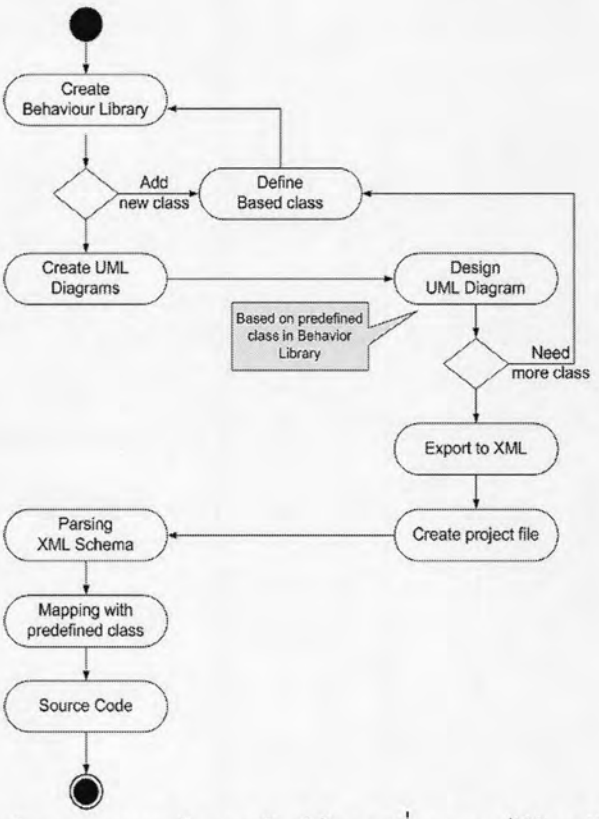
บทที่ 3
การออกแบบขั้นตอน และข้อกำหนดการสร้างชิ้นส่วนของงานวิจัย

ในบทนี้จะกล่าวถึงการออกแบบขั้นตอนการแปลงแผนภาพยูเอ็มแอลโดยใช้คลาสที่ออกแบบไว้ในคลังพฤติกรรม และข้อกำหนดการสร้างชิ้นส่วนของงานวิจัย เพื่อใช้เชื่อมโยงความหมายของแผนภาพยูเอ็มแอลเข้ากับส่วนปฏิบัติการของคลาสที่บันทึกไว้ในคลังพฤติกรรม ในการสร้างเป็นรหัสต้นฉบับ

3.1 การออกแบบขั้นตอนการแปลงแผนภาพยูเอ็มแอล โดยใช้คลาสที่ออกแบบไว้ในคลังพฤติกรรม

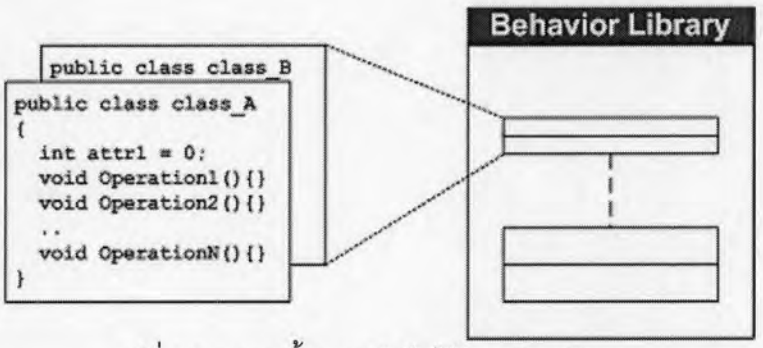
การออกแบบขั้นตอนการแปลงแผนภาพยูเอ็มแอล โดยใช้คลาสที่ออกแบบไว้ในคลังพฤติกรรม มีขั้นตอนการดำเนินการแสดงดังรูปที่ 3.1



รูปที่ 3.1 ขั้นตอนการแปลงแผนภาพยูเอ็มแอล โดยใช้คลาสที่ออกแบบไว้ในคลังพฤติกรรม

สามารถอธิบายลำดับขั้นตอนการดำเนินการได้ดังนี้

1. สร้างคลังพฤติกรรม เพื่อใช้เก็บคลาส ที่ต้องการใช้เป็นคลาสพื้นฐานสำหรับการออกแบบแผนภาพยูเอ็มแอล
2. ออกแบบคลาสพื้นฐานที่ใช้ในการสร้างแผนภาพยูเอ็มแอล แล้วบันทึกไว้ในคลังพฤติกรรม ดังรูปที่ 3.2



รูปที่ 3.2 คลาสพื้นฐานบันทึกไว้ในคลังพฤติกรรม

3. สร้างแผนภาพยูเอ็มแอล ประกอบด้วย แผนภาพคลาส แผนภาพสเตทชาร์ท และแผนภาพซีเควนซ์ จากคลาสพื้นฐานที่บันทึกอยู่ในคลังพฤติกรรม
4. ในกรณีที่ต้องการใช้คลาสพื้นฐานใหม่ ๆ นอกเหนือจากที่บันทึกในคลังพฤติกรรม ให้ดำเนินการออกแบบคลาสพื้นฐานใหม่ แล้วบันทึกเพิ่มเติมลงในคลังพฤติกรรม และกลับไปดำเนินการตามขั้นตอนการสร้างแผนภาพยูเอ็มแอลในขั้นตอนที่ 3 อีกครั้ง
5. เมื่อสร้างแผนภาพยูเอ็มแอลได้ครบตามความต้องการ ให้นำออกแผนภาพยูเอ็มแอลเป็นเอกสารเอกซ์เอ็มแอล
6. สร้างโพรเจกต์ไฟล์แล้วบันทึกรายชื่อแผนภาพยูเอ็มแอลที่ได้จากผลการดำเนินการในข้อ 5 เพื่อระบุรายการแผนภาพยูเอ็มแอลที่ต้องการใช้สำหรับโพรเจกต์นี้
7. วิเคราะห์แผนภาพยูเอ็มแอลในรูปแบบเอกสารเอกซ์เอ็มแอลที่บรรจุอยู่ในโพรเจกต์ และทำการจัดผังคำร่วมกับคลาสพื้นฐานที่ออกแบบไว้ในคลังพฤติกรรมเป็นรหัสต้นฉบับ

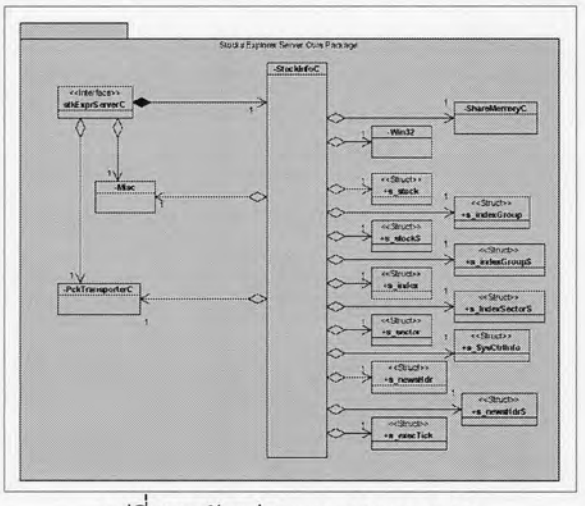
3.2 ข้อกำหนดการสร้างแผนภาพยูเอ็มแอล

แผนภาพยูเอ็มแอลที่ใช้สำหรับงานวิจัยนี้ สร้างด้วย โปรแกรมวิชวลพาราดาม คอมมูนิตี อิดิชัน รุ่น 6.0.2 [18] ประกอบไปด้วย แผนภาพแพกเกจ แผนภาพคลาส แผนภาพซีเควนซ์ และแผนภาพสเตทชาร์ท การกำหนดคุณสมบัติของแผนภาพจะต้องปฏิบัติตามข้อกำหนดดังนี้

3.2.1 ข้อกำหนดสำหรับแผนภาพแพกเกจ (Package Diagram) [19]

แผนภาพแพกเกจ แสดงตัวอย่างดังรูปที่ 3.3 ใช้เป็นแผนภาพที่แสดงรายการโปรแกรมต่อประสาน (Interface) ที่อ้างถึง เป็นส่วนที่แสดงรายการคลาสอินเตอร์เฟส (Class Interface) ที่ต้องการใช้ในการแปลโปรแกรม (Compile) ร่วมกับระบบงานที่สร้างขึ้น

สำหรับงานวิจัยนี้ กำหนดให้ระบบงานที่สร้างจากแผนภาพยูเอ็มแอล สามารถมีแผนภาพแพกเกจประกอบจำนวนหนึ่งแผนภาพ เพื่อระบุรายการแฟ้มข้อมูลที่บรรจุรายการชิ้นส่วนของคลาสอินเตอร์เฟสที่ใช้ภายในระบบงาน



รูปที่ 3.3 ตัวอย่างแผนภาพแพกเกจ

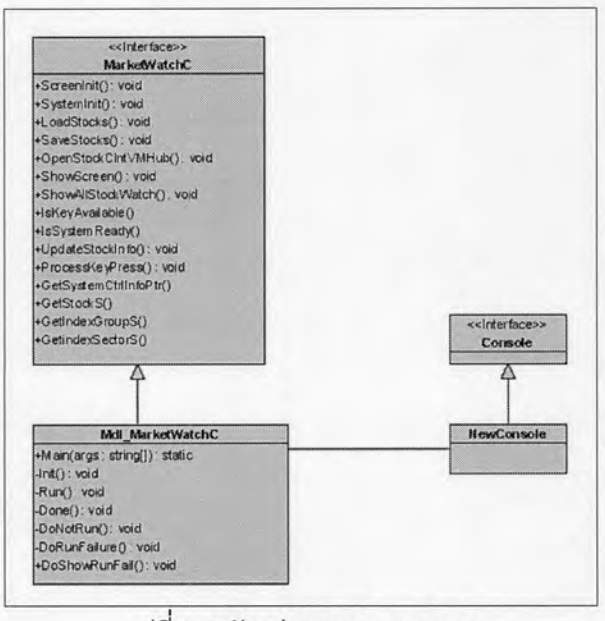
การใช้คลาสอินเตอร์เฟส สามารถสร้างด้วยสัญลักษณ์แผนภาพคลาส โดยกำหนดคุณสมบัติ ดังแสดงในตารางที่ 3.1

ตารางที่ 3.1 ตัวอย่างการกำหนดคุณสมบัติให้กับคลาสอินเตอร์เฟสสำหรับแผนภาพแพกเกจ

ชื่อคุณสมบัติ	รายละเอียด
Name	ระบุชื่อคลาสที่ใช้เป็นคลาสอินเตอร์เฟส
StereoTypes	ระบุเป็น Interface
ระบุข้อกำหนดคุณลักษณะ (Specification) ของ Tagged Values	
ชื่อคุณสมบัติ	รายละเอียด
Name	ระบุคำเฉพาะ SourceFile
Type	ระบุชนิดข้อมูลแบบข้อความ (Text)
Value	ระบุชื่อแฟ้มข้อมูลที่ใช้จัดเก็บคลาสอินเตอร์เฟสนี้

3.2.2 ข้อกำหนดสำหรับแผนภาพคลาส (Class Diagram) [19]

แผนภาพคลาส แสดงตัวอย่างดังรูปที่ 3.4 เป็นแผนภาพที่ใช้ระบุคลาส (Class) ที่สามารถใช้งานได้ภายในระบบงานสำหรับโพรเจกต์แต่ละโพรเจกต์ที่สร้างขึ้น

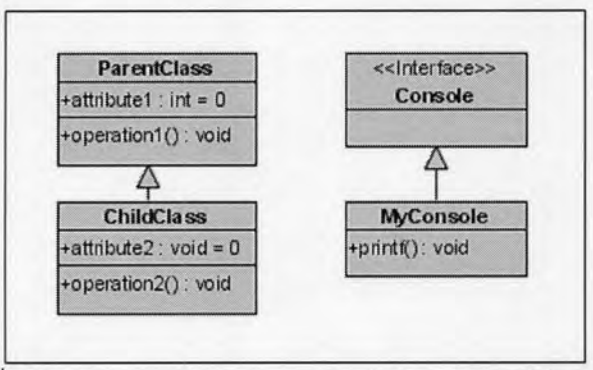


รูปที่ 3.4 ตัวอย่างแผนภาพคลาส

แผนภาพคลาส จะให้ความหมายเชิงโครงสร้าง กล่าวคือ ใช้อธิบายความสัมพันธ์ระหว่างคลาส และกำหนดรายการแอทริบิวต์ และโอเปอร์เรชันของคลาสที่แสดงอยู่ในแผนภาพ

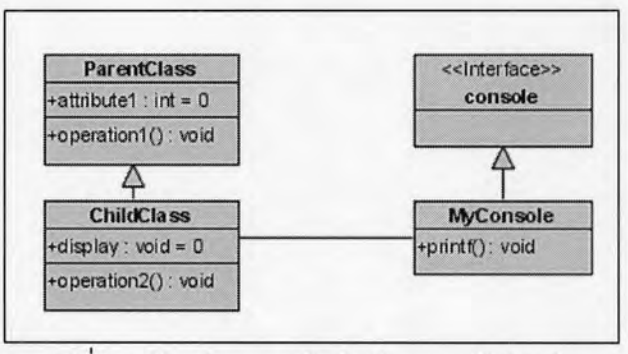
การสร้างแผนภาพคลาส มีข้อกำหนดในการสร้าง ดังนี้

- รูปสัญลักษณ์คลาส ต้องกำหนดรายการแอทริบิวต์ และรายการโอเปอร์เรชัน
- รูปสัญลักษณ์คลาส สามารถกำหนดคุณลักษณะเป็นชนิดคลาสอินเตอร์เฟส
- คลาสอินเตอร์เฟส ต้องไม่แสดงรายการแอทริบิวต์ และโอเปอร์ชัน
- ความสัมพันธ์แบบเจเนอรัลไลเซชัน (Generalization) เพื่อสืบทอดคุณสมบัติของคลาสแม่ แสดงดังรูป 3.5 สามารถใช้งานกับคลาสอินเตอร์เฟส



รูปที่ 3.5 ตัวอย่างการใช้งานความสัมพันธ์แบบเจเนอรัลไลเซชัน

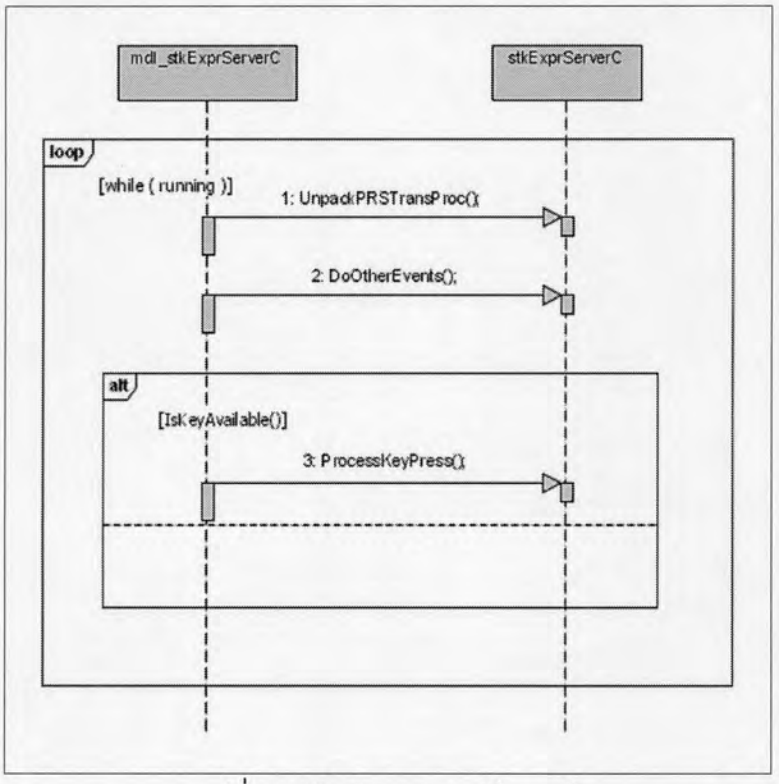
- ความสัมพันธ์แบบแอสโซซิเอชัน แสดงดังรูป 3.6 ใช้เพื่อเชื่อมโยงแสดงความสัมพันธ์ระหว่างคลาส



รูปที่ 3.6 ตัวอย่างความสัมพันธ์แบบแอสโซซิเอชัน

3.2.3 ข้อกำหนดสำหรับแผนภาพซีเควนซ์ (Sequence Diagram) [19]

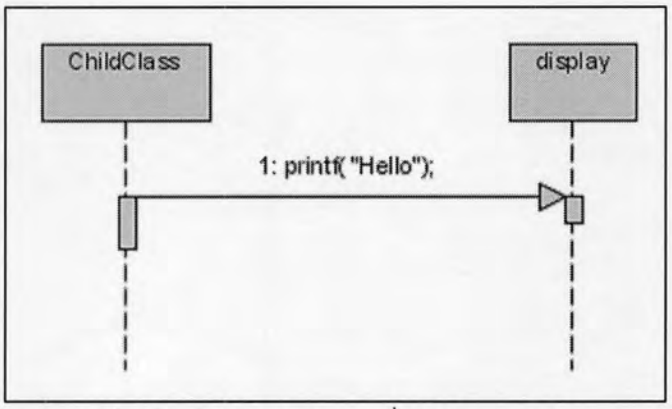
แผนภาพซีเควนซ์ แสดงตัวอย่างดังรูป 3.7 เป็นแผนภาพที่ แสดงการปฏิสัมพันธ์ ระหว่างคลาส โดยแสดงการติดต่อกันระหว่างคลาส ด้วยการส่งสาร จากคลาสหนึ่ง ไปอีกคลาสหนึ่ง ตามลำดับที่กำหนด ซึ่งใช้บรรยายกิจกรรมทั้งหมดที่เกิดขึ้นในสถานการณ์ (Scenario) หนึ่ง ๆ



รูปที่ 3.7 ตัวอย่างแผนภาพซีเควนซ์

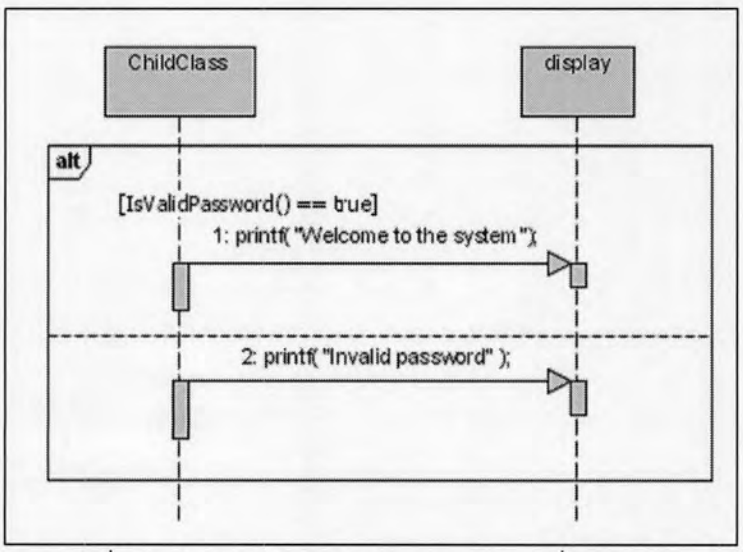
การสร้างแผนภาพซีเควนซ์ มีข้อกำหนดในการดังนี้

- คลาสที่ปรากฏในแผนภาพจะต้องเป็นคลาสที่ระบุอยู่ในแผนภาพคลาส ภายในโพรเจกต์ หรือระบบงานเดียวกันเท่านั้น
- คลาสที่ปรากฏในแผนภาพสามารถแสดงอยู่ในรูปของอินสแตนท์ที่มีชนิดเป็นคลาสที่ระบุในแผนภาพคลาส ดังรูป 3.8 โดยใช้สัญลักษณ์รูปลูกศรแทนการส่งสารจากคลาสที่เรียกใช้โอเปอร์เรชันไปยังคลาสหรือออปเจ็กต์ปลายทาง



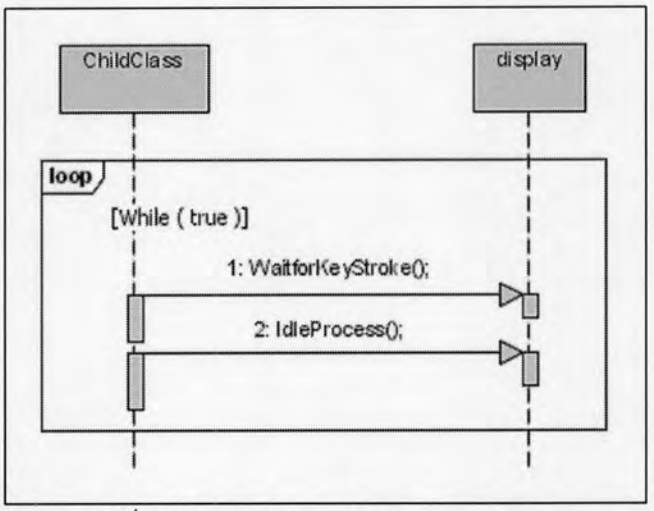
รูปที่ 3.8 ตัวอย่างการเรียกใช้ออปเจ็กต์ display ที่เป็นอินแสตนทของคลาส MyConsole

- สนับสนุนการใช้งานแบบมีขั้นตอนการตัดสินใจด้วยกรอบทางเลือก (Alternative Comblined Fragment Frame) ดังรูป 3.9 ด้วยประโยคเงื่อนไข IF..ELSE เพื่อตัดสินใจเลือกกระทำเมื่อเงื่อนไขที่กำหนดเป็นจริง หรือเลือกทางเลือกอื่นในกรณีที่เงื่อนไขเป็นเท็จ



รูปที่ 3.9 ตัวอย่างการกำหนดกรอบทางเลือกเพื่อตัดสินใจ

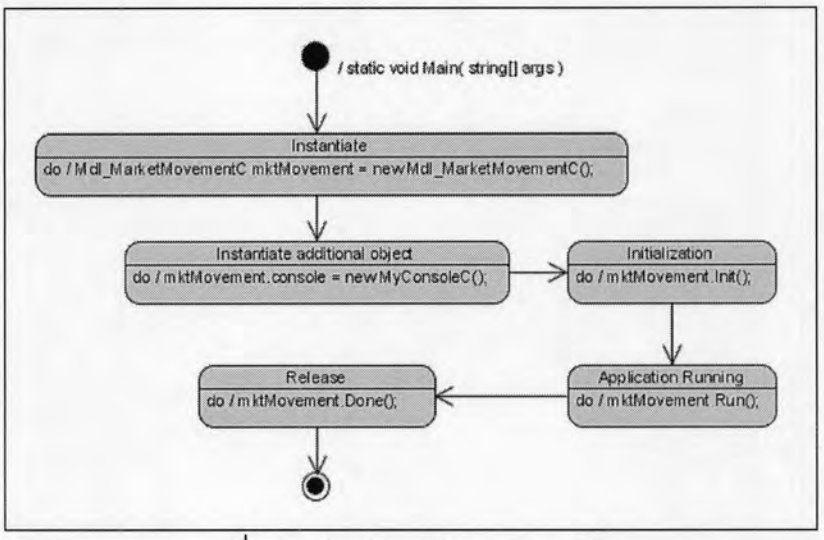
- สนับสนุนการใช้งานแบบมีกรอบวนรอบ (Loop) ดังรูปที่ 3.10 โดยกำหนดให้ใช้คำสั่งวนรอบ WHILE-LOOP หรือคำสั่งวนรอบ FOR-LOOP เพื่อกำหนดเงื่อนไขการดำเนินการวนรอบเท่านั้น



รูปที่ 3.10 ตัวอย่างการกำหนดกรอบวงรอบ

3.2.4 แผนภาพสเตทชาร์ท (Statechart Diagram) [19]

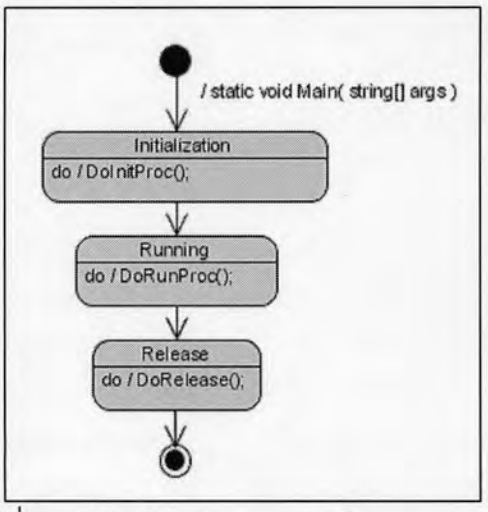
แผนภาพสเตทชาร์ท เป็นแผนภาพที่ใช้อธิบายลำดับการดำเนินการของโปรแกรม โดยกำหนดจุดเริ่มต้นการดำเนินงานของสเตทด้วยสัญลักษณ์สเตทเริ่มต้นต่อเชื่อมด้วยเส้นแทรนซิชันที่ระบุเส้นทางการดำเนินการด้วยหัวลูกศรชี้ไปยังสเตท ในลำดับการทำงานถัดไป และสิ้นสุดการดำเนินการโดยต่อเชื่อมเส้นแทรนซิชันไปยังสัญลักษณ์จุดสิ้นสุด แสดงตัวอย่างดังรูปที่ 3.11



รูปที่ 3.11 ตัวอย่างแผนภาพสเตทชาร์ท

การสร้างแผนภาพสเตทชาร์ท มีข้อกำหนดในการสร้างดังนี้

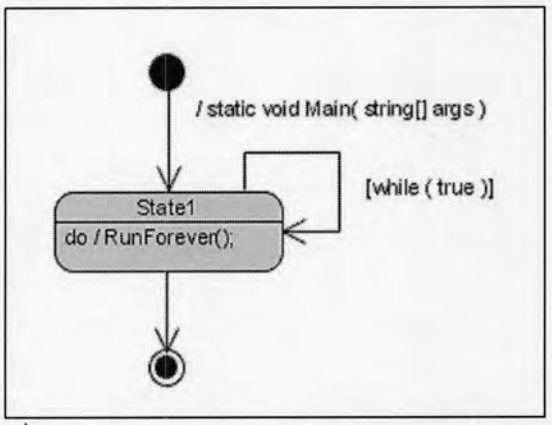
- แผนภาพสเตทชาร์ทจะต้องมีสัญลักษณ์สเตทเริ่มต้น เพื่อแสดงจุดเริ่มต้นการทำงานเสมอ โดยในแผนภาพหนึ่ง ๆ จะมีสัญลักษณ์สเตทเริ่มต้นได้เพียงตำแหน่งเดียว
- แผนภาพจะต้องมีสัญลักษณ์สเตทสุดท้าย เพื่อแสดงจุดสิ้นสุดการทำงานเสมอ โดยแผนภาพหนึ่ง ๆ จะสามารถมีสัญลักษณ์สเตทสุดท้าย เพื่อแสดงจุดสิ้นสุดได้หลายตำแหน่ง ขึ้นอยู่กับเส้นแทรนซิชันที่แตกย่อยด้วยเงื่อนไขที่กำหนดขึ้น
- แผนภาพใด ๆ ที่มีเส้นแทรนซิชันออกจากสเตทเริ่มต้น และระบุชื่อโอเปอร์เรชันในพารามิเตอร์ชื่อเอฟเฟค (Effect) จะหมายถึงการสร้างเมธอด หรือฟังก์ชัน ที่มีจุดสิ้นสุดของโอเปอร์เรชันที่ตำแหน่งสเตทสุดท้าย ดังรูปที่ 3.12



รูปที่ 3.12 ตัวอย่างรูปวงจรสมบูรณ์ของสเตทชาร์ท

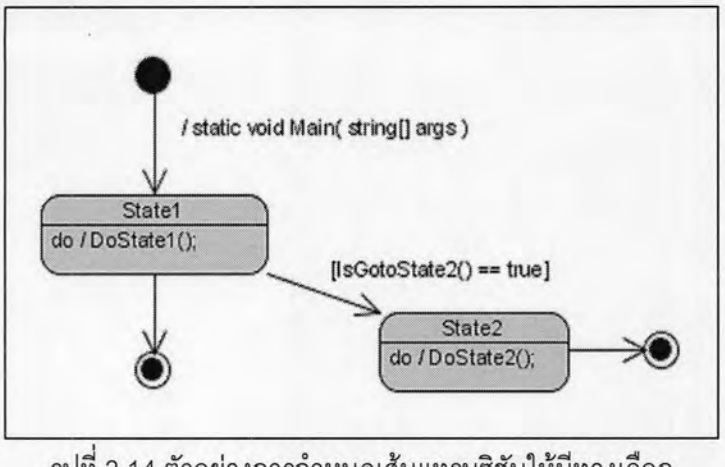
- การตีความเส้นแทรนซิชัน จะเริ่มต้นจากเส้นแทรนซิชันที่ชี้ตัวเองก่อนเสมอ
- การกำหนดเมธอดให้กับสเตทกำหนดให้ระบุในส่วนกำหนดพารามิเตอร์ชื่อ DoActivity ของสเตทนั้น ๆ โดยอนุญาตให้บรรจุคำสั่งได้มากกว่าหนึ่งคำสั่ง ด้วยการใช้เครื่องหมายอัฒภาค (semi-colon) คั่นชุดคำสั่งแต่ละคำสั่งออกจากกัน

- สัญลักษณ์เส้นแทรนซิชันชี้ตนเอง (Self-Transition) ในงานวิจัยนี้กำหนดให้เป็นการวนรอบสเตทด้วยเงื่อนไขวนรอบ WHILE-LOOP หรือวนรอบ FOR-LOOP ที่กำหนดให้เท่านั้น ดังรูปที่ 3.13



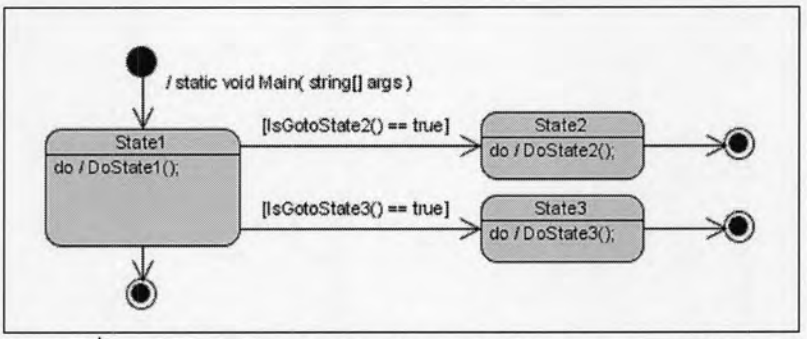
รูปที่ 3.13 ตัวอย่างรูปวงจรวนรอบชนิด Self-Transition

- สเตทใด ๆ ที่มีเส้นแทรนซิชันออกจากสเตทมากกว่าหนึ่งเส้น หรือมีหลายเส้นทางเลือก (Branching) หมายความว่ามีเงื่อนไขใด ๆ ก่อให้เกิดเส้นทางเลือกอื่นจะต้องระบุเงื่อนไขให้กับเส้นแทรนซิชันนั้น ๆ เสมอ
- การกำหนดเงื่อนไขให้กับเส้นแทรนซิชันสามารถกำหนดให้กับพารามิเตอร์ Guard เพื่อสร้างเงื่อนไขในการตัดสินใจ ดังรูปที่ 3.14 เงื่อนไขที่กำหนดให้กับเส้นแทรนซิชันนี้ไม่สนับสนุนการกำหนดแบบมีทางเลือกอื่น (Else-Condition)



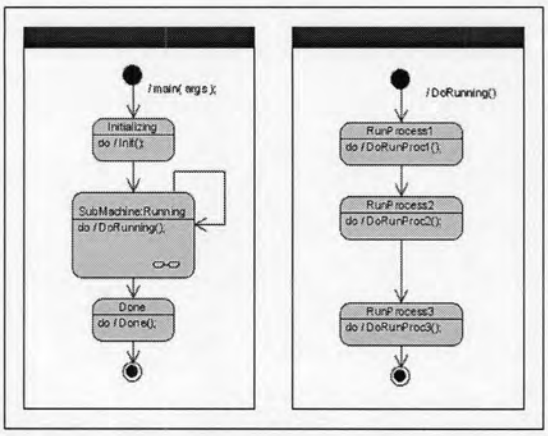
รูปที่ 3.14 ตัวอย่างการกำหนดเส้นแทรนซิชันให้มีทางเลือก

- เส้นแทรนซิชันสนับสนุนการทำงานแบบหลายทางเลือก (Multi-Branching) โดยกำหนดเงื่อนไขให้กับทุกทางเลือก ดังรูปที่ 3.15



รูปที่ 3.15 ตัวอย่างการกำหนดเส้นแทรนซิชันแบบหลายทางเลือก

- การตีความเมธอดที่เกิดขึ้นในสเตทจะครอบคลุมถึงเส้นแทรนซิชันที่มีเงื่อนไขประกอบ โดยเงื่อนไขดังกล่าวจะถูกตีความและบรรจุเป็นรหัสต้นฉบับอันดับสุดท้ายก่อนออกจากสเตท
- แผนภาพสเตทชาร์ทสนับสนุนการใช้งานร่วมกับสเตทย่อย (SubMachine State) โดยกำหนดให้สเตทย่อยเป็นตัวแทนของโปรแกรมย่อยที่แบ่งแยกออกจากสเตทหลัก การสร้างส่วนการทำงานของสเตทย่อยจึงกำหนดให้สร้างแผนภาพแยกออกจากแผนภาพสเตทชาร์ทหลัก ดังรูปที่ 3.16

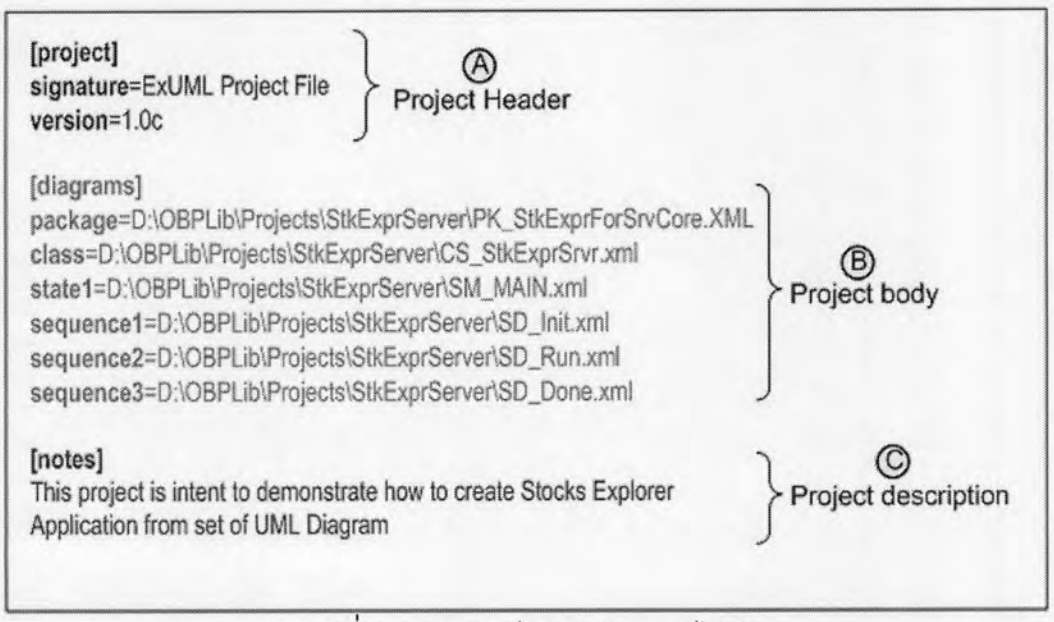


รูปที่ 3.16 ตัวอย่างการกำหนดสเตทย่อย (SubMachine State)

3.3 ข้อกำหนดการนำเข้าแผนภาพยูเอ็มแอล

การนำเข้าแผนภาพยูเอ็มแอลในรูปแบบเอกสารเอกซ์เอ็มแอลเพื่อแปลงเป็นรหัสต้นฉบับ จะต้องปฏิบัติตามข้อกำหนด ดังนี้

- สร้างโพรเจกต์ไฟล์ในรูปแบบเอกสารชนิดข้อความ (Text File) เพื่อบันทึกรายการแฟ้มเอกสารเอกซ์เอ็มแอลที่ต้องการใช้ในการแปลโปรแกรมร่วมกัน
- บันทึกรายการแฟ้มเอกสารเอกซ์เอ็มแอลไว้ในหัวข้อแผนภาพ (Diagram) ด้วยคำเฉพาะแต่ละชนิดของแผนภาพ ไว้ในโพรเจกต์ไฟล์ ดังรูปที่ 3.17



รูปที่ 3.17 รายละเอียดของโพรเจกต์ไฟล์

ตารางที่ 3.2 แสดงความหมาย และรายละเอียดของส่วนต่าง ๆ และป้ายระบุสำหรับโพรเจกต์ไฟล์ที่สร้างขึ้น

ตารางที่ 3.2 ความหมาย และรายละเอียดของป้ายระบุในโพรเจกต์ไฟล์

ส่วน	ป้ายระบุ	รายละเอียด
[project]		เป็นส่วนหัวของโพรเจกต์ (Project Header) สำหรับระบุลายเซ็น และเวอร์ชันของโพรเจกต์ไฟล์ ประกอบไปด้วยคำหลัก (keyword) 2 ส่วน คือ
	Signature	ใช้ระบุลายเซ็น ของโพรเจกต์ไฟล์ โดยต้องกำหนดเป็น "ExUML Project File"
	Version	ใช้ระบุเวอร์ชันของโพรเจกต์ไฟล์ ในปัจจุบันรองรับการทำงานด้วยไฟล์เวอร์ชัน 1.0c
[diagrams]		เป็นส่วนรายการแฟ้มข้อมูล (Project body) สำหรับระบุรายชื่อแผนภาพที่ใช้แปลโปรแกรมสำหรับโพรเจกต์นี้ ประกอบไปด้วย คำหลัก สำหรับแผนภาพ 4 แผนภาพ คือ
	package	ใช้ระบุชื่อแฟ้มข้อมูลเอกซ์เอ็มแอลชนิดแพกเกจ
	class	ใช้ระบุชื่อแฟ้มข้อมูลเอกซ์เอ็มแอลชนิดคลาส
	State*X*	ใช้ระบุชื่อแฟ้มข้อมูลเอกซ์เอ็มแอลชนิดสเตท โดย *X* คือลำดับของแฟ้มข้อมูลที่ใช้ โดยปกติในโพรเจกต์หนึ่ง จะมีแฟ้มข้อมูลสเตทเพียงแฟ้มเดียว ยกเว้นในกรณีที่มีการใช้งานสเตทย่อย
	Sequence*X*	ใช้ระบุชื่อแฟ้มข้อมูลเอกซ์เอ็มแอลชนิดซีเควนซ์ โดย *X* คือลำดับของแฟ้มข้อมูลที่ใช้ ในโพรเจกต์หนึ่ง ๆ สามารถมีแฟ้มข้อมูลซีเควนซ์ได้หลายแฟ้ม
[notes]		เป็นส่วนรายการหมายเหตุ (comment) หรือบันทึก (note) เพื่อใช้อธิบายจุดประสงค์ หรือเป้าหมายของโพรเจกต์

- บันทึกแฟ้มข้อมูลโพรเจกต์ไฟล์ด้วยนามสกุลอีเอกซ์พี (EXP : Executable UML Project File) เพื่อที่จะสามารถ นำไปแปลเป็นรหัสต้นฉบับ ด้วยโปรแกรมแปลงแผนภาพยูเอ็มแอล ตามวัตถุประสงค์ของงานวิจัยนี้