

บทที่ 5

หลักการใช้อยูเอ็มแอลเพื่อการออกแบบ

ภาษายูเอ็มหรือยูเอ็มแอล (Unified Modeling Language : UML) คือภาษาภาพหรือกราฟิก (graphical language) นักวิเคราะห์และออกแบบระบบใช้เป็นเครื่องมือช่วยในการวิเคราะห์และออกแบบซอฟต์แวร์เชิงวัตถุ เพื่อแสดงการทำงานของระบบด้วยไดอะแกรมหรือแผนภาพต่างๆ โดยไดอะแกรมมีหลายแบบซึ่งแต่ละแบบจะมีภาพความหมายและวัตถุประสงค์การใช้งานที่แตกต่างกัน ยูเอ็มแอลเป็นชุดปฏิบัติงานที่ได้รับการยอมรับจากทีมงานพัฒนาซอฟต์แวร์เชิงวัตถุว่าดีที่สุดในเฉพาะระบบงานขนาดใหญ่และมีความซับซ้อนเพราะสามารถนำมาสร้างแบบจำลองทางธุรกิจเพื่อเป็นต้นแบบในการพัฒนาระบบ นิยมเรียกยูเอ็มแอล มากกว่าภาษายูเอ็ม

5.1 ประวัติยูเอ็มแอล

ยูเอ็มแอลเป็นภาษาภาพที่เริ่มใช้ครั้งแรกในโรงงานอุตสาหกรรม ก่อนการพัฒนาให้เป็นยูเอ็มแอลมาตรฐานที่ใช้กับการพัฒนาระบบซอฟต์แวร์เชิงวัตถุ ยูเอ็มแอลในปัจจุบันถือกำเนิดมาจากบุคคล 3 ท่านคือ

- เกรดี บูช (Grady Booch) สร้างสัญกรณ์บูช (Booch notation)
- อีวา เจคอบสัน (Ivar Jacobson) สร้างเทคนิควิศวกรรมซอฟต์แวร์เชิงวัตถุ นิยมเรียกสั้นๆ ว่าโอโอเอสอี (Object Oriented Software Engineering : OOSE) สำหรับใช้ในบริษัทอีริคสัน (Eriksson)
- เจมส์ รัมบอช (James Rumbaugh) สร้างเทคนิคการสร้างแบบจำลองวัตถุ นิยมเรียกสั้นๆ ว่าโอเอ็มที (Object Modeling Technique : OMT) สำหรับใช้ในบริษัทเจเนอรัลอิเล็กทริก (General Electric)

ปี คศ.1994 อีวา เจคอบสัน และเกรดี บูช ได้มาร่วมงานกันและมีแนวคิดที่เป็นต้นกำเนิดของเมธอดเรียกว่าเมธอดยูนิฟาย “Unified Method” ต่อมาในปี คศ.1995 อีวา เจคอบสัน เข้ามาเพิ่มในกลุ่มอีก 1 คนรวมเป็น 3 คน โดย อีวา เจคอบสัน เพิ่มแนวคิดเกี่ยวกับเทคนิคโอโอเอสอีเพื่อผสานเข้าในเมธอด ซึ่งแนวคิดของโอโอเอสอีในยูเอ็มแอลคือ ยูสเคส (Use Case) ซึ่งเป็นหลักการสำคัญของยูเอ็มแอลในปัจจุบัน ต่อมากลุ่มการจัดการวัตถุหรือโอเอ็มจี (Object Management Group : OMG²) ซึ่งเป็นองค์กรมาตรฐานโรงงานที่จัดตั้งขึ้นโดยไม่หวังผลกำไร ดำเนินการจัดทำภาษาโมเดลมาตรฐานสำหรับโรงงาน และ

ได้รับการพัฒนาแบบต่อยอดจากโอเอ็มจีเป็นยูเอ็มแอล ปัจจุบันโอเอ็มจีเป็นเจ้าของยูเอ็มแอลสามารถดาวน์โหลดการใช้ได้ฟรีที่เว็บไซต์ <http://www.omg.org/> หรือดาวน์โหลดจากเว็บไซต์ <http://www.uml.org/>



ภาพที่ 5.1 ภาพจากเว็บไซต์ <http://www.uml.org/>

5.2 ภาษากราฟิกชื่อยูเอ็มแอล

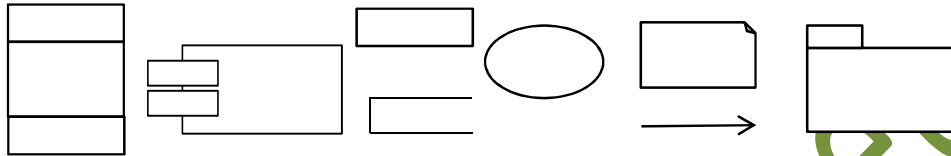
ภาษากราฟิกมีมาตั้งแต่โบราณอาจเป็นภาษาเฉพาะกลุ่ม หรือเป็นภาษาที่รู้จักกันโดยทั่วไป ในปัจจุบันภาษาภาพสากลเช่นภาพที่มีลักษณะผู้หญิง แทนห้องน้ำหญิง ภาพที่มีลักษณะผู้ชาย แทนห้องน้ำชาย ในหมู่ผู้ขับซัปรถยนต์จะมีภาพต่างๆ ที่สื่อความหมายเพื่อการขับซัปรถยนต์ เช่นป้ายห้ามจอด ป้ายยูเทิร์น ป้ายห้ามเลี้ยว เป็นต้น แม้กระทั่งป้ายที่เห็นได้ทั่วไป เช่นป้ายห้ามสูบบุหรี่ ป้ายร้านอาหาร ป้ายร้านกาแฟ ป้ายห้ามใช้เสียง ป้ายห้ามใช้โทรศัพท์ ป้ายห้ามถ่ายรูป จะเห็นได้ว่าป้ายทุกป้ายที่กล่าวมาเป็นป้ายมาตรฐานที่สื่อความหมายให้เข้าใจตรงกัน แม้กระทั่งแปลนบ้านที่นิยมเรียกกันว่าพิมพ์เขียวซึ่งมีสัญลักษณ์แสดงตำแหน่งประตู บันได หรืออื่นๆ ที่ผู้สร้างบ้านมีความเข้าใจตรงกัน สามารถนำไปเป็นแบบสำหรับสร้างบ้านที่ต้องการได้ ภาษากราฟิกยูเอ็มแอลมาตรฐานก็มีลักษณะเช่นเดียวกันคือมีภาพและสัญลักษณ์สำหรับสื่อความหมายสำหรับใช้ในการพัฒนาและออกแบบระบบซึ่งนักพัฒนาซอฟต์แวร์เชิงวัตถุต้องศึกษาถึงความหมายของสัญลักษณ์ต่างๆ เพื่อการใช้งานที่ถูกต้องตรงตามมาตรฐานและตรงความต้องการ

สิ่งสำคัญเบื้องต้นเกี่ยวกับการเรียนรู้ยูเอ็มแอลเพื่อให้สามารถนำยูเอ็มแอลไปใช้งานได้โดยทั่วไปนั้นควรทราบเกี่ยวกับ 4 องค์ประกอบหลัก คือ

(1) องค์ประกอบพื้นฐานของบล็อก (block) และยูเอ็มแอลแต่ละประเภทมีการใช้งานอย่างไร

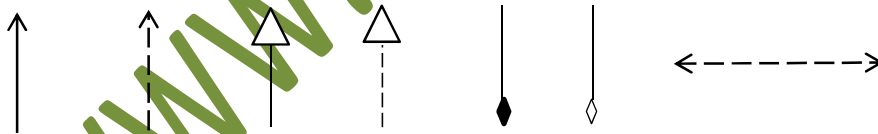
การสร้างบล็อกยูเอ็มแอล มีรายละเอียดสำหรับส่วนประกอบของการสร้างที่เหมือนกันคือ

(1.1) ส่วนของแบบ (things) ของบล็อก ซึ่งเป็นส่วนสำคัญที่สุด โดยส่วนประกอบของแบบสามารถเป็น แบบโครงสร้าง (structure things) แบบพฤติกรรม (behavior things) แบบการจัดกลุ่ม (grouping things) และแบบการเขียนหมายเหตุประกอบ (annotation things)



ภาพที่ 5.2 ตัวอย่างกราฟิกแสดงส่วนของแบบ

(1.2) ส่วนของความสัมพันธ์ เป็นส่วนที่แสดงความสัมพันธ์ขององค์ประกอบต่างๆ โดยความสัมพันธ์ที่แสดงนั้นจะแสดงให้เห็นถึงการทำงานของซอฟต์แวร์ที่ต้องการพัฒนา เช่นความสัมพันธ์การรวมกลุ่ม (association) ที่เป็นความสัมพันธ์ในแบบภาพรวมกลุ่ม (aggregation) ซึ่งเป็นความสัมพันธ์แบบเป็นส่วนหนึ่งของสามารถแยกออกจากกันได้ หรือความสัมพันธ์ในแบบเกี่ยวข้งกัน (composition) ไม่สามารถแยกออกจากกันได้ ความสัมพันธ์ในแบบขึ้นอยู่กับ (dependency) ความสัมพันธ์ในแบบสืบทอดคุณสมบัติ (generalization)



ภาพที่ 5.3 ตัวอย่างกราฟิกแสดงส่วนของความสัมพันธ์

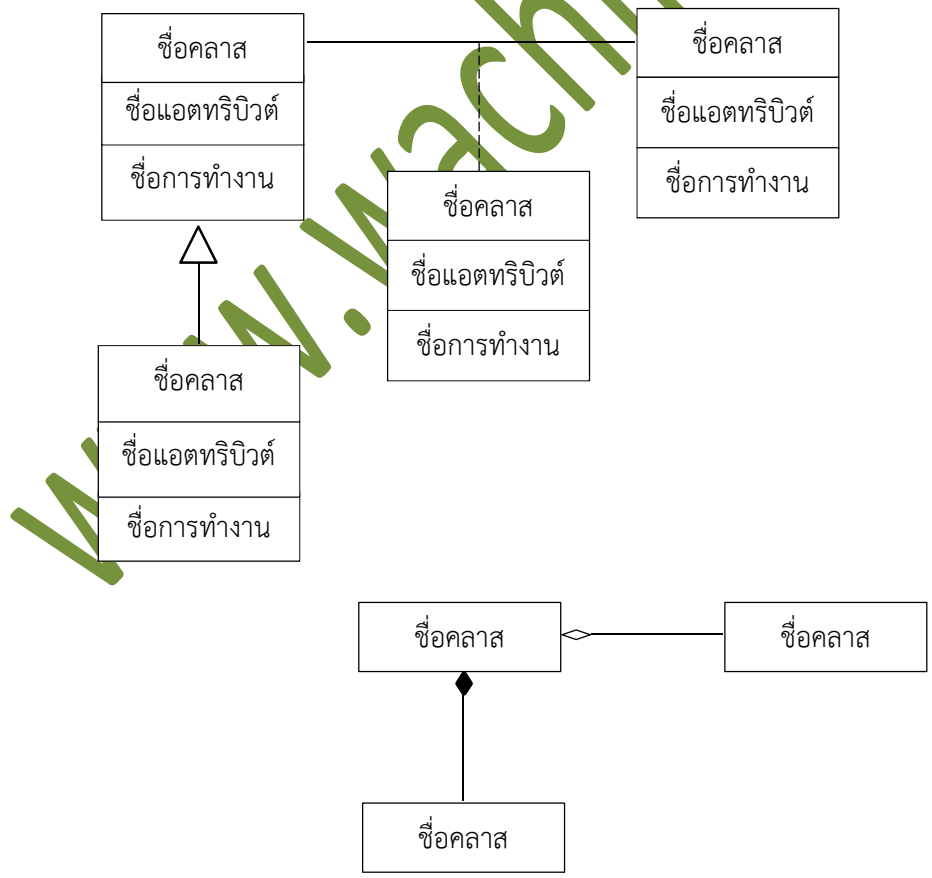
(1.3) ไดอะแกรมหรือแผนภาพ จะเป็นส่วนแสดงการทำงานของระบบ โดยแสดงให้เห็นถึงความสัมพันธ์ของทุกองค์ประกอบทั้งซอฟต์แวร์และฮาร์ดแวร์

- (2) การเชื่อมต่อบล็อกแต่ละแบบเพื่อการใช้งาน
- (3) ระบบที่ต้องการออกแบบและความเชื่อมต่อกับยูเอ็มแอล
- (4) การแปลงระบบที่ออกแบบด้วยยูเอ็มแอลเป็นซอฟต์แวร์เชิงวัตถุ

ถึงแม้ว่ายูเอ็มแอลจะเป็นภาษามาตรฐานก็ตาม แต่ปัจจุบันมีการพัฒนายูเอ็มแอลหลายรุ่น ซึ่งแต่ละรุ่นมีพื้นฐานหลักการสร้างจะเหมือนกันแต่มีความแตกต่างย่อยในรายละเอียด ในที่นี้ขอเสนอเกี่ยวกับยูเอ็มแอล 2.0 ตามมาตรฐานของโอเอ็มจีซึ่งแบ่งไดอะแกรมออกเป็น 3 ประเภท โดยแต่ละประเภทแบ่งเป็นไดอะแกรมดังนี้

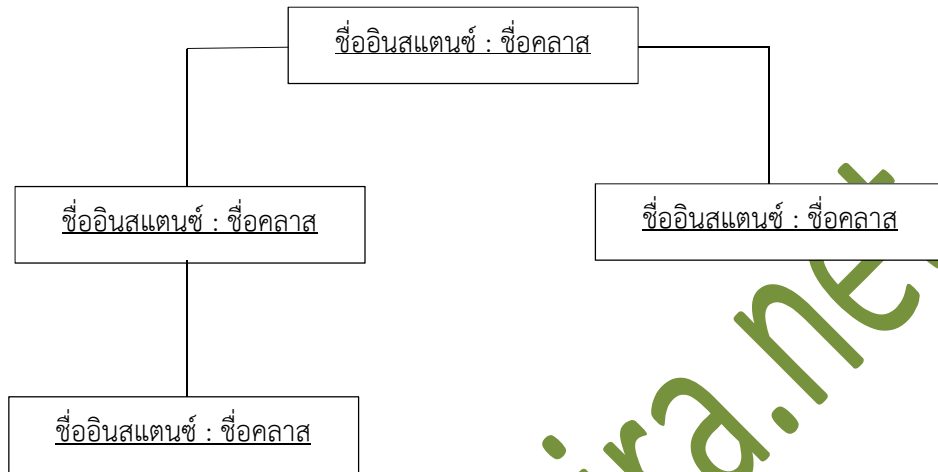
(1) แผนภาพโครงสร้างหรือสตรัคเจอร์ไดอะแกรม (Structure Diagrams) เป็นไดอะแกรมที่แสดงโครงสร้างหลักหรือส่วนคงที่ในระบบ แบ่งเป็นไดอะแกรมจำนวน 6 ไดอะแกรมคือ

(1.1) แผนภาพคลาสหรือคลาสไดอะแกรม (Class Diagram) ใช้สำหรับพัฒนาระบบ เป็นการแสดงคลาสของระบบและความสัมพันธ์แบบต่างๆ เช่น แอสโซซิเอชันหรือการเชื่อมต่อระหว่าง 2 คลาส อินเฮอริเทนซ์หรือการถ่ายทอด แอครีเกชันหรือการรวมตัว ซึ่งเป็นการแสดงมุมมองของระบบในแบบวัตถุโดยแสดงถึงคุณลักษณะและการดำเนินงานรวมไปถึงแสดงความสัมพันธ์ของวัตถุ



ภาพที่ 5.4 ตัวอย่างแผนภาพคลาสหรือคลาสไดอะแกรม

(1.2) แผนภาพวัตถุหรือออบเจกต์ไดอะแกรม (Object Diagram) ใช้สำหรับอธิบายอินสแตนซ์ของคลาสไดอะแกรม การใช้คลาสไดอะแกรมคล้ายกับการใช้ออบเจกต์ไดอะแกรมแต่ใช้เพื่อสร้างต้นแบบ (prototype) ของระบบในมุมมองการปฏิบัติ (practical perspective)



ภาพที่ 5.5 ตัวอย่างแผนภาพวัตถุหรือออบเจกต์ไดอะแกรม

(1.3) แผนภาพส่วนประกอบหรือคอมโพเนนต์ไดอะแกรม (Component Diagram) เป็นไดอะแกรมมุมมองการดำเนินการของระบบสำหรับแทนกลุ่มขององค์ประกอบเช่น คลาส อินเตอร์เฟส หรือ ความร่วมมือ (collaborations) และความสัมพันธ์

ภาพที่ 5.6 ตัวอย่างแผนภาพวัตถุหรือออบเจกต์ไดอะแกรม

ที่มา : <http://www.agilemodeling.com/artifacts/componentDiagram.htm>, 2556

(1.4) แผนภาพโครงสร้างคอมโพสิทหรือคอมโพสิทสตรัคเจอร์ไดอะแกรม (Composite Structure Diagram)

(1.5) แผนภาพแพคเกจหรือแพคเกจไดอะแกรม (Package Diagram)

....

www.wachira.net

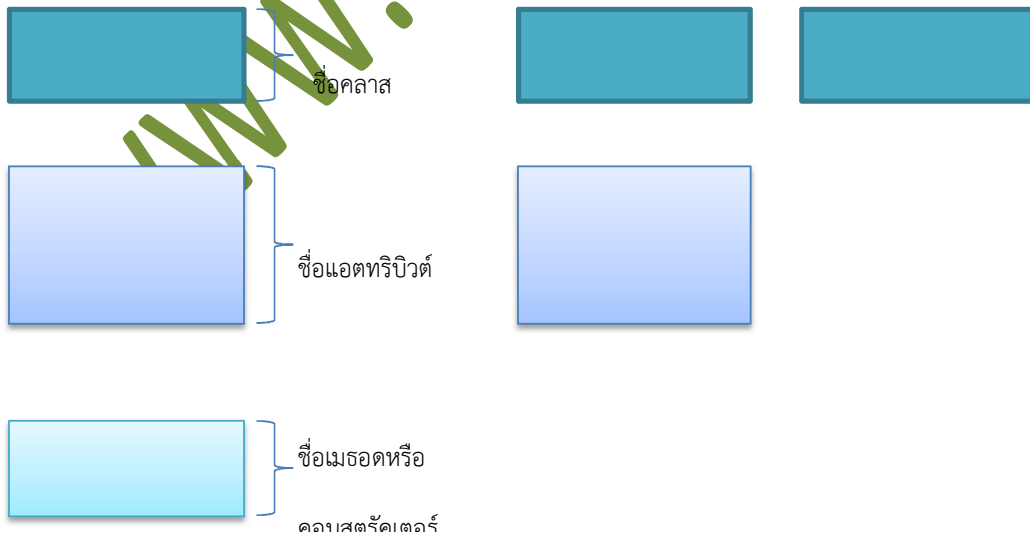
5.3 คลาสไดอะแกรม

คลาสไดอะแกรม (class diagrams) เป็นไดอะแกรมสำหรับแทนคลาสในการพัฒนาซอฟต์แวร์เชิงวัตถุซึ่งเป็นไดอะแกรมที่พบบ่อยที่สุดในการเขียนยูเอ็มแอล เนื่องจากเป็นมุมมองเชิงวัตถุที่ใกล้เคียงกับความเป็นจริงมากที่สุดโดยแสดงคลาส อินเทอร์เฟซ ความร่วมมือและความสัมพันธ์

....วิเคราะห์และออกแบบ...

5.4.1 การเขียนคลาสไดอะแกรม

การเขียนคลาสไดอะแกรมเต็มรูปแบบจะประกอบด้วย 3 ส่วน ซึ่งอาจเขียนเพียง 2 ส่วนแรก หรือ 1 ส่วนแรก ตามวัตถุประสงค์ในการใช้งาน โดยส่วนที่ 1 คือส่วนของชื่อคลาส ส่วนที่ 2 คือส่วนของแอดทริบิวต์ของชื่อคลาสและสัญลักษณ์เพื่อแสดงการเข้าถึง (visibility) แอดทริบิวต์ ส่วนที่ 3 คือส่วนของเมธอดหรือคอนสตรัคเตอร์ของคลาสและสัญลักษณ์เพื่อแสดงการเข้าถึงเมธอดหรือคอนสตรัคเตอร์



ก.

ข.

ค.

ภาพที่ 5.7 ตัวอย่างโครงสร้างคลาสไดอะแกรม

- ก. คลาสไดอะแกรม 3 ส่วน
- ข. คลาสไดอะแกรม 2 ส่วนแรก
- ค. คลาสไดอะแกรม 1 ส่วนแรก

จากภาพที่ 5.5 ขนาดและความกว้างของแต่ละส่วนขึ้นกับข้อมูลที่อยู่ภายในของแต่ละส่วน (คลาสไดอะแกรมของยูเอ็มแอลไม่มีสีในที่นี้ใช้สีเพื่อให้ง่ายต่อการแยกความแตกต่างของแต่ละส่วน) ในส่วนของชื่อแอตทริบิวต์และชื่อเมธอดหรือคอนสตรัคเตอร์ จะมีสัญลักษณ์ - + # ~ หน้าส่วนของชื่อตามประเภทโดย - คือ private + คือ public # คือ protected. ส่วน ~ คือ package ซึ่งจะอยู่ในส่วนของชื่อเมธอดหรือคอนสตรัคเตอร์ (กรณีไม่ระบุจะมีค่าปกติคือ -)

รูปแบบการใช้งานแอตทริบิวต์ คือ

การเข้าถึง แอตทริบิวต์ : ชนิด

ตัวอย่าง

code : string

+ name : string

รูปแบบการใช้งานเมธอดหรือคอนสตรัคเตอร์ คือ

การเข้าถึง เมธอด (รายการพารามิเตอร์) : ชนิดการคืนกลับของนิพจน์

ตัวอย่าง

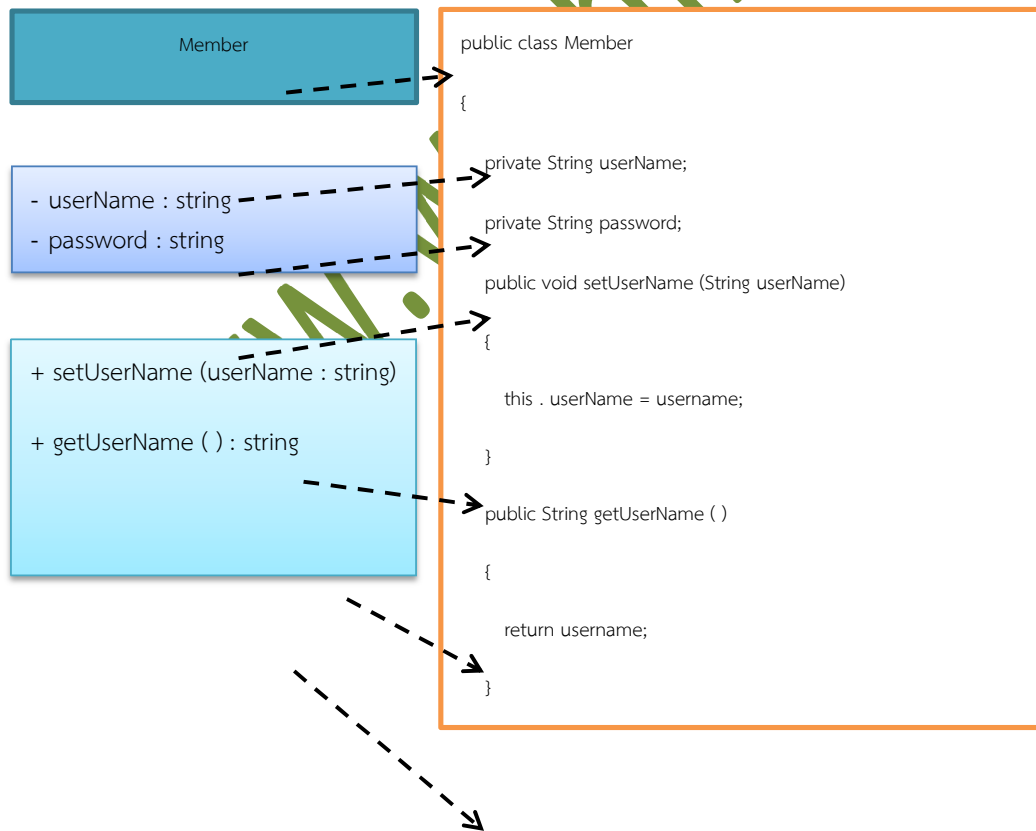
+ met1 (x : b) : double

+ setUsername (userName : string)

+ getUsername () : string

ตารางที่ 5.1 สัญลักษณ์แสดงการเข้าถึงแอตทริบิวต์หรือเมธอด ของคลาสไดอะแกรม

สัญลักษณ์	การเข้าถึง	ความหมาย	ตัวอย่าง
+	public		<div style="border: 1px solid black; padding: 5px;"> ClassName +att1 : type +att2 : type - m1() + m2(): type </div>
-	private		
#	protected		
~	package		



5.4.2 ความสัมพันธ์ของคลาสไดอะแกรม

คลาสของระบบ...

ความสัมพันธ์ระหว่างคลาส

(1) Association เป็นความสัมพันธ์ระหว่างคลาสตั้งแต่ 2 คลาส กรณีคลาสหนึ่งคลาสเป็น instance ของคลาสอีกคลาสหนึ่ง...

(2) Aggregation เป็นการแสดงความสัมพันธ์คลาสที่เป็นส่วนประกอบของคลาสอื่น

(3) Generalization เป็นการแสดงความสัมพันธ์ระหว่างซูเปอร์คลาสและสับคลาส

การแสดง multiplicities

(1) 0..1

(2) 0..*

(3) 1..5

(4) 1..*

(5) 4..*,

5.4.3

Instance diagram

instanceName:className

5.5 อินแอริแทนซ์ไดอะแกรม

Jung Woo UML Diagrams

<http://www.csee.umbc.edu/courses/undergraduate/CMSC445/Fall06/UML%20Diagrams.ppt>

- <http://www.agilemodeling.com/>
- <http://www.visual-paradigm.com/VPGallery/diagrams/index.html>
- <http://bdn.borland.com/article/0,1410,31863,00.html>
- http://en.wikipedia.org/wiki/Unified_Modeling_Language
- http://pigseye.kennesaw.edu/~dbraun/csis4650/A&D/UML_tutorial/index.htm

บรรณานุกรม

Scott, W. Ambler. (2004). **UML 2 Component Diagrams**. Retrieved May 16, 2013, from <http://www.agilemodeling.com/artifacts/componentDiagram.htm>.

ทีมนักวิจัยสถาบันนวัตกรรมเทคโนโลยีสารสนเทศคอมพิวเตอร์
มหาวิทยาลัยเกษตรศาสตร์. (2552). **การออกแบบระบบงานและการเชื่อมโยง
แลกเปลี่ยนข้อมูลด้วย UML** วันที่ 11-12 พฤศจิกายน 2552 (หน้า 1-128)
กรุงเทพฯ : มหาวิทยาลัยเกษตร.
[http://egif.mict.go.th/phase3/document/20091111-
%20course4%20v1.1.pdf](http://egif.mict.go.th/phase3/document/20091111-%20course4%20v1.1.pdf)

แนวคิดเชิงวัตถุ [http://cit.snru.ac.th/UserFiles/File/OOAD1_51/Unit02\(1\).pdf](http://cit.snru.ac.th/UserFiles/File/OOAD1_51/Unit02(1).pdf)

www.wachira.net