application server, และ library ต่าง ๆ มากมายสำหรับจาวาที่เราสามารถใช้งานได้โดยไม่ต้องเสียค่าใช้จ่าย ทำให้เราสามารถลดค่าใช้จ่ายที่ต้องเสียไปกับการซื้อ tool และ s/w ต่าง ๆ

### ข้อเสียของ ภาษา Java

- ทำงานได้ช้ากว่า native code (โปรแกรมที่ compile ให้อยู่ในรูปของภาษาเครื่อง) หรือโปรแกรมที่เขียนขึ้นด้วยภาษาอื่น อย่างเช่น C หรือ C++ ทั้งนี้ก็เพราะว่าโปรแกรมที่เขียนขึ้นด้วยภาษาจาวาจะถูกแปลงเป็นภาษากลาง ก่อน แล้วเมื่อโปรแกรมทำงานคำสั่งของภาษากลางนี้จะถูกเปลี่ยนเป็นภาษาเครื่องอีก ทีหนึ่งทีละคำสั่ง (หรือกลุ่มของคำสั่ง) ณ runtime ทำให้ทำงานช้ากว่า native code ซึ่งอยู่ในรูปของภาษาเครื่องแล้วตั้งแต่ compile โปรแกรมที่ต้องการความเร็วในการทำงานจึงไม่นิยมเขียนด้วยจาวา

- เครื่องมือที่มีในการใช้พัฒนาโปรแกรมจาวามักไม่ค่อยเก่ง ทำให้หลายอย่างโปรแกรมเมอร์จะต้องเป็นคนทำเอง ทำให้ต้องเสียเวลาทำงานในส่วนที่ tool ทำไม่ได้ ถ้าเราดู tool ของ MS จะใช้งานได้ง่ายกว่าและพัฒนาได้เร็วกว่า (แต่เราต้องซื้อ tool ของ MS และก็ต้องรันบน platform ของ MS)

สำหรับการพัฒนาภาษา Java สามารถพัฒนา Application ได้หลากหลายรูปแบบมาก เช่น Application ที่ทำงานบน Windows, Mac, Linux หรือบน Web Application (JSP Java Servlet) และที่กำลังมาแรงสุดในตอนนี้คือ การพัฒนา Application บน Mobile ซึ่งในปัจจุบันสามารถพัฒนาได้บน Android และ BlackBerry และในอนาคตจะยังมีตามมาอีกหลายตัวแน่นอน

ดังนั้นในภาษา Java จะมีรุ่นที่เป็น SDK อยู่หลายตัว อาทิเช่น J2SE, J2EE, J2ME หรือ SE, EE, ME โดยรุ่นต่างๆเหล่านี้จะเป็นตัวบอกการพัฒนาใน Platform ต่างๆ เช่น

- J2SE ปัจจุบันเปลี่ยนชื่อเป็น SE (Standard Edition) ไว้สำหรับพัฒนาโปรแกรมบนคอมพิวเตอร์ตั้งโต๊ะทั่วไป
- J2EE ปัจจุบันเปลี่ยนชื่อเป็น EE (Enterprise Edition) ไว้สำหรับพัฒนาโปรแกรมในองค์กรใหญ่ๆ หรือมีขอบเขตของโครงการกว้างมาก
- J2ME ปัจจุบันเปลี่ยนชื่อเป็น ME (Micro Edition) ไว้สำหรับพัฒนาโปรแกรมบนอุปกรณ์พกพา เช่น โทรศัพท์มือถือ หรือพีดีเอ

ซึ่งปกติแล้วในการพัฒนา Application ด้วยภาษา Java ทั่วๆไปจะใช้รุ่น SE (Standard Edition) จะมี JDK (Java Development Kit) ที่ประกอบไปด้วย compiler และ debugger ของภาษา Java สำหรับนักพัฒนา JRE (Java Runtime Environment) ซึ่งเป็นสิ่งที่รวม library ต่างๆสำหรับการรันโปรแกรมที่พัฒนาด้วย Java ซึ่งถ้าติดตั้ง JDK เพียงตัวเดียวก็จะมี JRE รวมอยู่ด้วย

## 2. JavaFX



JavaFX 8 เป็น Library มาตรฐานน้องใหม่ที่มาพร้อมกับการติดตั้งระบบ Java 8 ไว้ใช้งานแทนที่การใช้งาน Java Swing เดิมที่ใช้สำหรับการเขียนคอมพิวเตอร์กราฟฟิกส์สำหรับสื่อสารกับผู้ใช้หรือที่เรียกว่า GUI (Graphic User Interface) นั่นเอง

JavaFX เป็น Library มาตรฐาน เราจึงสามารถพัฒนาขึ้นมาด้วยภาษา Java ได้หรือแม้แต่ใช้พวก Visage, Groovy, Scala หรือ Kotlin ก็ตาม



เราสามารถเขียนโปรแกรมโดยใช้ Library จาก JavaFX เพียงอย่างเดียวหรือจะใช้ไฟล์ FXML เป็นตัวช่วยในการจัดวางลักษณะเครื่องมือทางกราฟฟิกส์ต่างๆบนหน้าจอ ซึ่ง FXML ก็มีลักษณะของ XML อยู่ด้วยทำให้การเรียนรู้และใช้งานง่าย และที่เด็ดสุดก็เรื่องการมี SceneBuilder เป็นตัวช่วยในการจัดวางเครื่องมือเครื่องมือหรือการจัดหน้าจอปุ่มต่างๆแบบกราฟิก wysiwyg นั่นเอง

JavaFX รองรับ data binding, multimedia และสามารถเขียน Web Application ได้ด้วย แถมความสามารถในการจัดการกราฟฟิกส์และ animation สามารถเอาไปทำเกมหรือ Presentation แบบอื่นๆได้อีกด้วย

GUI หรือเครื่องมือทางกราฟฟิกส์ที่สามารถมองเห็นแสดงผลรูปหรือข้อความ ใช้เมาส์คลิกหรือมีสัมผัส มีช่องเติมข้อความ หรือปุ่มเลื่อนปุ่มกดแบบต่างๆ ฯลฯ ในการเขียนโปรแกรมด้วย Java สิ่งเหล่านี้จะเป็น Object ที่เหมือนเอามาแขวนอ้างอิงกันไว้ในแบบที่เรียกว่า Scene graph ส่วนที่เล็กที่สุดของ scene graph เรียกว่า node ซึ่งก็คือบรรดาปุ่ม ข้อความ รูปเหลี่ยมต่างๆ ภาพ และสื่อต่างๆ ดังที่กล่าวไป

โครงสร้างของ **JavaFX platform** เป็นดังแผนภาพนี้

JavaFX Public API			
Quantum Toolkit			
Prism	Glass Windowing Toolkit	Media Engine	Web Engine

#### - Prism

Prismคือส่วนระบบของ Java ที่เป็นประสานงานกับตัวเร่งทางกราฟฟิกส์ด้วย Hardware Driver ใช้เขียน GUI ต่างๆ ถ้าระบบที่ใช้ไม่มีฮาร์ดแวร์พิเศษทางกราฟฟิกส์เช่นพวก DirectX หรือ OpenGL แล้ว Java ก็จะใช้ Java 2D สำหรับการวาด GUI ในทางเทคนิค การวาด GUI เราจะเรียกว่า Render

#### - Glass Windowing Toolkit

Glass Windowing Toolkitเป็นตัวเชื่อมระบบวินโดวส์ของ OS ที่ Java อาศัยอยู่กับ Application ของเราเพื่อการแสดงผลทางกราฟฟิกส์และการจัดลำดับเหตุการณ์สำหรับการแสดงผลหรือการตอบสนอง (ทางเทคนิคเรียกว่า Eventโดยที่การทำงานในส่วนการจัดการ event นี้เราเรียกว่า JavaFX Application Thread ซึ่งมีเพียง Thread เดียวในระบบ

Prism จะใช้ Thread ในการ render แยกออกมาจาก JavaFX Application Thread โดยใช้ event ตัวที่เรียกว่า pulse event ซึ่งจะ synchronized การวาดร่วมกับ event อื่นที่จัดการโดย Java Application Thread ด้วยความเร็วสูงสุดที่ 60 เฟรมต่อวินาที

#### - Media Engine

Media Engine ใช้ในการจัดการวิดีโอและเสียงโดยอาศัย Codec ของ Java และปล่อยให้ JavaFX Application Thread เป็นตัว synchronize เพื่อให้ทำงานร่วมกับระบบรวมได้ ซึ่ง Media Engine ของ Java จะใช้ GStreamer เป็นตัวจัดการกลไกการทำงานของวิดีโอและเสียงอีกทีหนึ่ง

#### - Web Engine

**Web Engine** จะใช้ WebKit ซึ่งรองรับการแสดงผลพวก HTML5 CSS JavaScript และ DOM แล้วส่งต่อให้ Prism ในการวาดลงบนจอ

#### - Quantum Toolkit

**Quantum Toolkit** เป็น abstract ใช้สร้าง JavaFX เพื่อเชื่อมการทำงาน Prism, Glass Windowing Toolkit, Media Engine และ Web Engine เข้าด้วยกัน

### 3. Eclipse JDT

Eclipse คือโปรแกรมที่ใช้สำหรับพัฒนาภาษา Java ซึ่งโปรแกรม Eclipse เป็นโปรแกรมหนึ่งที่ใช้ในการพัฒนา Application Server ได้อย่างมีประสิทธิภาพ และเนื่องจาก Eclipse เป็นซอฟต์แวร์ OpenSource ที่พัฒนาขึ้นเพื่อใช้โดยนักพัฒนาเอง ทำให้ความก้าวหน้าในการพัฒนาของ Eclipse เป็นไปอย่างต่อเนื่องและรวดเร็ว

Eclipse มีองค์ประกอบหลักที่เรียกว่า Eclipse Platform ซึ่งให้บริการพื้นฐานหลักสำหรับรวบรวมเครื่องมือต่างๆจากภายนอกให้สามารถเข้ามาทำงานร่วมกันในสภาพแวดล้อมเดียวกัน และมีองค์ประกอบที่เรียกว่า Plug-in Development Environment (PDE) ซึ่งใช้ในการเพิ่มความสามารถในการพัฒนาซอฟต์แวร์มากขึ้น เครื่องมือภายนอกจะถูกพัฒนาในรูปแบบที่เรียกว่า Eclipse plug-ins ดังนั้นหากต้องการให้ Eclipse ทำงานใดเพิ่มเติม ก็เพียงแค่พัฒนา plugin สำหรับงานนั้นขึ้นมา และนำ Plug-in นั้นมาติดตั้งเพิ่มเติมให้กับ Eclipse ที่มีอยู่เท่านั้น Eclipse Plug-in ที่มีมาพร้อมกับ Eclipse เมื่อเรา download มาครั้งแรกก็คือ องค์ประกอบที่เรียกว่า Java Development Toolkit (JDT) ซึ่งเป็นเครื่องมือในการเขียน และ Debug โปรแกรมภาษา Java

ข้อดีของโปรแกรม Eclipse คือ ติดตั้งง่าย สามารถใช้ได้กับ J2SDK ได้ทุกเวอร์ชันรองรับ ภาษาต่างประเทศอีกหลายภาษา มี plugin ที่ใช้เสริมประสิทธิภาพของโปรแกรม สามารถทำงานได้กับไฟล์ หลายชนิด เช่น HTML, Java, C, JSP, EJB, XML และ GIF และที่สำคัญเป็น Freeware (ให้ใช้งานได้ 90 วัน ถ้าจะใช้งานเต็มประสิทธิภาพต้องเสียค่าใช้จ่ายภายหลัง) ใช้งานได้กับระบบปฏิบัติการ Windows, Linux และ Mac OS

#### 4. RichtextFX

RichtextFX ทำให้กล่องข้อความของ JavaFX ใช้งานได้มีประสิทธิภาพมากขึ้น ช่วยให้นักพัฒนา สามารถกำหนดช่วงของข้อความที่แสดงในเนื้อหา ขอบเขตเครื่องหมาย ตัวอักษร โดยไม่ต้องใช้ HTMLEditor โดยตัวของ RichtextFX จะไม่ขึ้นอยู่กับรูปแบบ MVC (Model View Controller) Text-Highlight

### 3. รายละเอียดพัฒนาโปรแกรมในเชิงเทคนิค

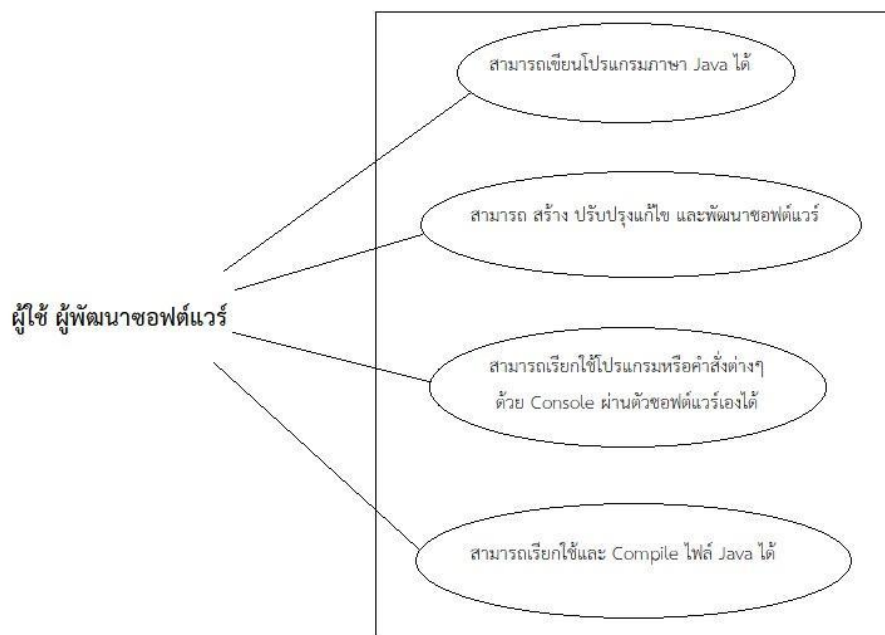
#### 1. Input Specification / Output Specification

ตัวซอฟต์แวร์จะรับ Event ต่างๆ การพิมพ์ในส่วนของหน้าสำหรับพัฒนาโปรแกรม คำสั่งใน Console หรือคำสั่งกด จากเมาส์ หรือคีย์บอร์ด และแสดงผลเป็นสิ่งที่ Compile ได้จาก Input ผ่านส่วนพัฒนาโปรแกรม หรือคำสั่งที่ทำงานผ่าน Console

#### 2. Functional Specification

##### - บทนำ

Signy IDE เป็นซอฟต์แวร์สำหรับช่วยเหลือผู้ที่พัฒนาซอฟต์แวร์ ทำงานโดยการเขียนโปรแกรมและ Compile มีระบบการจัดการ เครื่องมือต่างๆ และทำงานได้รวดเร็ว



## - Use Case

Use case Name	ซอฟต์แวร์ Integrated Development Environment ที่ใช้จัดการกับ ภาษา Java
Actor	ผู้ใช้ หรือ ผู้พัฒนาซอฟต์แวร์
Description	Text Editor, Compiler, Console, Outline , File Explorer
Abstract use case	เมนูช่วยเหลือในการเช็ค และ จัดระเบียบ
Main Flow	Actor หรือ ผู้ใช้ จะทำการเขียนโค้ดลงไปในส่วนของ Text Editor โดยตัว ซอฟต์แวร์จะมีฟังก์ชันการทำงาน Compiler และสามารถรันคำสั่งใน Console ในการเพื่อใช้คำสั่งต่างๆ และผู้ใช้สามารถเรียกเปิดไฟล์หรือ Save ไฟล์ที่เขียน ผ่านเมนูต่างๆภายในซอฟต์แวร์ และมีเครื่องมือช่วยเหลือในการทำงาน เช่นการ จัดระเบียบโค้ด

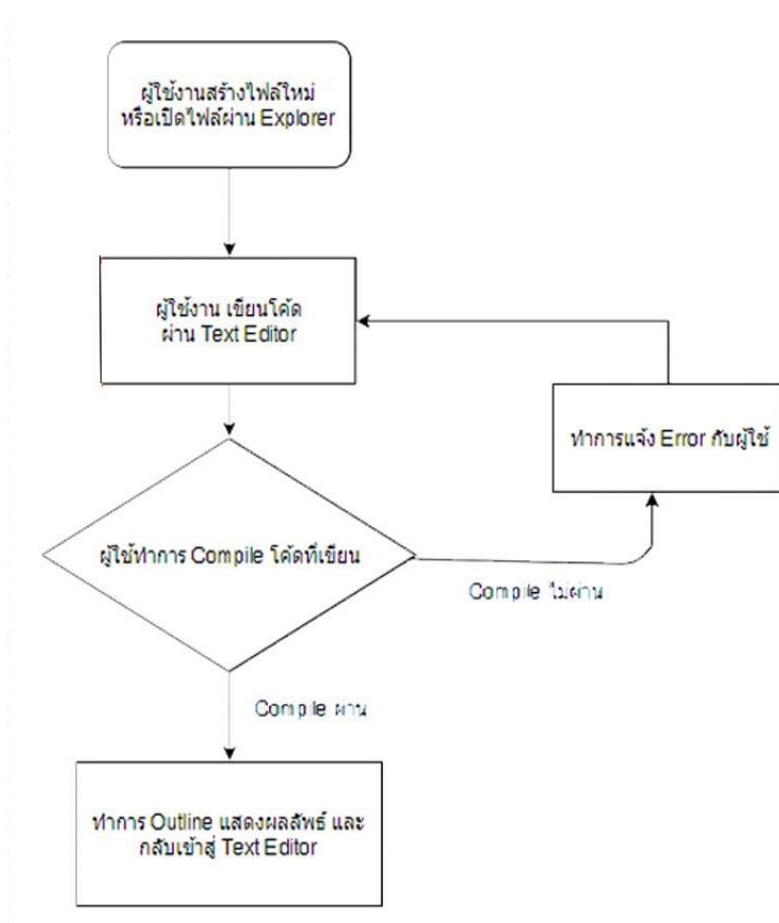
## - ส่วนข้อมูลนำเข้า

### Form Elements:

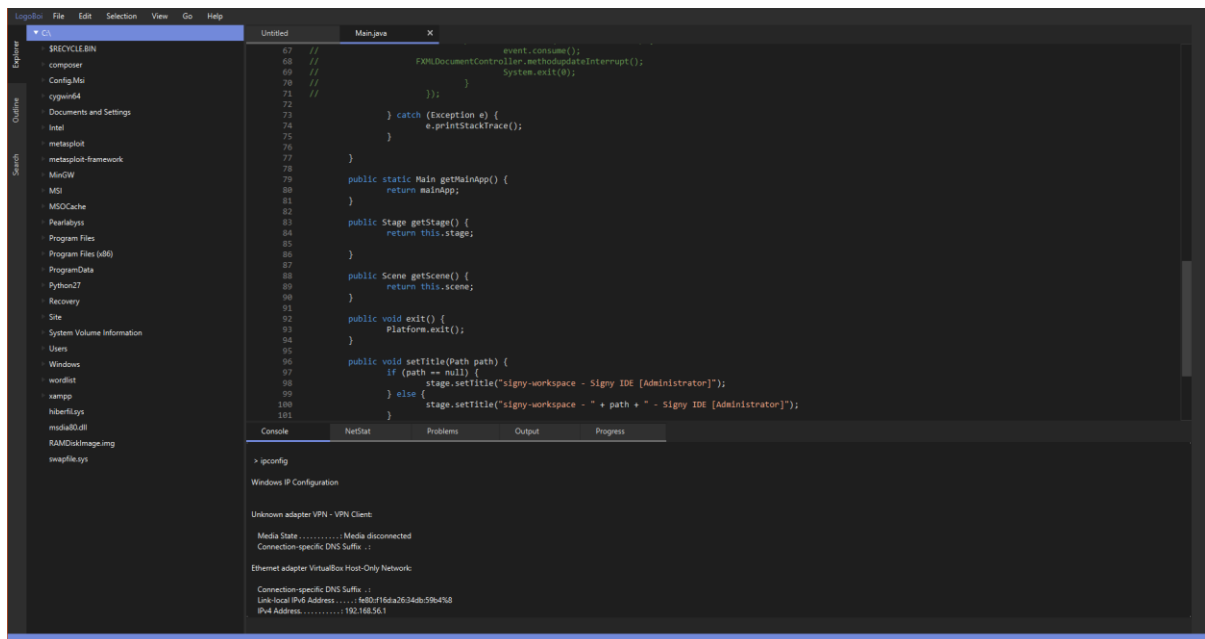
Field Label	Description	Data Type	Editable	UI Control
Text Editor	เขียนโค้ดภาษา Java	ตัวอักษรและตัวเลข	ได้	Textarea
Console	ใช้เรียกคำสั่ง เช่น การ Compile	ตัวอักษรและตัวเลข	ได้	Textfield
Outline	แสดงค่าผลลัพธ์	Tree	ขึ้นอยู่กับ Editor	Treeview
Explorer	ค้นหา/เปิด ไฟล์ใน ตำแหน่งที่กำหนด	Tree <File>	ขึ้นอยู่กับ Directory	Treeview
Search	ค้นหาคำที่เกี่ยวข้อง	ตัวอักษรและตัวเลข	ได้	Textfield

## Menu Elements:

Menu Label	Description
File	จัดการตัวไฟล์ เช่นการ Save , Open ไฟล์
Edit	ช่วยเหลือในการแก้ไข Text เช่น Cut Copy Replace
Selection	เลือกส่วนของ Text ให้ง่ายต่อการแก้ไข
View	เปิด ปิด แถบเครื่องมือ
Help	เกี่ยวกับผู้จัดทำซอฟต์แวร์และเอกสารช่วยเหลือ



### 3. Design อย่างคร่าวๆของโปรแกรม



### 4. ขอบเขต และข้อจำกัดของโปรแกรม

#### 1 Spec ของเครื่องคอมพิวเตอร์

- OS: Microsoft Windows 7, 8.1, 10 with Java Runtime Environment 10 Processor
- Processor: Processor with Duo Core or better
- Memory: 1 GB RAM Disk Space
- Disk Space: 1 GB free space while running
- Java Runtime Environment 10 is required


#### 2 ผู้ใช้โปรแกรมต้องมีความรู้ในภาษา Java และการเรียกคำสั่ง Console

## 4. คู่มือ / ฟังก์ชันการทำงาน

### 1. การติดตั้ง Java Platform Standard Edition 10

ขั้นตอนที่ 1 ไปที่ลิงค์ด้านล่าง

<https://www.oracle.com/technetwork/java/javase/downloads/java-archive-javase10-4425482.html>

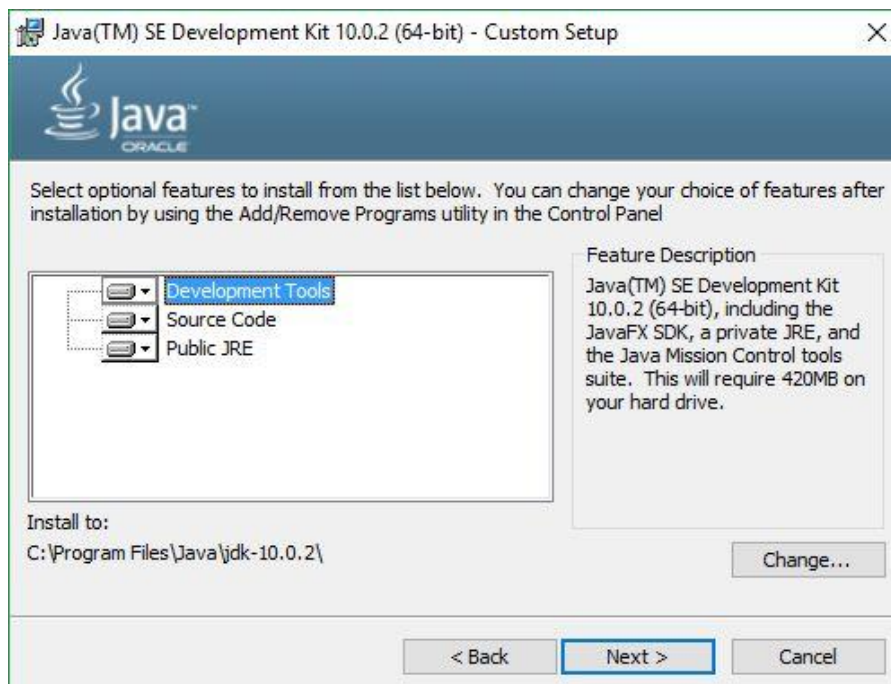
Java SE Development Kit 10.0.2		
You must accept the <a href="#">Oracle Binary Code License Agreement for Java SE</a> to download this software.		
<input type="radio"/> Accept License Agreement <input checked="" type="radio"/> Decline License Agreement		
Product / File Description	File Size	Download
Linux	306 MB	 <a href="#">jdk-10.0.2_linux-x64_bin.rpm</a>
Linux	338.43 MB	 <a href="#">jdk-10.0.2_linux-x64_bin.tar.gz</a>
macOS	395.46 MB	 <a href="#">jdk-10.0.2_osx-x64_bin.dmg</a>
Solaris SPARC	207.07 MB	 <a href="#">jdk-10.0.2_solaris-sparcv9_bin.tar.gz</a>
Windows	390.25 MB	 <a href="#">jdk-10.0.2_windows-x64_bin.exe</a>
<a href="#">Back to top</a>		

ขั้นตอนที่ 2 ด้านล่างจะมีหน้าต่างให้เลือก Download Java SE 10.0.2 เลือก Accept License Agreement เพื่อขออนุญาต Download จากนั้นให้คลิก Downloadตามระบบปฏิบัติการที่เราใช้

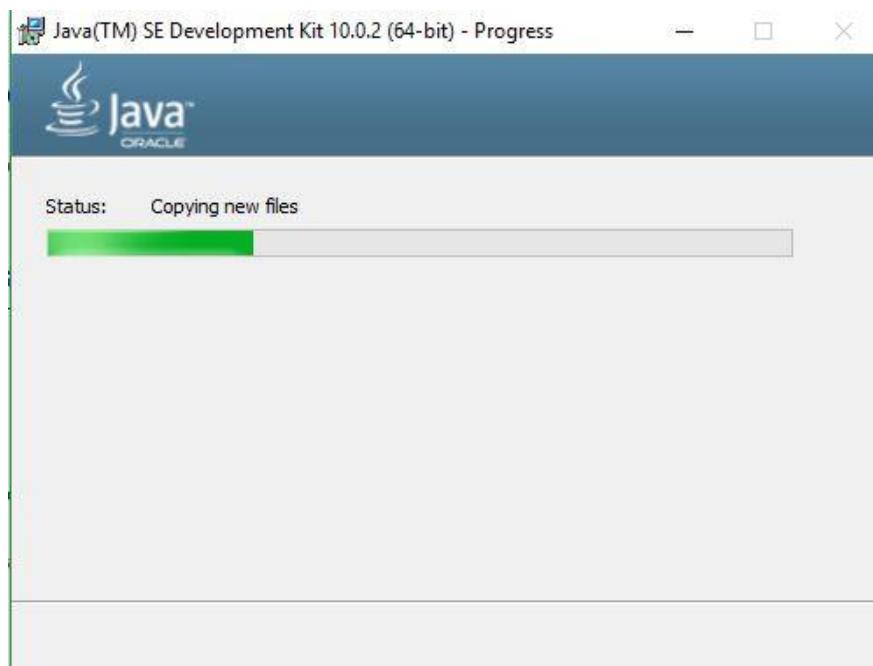


ขั้นตอนที่ 3 จะปรากฏหน้าต่างขึ้น ให้กด Next > เข้าสู่หน้าต่างติดตั้ง



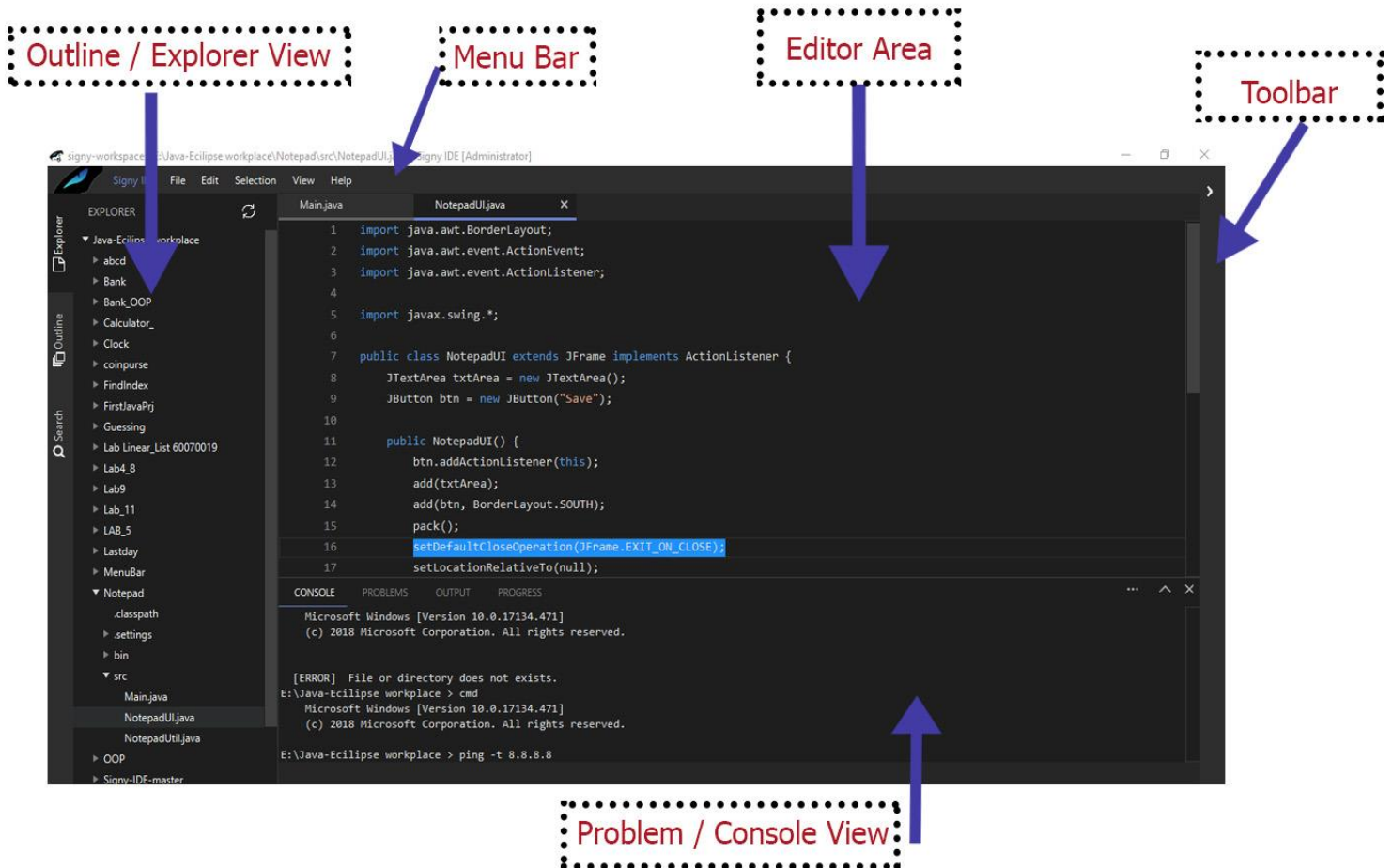


ขั้นตอนที่ 4 เลือก Development Tools แล้วกด Next > แล้วรอจนเสร็จ





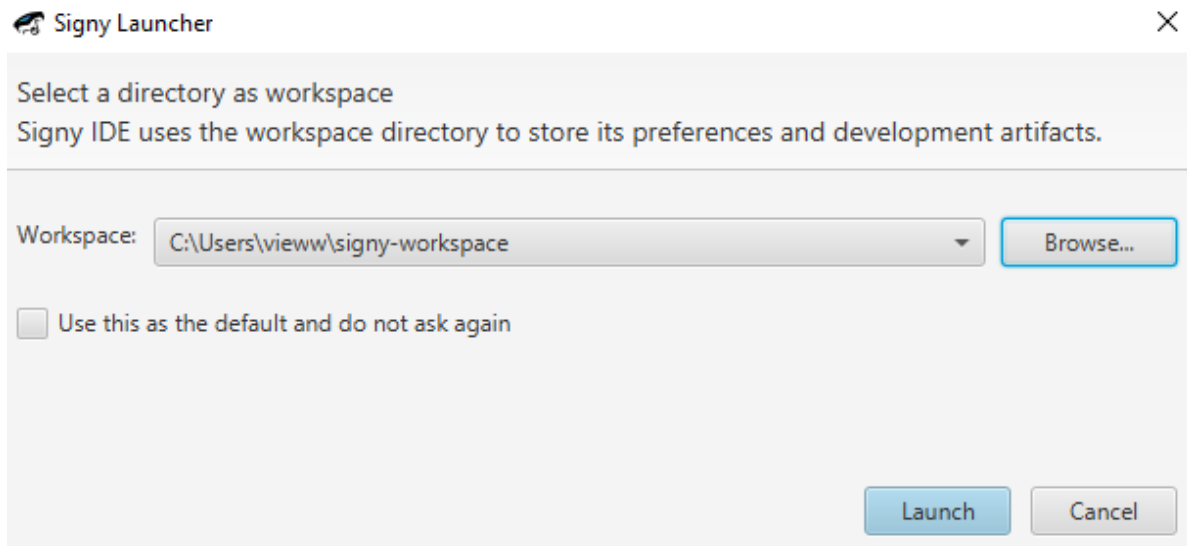
## 2. ส่วนประกอบต่างๆ ของซอฟต์แวร์ Signy IDE (Mock Up)



ICON ของ Signy IDE

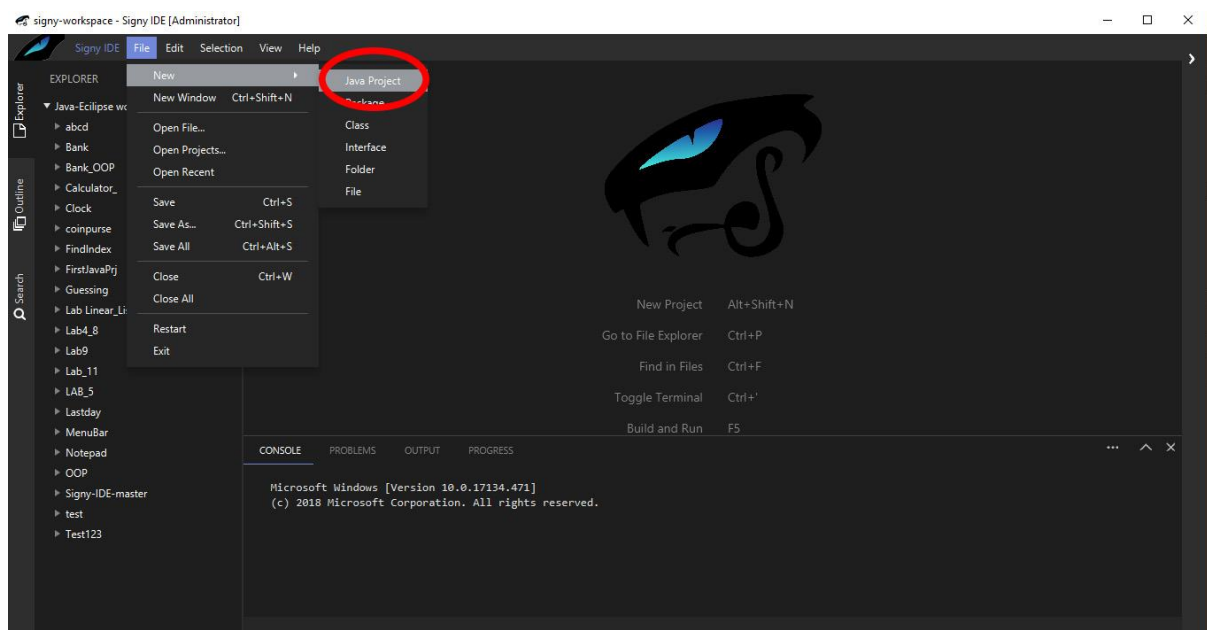
### 3. การใช้งาน และการคอมไพล์เพื่อทดสอบโค้ด

เปิดโปรแกรมขึ้นมา ให้เลือก Directory ที่ต้องการบรรจุไฟล์

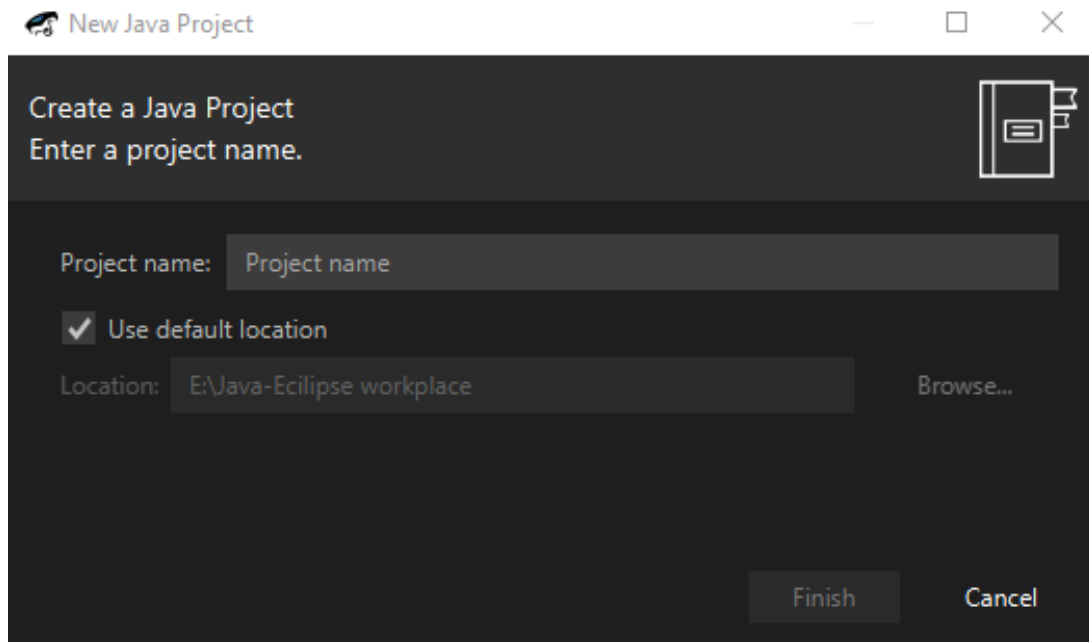


### การสร้าง Java Project

ไปที่แถบ Menu bar เลือกที่ File >> New >> Java Project

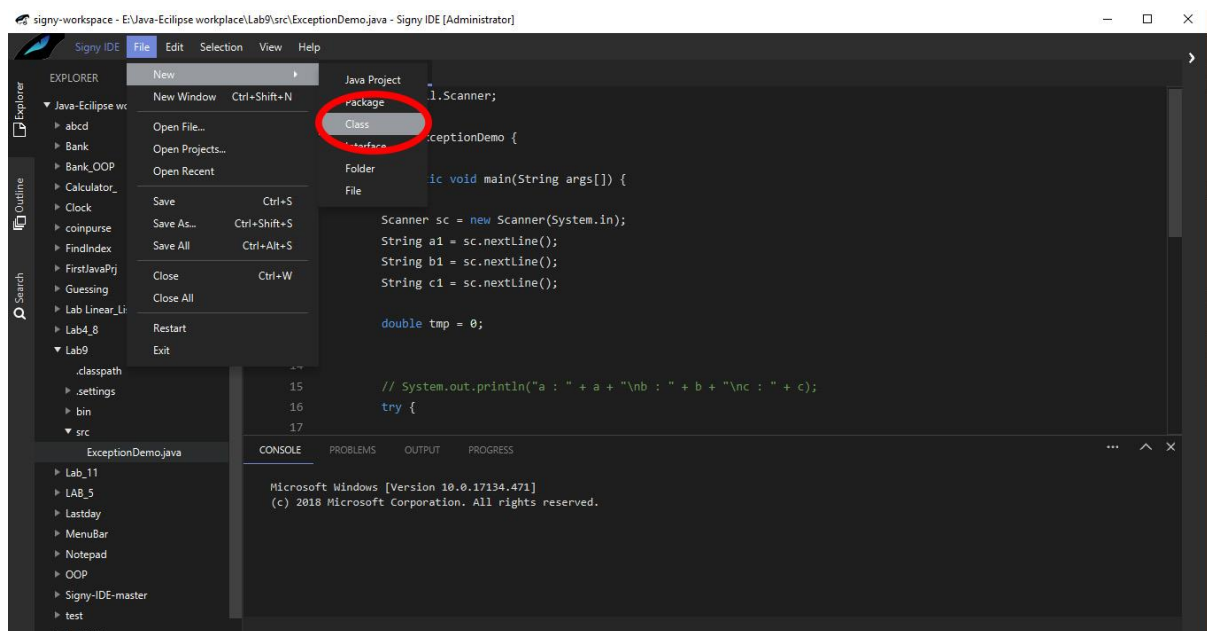


ตั้งชื่อ Java Project ให้เรียบร้อย จากนั้นกด Finish

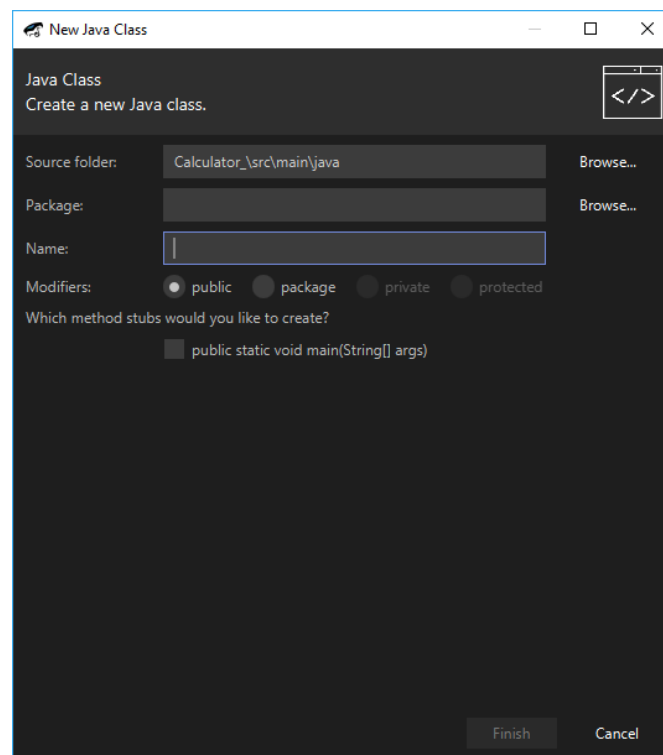


## การสร้าง Class

ไปที่แถบ Menu bar เลือกที่ File >> New >> Class

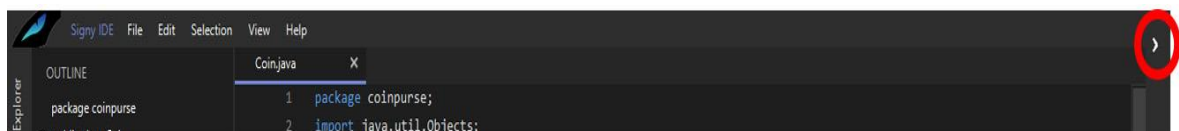


ตั้งชื่อ Class และเซตค่าต่างๆของ Class จากนั้นกด Finish

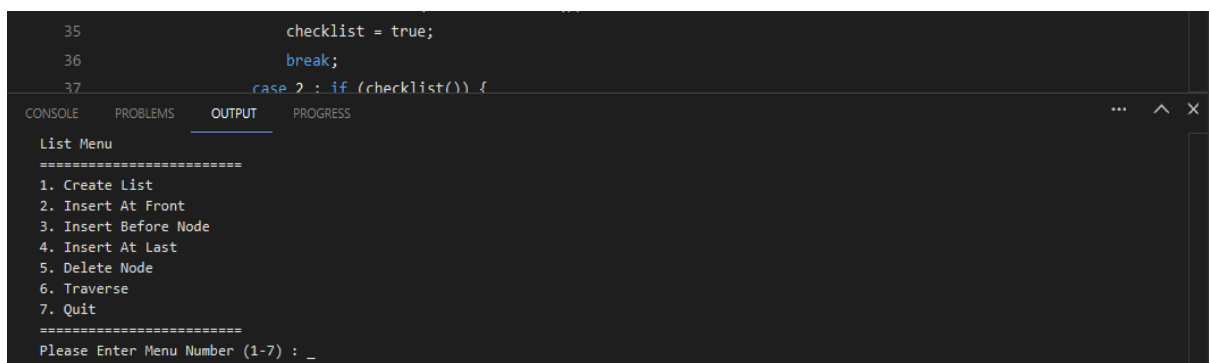


การคอมไพล์เพื่อทดสอบ Class

ไปที่แถบ Tool bar กดปุ่มมุมขวาบนเพื่อคอมไพล์



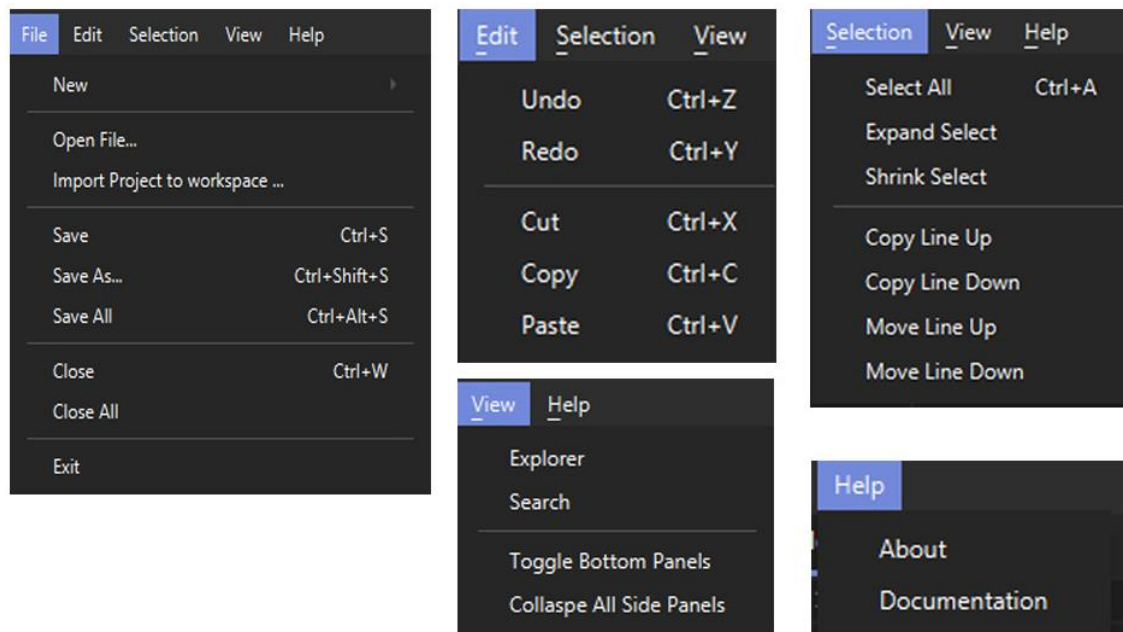
แสดงผลที่ Toggle Bottom Panel ด้านล่าง



#### 4. เครื่องมือช่วยเหลือในการเขียนซอฟต์แวร์

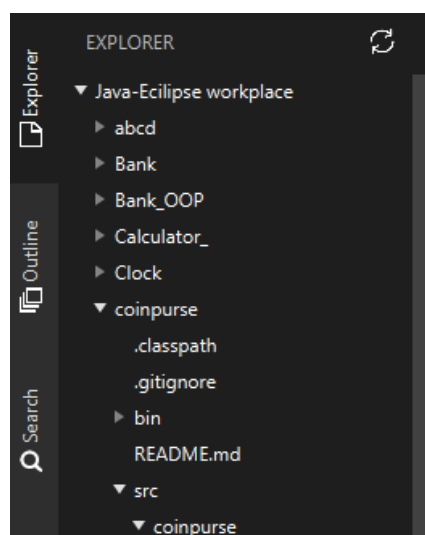
##### Menu bar

- Menu bar >> File คำสั่งในการจัดการไฟล์เช่นการเปิดปิดไฟล์ต่างๆ
- Menu bar >> Edit การจัดการ Text หรือ History
- Menu bar >> Selection เครื่องมือช่วยการจัดการข้อความที่ Select
- Menu bar >> View เปิด หรือ ปิด Toggle Bottom Panel หรือ Side Panel
- Menu bar >> About ส่วนติดต่อผู้จัดทำ ดูหน้าเอกสาร หรือ ไปหน้าเว็บไซต์ API

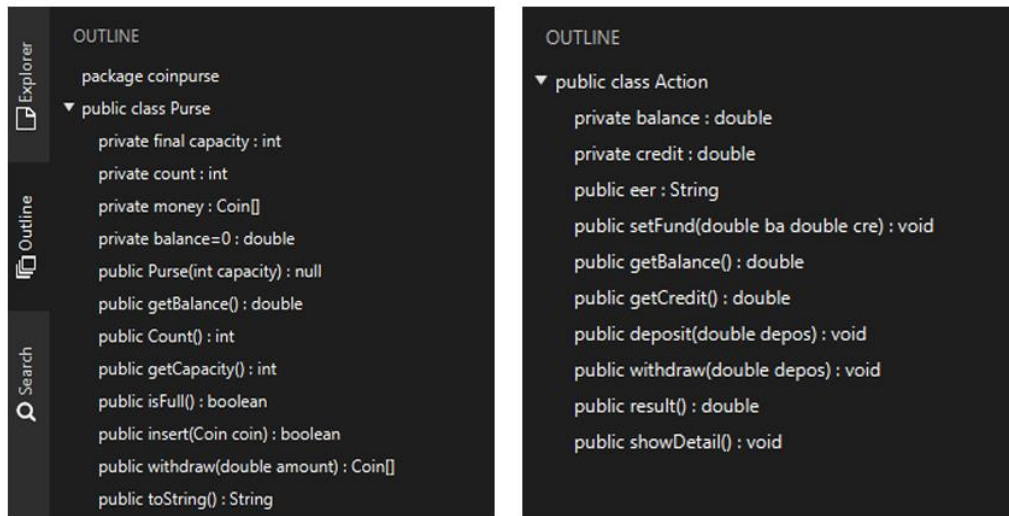


##### เครื่องมือช่วยเหลือ ด้าน Side Panel

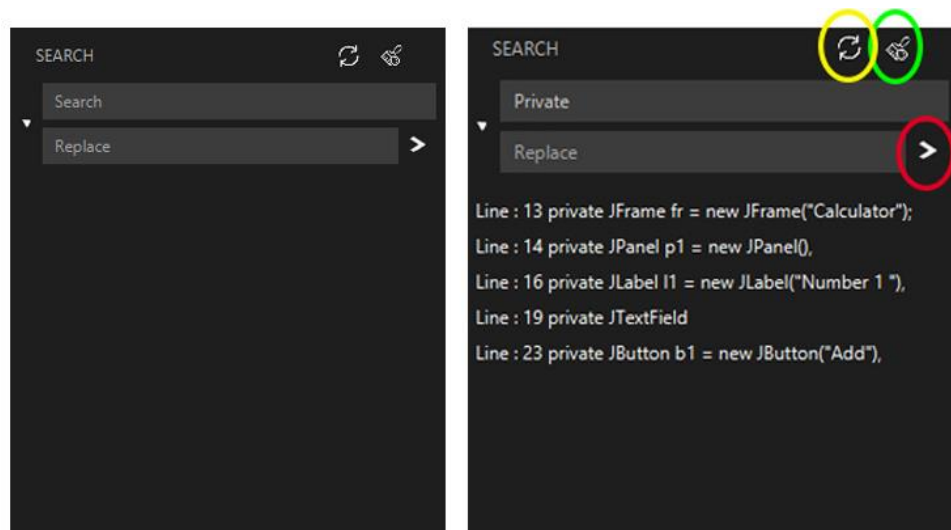
- Explorer แถบเครื่องมือ เพื่อเรียกดูไฟล์ Browse หรือค้นหาไฟล์ภายในเครื่อง



- **Outline** แยก Method, Object, Class ของไฟล์ ให้เข้าถึงได้ง่าย



- **Find & Replace** เครื่องมือช่วยเหลือในการ ค้นหา คำ แทนที่คำ หรือเข้าดูตำแหน่งของคำ



**สีเหลือง** : ทำการค้นหาคำที่ค้นหา แสดงผลออกมาในแต่ละบรรทัด สามารถ Direct ไปคำนั้นๆได้เลย

**สีเขียว** : ทำการ Clear Text ที่อยู่ภายใน TextField

**สีแดง** : ทำการ Replace คำทั้งหมด



## เอกสารอ้างอิง

Getting Started with JavaFX (September 2013). Gail Chappell. From [https://docs.oracle.com/javafx/2/get\\_started/jfxpub-get\\_started.htm](https://docs.oracle.com/javafx/2/get_started/jfxpub-get_started.htm)

RichtextFX (September 2013). Tomas Mikula. From <https://github.com/FXMisc/RichTextFX>

Java Introduction (14 March 2017). From <https://mindphp.com/คู่มือ/73-คืออะไร/2185-java-คืออะไร.html>

Java Platform Standard Edition 10 (1993) from <https://docs.oracle.com/javase/10/docs/api/overview-summary.html>

Eclipse JDT – Abstract Syntax Tree (AST) and the Java Model (23 June 2018) Lars Vogel, Simon ScholzFabian Pfaff (c). From <http://www.vogella.com/tutorials/EclipseJDT/article.html>

About SQLite (9 April 2000). Richard Hipp. From <https://www.sqlite.org/about.html>

JavaFX tutorial (April 2013). Monica Pawlan. From <https://www.tutorialspoint.com/javafx/>