Aggregation Abstraction

Week 05

Object-Oriented Analysis and Design

วัตถุประสงค์

- อธิบายหลักการแยกและประกอบคลาสด้วยวิธีการ Aggregation Association ได้
- อธิบายและใช้งาน Cardinality, Required และ Optional Components ได้

ทบทวนของเก่า

- ครั้งที่แล้ว เราได้ศึกษาเรื่อง Classification Abstraction ไปแล้ว
  - เป็นกระบวนการสร้าง Class จาก Object ต่าง ๆ ที่มีอยู่ใน Problem Domain จนได้ Class
  - ยังไม่มีความสัมพันธ์ระหว่าง Class ต่าง ๆ เหล่านั้น
  - ในบางครั้ง class หนึ่งๆ อาจประกอบด้วย Class อื่นๆ หลายคลาส
  - ในการทำ Classification เราจะเขียนลูกศรเส้นประ โดยมีทิศทางจากวัตถุไปหาคลาส

Aggregation Abstraction

- ในโลกความจริง วัตถุจะเกิดจากการประกอบกันเข้าของวัตถุหลายๆ ชนิด
  - ประกอบแบบไม่สามารถแยกชิ้นส่วน (มาใช้งาน) ได้
    - เช่น คอนกรีต ประกอบด้วยหิน ทราย ปูนซิเมนต์ และ น้ำ (เราไม่สามารถเอาปูนซิเมนต์ออกมาจากคอนกรีต เพื่อใช้งานใหม่ได้)
  - ประกอบแบบแยกชิ้นส่วน (มาใช้งาน) ได้
    - เช่น โคมไฟ ประกอบหลอดไฟ สวิตช์ สายไฟ สตาร์ทเตอร์ บัลลาสต์ (เราสามารถแยกส่วนประกอบต่างๆ ไปใส่ในโคมไฟอื่น หรือนำไปใช้ที่อื่นได้ หากมีขนาดเท่ากัน)

กรณีศึกษา

- ให้นักศึกษา List วัตถุที่เกิดจากการรวมกันของวัตถุอื่น
  - แบบแยกส่วนนำมาใช้ใหม่ได้
  - แบบไม่สามารถแยกส่วนนำมาใช้ใหม่ได้

Concept ของวัตถุแบบ Aggregation

- เมื่อนำ Class มาประกอบกันแบบ Aggregation จะทำให้เกิด Concept ที่ต่างออกไปแก่ Class ใหม่
  - การนำ ทราย หิน ปูน น้ำ มาประกอบเป็นคอนกรีต จะได้วัตถุที่มี Concept ต่างไปโดยสิ้นเชิง
  - การนำ โต๊ะ เก้าอี้ กระดาน มาประกอบเป็นห้องเรียน จะต่างจากการนำโต๊ะ เก้าอี้ มาประกอบกันเป็นห้องรับประทานอาหาร

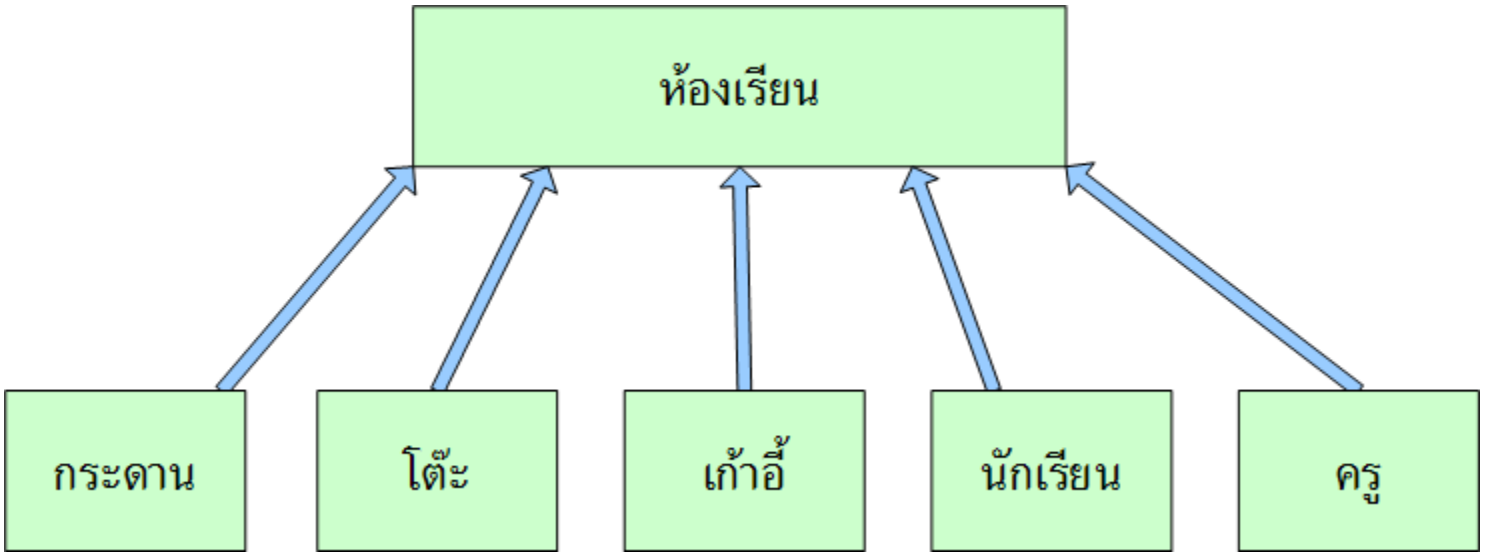
Composition VS. Decomposition

- Composition : การนำ Class มาประกอบกันเพื่อให้ได้ Class ใหม่ตาม Concept ที่กำหนด
  - การนำ ล้อรถ เครื่องยนต์ ตัวถัง ระบบขับเคลื่อน มารวมกัน จะทำให้ได้คลาส รถยนต์
- Decomposition : การจำแนก Class เพื่อให้รู้ว่า Class ที่มี Concept นั้น ประกอบด้วยคลาสอะไรบ้าง
  - เช่น เมื่อกำหนด Concept ของรถยนต์ เราก็จะทราบว่าควรมีล้อ เครื่องยนต์ ฯลฯ

ตัวอย่าง 4.1

- “ห้องเรียนประกอบไปด้วย กระดานดำ 1 กระดาน มีเก้าอี้และโต๊ะจำนวนหนึ่ง มีนักศึกษา มีอาจารย์”
- จากข้อความข้างต้น สามารถสรุปได้ว่า class กระดานดำ class โต๊ะ class เก้าอี้ class นักศึกษา class อาจารย์ เมื่อนำมารวมกันจะได้ class ใหม่ คือ class ห้องเรียน (Concept ต่างไปจากเดิม)

Composition



Decomposition

- Class ห้องเรียนสามารถแบ่งออกได้เป็น
  - Class กระดานดำ
  - Class โต๊ะ
  - Class เก้าอี้
  - Class นักศึกษา
  - Class อาจารย์

Decomposition

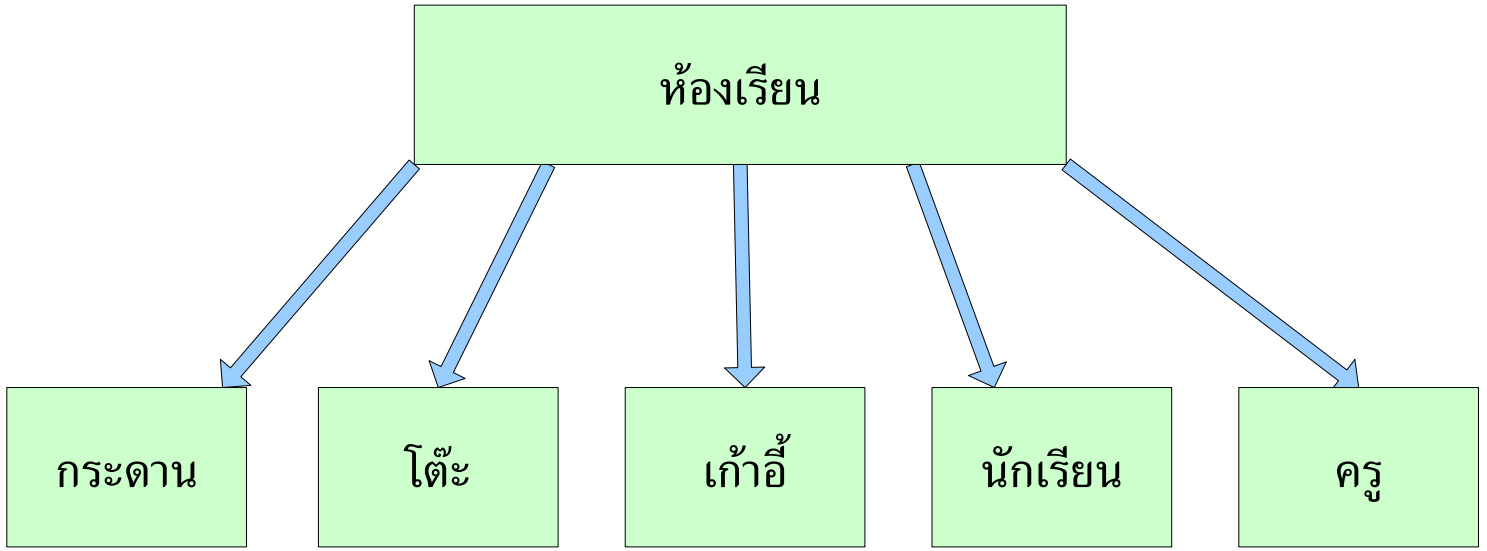
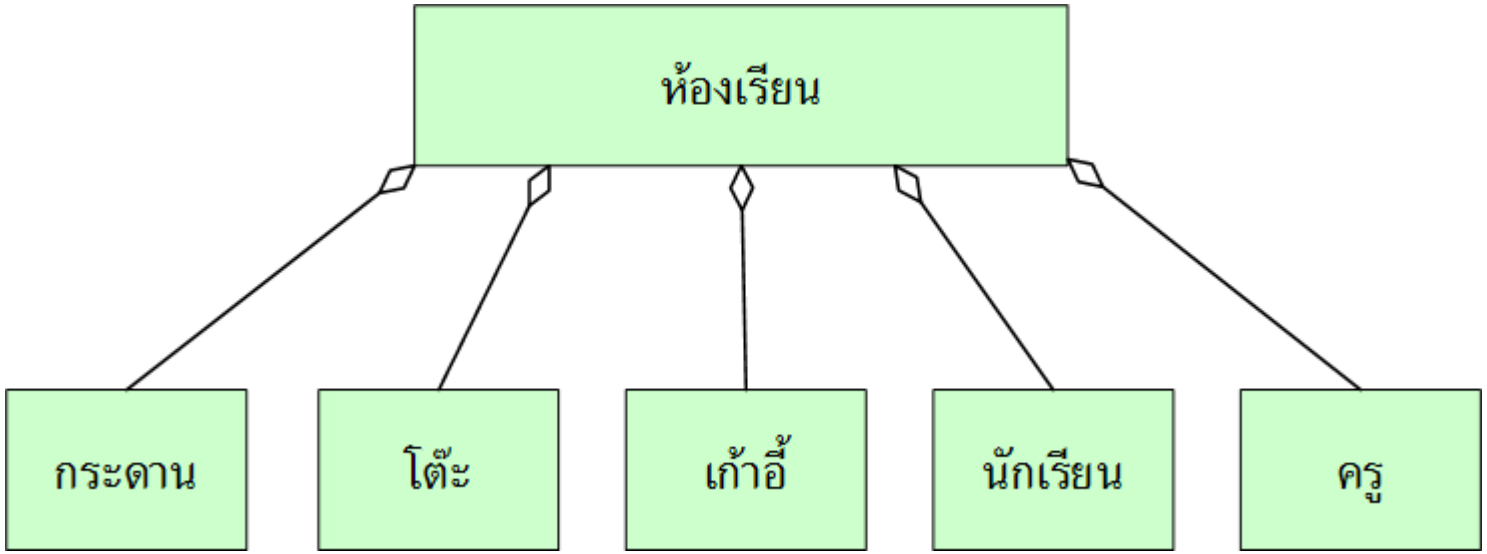


diagram ของ Aggregation Abstraction

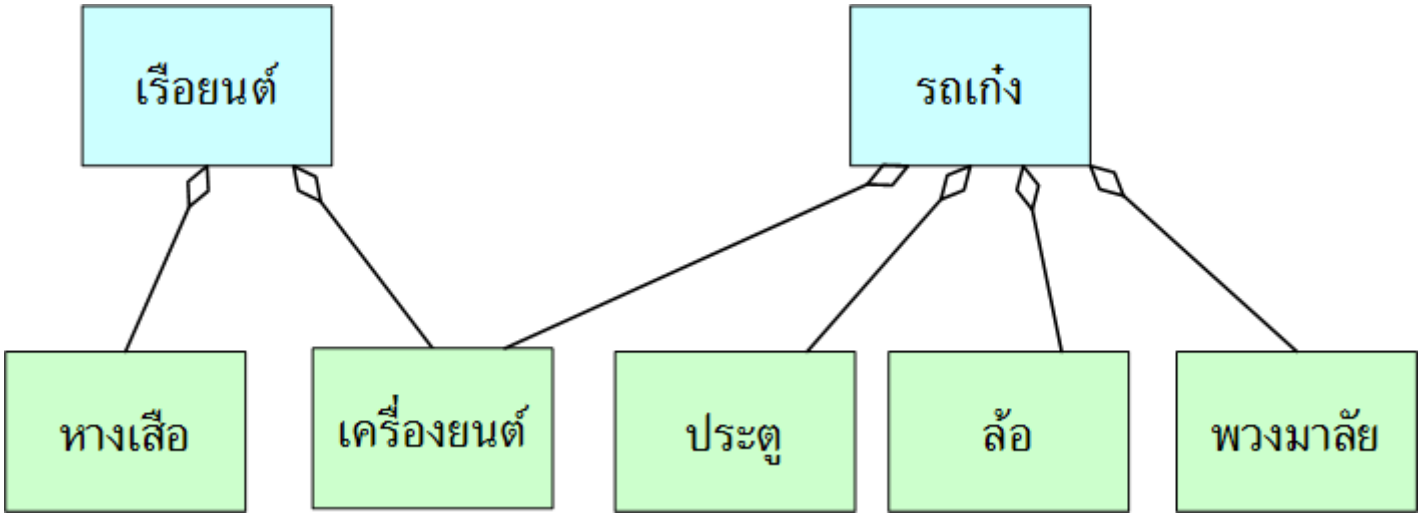
- ใช้เส้นตรงที่มีหัวสี่เหลี่ยมขนมเปียกปูน

  - ลากจาก Class ย่อย ไปยัง Class หลัก



Advances Aggregation Abstraction

- อาจมี Class ที่เป็น Class ย่อยของหลายๆ คลาสใหญ่ซึ่งมี Concept ต่างกัน



สรุป Aggregation Abstraction

- คือ การ พยายามตอบคำถามที่ว่า มี class ใดเป็นส่วนประกอบ (Is part of) ของ class อื่นหรือไม่ และที่สำคัญ “การประกอบกันของ class ต้องทำให้เกิด class ใหม่ ซึ่งมี concept ใหม่ด้วย”
- ในทาง object orientation นั้น การแสดงสัญลักษณ์เพื่อแสดง Aggregation Abstraction ของ class นั้น ทำได้โดยการโยงลูกศรเป็นสี่เหลี่ยมขนมเปียกปูน จาก class ย่อยหรือ class ที่เป็นส่วนประกอบ (Composite class) ไปยัง Class หลัก (Main Class)

กิจกรรม

- ให้นักศึกษาวาดแผนภาพ Composition ของเครื่องคอมพิวเตอร์
- ให้นักศึกษาวาดแผนภาพ Decomposition ของหนังสือ 1 เล่ม

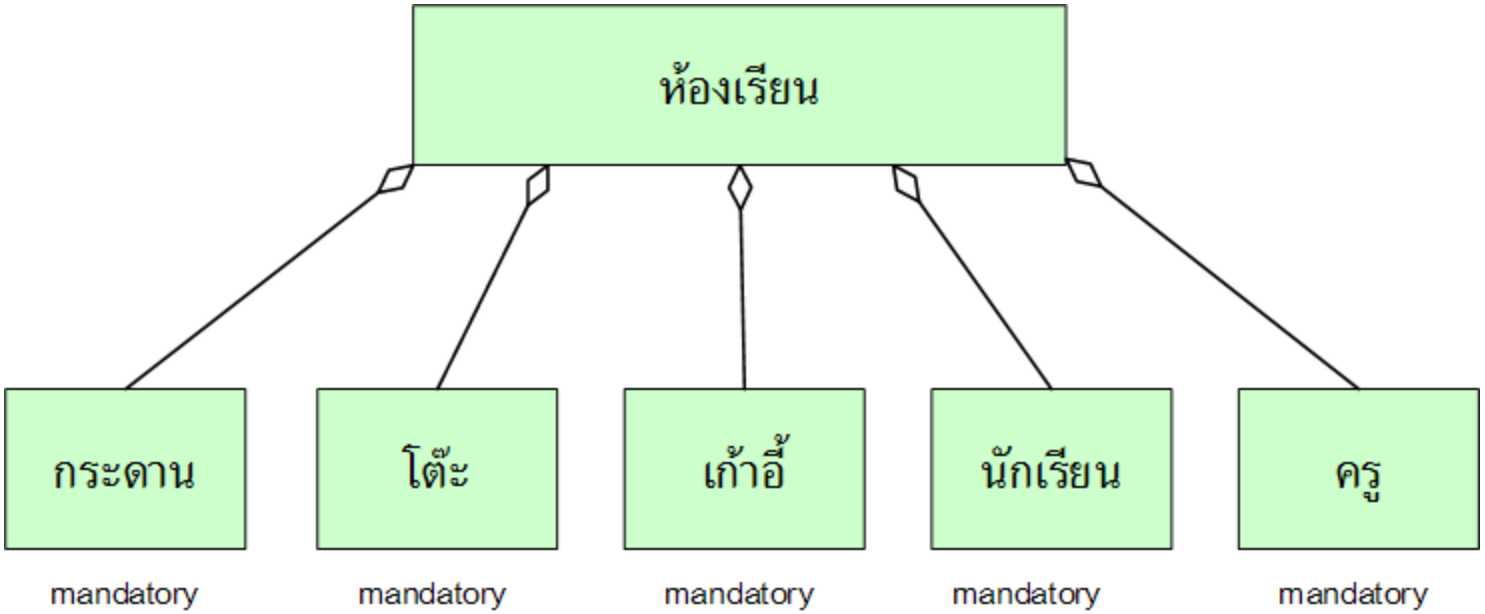
Cardinality, Required & Optional Components

- การประกอบกันของ class หรือความสัมพันธ์เชิง is part of
  - อาจจะประกอบไปด้วย class ย่อย (Composite class) ชนิดที่หนึ่ง เพียงชิ้นเดียว
  - class ย่อยชนิดที่สอง จำนวน 4 ชิ้นขึ้นไป
  - Class ย่อยชนิดที่สาม ไม่จำกัดจำนวน (หรืออาจไม่มีเลยก็ได้)
- สิ่งที่ใช้ในการแสดงจำนวนสมาชิกของ Object ในความสัมพันธ์ ดังกล่าวนี้เรียกว่า Cardinality

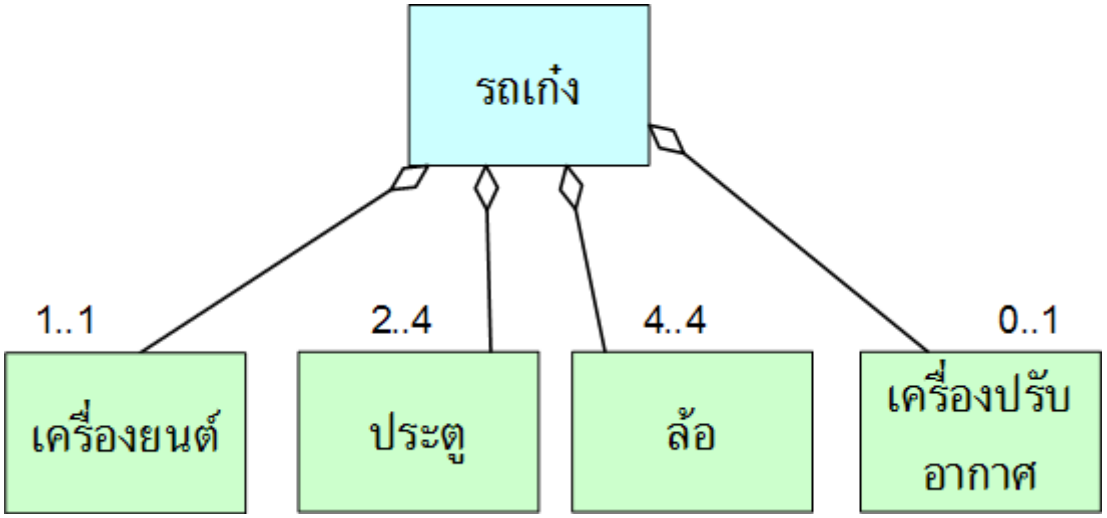
Cardinality, Required & Optional Components

- ในทาง OO นิยมเรียก Class ย่อย ว่า Component
- ส่วนประกอบที่**จำเป็น**ต้องมี เรียกว่า Required หรือ Mandatory Component
  - รถยนต์จำเป็นต้องมีเครื่องยนต์ ถ้าไม่มีเครื่องยนต์ รถยนต์ก็ไม่สามารถวิ่งได้
- ส่วนประกอบที่**ไม่จำเป็น**ต้องมี เรียกว่า Optional Component
  - เครื่องปรับอากาศในรถยนต์ไม่จำเป็นต้องมีก็ได้ ถึงไม่มีเครื่องปรับอากาศรถยนต์ก็ยังสามารถวิ่งได้

Cardinality, Required & Optional Components



Cardinality



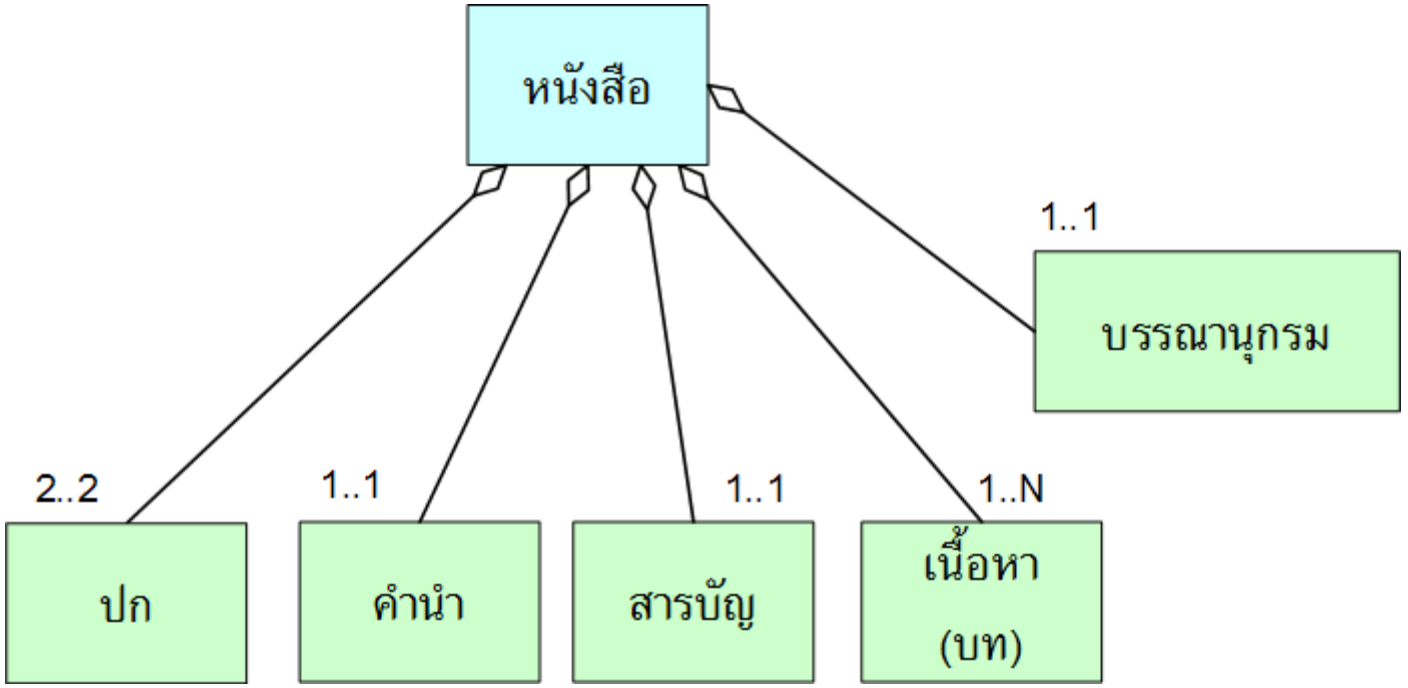
Maximum & Minimum Cardinality

- Maximum Cardinality (Max-card): จำนวนมากที่สุดของ Components ที่สามารถมีได้
  - เท่ากับ N
- Minimum Cardinality (Min-card): จำนวนน้อยที่สุดของ Components ที่สามารถมีได้
  - เท่ากับ 0 (ศูนย์)

การอ่าน Cardinality

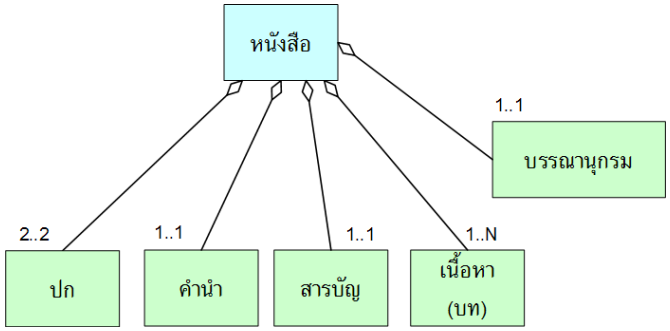
- <maximum | minimum> Cardinality ของ <ชื่อ Component> ใน aggregation <ชื่อ Class หลัก>-<ชื่อ component> มีค่าเท่ากับ <ค่าของ cardinality> เช่น
  - Minimum Cardinality ของ ประตู ใน Aggregation รถ-ประตู ,มีค่าเป็น 2 และ Maximum Cardinality ของ ประตู ใน Aggregation รถ-ประตู ,มีค่าเป็น 2
  - Minimum Cardinality ของ นักเรียน ใน aggregation ห้องเรียน-นักเรียน มีค่าเป็น 0 และ Maximum Cardinality ของ นักเรียน ใน aggregation ห้องเรียน-นักเรียน มีค่าเป็น n เมื่อ n เป็นจำนวนใดๆ

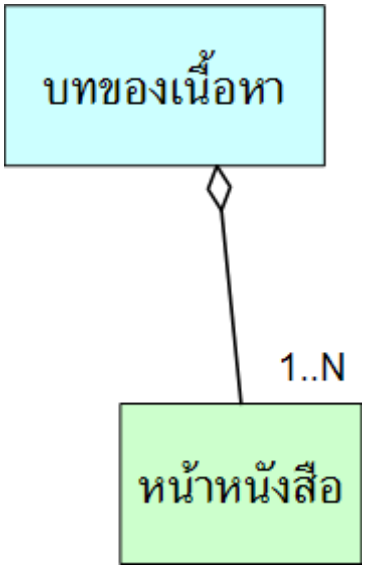
ตัวอย่าง Aggregation ของคลาส หนังสือ (1)

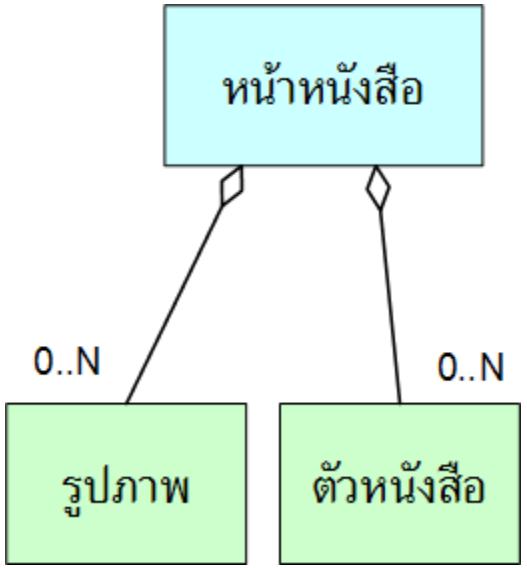


ให้นักศึกษาพยายามอธิบาย cardinality เป็นประโยคคำพูด

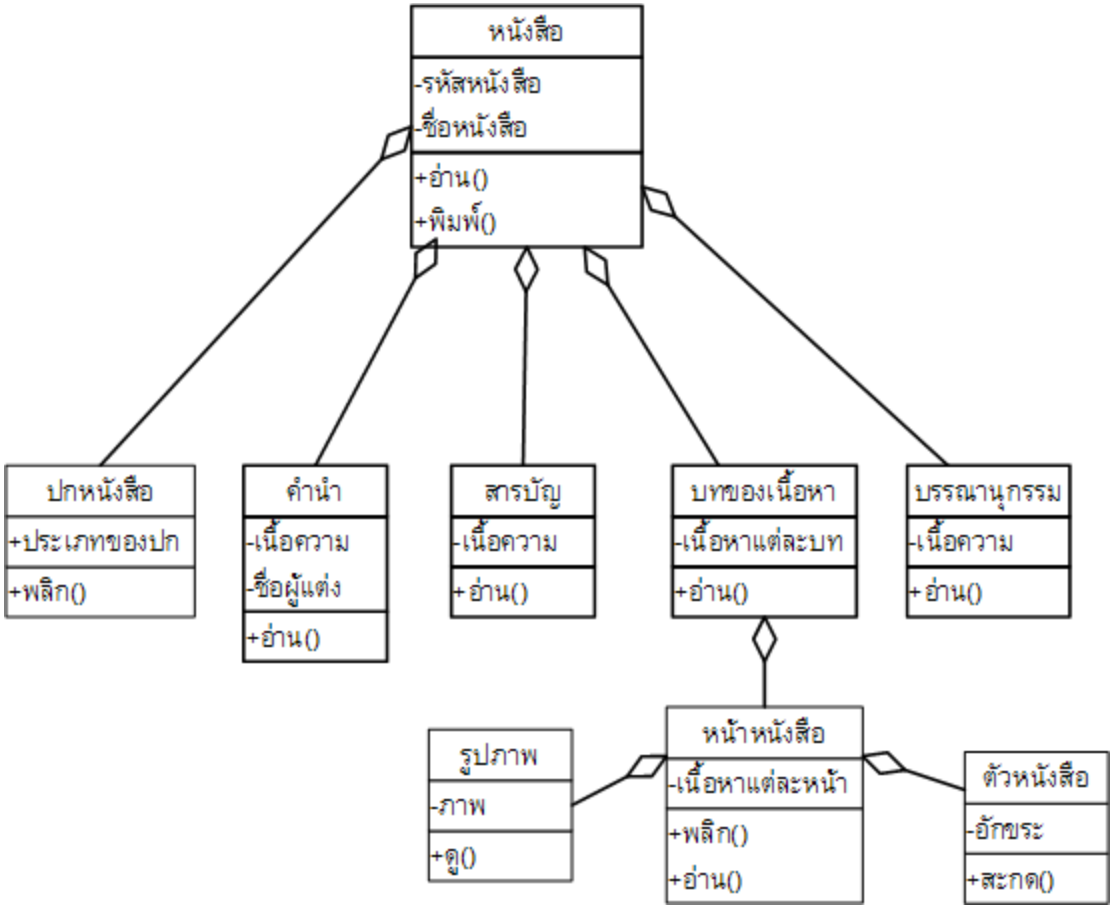
ตัวอย่าง Aggregation ของคลาส หนังสือ (2)







เพิ่ม Attribute และ Method ให้กับ Aggre. Class หนังสือ



แบบฝึกหัด / กิจกรรมอื่นๆ

- จงเขียน class diagram ที่แสดง Aggregation Abstration ของ Computer PC
- จงเขียน class diagram ที่แสดง Aggregation Abstration ของ โทรศัพท์มือถือ (Mobile)
- จงเขียน class diagram ที่แสดง Aggregation Abstration ของ บัญชีธนาคาร (Bank Account)

References

- กิตติพงษ์ กลมกล่อม, "พื้นฐานการวิเคราะห์และออกแบบระบบเชิงวัตถุด้วย UML", สำนักพิมพ์ เคทีพี, 2552.
- พนิดา พานิชกุล, "การพัฒนาระบบเชิงวัตถุด้วย UML", สำนักพิมพ์ เคทีพี, 2552.
- พนิดา พานิชกุล, "Object-Oriented ฉบับพื้นฐาน" , สำนักพิมพ์ เคทีพี, 2548.