

OOAD



www.siam2dev.com

วิชา การวิเคราะห์และออกแบบระบบเชิงวัตถุ (4122506)

Lec02_2 : แอกริเกชันแอบแทรกชัน

(Aggregation Abstraction)

By

ผู้ช่วยศาสตราจารย์ ดร. นัฐพงศ์ ส่องเนียม

Asst. Dr. Nattapong Songneam

เว็บไซต์ :: <http://www.siam2dev.com>

อีเมล :: siam2dev@gmail.com



ส่งงานกลุ่มครั้งที่ 1

- **Business Rule**เงื่อนไขทางธุรกิจของระบบงานที่พัฒนาเขียน/พิมพ์เป็นรายงาน กลุ่ม 10 คณะ ส่ง สัปดาห์หน้า
 - ปก
 - ชื่อระบบงาน
 - ผู้จัดทำ
 - ส่งรายวิชา สาขา คณะ ม.
 - คำนำ
 - สารบัญ
 - บทที่ 1 บทนำ
 - หลักการและเหตุผล
 - Business Rule ของระบบ

ไม่น้อยกว่า 10-20 หน้า

ส่งงานกลุ่มครั้งที่ 1

1. กำหนด ชื่องาน
2. สมาชิก
3. ความหมายของ Business Rule
4. ความเป็นมา ของ บ. ตัวอย่าง ร้านขายกาแฟ และเครื่องดื่ม CAFÉ'
5. แผนธุรกิจ
6. และเขียนมาเป็น Business Rule ของ บ. ของกลุ่ม
7. และ Function อย่างน้อย 10 Function
8. ไม่น้อยกว่า 10 หน้ากระดาษ

การกำหนดชื่อเรื่อง / ชื่อระบบงาน

- คลินิก
- บ. Toyota

ตัวอย่าง : การวิเคราะห์และออกแบบระบบ การจัดการร้านคลินิกขายยา
กรณีศึกษา ABC ฟาร์มาซี จำกัด

การวิเคราะห์และออกแบบระบบบริหารจัดการร้านสะดวกซื้อ กรณีศึกษา ...ร้าน ABC Express จำกัด

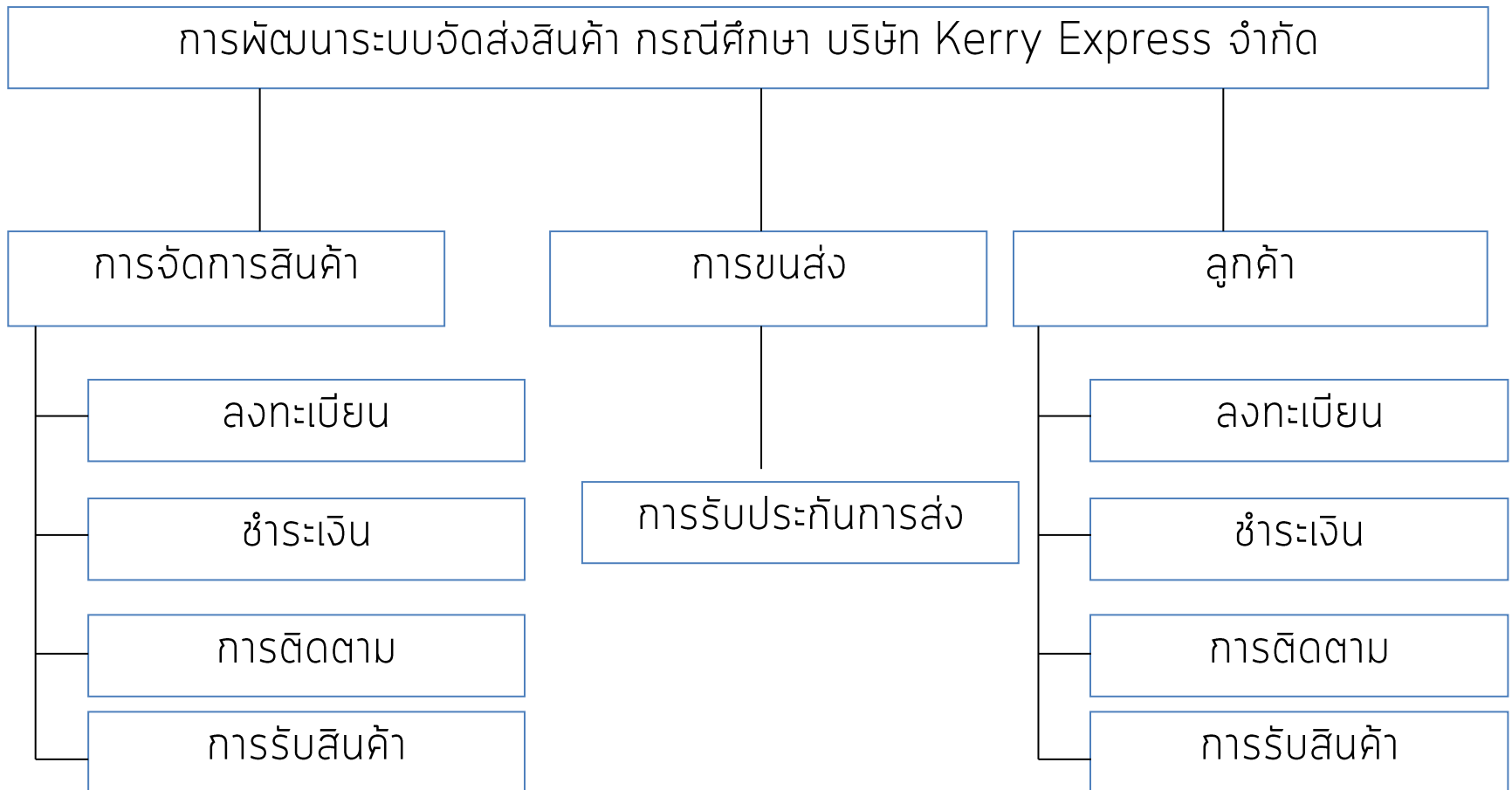
การวิเคราะห์และออกแบบระบบจ่ายเงินเดือนของ บ. Toyota

การวิเคราะห์และออกแบบระบบจัดการสินค้าคงคลัง บริษัท ABC Industry จำกัด

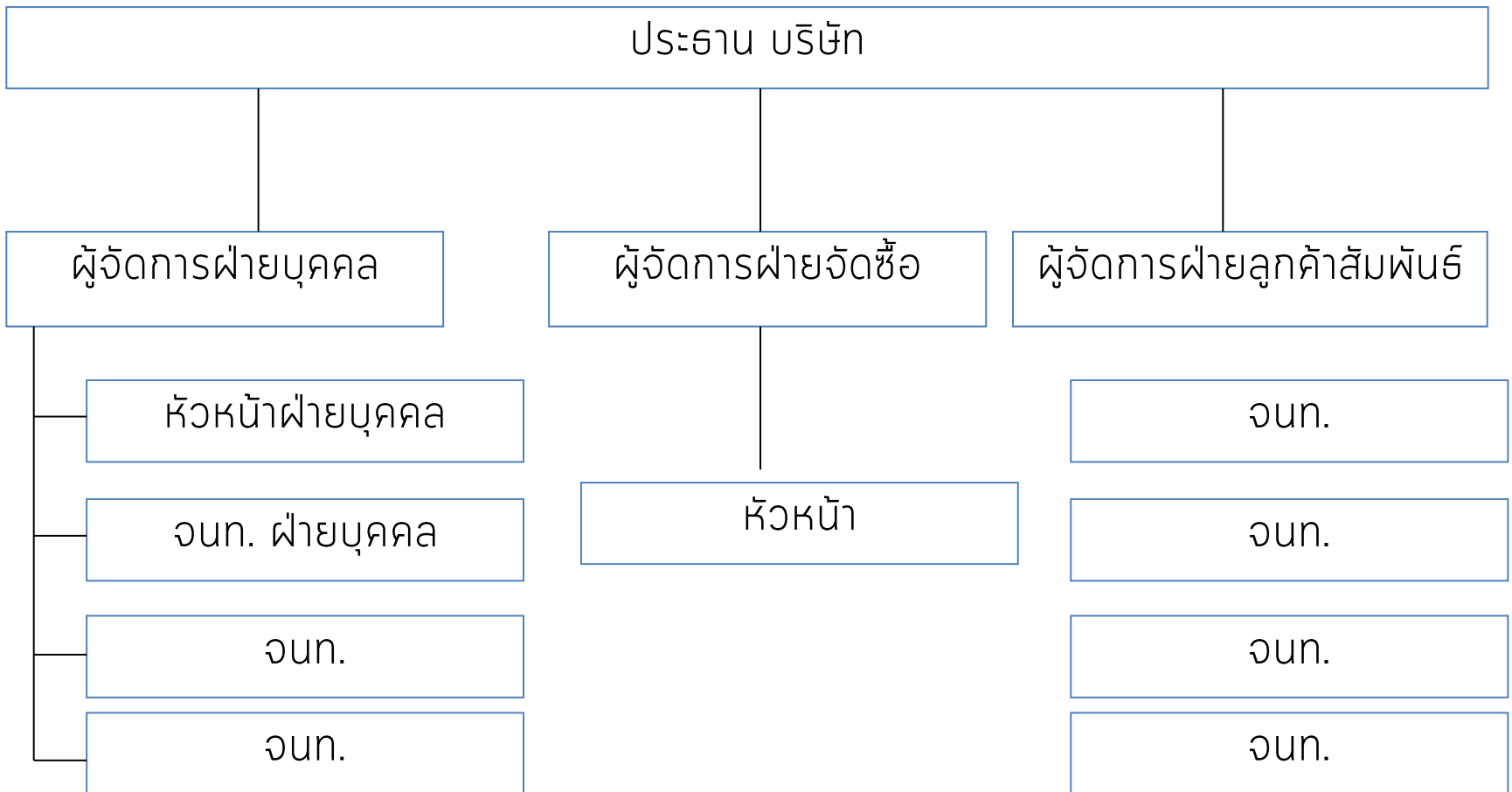
การออกแบบและพัฒนาระบบจัดการสินค้าคงคลัง บริษัท ABC Industry จำกัด

การออกแบบและพัฒนาระบบจัดส่งสินค้า กรณีศึกษา บริษัท Kerry Express จำกัด

ตัวอย่างของBusiness Rule



Organization Chart



Business Rule

กรณีศึกษา Kerry Express

- ระบบจัดส่งสินค้า บริษัท Kerry Express จำกัด
- บ. ดำเนินธุรกิจ อะไร ประเภทใด บริษัทดำเนินธุรกิจในการจัดส่งสินค้าให้กับลูกค้าทั้งภายในและต่างประเทศ
- เริ่มดำเนินการตั้งแต่เมื่อไหร่
- มีจำนวนพนักงานเท่าไหร่ มีกี่แผนก อะไรบ้าง
- มีกี่สาขา
- การจัดส่ง ดำเนินการอย่างไรตั้งแต่รับสินค้ามา และ จัดส่งเรียบร้อยถึงมือลูกค้า ทำอย่างไร
- วิสัยทัศน์ พันธกิจ
- การคิดราคา ภายใน กทม. ต่างจังหวัด ต่างประเทศ ?
- คิดตามน้ำหนัก คิดตามจำนวนกล่อง ...
- การรับประกัน สินค้า หรือไม่...

กรณีศึกษา Kerry Express

- บริษัท Kerry Express เป็นหนึ่งในบริษัทด้านการขนส่งและโลจิสติกส์ที่มีบริการในประเทศไทย โดยเน้นในการให้บริการด้านการจัดส่งพัสดุและสินค้าทั้งในระดับภูมิภาคและระดับประเทศ ด้วยการใช้เทคโนโลยีและระบบที่ทันสมัย เพื่อให้บริการที่มีคุณภาพและประสิทธิภาพสูงสุดสำหรับลูกค้าของตน
- นี่คือตัวอย่างของ Business Rule สำหรับกรณีศึกษา Kerry Express:
- **Business Rule: การเก็บค่าส่งพัสดุ**
- **รายละเอียด:** บริษัท Kerry Express จะเก็บค่าบริการส่งพัสดุตามน้ำหนักและขนาดของพัสดุ โดยมีอัตราค่าบริการที่แตกต่างกันตามประเภทของพัสดุและระยะทางที่จะส่ง
- **เงื่อนไข:**
- ค่าบริการจะถูกคำนวณโดยอ้างอิงจากน้ำหนักและขนาดของพัสดุ
- ระยะทางการส่งขึ้นอยู่กับพื้นที่ปลายทาง
- ลูกค้าสามารถตรวจสอบค่าบริการที่เว็บไซต์หรือแอปพลิเคชันของ Kerry Express ได้
- ค่าบริการอาจมีการเปลี่ยนแปลงตามโปรโมชั่นหรือส่วนลดที่บริษัทกำหนด

กรณีศึกษา Kerry Express

การดำเนินการ:

1. ลูกค้าจะกรอกข้อมูลพัสดุ เช่น น้ำหนักและขนาด และสถานที่จัดส่ง
2. ระบบจะคำนวณค่าบริการตามเงื่อนไขที่กำหนด
3. ลูกค้าจะได้รับรายละเอียดค่าบริการและตัวเลือกการจ่ายเงิน {โอนเงิน , QR-code , เงินสด}

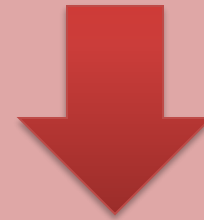
ผลลัพธ์:

- ลูกค้าจะได้รับข้อมูลค่าบริการที่ต้องชำระสำหรับการส่งพัสดุ
- บริษัท Kerry Express จะได้รับรายได้จากค่าบริการตามที่คำนวณไว้
- นี่เป็นเพียงตัวอย่างหนึ่งของ Business Rule สำหรับกรณีศึกษา Kerry Express โดยอาจมี Business Rule อื่น ๆ ที่เกี่ยวข้องกับด้านการบริการส่งของและโลจิสติกส์ในบริษัทนี้อีกมากมาย

สอบระหว่างภาค

วันนี้ บทที่ 2.2

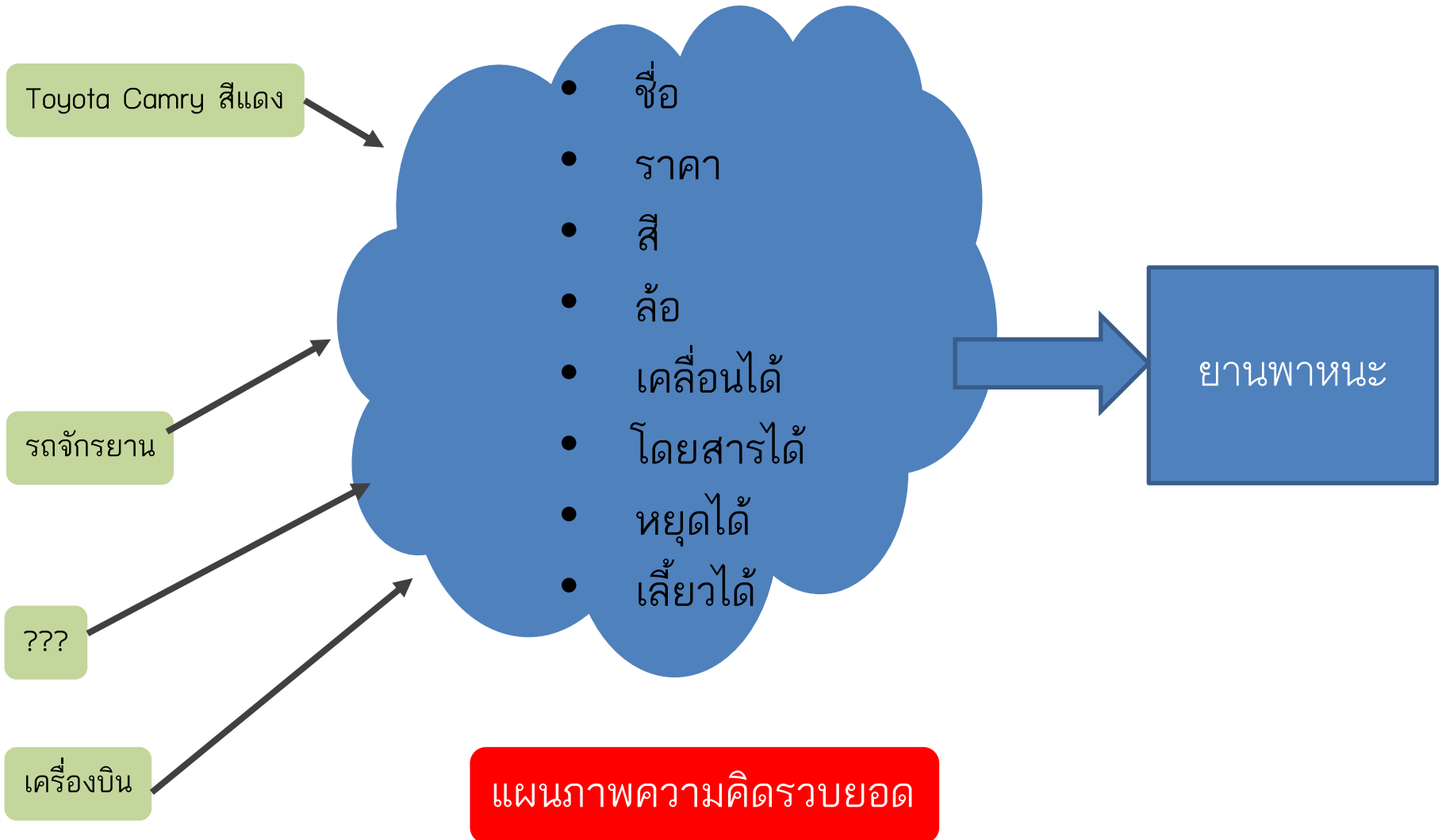
- ข้อสอบเป็นข้อเขียน
- ประมาณ 5 -7 ข้อใหญ่
- 60 คะแนน คิดเป็น 20%
- เนื้อหาที่ สอบ บทที่ 1 , 2 , 2.1 , 2.2 , 2.3 , 2.4 , 4 , 5



Lec02_2 : Aggregation Abstraction

- ในบทที่แล้วได้พูดถึง Classification Abstraction ไปแล้ว ซึ่งเป็นกระบวนการสร้าง Class จาก Object ต่าง ๆ ที่มีอยู่ใน Problem Domain จนได้ Class พื้นฐานที่จะนำไปสู่กระบวนการในขั้นตอนต่อไป แต่ยังขาดความสัมพันธ์ระหว่าง Class ต่าง ๆ ที่เราสร้างขึ้น
- จากสภาพความเป็นจริงในโลกนี้จะพบว่ายังมีวัตถุหลาย ๆ อย่างที่เกิดจากการรวมตัวของวัตถุอื่น ๆ เช่น คนเกิดจากการรวมกันของ วัตถุ ลำตัว วัตถุแขน วัตถุขา วัตถุหัว หรือ คอมพิวเตอร์เกิดจากการรวมของ case , main board , monitor , drive หรือแม้แต่โลกเราก็ยังประกอบไปด้วย ดิน น้ำ อากาศ และแร่ธาตุ ต่าง ๆ เป็นต้น

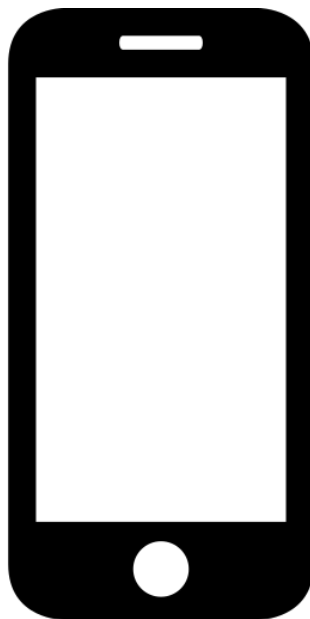
concept



Aggregation

- วัตถุใด ๆ ปกติจะประกอบไปด้วยวัตถุย่อย หรือ ส่วนประกอบย่อย เช่น
- โทรศัพท์มือถือ ก็ประกอบไปด้วย หน้าจอ ปุ่ม กล้อง ชิม แบต
- รถยนต์ ประกอบไปด้วย ล้อ เครื่องยนต์ ตัวถัง หลอดไฟ ที่นั่ง พวงมาลัย เป็นต้น

โทรศัพท์มือถือ



โทรศัพท์มือถือก็ประกอบไปด้วย หน้าจอ ปุ่ม กล้อง ชิป แบต

Abstraction

- Classification Abstraction หาดลลาจากวัตถุที่มีอยู่ในโดเมน
- **Aggregation Abstraction** คือ การพิจารณาส่วนประกอบย่อยของวัตถุ
- Generalization/specialization คือ แบ่งประเภทของคลาส
- Association Abstraction คือ การพิจารณาความสัมพันธ์หรือเกี่ยวข้องกันระหว่างคลาส

Aggregate/Composite

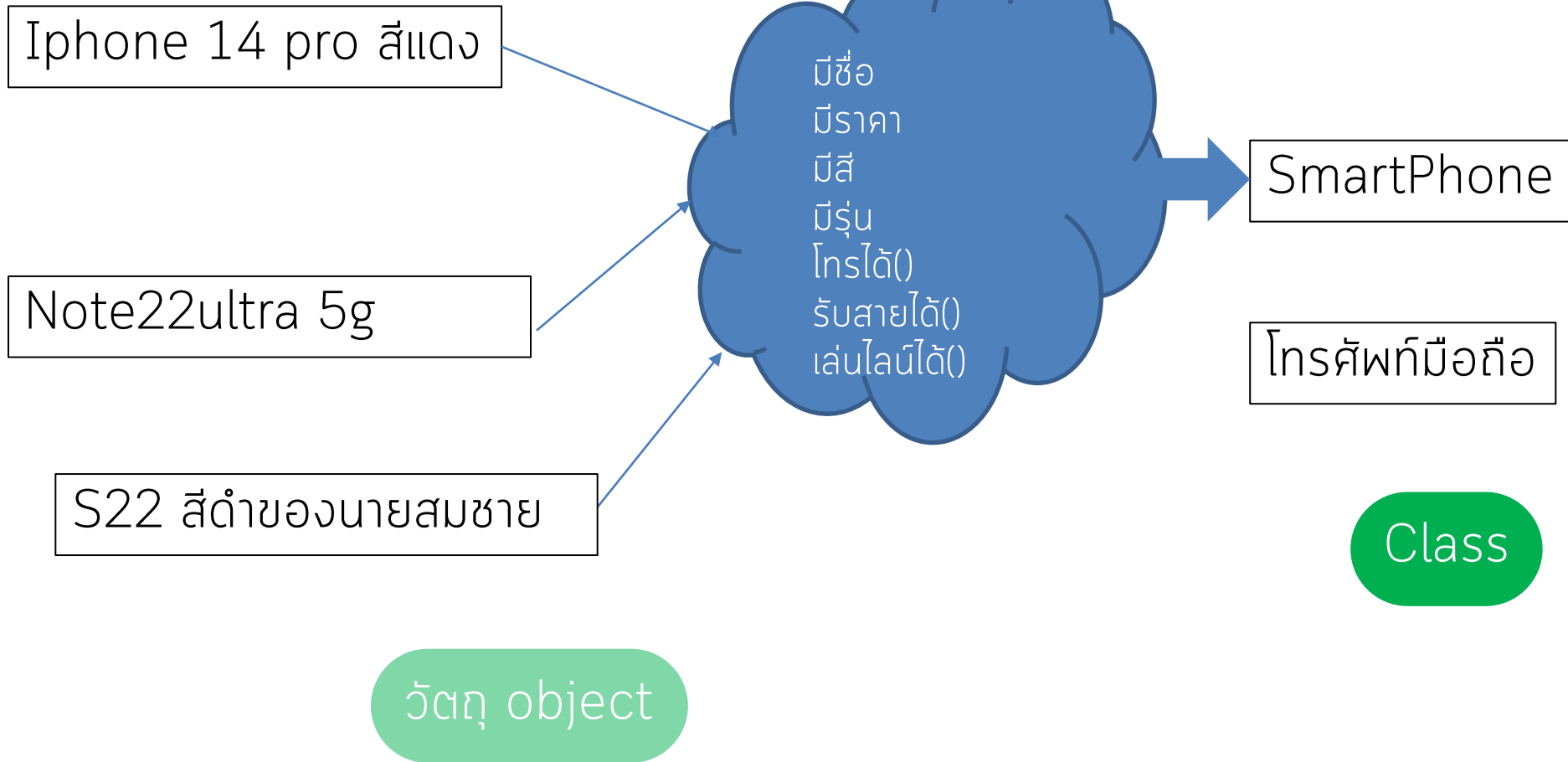
- แนวคิด ก็คือ วัตถุหนึ่ง สามารถมีส่วนประกอบเป็นวัตถุอื่น เป็นส่วนประกอบได้ เรียก การเป็นส่วนประกอบ ว่า aggregate หรือ composite
- คือ การพิจารณาที่วัตถุ หนึ่ง จะมีส่วนประกอบย่อยอะไรบ้าง เรียกการทำแบบนี้ว่า Aggregation Abstraction

Classification abstraction

คือการหาว่าวัตถุ อยู่ในคลาส การที่จะบอกว่าวัตถุไหนอยู่ในคลาสไหน จะต้องหาหรือสรุปความคิดรวบยอดเกี่ยวกับสิ่งนั้นให้ได้ความคิดรวบยอดก็คือ Concept

Concept ก็คือการบอกว่า วัตถุนั้นมีลักษณะอะไรบ้าง และมีฟังก์ชันใดบ้าง เรียกสิ่งนั้นว่าอะไรซึ่งก็คือ การหา attributes และฟังก์ชัน ของวัตถุหรือ คลาสนั้นเอง

การให้นิยามหรือความคิดรวบยอดเกี่ยวกับวัตถุ



อุปกรณ์การเรียน

ถ้า

เครื่องเขียน

อุปกรณ์คอมพิวเตอร์

- เจอ ปากกา ดินสอ เมาส์ โปรเจ็คเตอร์ อยู่ในกลุ่มเดียวกันหรือไม่

- ถ้าไม่อยู่
- ถ้าอยู่ เรียกว่า อะไร

แมว

สิ่งที่อยู่ในโลก

เสือด่า

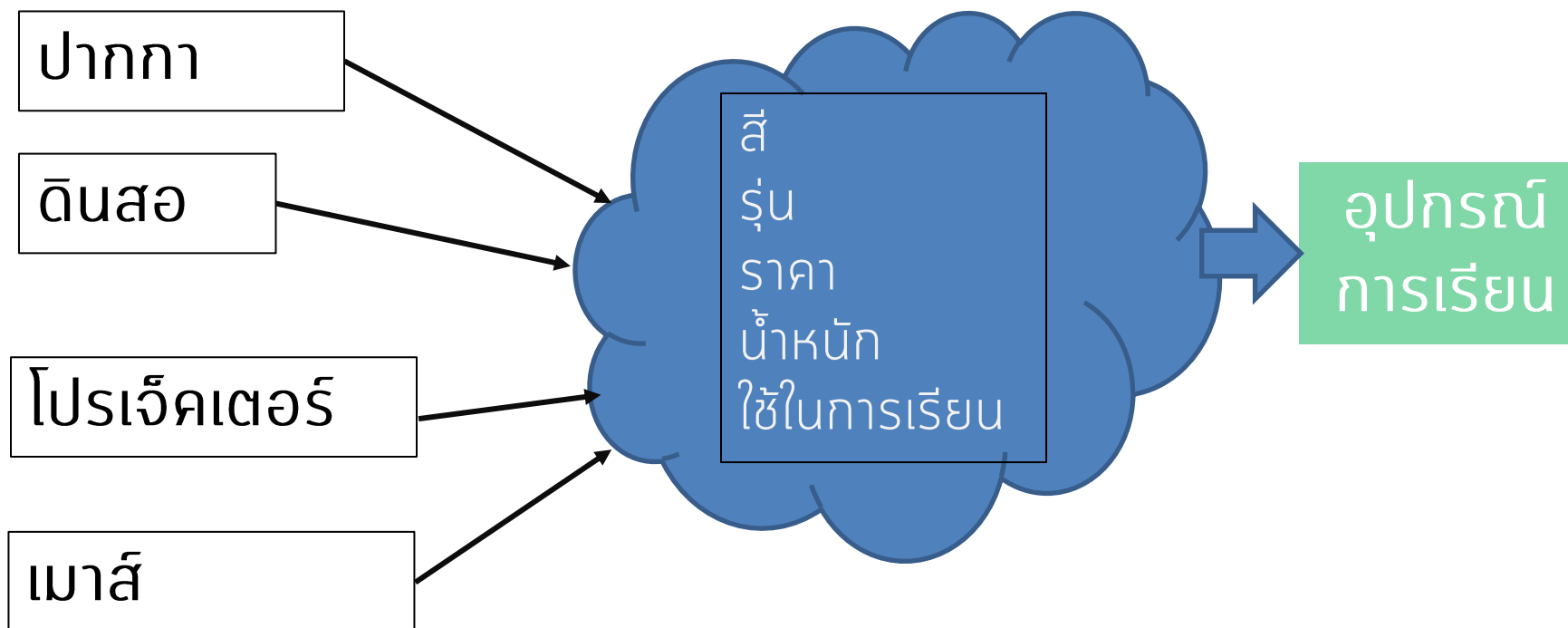
สิ่งของ

สิ่งไม่มีชีวิต

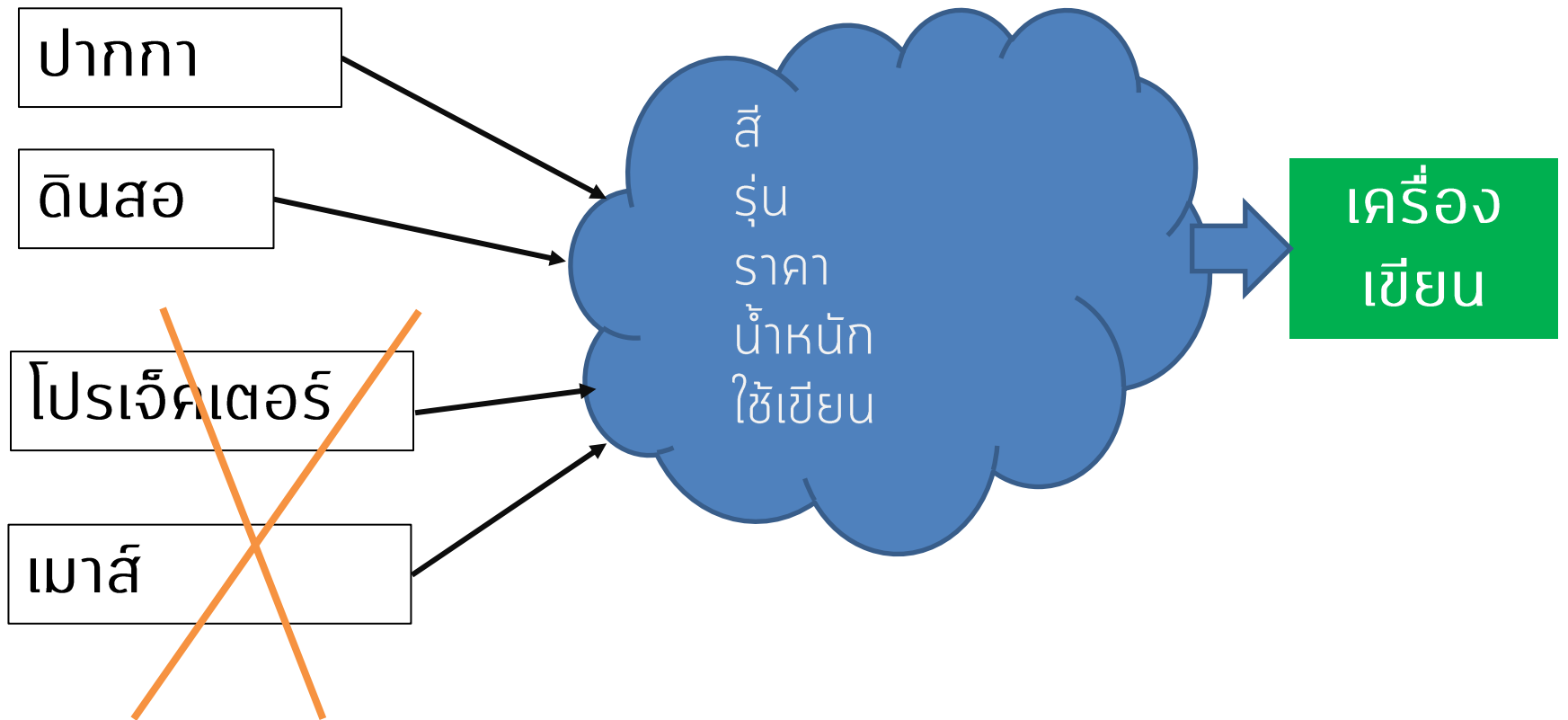
สินค้า

- ไม่มีประโยชน์
- มีประโยชน์

เจอ ปากกา ดินสอ เมสส์ โปรเจ็คเตอร์ อยู่ในกลุ่มเดียวกันหรือไม่



เจอ ปากกา ดินสอ เมาส์ โปรเจ็คเตอร์ อยู่ในกลุ่ม
เดียวกันหรือไม่



class

class

เป็นส่วนของ

เครื่องเขียน

อุปกรณ์การเรียน

ดินสอ

ดินสอสีแดง

ดินสอของอาจารย์ นัฐพงศ์

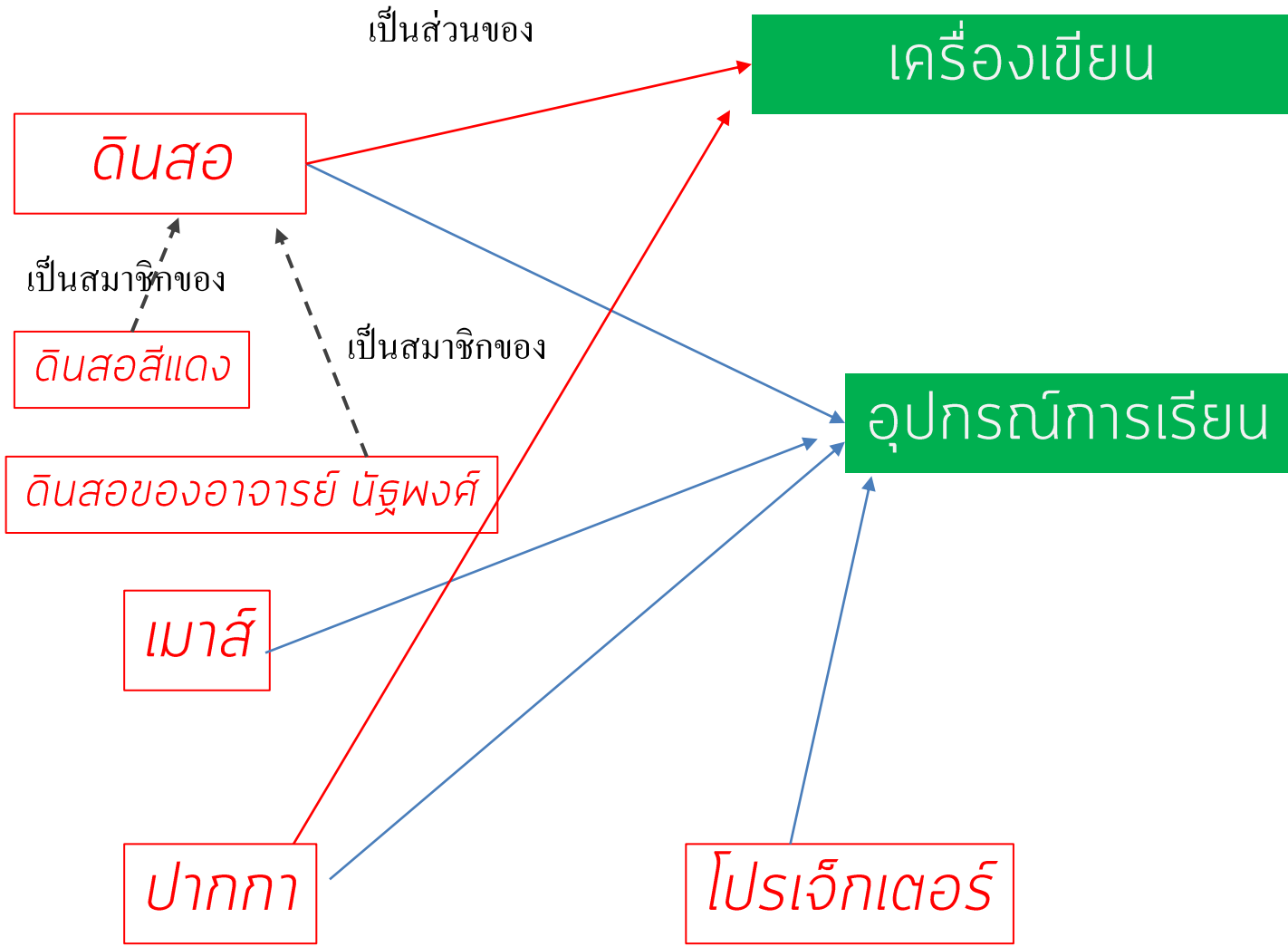
เมาส์

ปากกา

โปรเจกเตอร์

เป็นสมาชิกของ

เป็นสมาชิกของ



เครื่องเขียน

เมาส์

พริ้นเตอร์

อุปกรณ์คอมพิวเตอร์

class

class

แบ็นพิมพ์

ดินสอ

ปากกา

ยางลบ

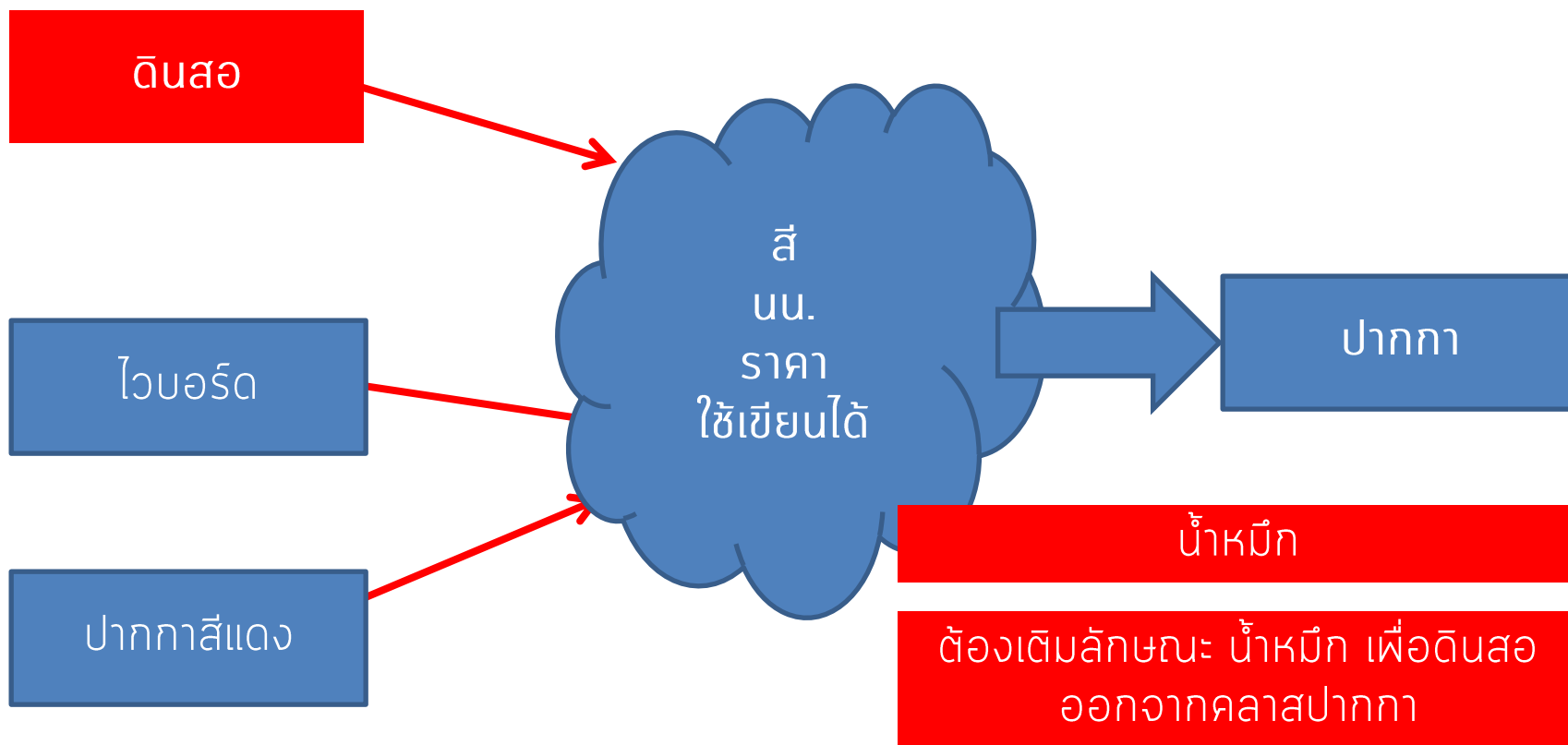
ไม้บรรทัด

มีชื่อ
ยี่ห้อ
ราคา
+ใช้สำหรับเขียน()

เครื่องเขียน

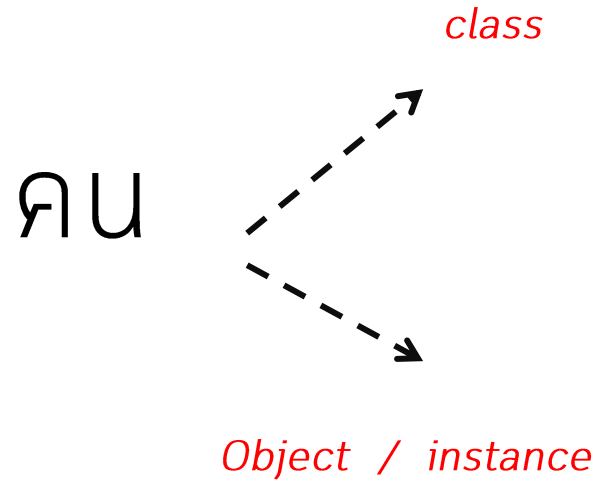
- ถ้าเราเจอ สิ่งหนึ่ง มี สี นน. ราคา และใช้เขียน เรียกว่าปากกา
- ถ้าปากกาสีแดง ไบบอร์ด ดินสอ ก็ อยู่ในกลุ่มนี้

เมาส์ เมาส์ อยู่ในกลุ่ม หรือไม่



Reviews

- Objects ?
- Class ?



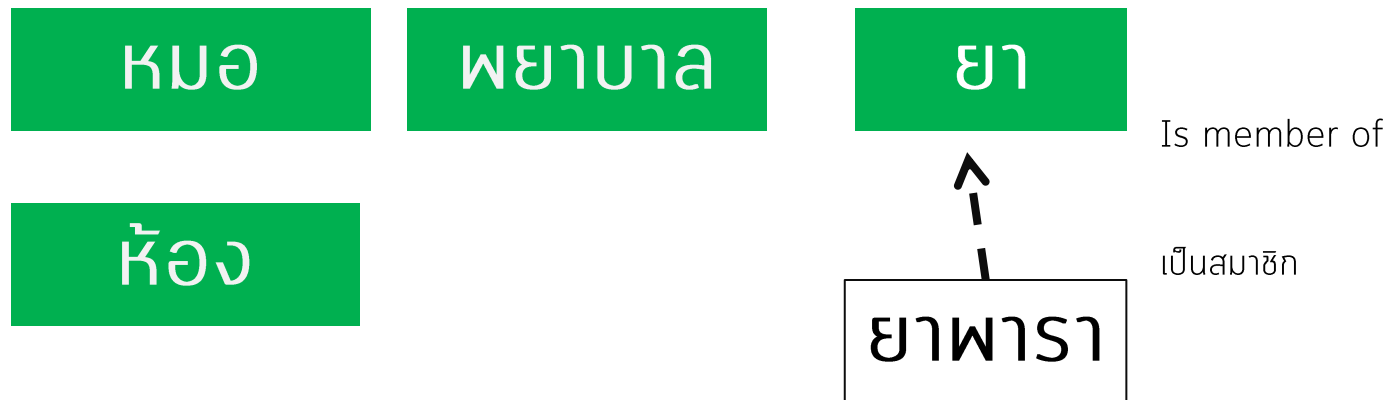
Problem Domain = ขอบเขตระบบ/สิ่งที่เราสนใจ

จะได้คลาสต่างๆ fundamental classes

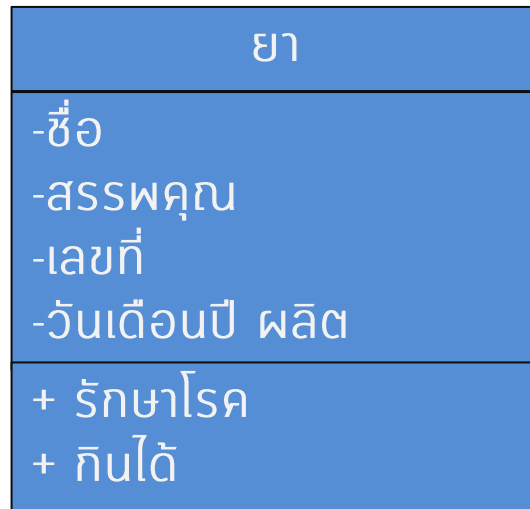
สมมุติ ทำ classification abs สำหรับ SW.

*** การเขียนแผนภาพคลาส class diagram สัมพันธ์

สิ่งที่จะได้ ก็คือคลาสต่างๆ ของ SW.



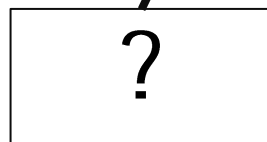
คลาส



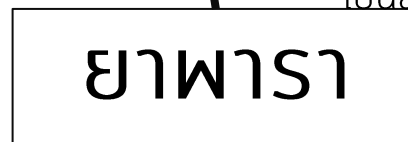
Class name

attribute

Function / Behavior / Operation

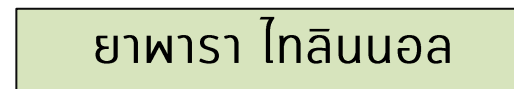


วัตถุ



Is member of

เป็นสมาชิก



จะได้คลาสต่างๆ fundamental classes

สมมุติ ทำ classification abs สำหรับ ระบบห้องสมุด

*** การเขียนแผนภาพคลาส class diagram สัมพันธ์

สิ่งที่จะได้ ก็คือคลาสต่างๆ ของ ห้องสมุด

บัตรสมาชิก

เจ้าหน้าที่

นักศึกษา

CD

??

สื่อการเรียน

ใบเสร็จ

หนังสือ

Is member of



เป็นสมาชิก

หนังสือ OOP

จะได้คลาสต่างๆ fundamental classes

สมมุติ ทำ classification abs สำหรับ จองห้องพักรงแรม

*** การเขียนแผนภาพคลาส class diagram สัมพันธ์

สิ่งที่ได้ ก็คือคลาสต่าง ๆ ของ จองห้องพักรงแรม

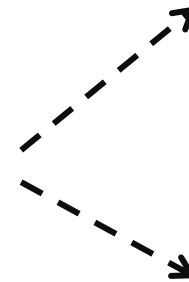


Reviews

- Objects ?
- Class ?

คุณ

class

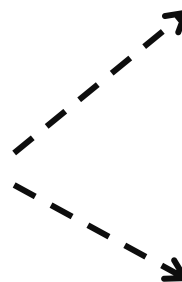


Object / instance

Problem Domain = ขอบเขตระบบ/สิ่งที่เราสนใจ

คุณ

class



ไม่ระบุเจาะจง

เจาะจง มากมาย ข คือ object

ขั้นตอน การทำ Classification Abs.

ขั้นตอนที่ 1 : กำหนด *Problem Domain*

คือหาคลาสๆ จากวัตถุต่างๆ ที่อยู่ใน *Problem domain*

ขั้นตอนที่ 2 : หารายชื่อ

ขั้นตอนที่ 3 : แยกแยะ รายชื่อ ว่า *object* หรือ *attributes* ของ *objects*

ขั้นตอนที่ 4 : ให้ *concept* การให้ความคิดรวบยอด เพื่อให้ได้ คลาส พื้นฐานต่างๆ

- วาดเป็นแผนภาพเพื่อแสดง *concept*

ขั้นตอนที่ 5 : จากนั้น ระบุ ว่าวัตถุใด อยู่ในคลาสใด

ขั้นตอนที่ 6 : จากใส่รายละเอียดของคลาส *attribute* , *function*

ขั้นตอนที่ 7 : วาดแผนภาพ แสดง *classification abstractions*

จะได้คลาสต่างๆ

Classification abs.

Class A
Class B
Class C
Class E
Class N

SW.

Aggregate

Class A

Class B

Class C

Class D

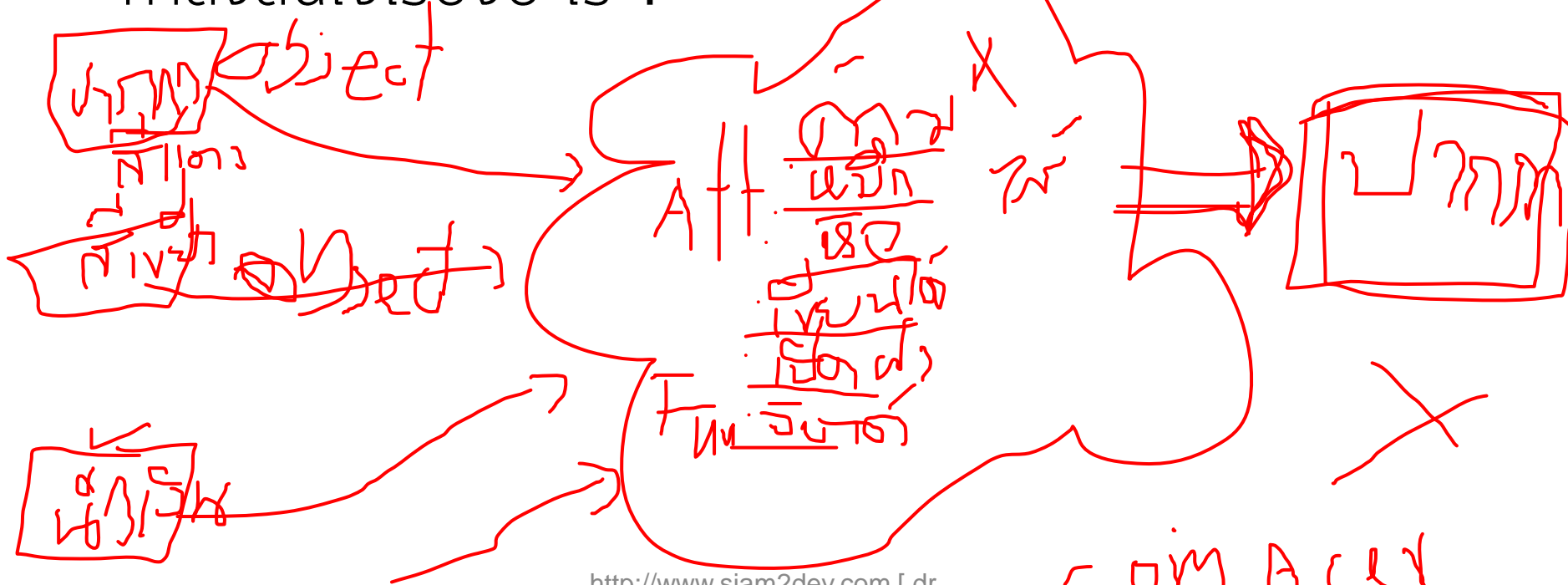
Class A

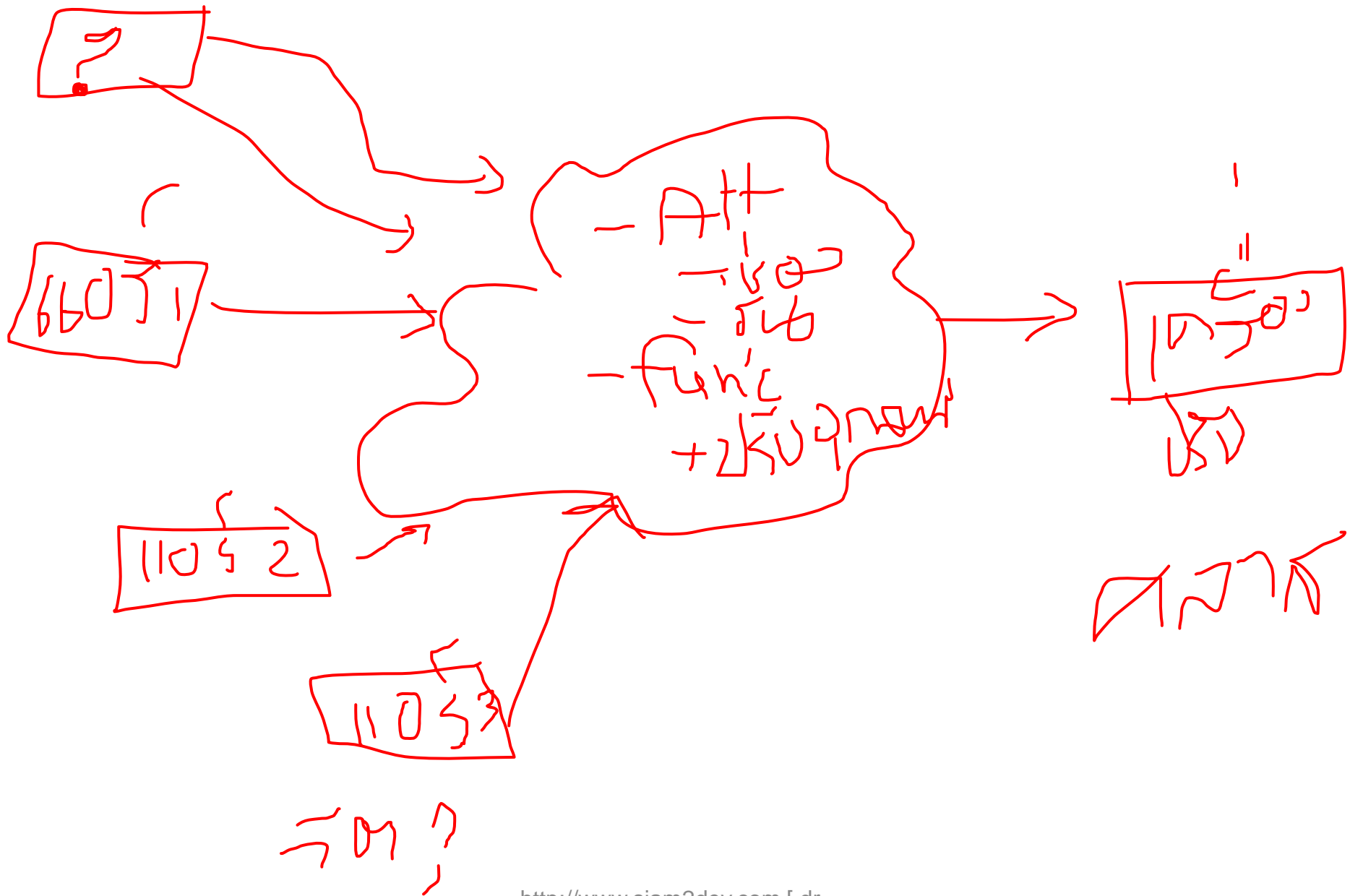
Class A1
Class A2
Class A3

Class B

วัตถุ ?

- หาคำนาม
- แล้วว่าอะไรคือวัตถุ คือแอทริบิวต์
- กำลังสนใจเรื่องอะไร ?





ความคิดรวบยอด(Concept)ของยานพาหนะ

รถยนต์ BMW สีแดง

รถยนต์

รถไฟ

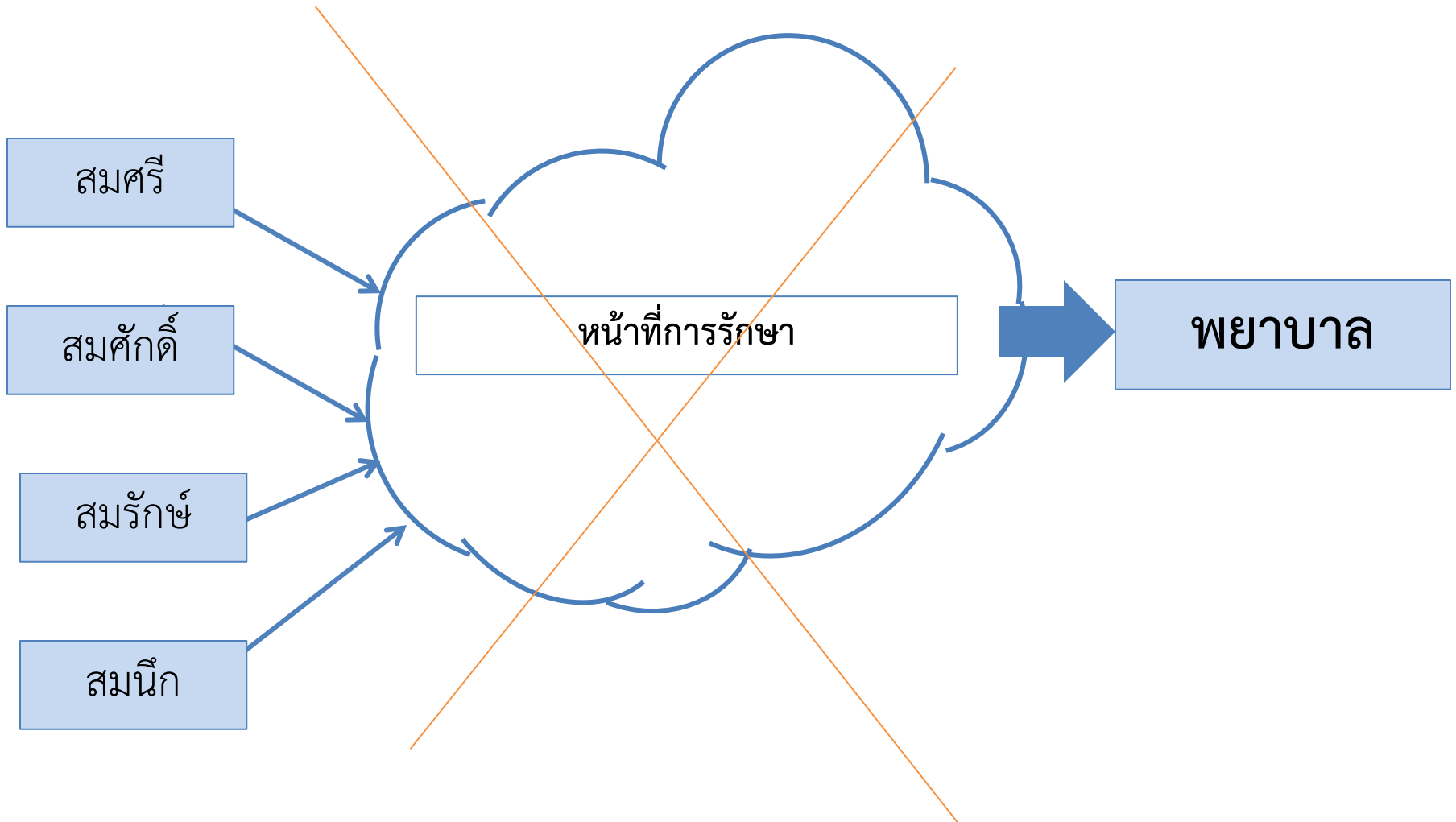
เรือ

เครื่องบิน

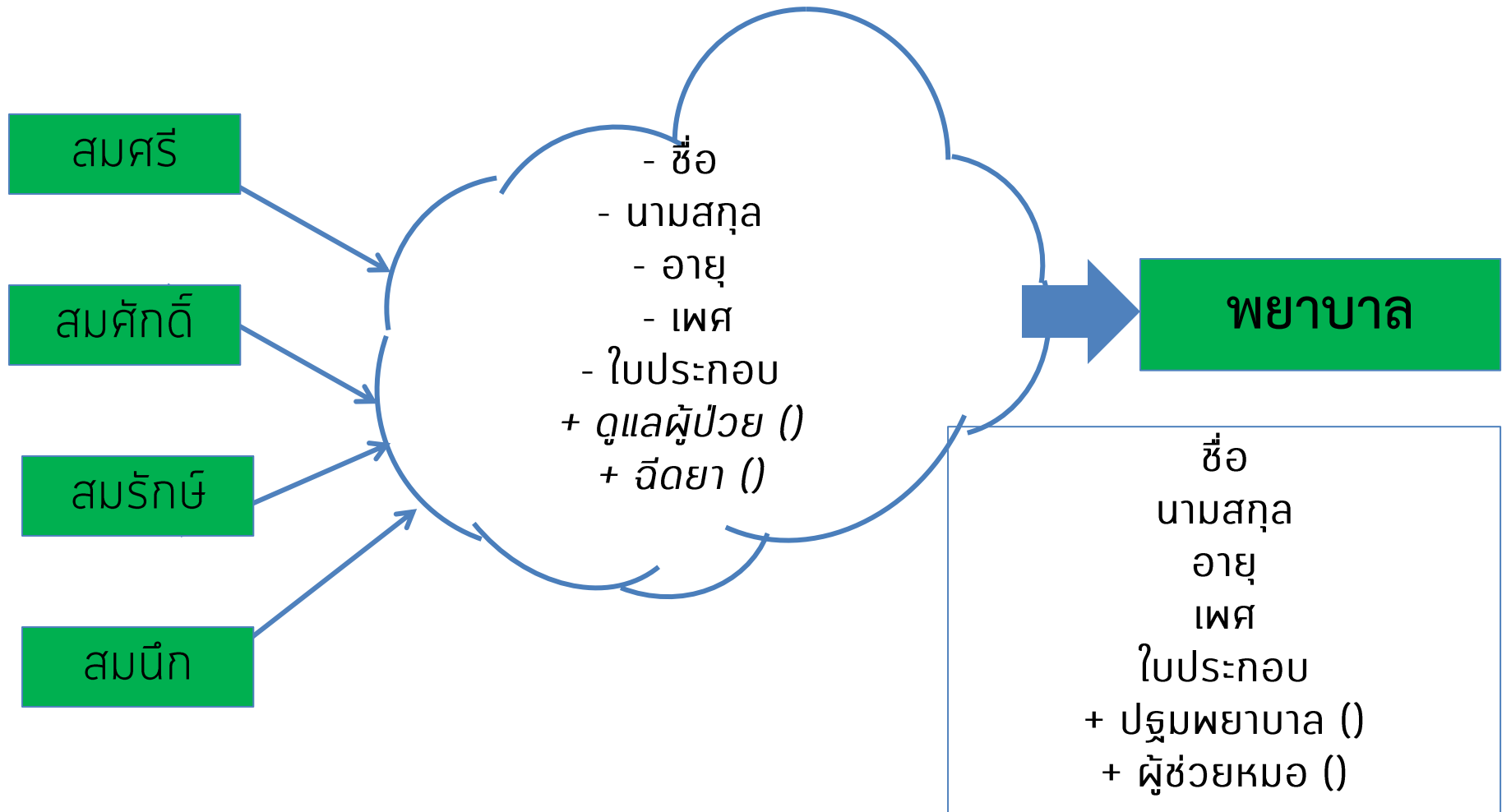
- มีที่นั่ง
โดยสารได้ ()
เคลื่อนที่ได้ ()
หยุดได้ ()
เลี้ยวได้ ()
ใช้น้ำมันหรือก๊าซเป็นเชื้อเพลิง ()

ยานพาหนะ

ความคิดรวบยอด(Concept)ของพยาบาล



ความคิดรวบยอด(Concept)ของพยาบาล



พยาบาล

attributes

functions

พยาบาล

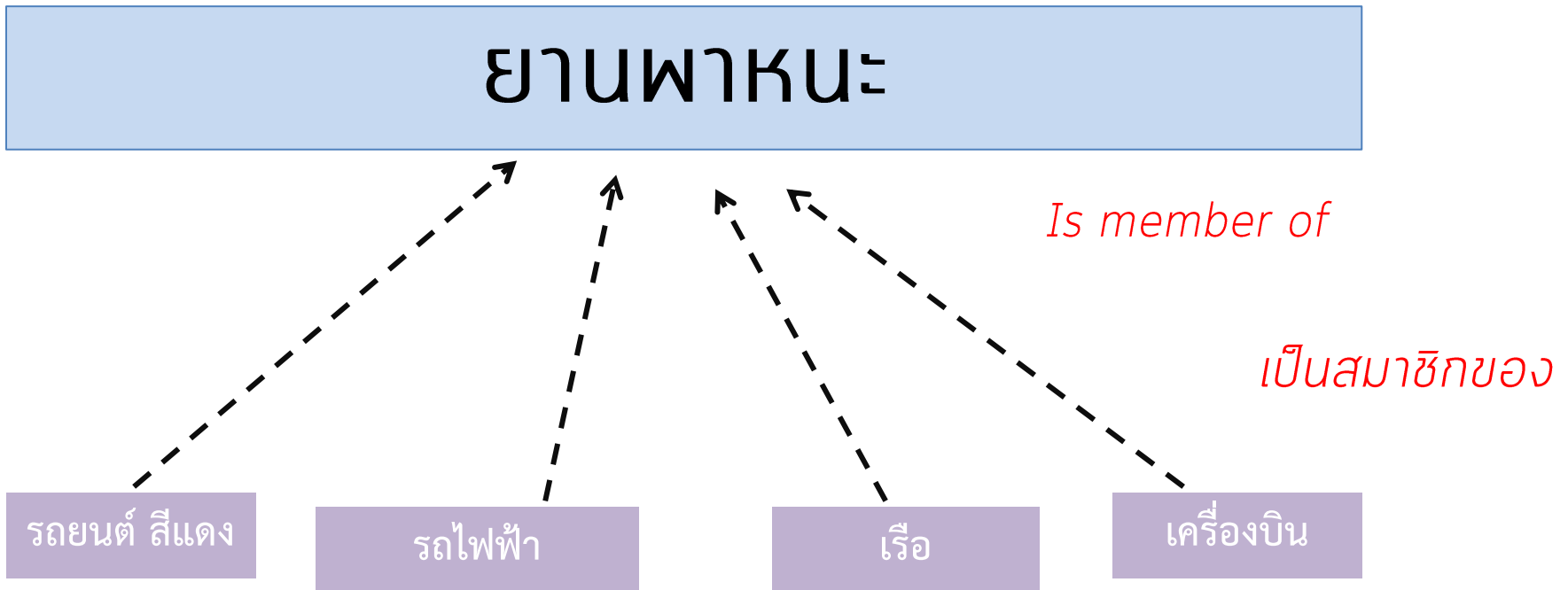
- ชื่อ
- # นามสกุล
- เพศ
- อายุ
- ใบประกอบวิชาชีพ

- + ดูแลคนไข้ ()
- + ฉีดยา()
- + สอบถามอาการ ()

คนไข้

- ชื่อ
- # นามสกุล
- เพศ
- อายุ

- + บอกอาการ ()



วัตถุ

รถยนต์ สีแดง ยี่ห้อ BMW เลขทะเบียน 4ก 4754

คลาส

รถยนต์

class

ยานพาหนะ

- ชื่อ
- ที่นั่ง

- + เคลื่อนที่ได้()
- + บอกชื่อ ()

Is member of

เป็นสมาชิกของ

รถยนต์ ของ
นาย สมศักดิ์

รถไฟฟ้า สายสี
แดง

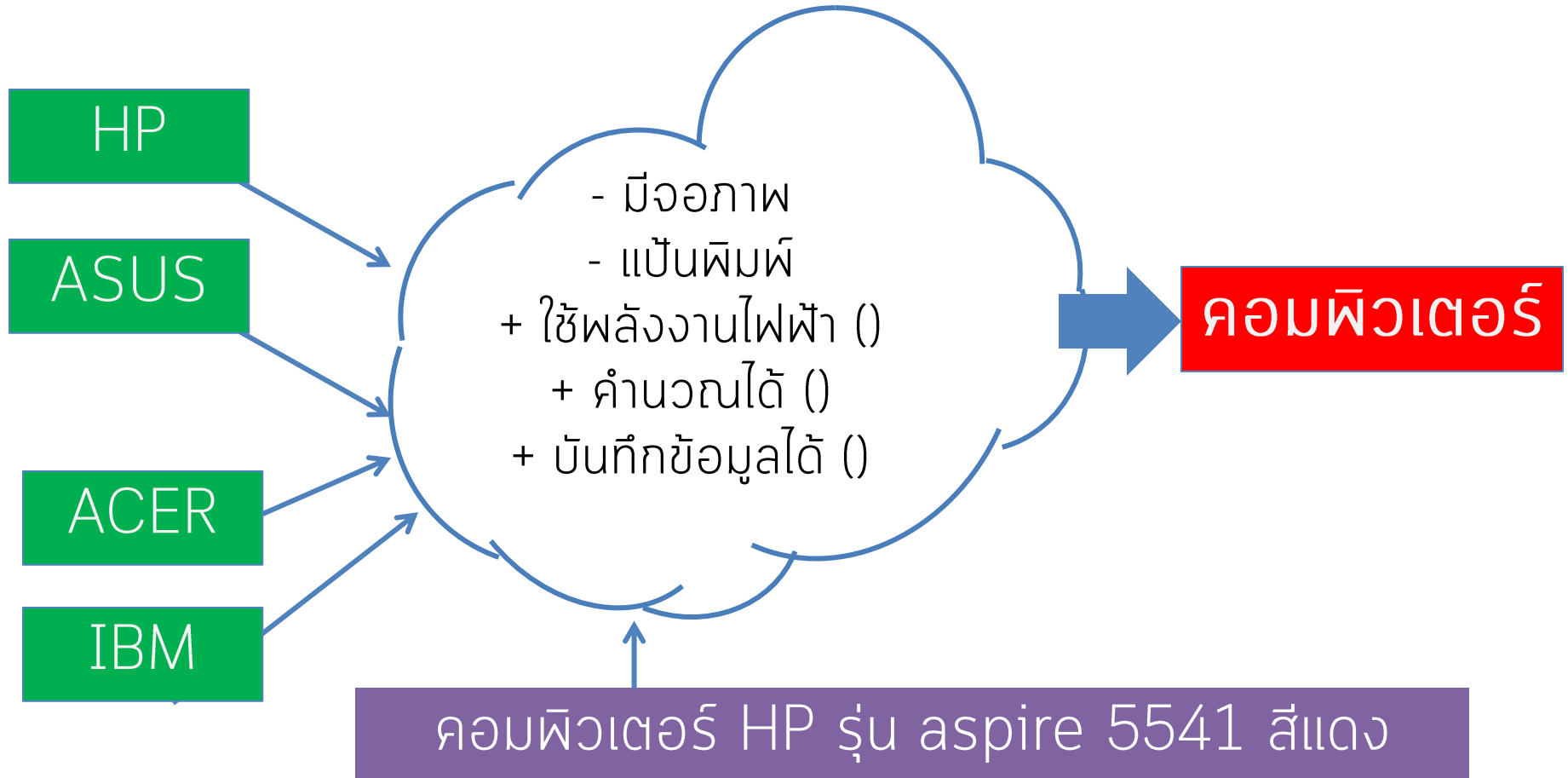
เรือ สีขาว

เครื่องบิน
Boing747 เที่ยวบิน
TG456

objects

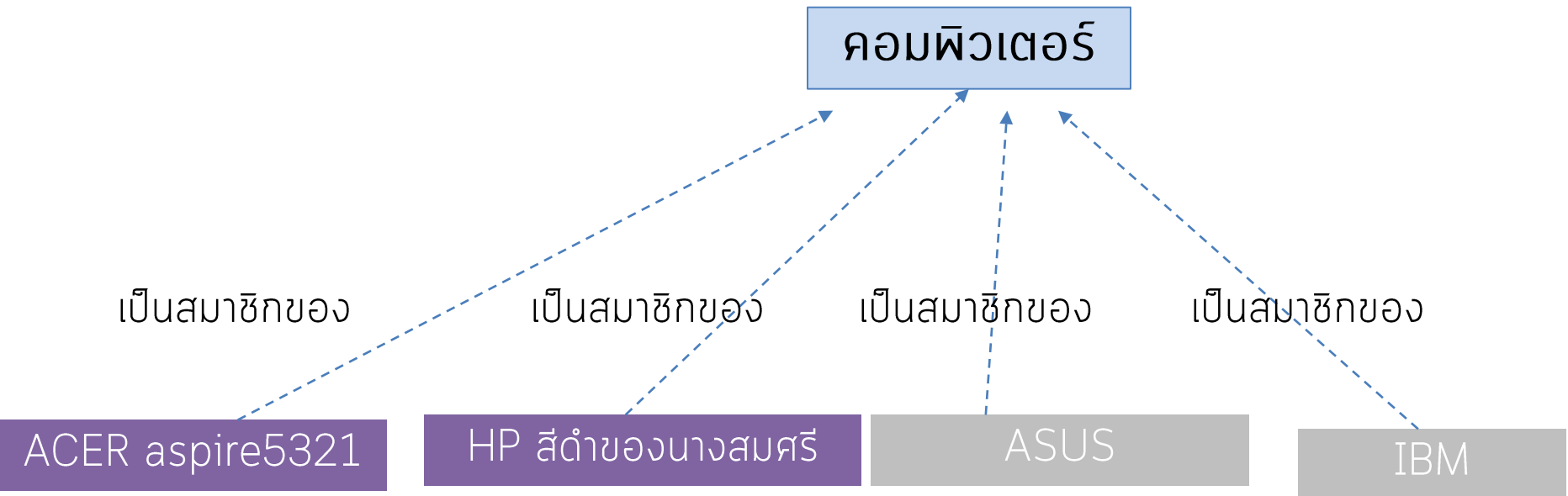
***ควรบอกวัตถุประสงค์ให้เจาะจง มากกว่า นี้ เช่น รถยนต์ ของนาย สมศักดิ์

ความคิดรวบยอดของคอมพิวเตอร์



***ควรบอกไว้ด้วยให้เจาะจง มากกว่า นี้ เช่น Acer aspire2000 สีแดง ของนายดำ

Classification abstractions



ACER Aspire 5:

หน้าจอ: 15.6 นิ้ว

ประมวลผล: Intel Core i5-1135G7

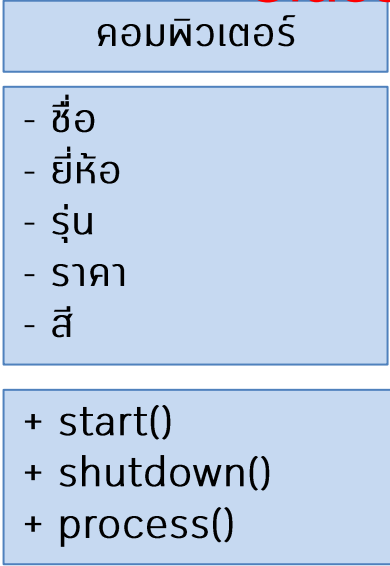
หน่วยความจำ: 8GB RAM, 512GB SSD

การ์ดจอ: NVIDIA GeForce MX350

ระบบปฏิบัติการ: Windows 10

Visibility / Accessibility

- Private
Protected
+ Public



Is member of

เป็นสมาชิกของ

เป็นสมาชิกของ

เป็นสมาชิกของ

เป็นสมาชิกของ

ACER aspire5321

HP สีดำนางงามสมศรี

ASUS

IBM

Encapsulation
Information Hiding

คน

คอมพิวเตอร์

- ชื่อ
- ยี่ห้อ
- รุ่น
- ราคา
+ สี

- Private
Protected
+ Public

+ start()
+ shutdown()
+ process()
+ กำหนดราคา()

Is member of

เป็นสมาชิกของ

เป็นสมาชิกของ

เป็นสมาชิกของ

เป็นสมาชิกของ

ACER
aspire532
1

HP สีดำนางงามสตรี

ASUS

IBM

ส่วนประกอบ (Aggregate)

โดยปกติแล้ววัตถุย่อย
ประกอบขึ้นมาจากหลาย
ส่วนประกอบ



จเขียนแผนภาพความคิดรวบยอด(concept) ของ นาฬิกาข้อมือ



ชื่อ-นามสกุล รหัส 13 หลัก

X บริษัทผลิต คือนาฬิกาข้อมือ ?

X คนใช้งาน คือนาฬิกาข้อมือ ?

X บอกเวลาเป็นฟังก์ชัน ?

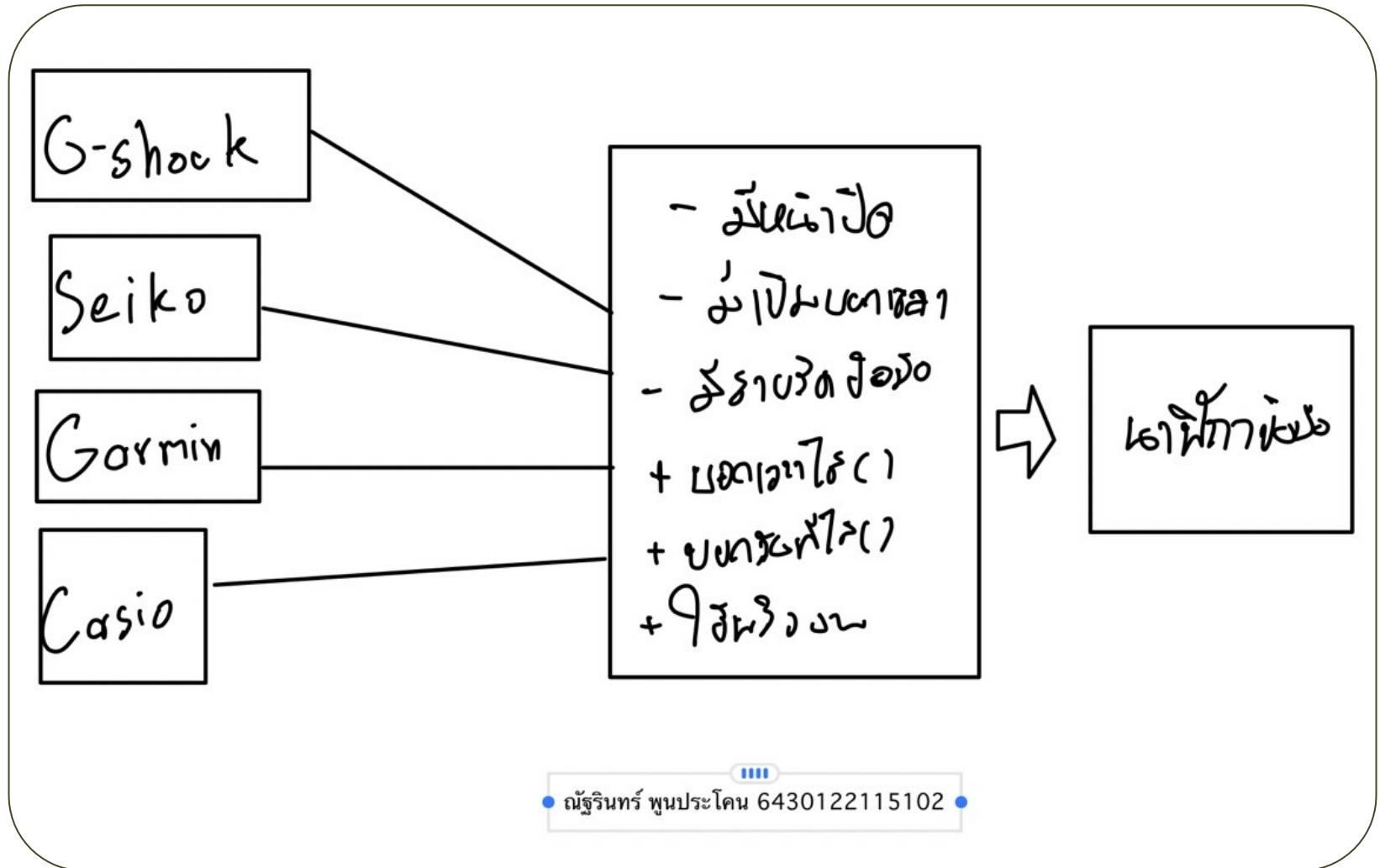
นาฬิกาข้อมือของ อ. นัฐพงศ์

นาฬิกาข้อมือสีแดง



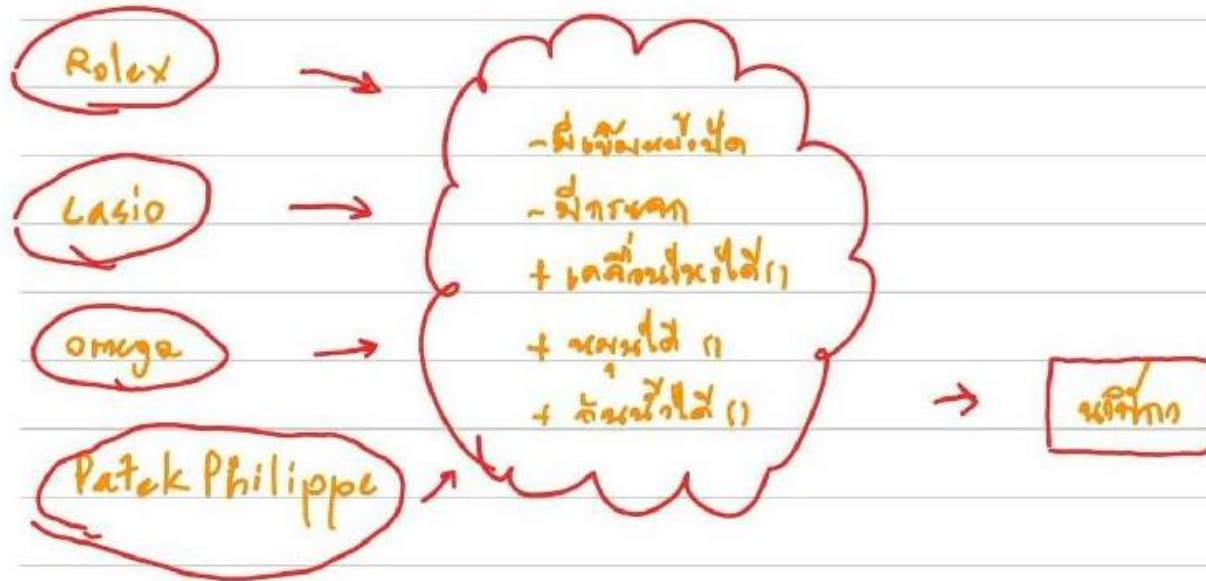
นาย วีรภัทร กิ่งแก้ว 6430122115111

การใส่แคยี่ห้อ ยังเป็นคลาส ไม่ใช่วัตถุดิบ



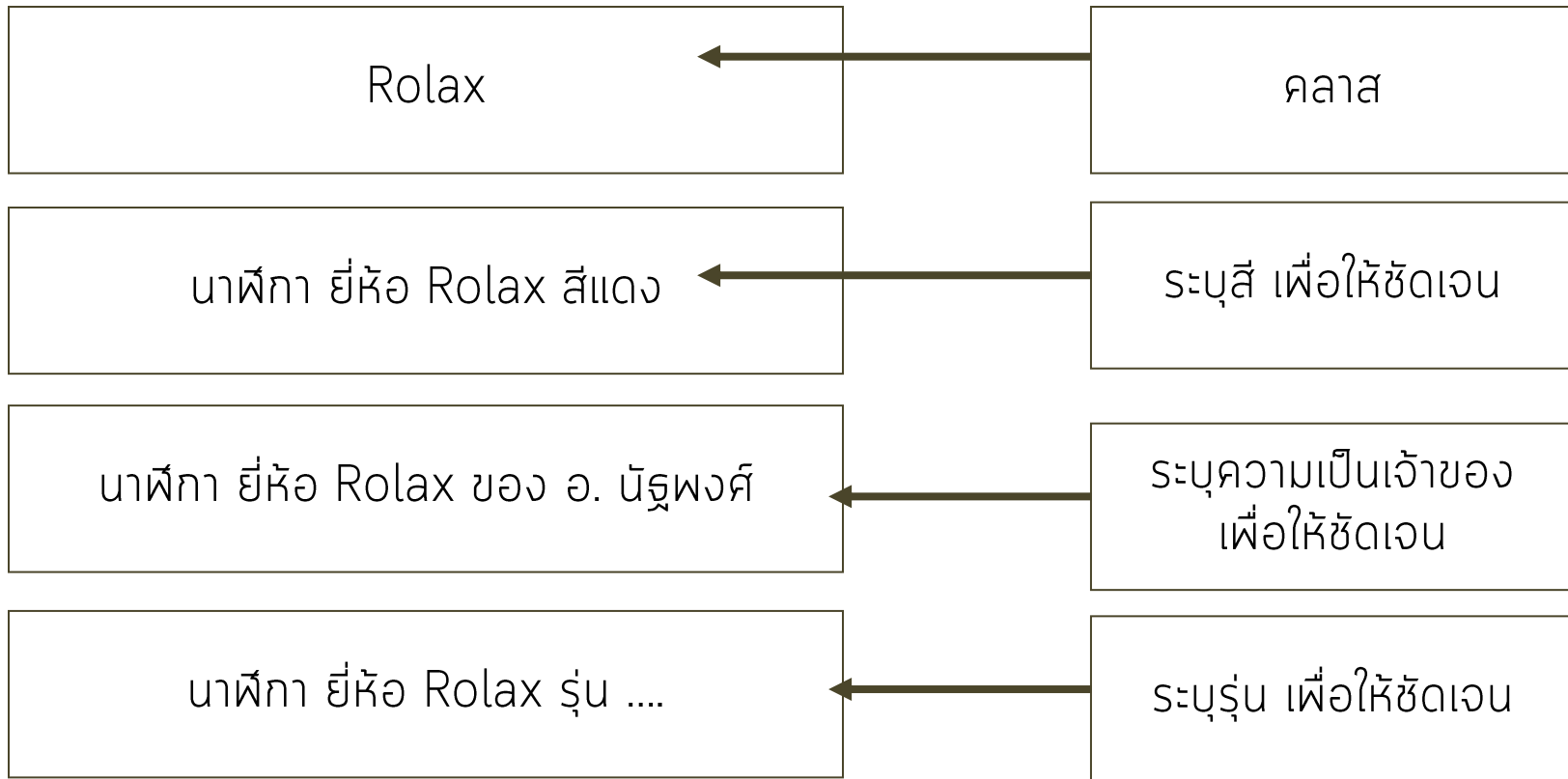
การใส่แคยี่ห้อ ยังเป็นคลาส ไม่ใช่วัตถุ

คอนเซ็ปต์ของแบรนด์ที่ทรงพลัง (Concept)



เทพสุวรรณ
6430122115112

การระบุคลาสให้เป็นวัตถุ



จเขียนแผนภาพความคิดรวบยอด(concept) ของ เสื้อเชิ้ต



ชื่อ-นามสกุล รหัส 13 หลัก

คลาสนาฬิกา

- หน้าปัด
- ถ่าน
- สายรัด
- นีอต
- ic
- เข็ม
- ตัวเลข

Problem Domain

Digital / Analog



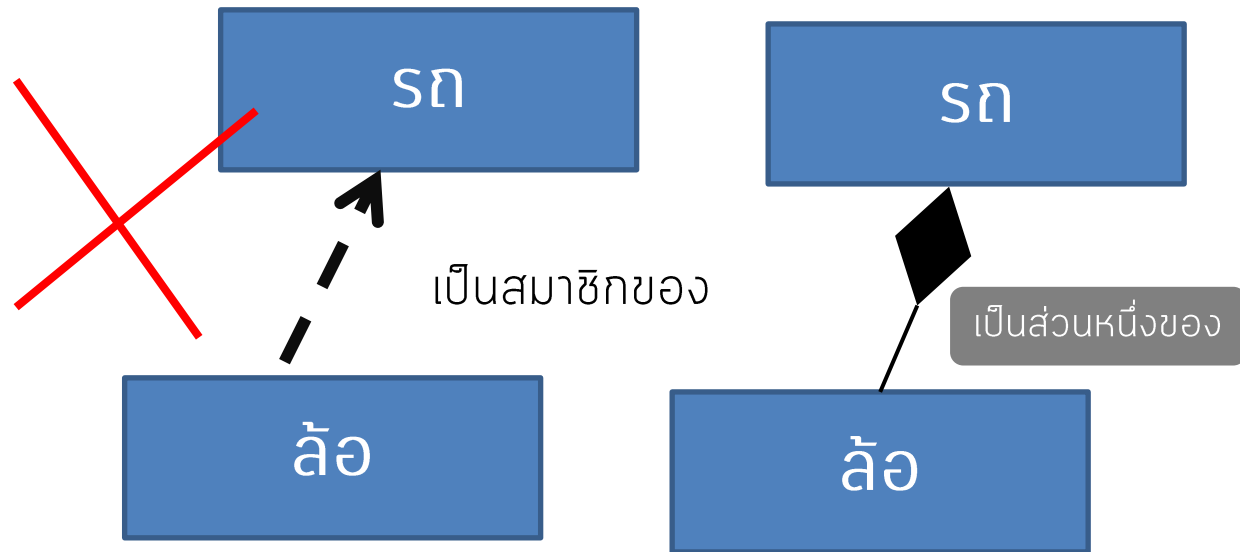
นาฬิกาข้อมือ มีส่วนประกอบอะไรบ้าง ?

- ถ่าน/แบตเตอรี่
- หน้าปัด
- สาย
- เข็มนาฬิกา หรือตัวเลขที่บอกเวลา

Problem Domain

รถยนต์ มีส่วนประกอบอะไรบ้าง ?

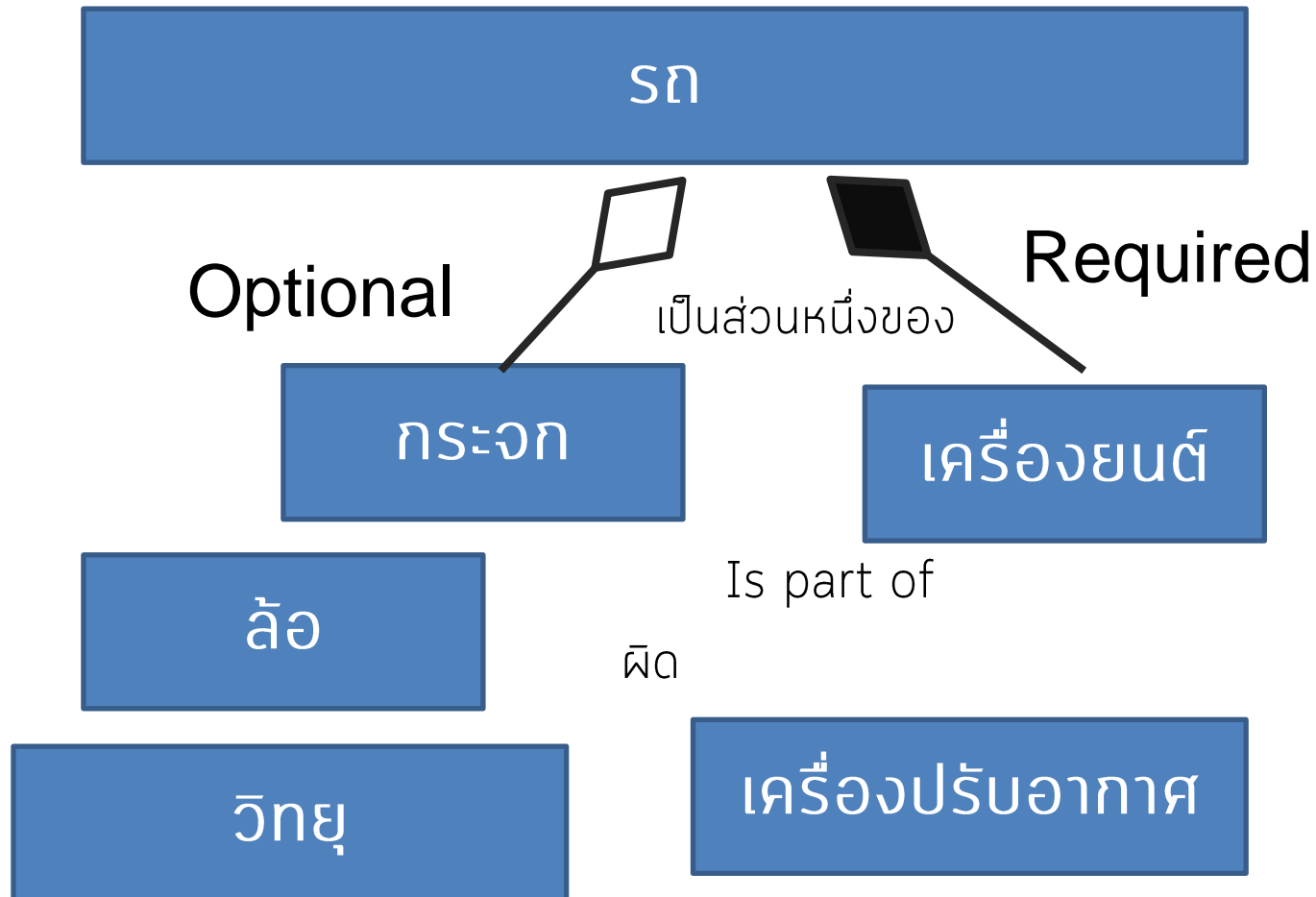
- ล้อ
- ไฟ
- กระจก
- เครื่องยนต์
- พวงมาลัย
- เบาะ
- ฯลฯ

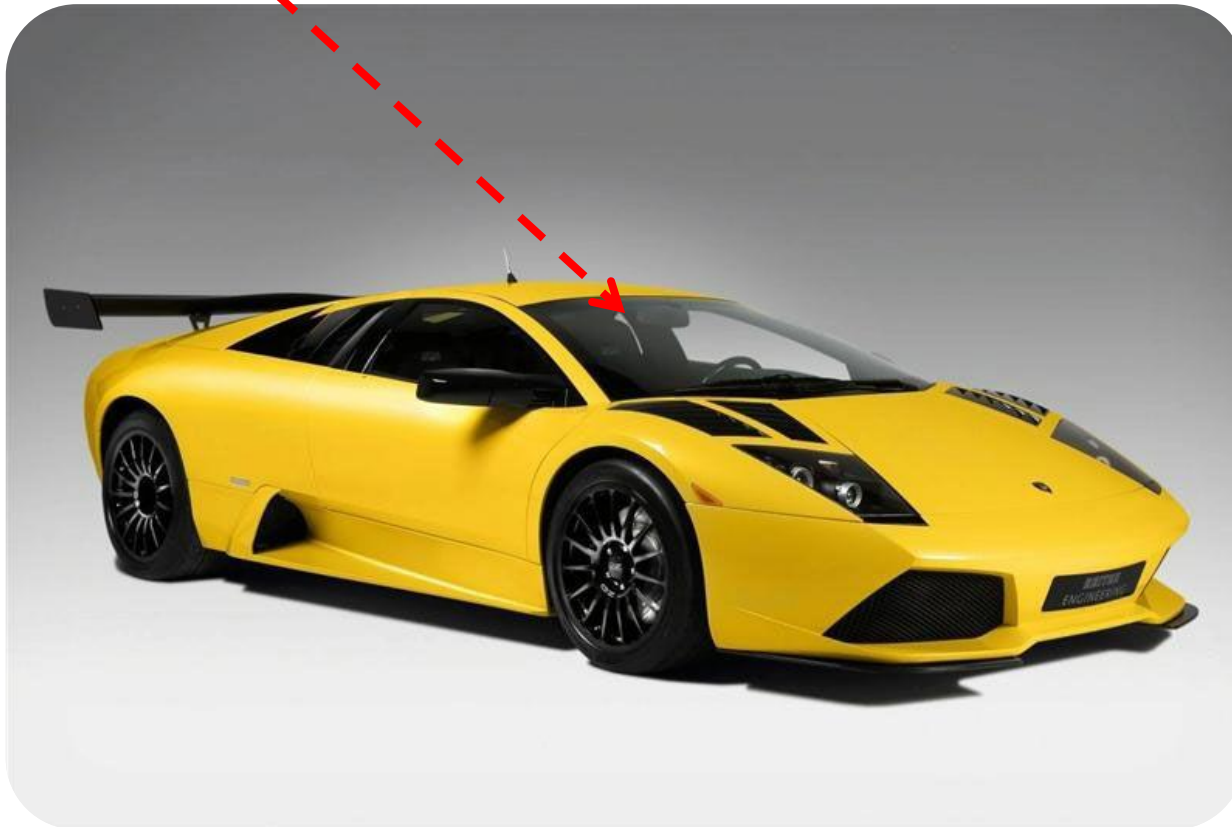


ผิด

รถยนต์ มีส่วนประกอบอะไรบ้าง ?

- ล้อ
- ไฟ
- กระจก
- เครื่องยนต์
- พวงมาลัย
- เบาะ
- ฯลฯ

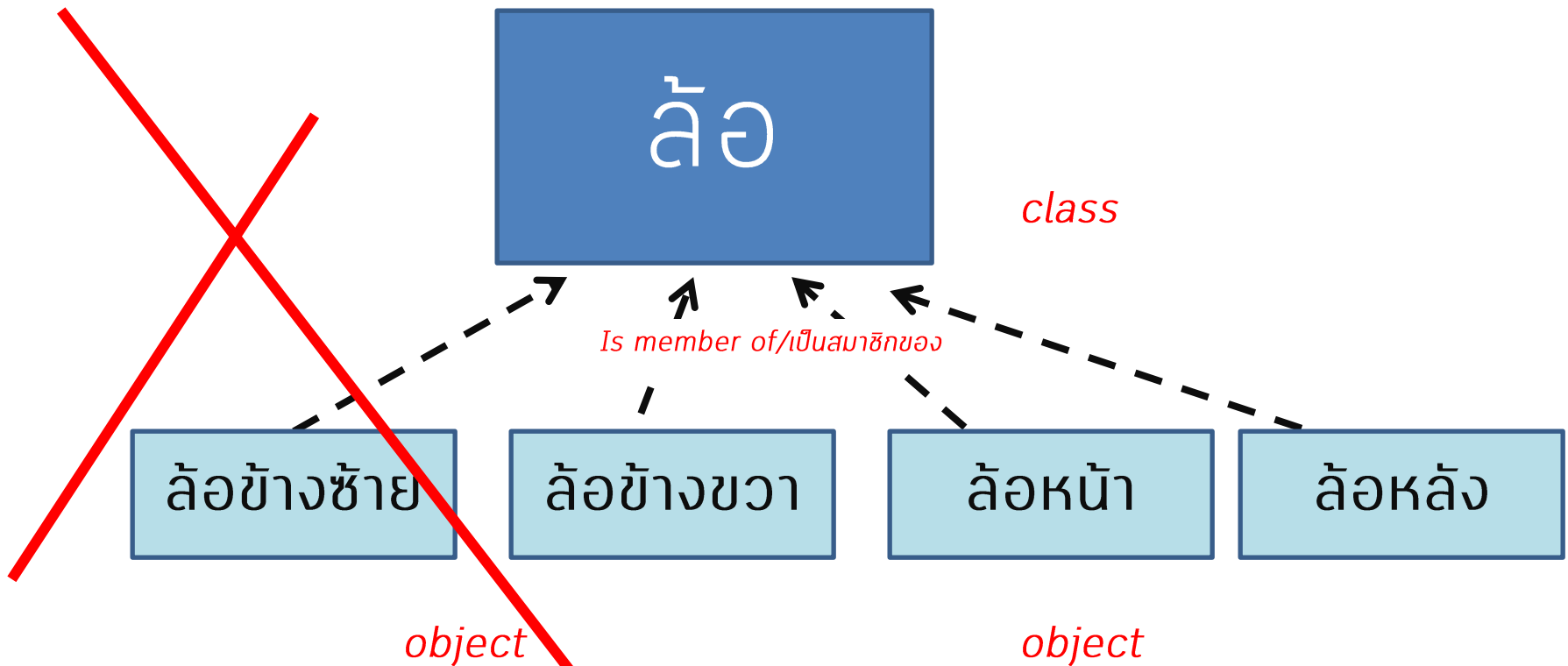




รถยนต์

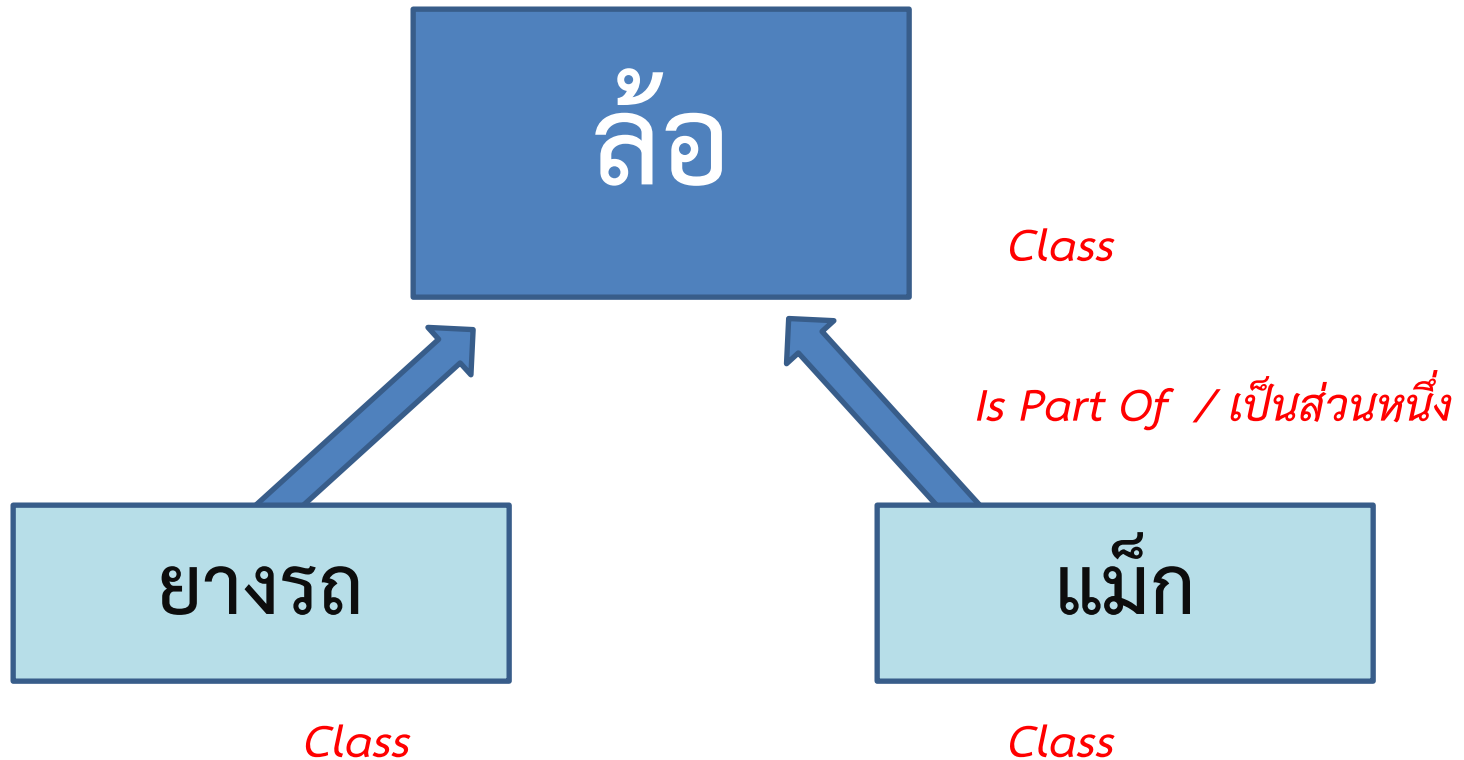
- ถ้ามีแค่ พวงมาลัย อย่างเดียว ไม่เรียกว่าเป็นรถยนต์
- ถ้ามีแค่ ล้อ อย่างเดียว ไม่เรียกว่าเป็นรถยนต์
- ถ้ามีแค่ กระจก อย่างเดียว ไม่เรียกว่าเป็นรถยนต์
- ถ้ามีแค่ ที่นั่ง อย่างเดียว ไม่เรียกว่าเป็นรถยนต์
- ดังนั้นรถยนต์ ก็คือสิ่งที่ประกอบไปด้วย ..ล้อ ที่นั่ง กระจก เครื่องยนต์ พวงมาลัย กระจัง (เนื่องจากส่วนประกอบของรถยนต์ มีหลายอย่าง จะให้บอกหมด ไม่ได้ เรียกว่า problem domain หมายถึง เราไม่สามารถสนใจทุกอย่างของรถยนต์ได้)

Classification Abstraction



*** สามารถอธิบายได้ว่า ไม่ว่าจะจะเป็นล้อข้างซ้าย และข้างขวา จัดว่าเป็นล้อรถ นั่นเอง

Aggregation Abstraction

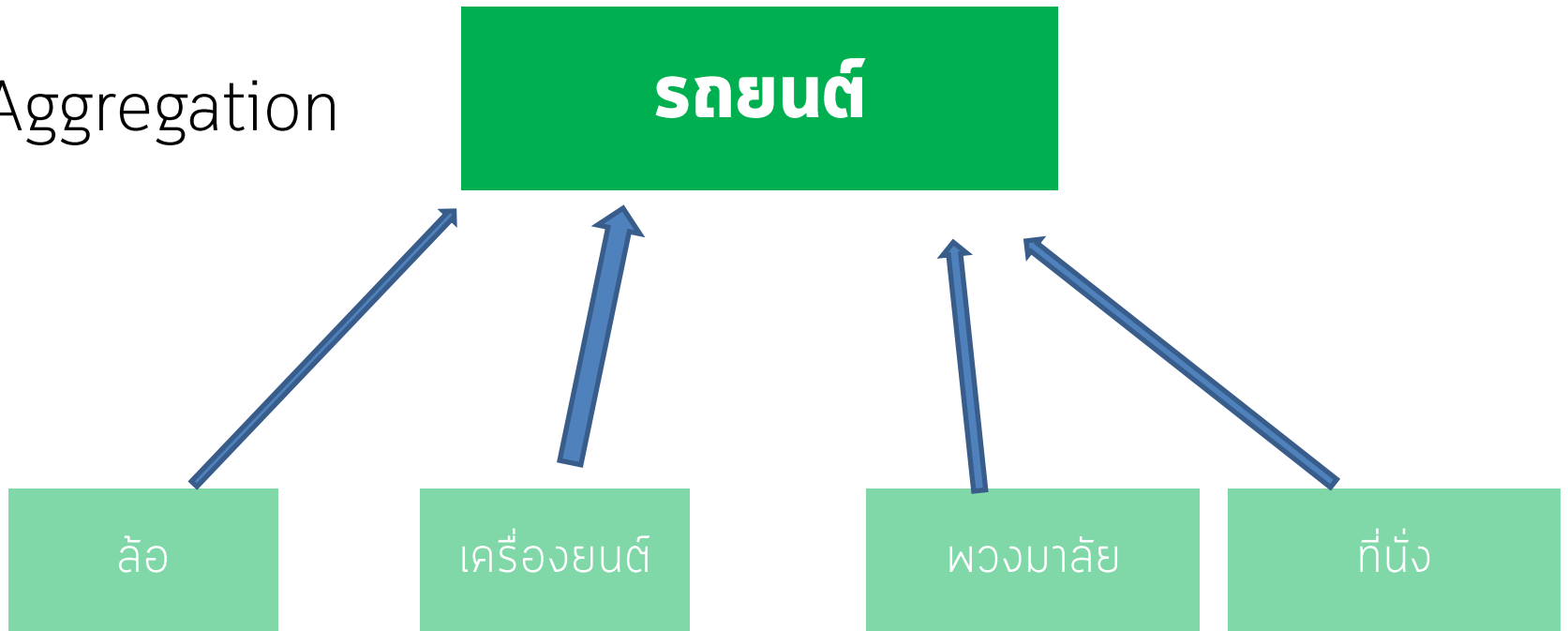


***** สามารถอธิบายได้ว่า ยางรถยนต์ ประกอบไปด้วย ยางรถ และ แม่เหล็ก**

ส่วนประกอบของรถยนต์



Aggregation



รถยนต์ไม่สามารถมีแค่อย่างเดียว หรือ ไม่สามารถมีพวงมาลัยอย่างเดียวได้

นำคลาสต่าง ๆ มารวมกันหรือประกอบกันขึ้นมาทำให้ได้ concept ใหม่ คือรถยนต์

Aggregation

รถยนต์

ได้ concept ใหม่
จากการรวมของ
คลาส ย่อย ๆ

ล้อ

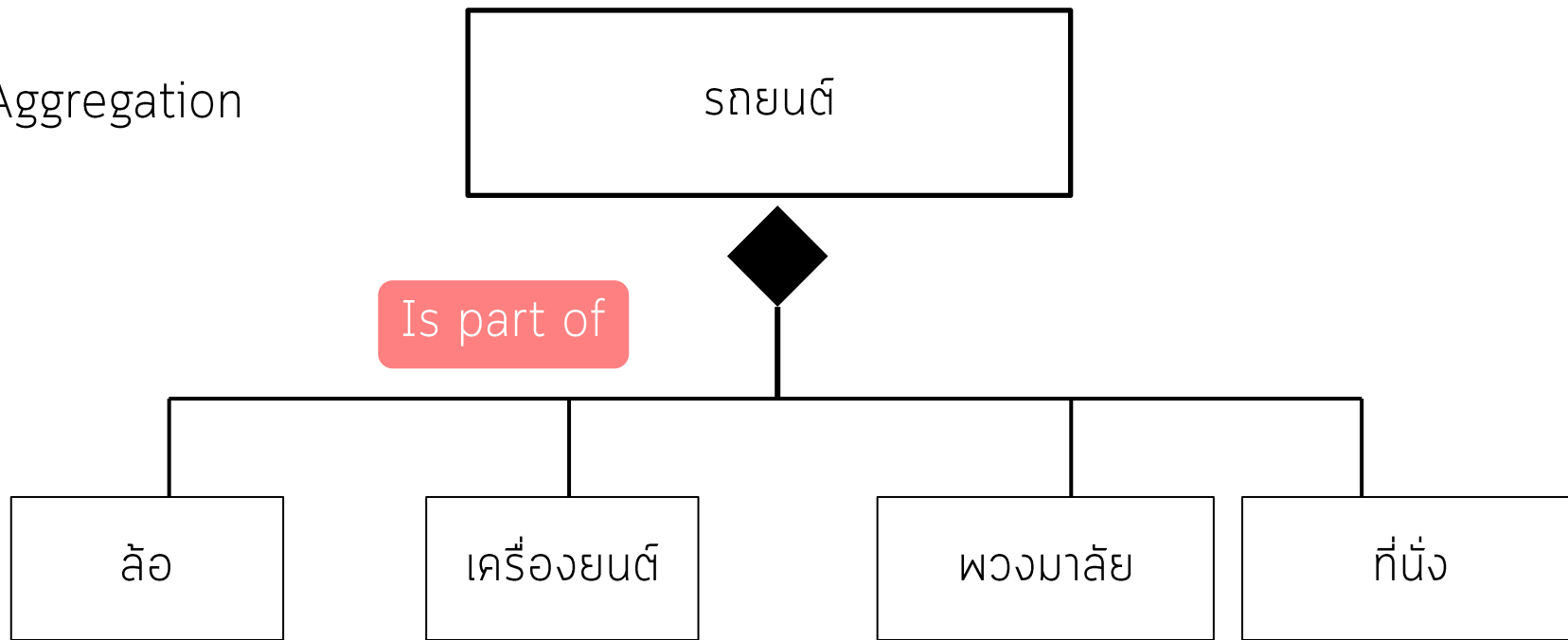
เครื่องยนต์

พวงมาลัย

ที่นั่ง

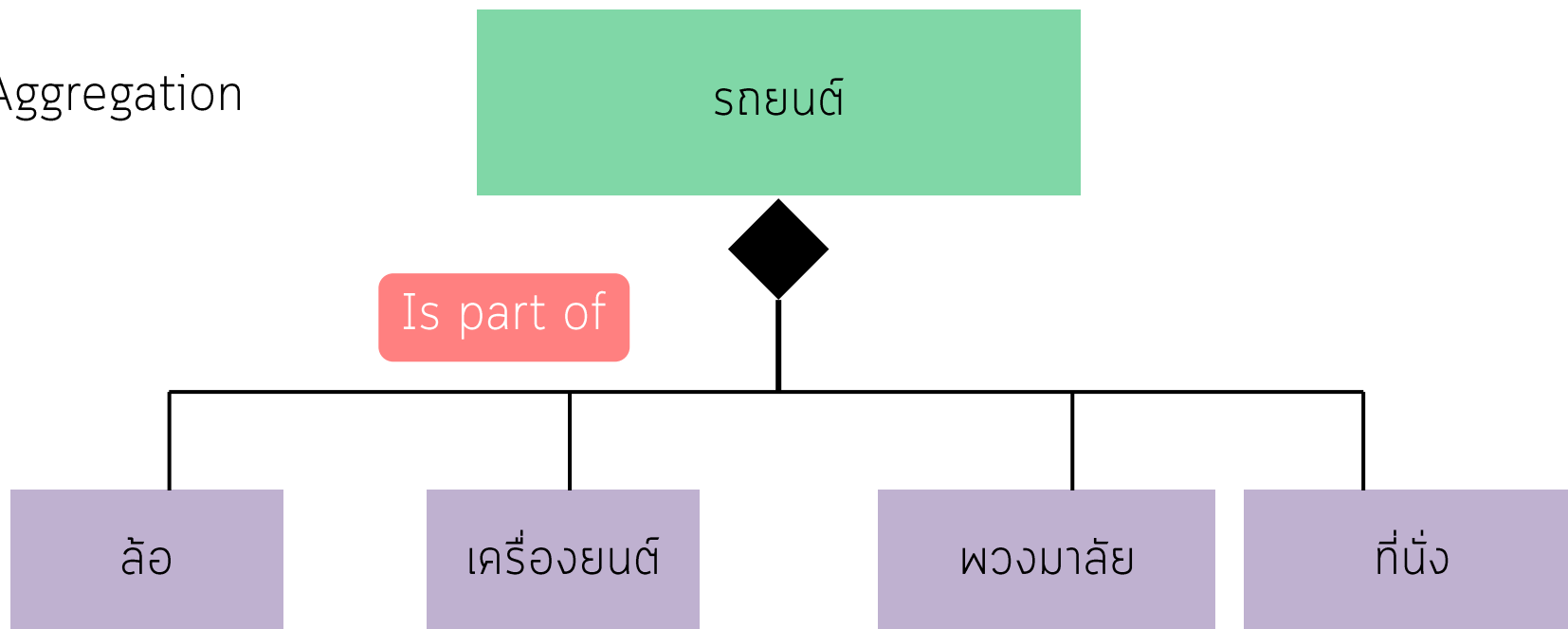
นำคลาสต่าง ๆ มารวมกันหรือประกอบกันขึ้นมาทำให้ได้ concept ใหม่ คือรถยนต์

Aggregation



นำคลาสต่าง ๆ มารวมกันหรือประกอบกันขึ้นมาทำให้ได้ concept ใหม่ คือรถยนต์

Aggregation



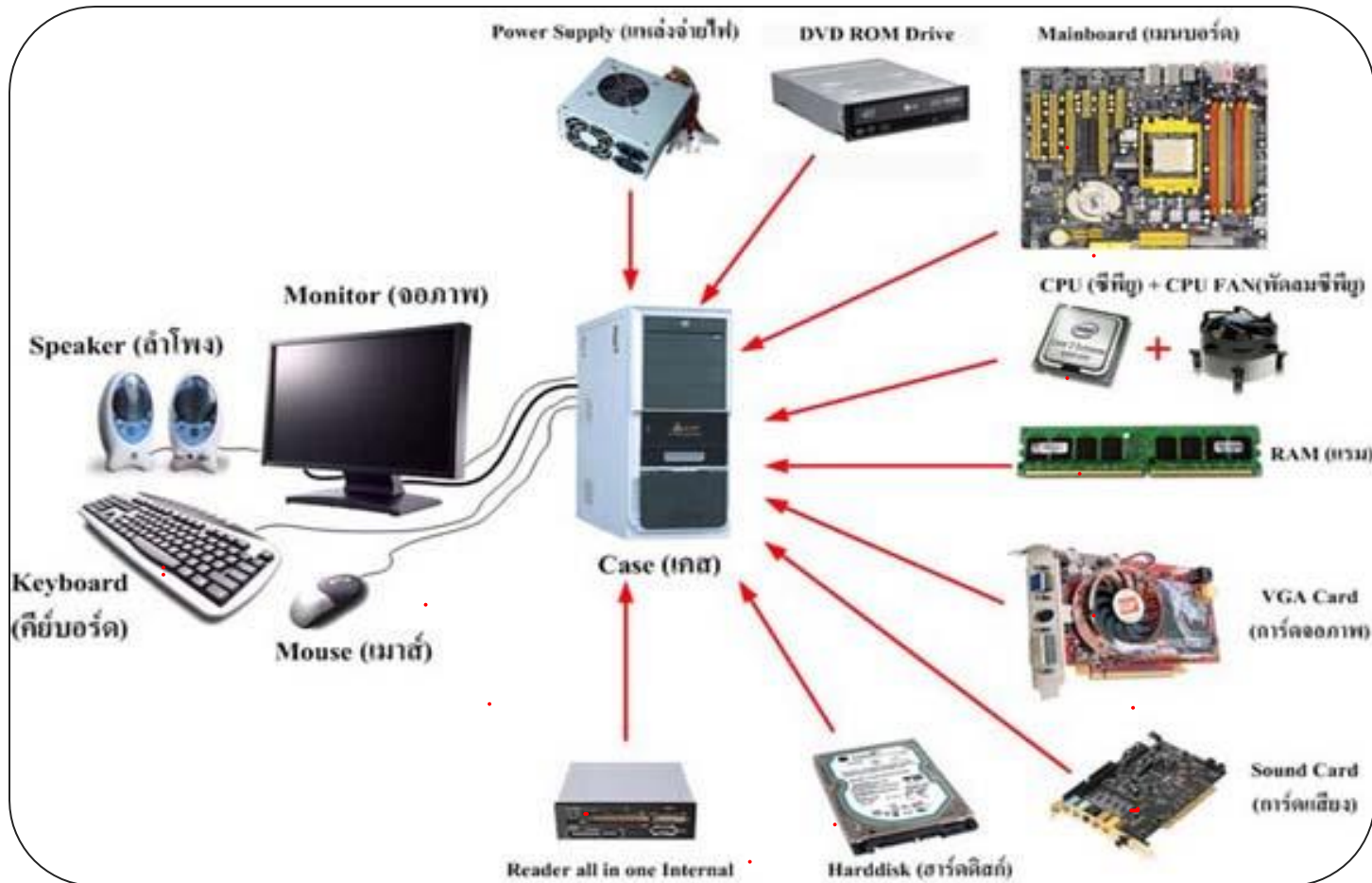
นำคลาสต่าง ๆ มารวมกันหรือประกอบกันขึ้นมาทำให้ได้ **concept** ใหม่ คือรถยนต์

PDT: Primitive Data Type
(integer , floating-point , String , date/time)

รถยนต์
<ul style="list-style-type: none">- รุ่น : String / Text- ยี่ห้อ : String / Text- ราคา : floating- ล้อ : Object- พวงมาลัย : Object- เครื่องยนต์ : Object
<ul style="list-style-type: none">+ กำหนดราคา (price)+ บอกรุ่น () : String+ บอกสี () : String+ บอกราคา () : float+ เลี้ยวได้ ()+ หยุดได้ ()

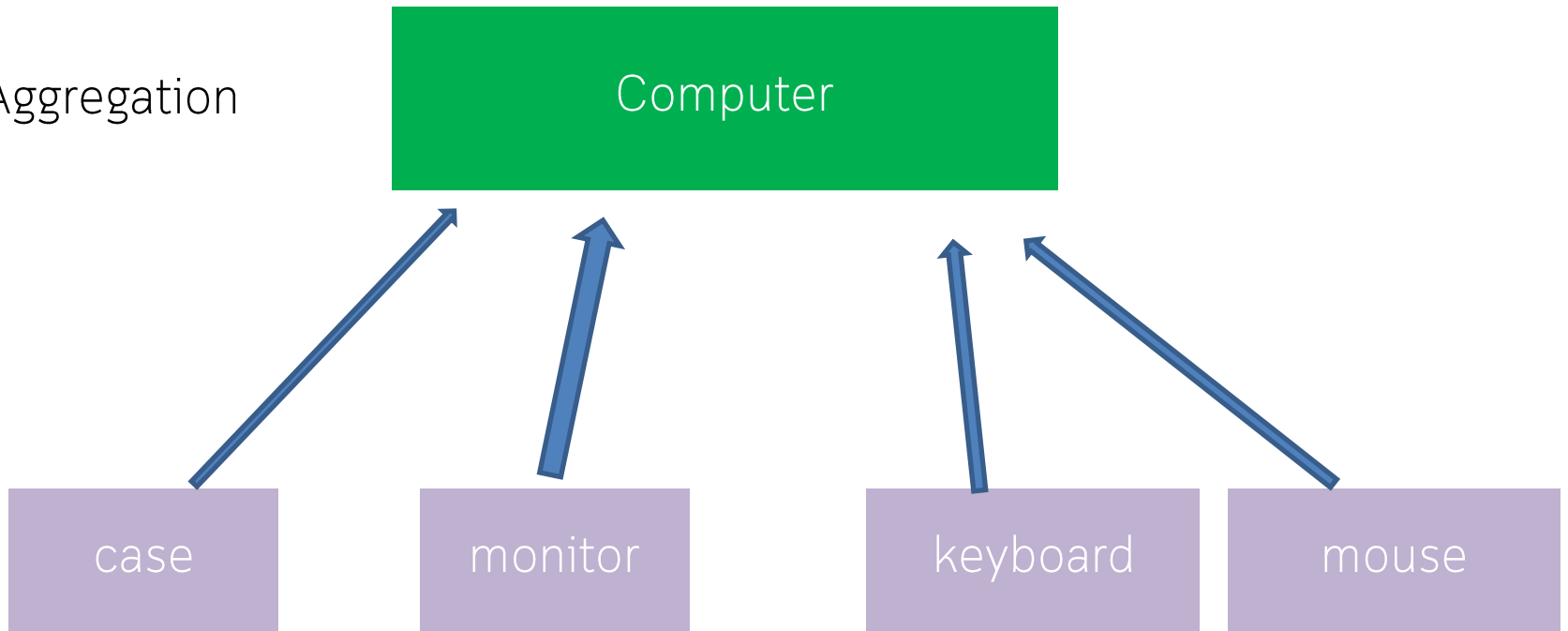
ADT: Abstract Data Type
(Class/ Object)

ส่วนประกอบของคอมพิวเตอร์



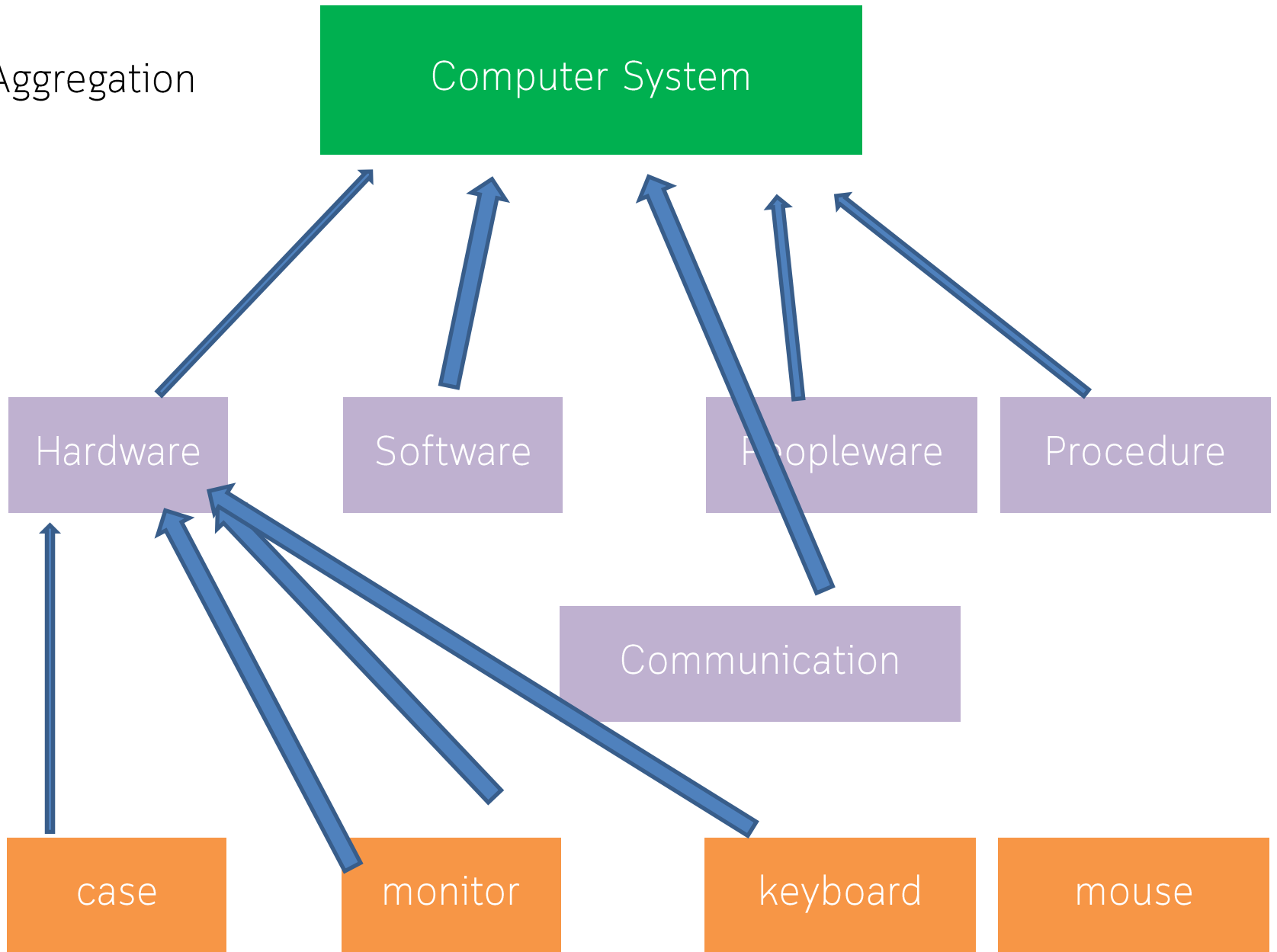
รูปภาพ : <http://gobgabsocool.blogspot.com/2013/03/blog-post.html>

Aggregation

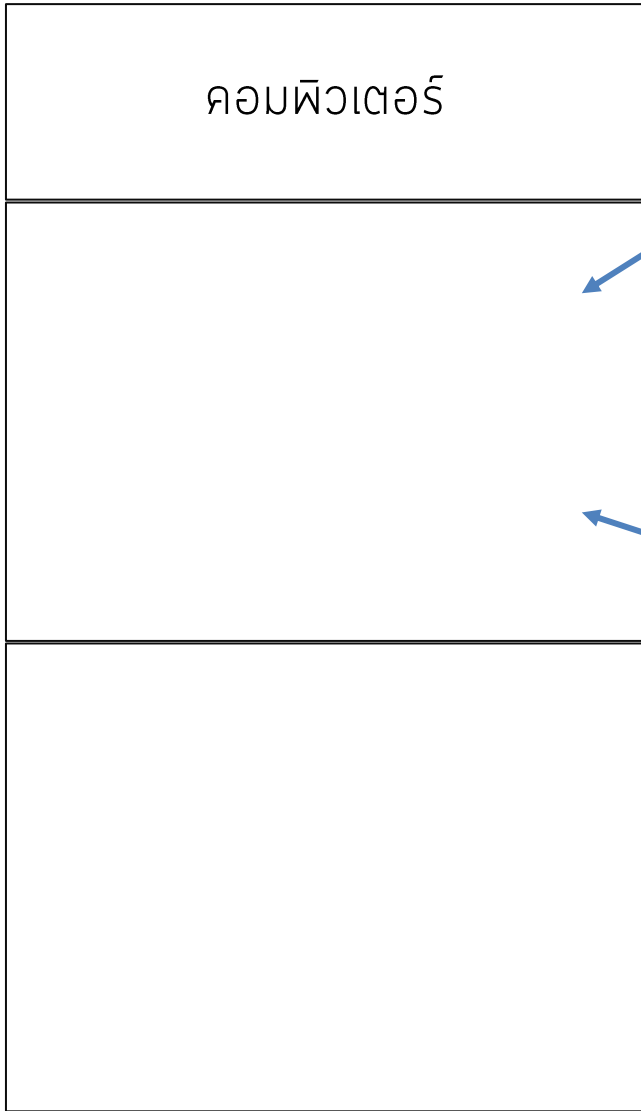


นำคลาสต่าง ๆ มารวมกันหรือประกอบกันขึ้นมาทำให้ได้ concept ใหม่ คือคอมพิวเตอร์

Aggregation

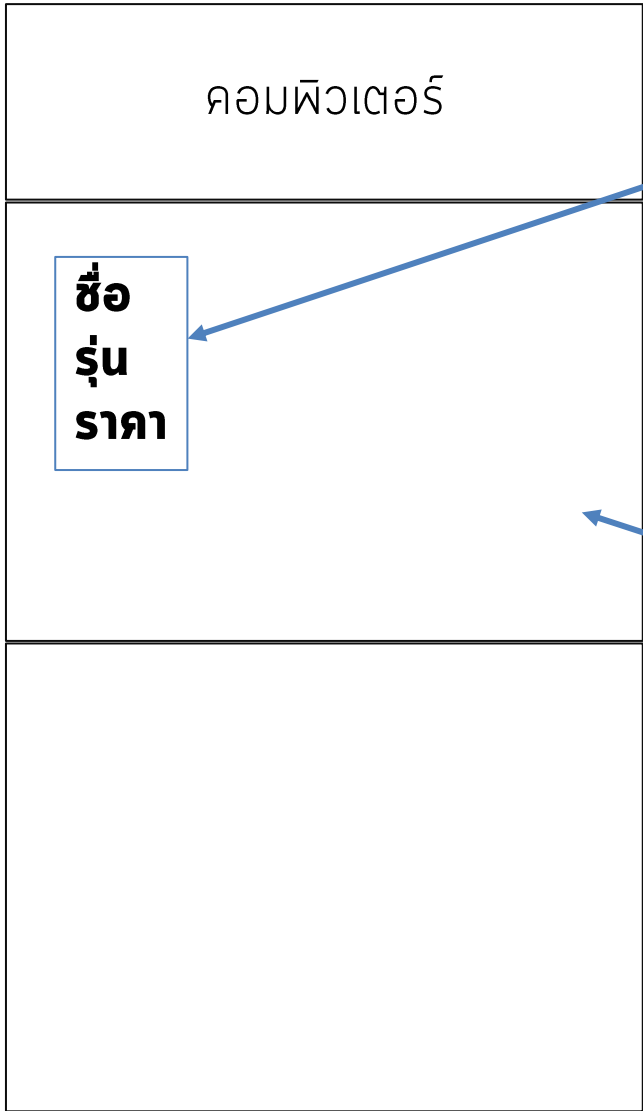


PDT: Primitive Data Type
(integer , floating-point , String , date/time)



ADT: Abstract Data Type
(Class/ Object)

PDT: Primitive Data Type
(integer , floating-point , String , date/time)



ADT: Abstract Data Type
(Class/ Object)

PDT: Primitive Data Type
(integer , floating-point , String , date/time)

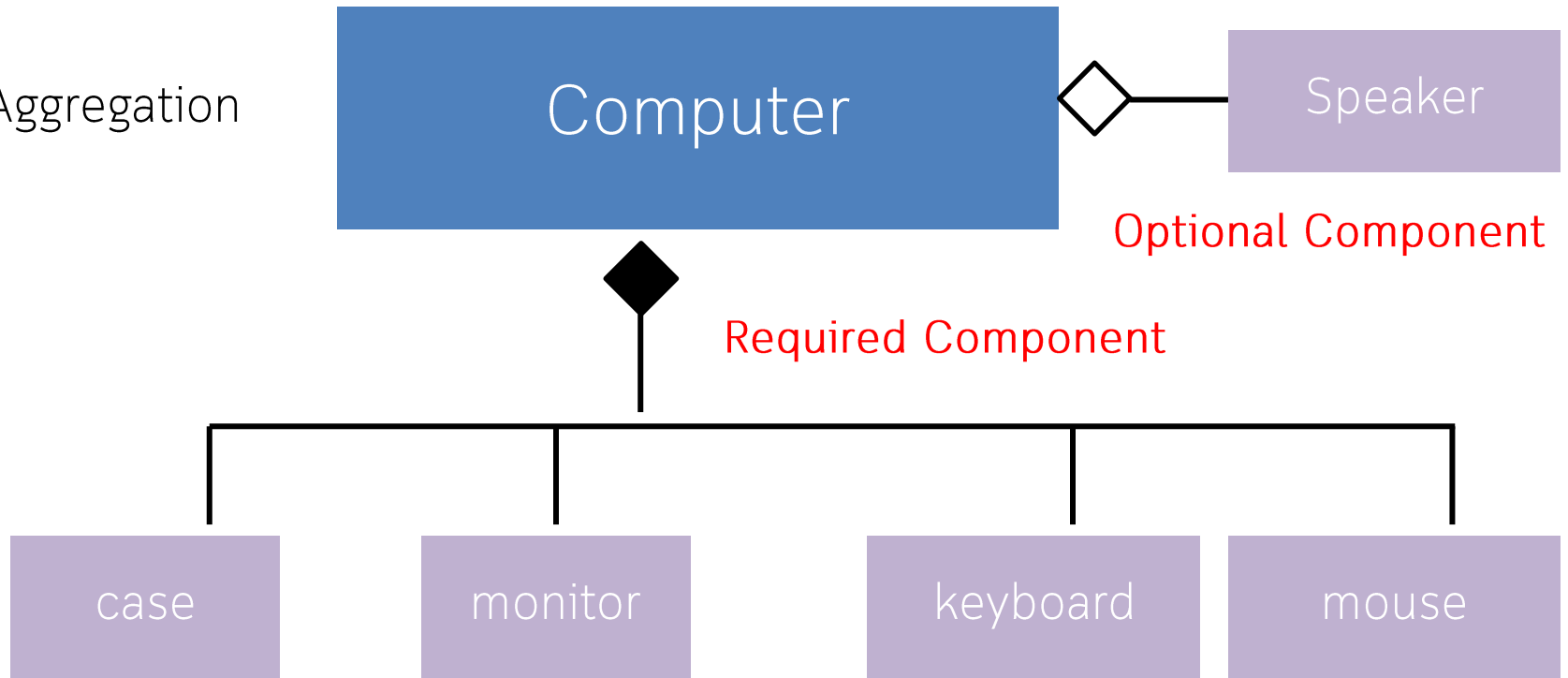
คอมพิวเตอร์

ชื่อ : Textual
รุ่น : Textual
ราคา : Floationg-Point

เมาส์ : Mouse
คีย์บอร์ด : Keyboard
จอภาพ : Monitor
ลำโพง : Speaker

ADT: Abstract Data Type
(Class/ Object)

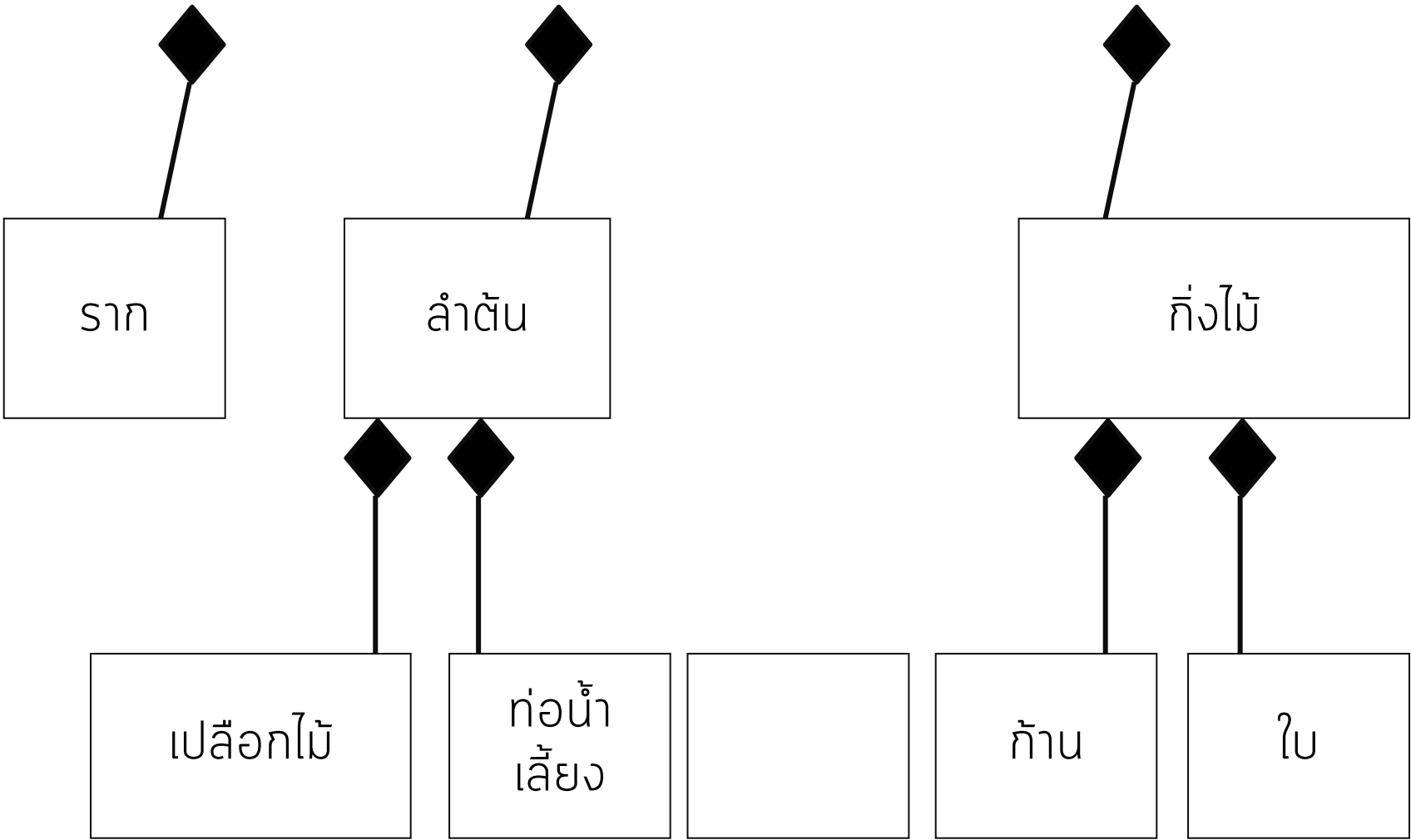
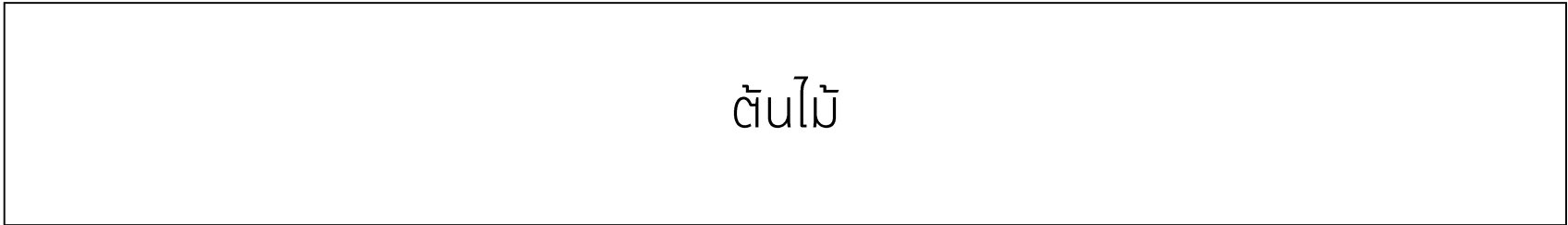
Aggregation



นำคลาสต่าง ๆ มารวมกันหรือประกอบกันขึ้นมาทำให้ได้ concept ใหม่ คือคอมพิวเตอร์

Aggregation Abstraction

- ส่วนประกอบชิ้นเล็ก ๆ รวมกันทำให้เกิด concept ใหม่ เรียกว่าการทำ aggregation
- ส่วนประกอบชิ้นใหญ่ขึ้น รวมกัน ทำให้เกิด concept ใหม่ เหมือนกัน ก็เรียกการทำ aggregation



TitleBar

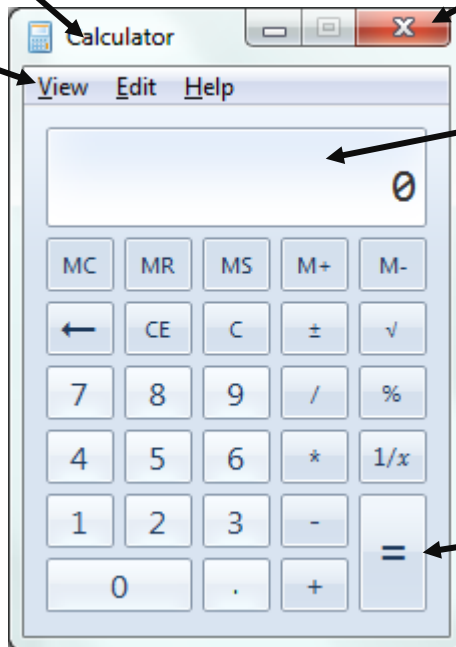
Control box

Menu bar

TextBox

M_view

M_edit



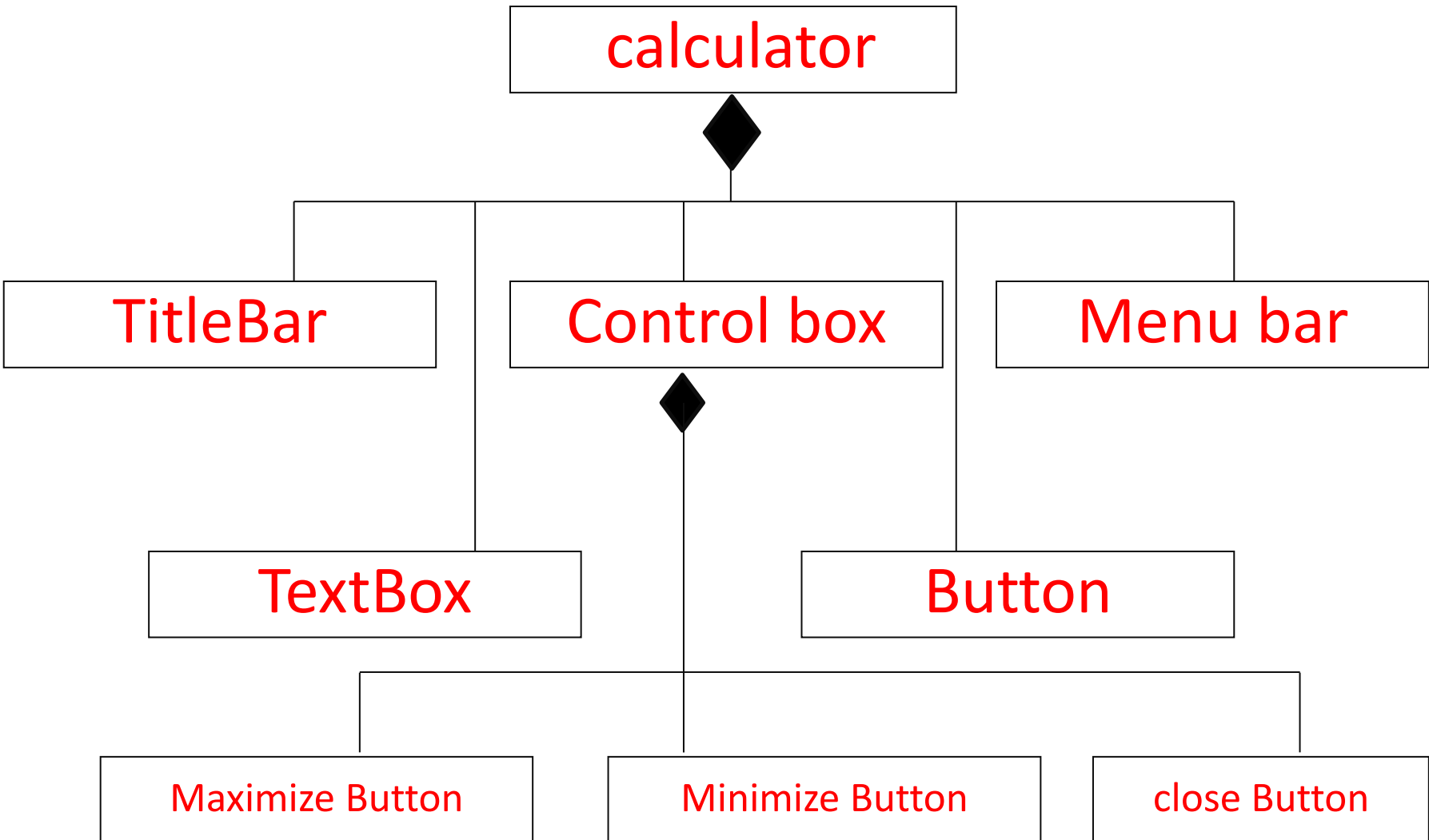
TxtResult

Button

b0

b1

มีข้อความ Text
และ คลิ๊กได้ button1_click()



Required/Optional Component

- Required Component คือ ส่วนประกอบที่จำเป็น
- Optional Component คือ ส่วนประกอบที่ไม่จำเป็น

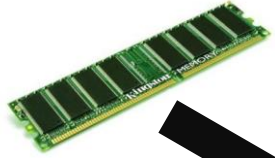


Required Component



Optional Component

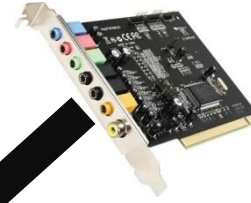
หน่วยความจำ(RAM)



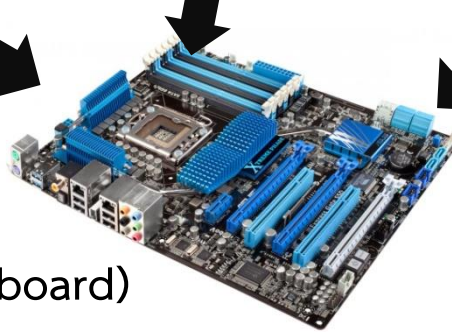
หน่วยประมวลผล(CPU)



ซาว์นการ์ด



เมนบอร์ด(Mainboard)



จอภาพ(Monitor)



คีย์บอร์ด(Keyboard)



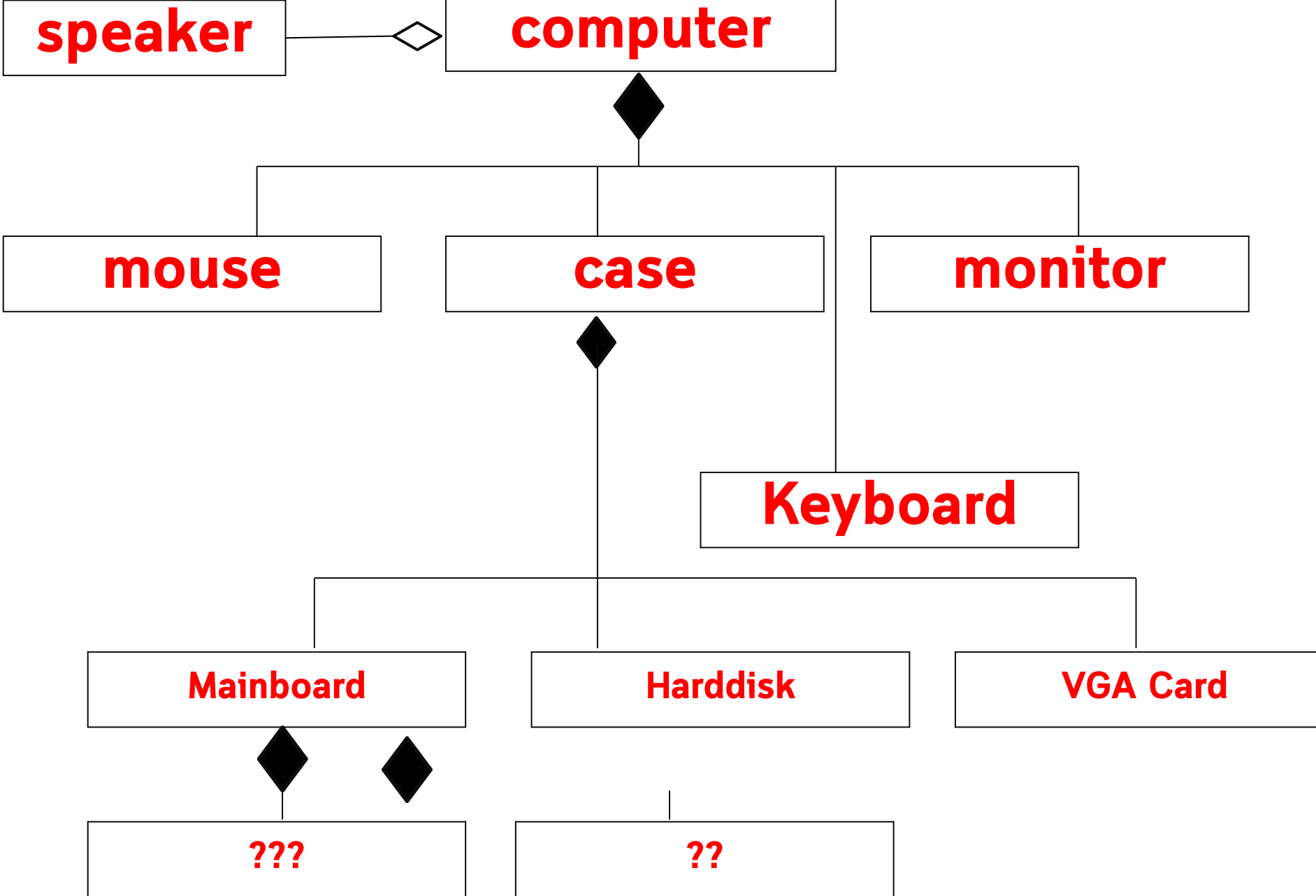
คอมพิวเตอร์(Computer)

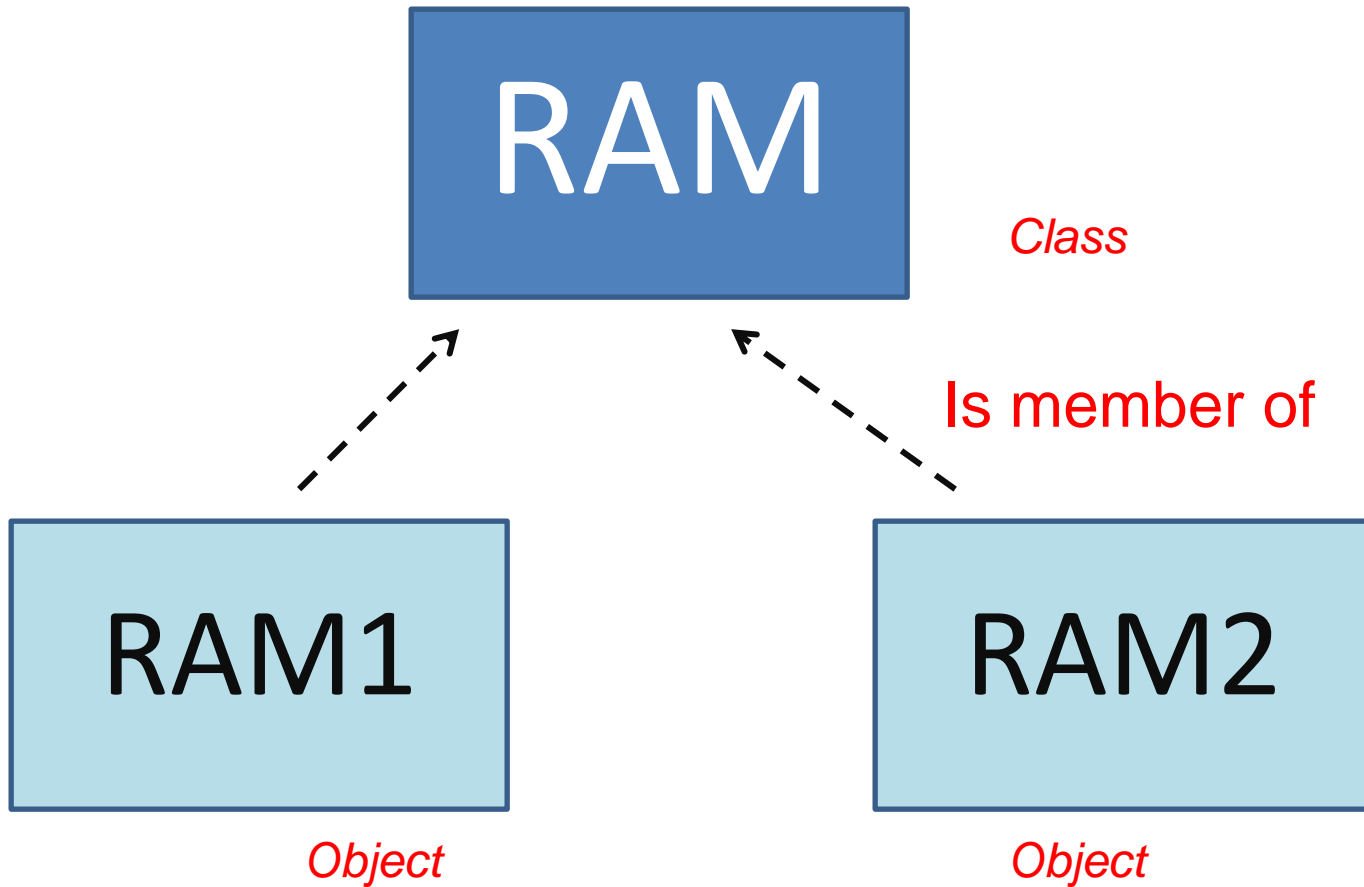


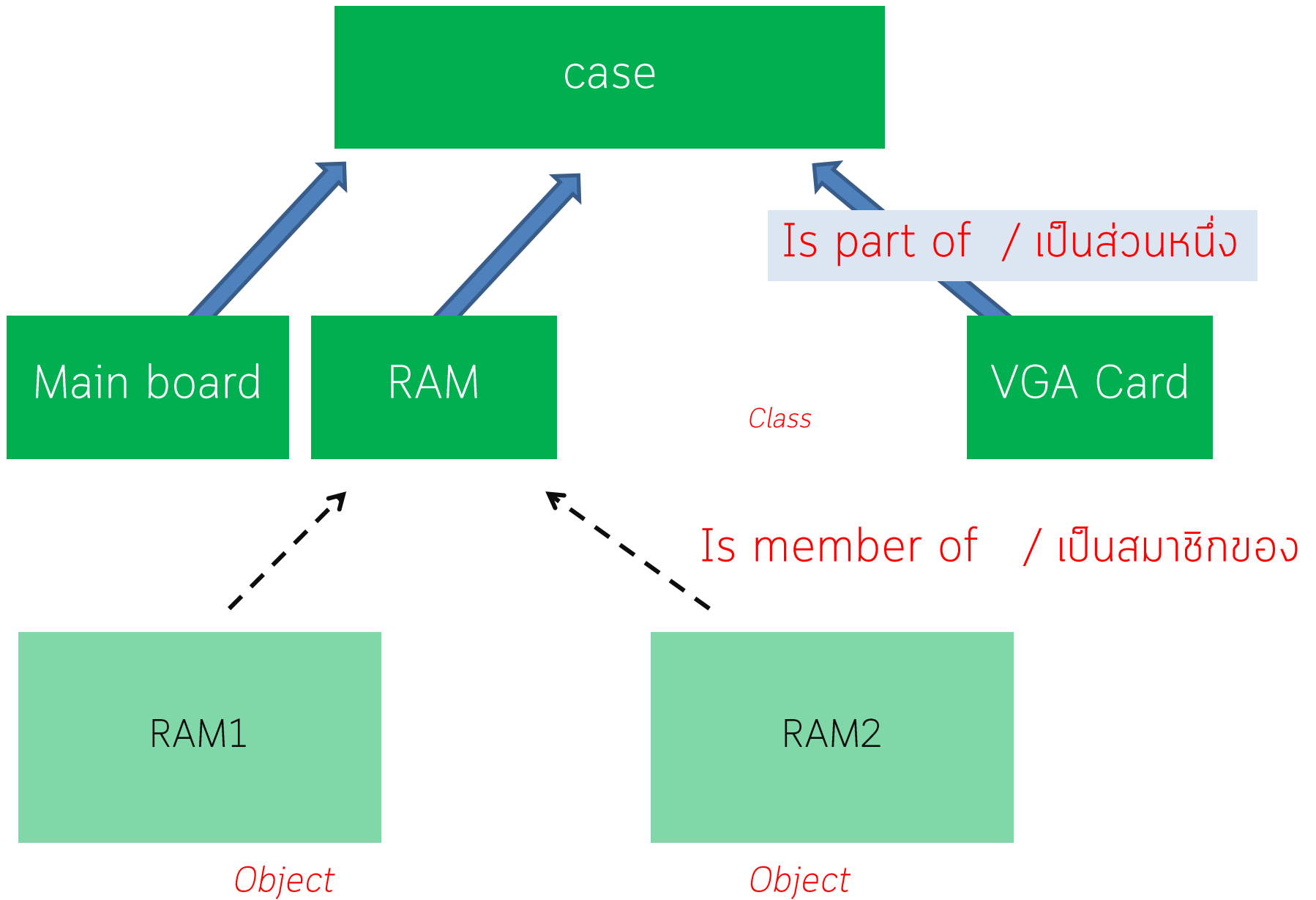
เมาส์(Mouse)



<http://www.siam2dev.com> [dr. nattapong songneam]







จงอธิบาย สิ่งต่อไปนี้ ในเชิงของ Aggregation Abstraction

- คน ก็ ประกอบไปด้วย หัว ลำตัว แขน ขา ในขณะที่หัวประกอบไปด้วย ผม ตา หู จมูก ปาก **สมอง**

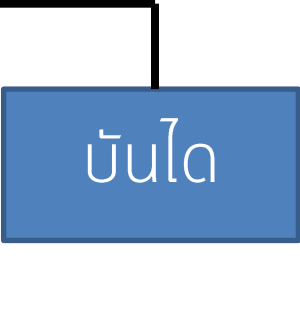
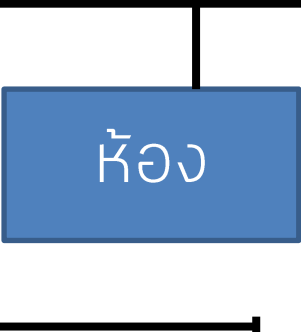
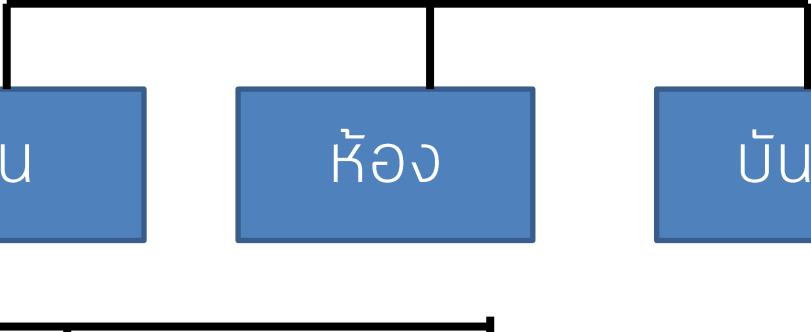
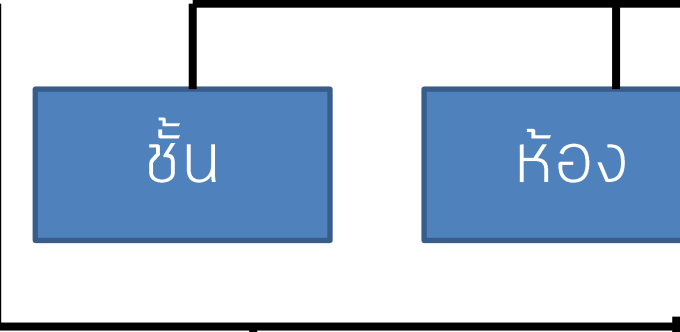
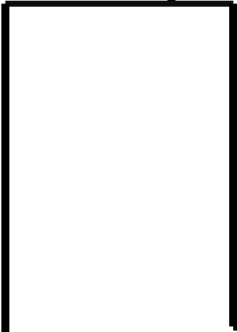
- โรงพยาบาล ---โรงพยาบาลประกอบไปด้วย หมอ พยาบาล คนไข้ ยา อาหาร ถ้าพิจารณาที่อาคารยังประกอบไปด้วยห้องต่าง ๆ ในขณะที่ห้องก็ยังประกอบไปด้วยอุปกรณ์ เติง เป็นต้น
- มหาวิทยาลัย --- ...
- วัด --- ...
- โทรศัพท์มือถือ--- ...
- คลินิกรักษาโรค --- ...

มหาวิทยาลัย



บุคลากร

อาคาร



ชั้น

ห้อง

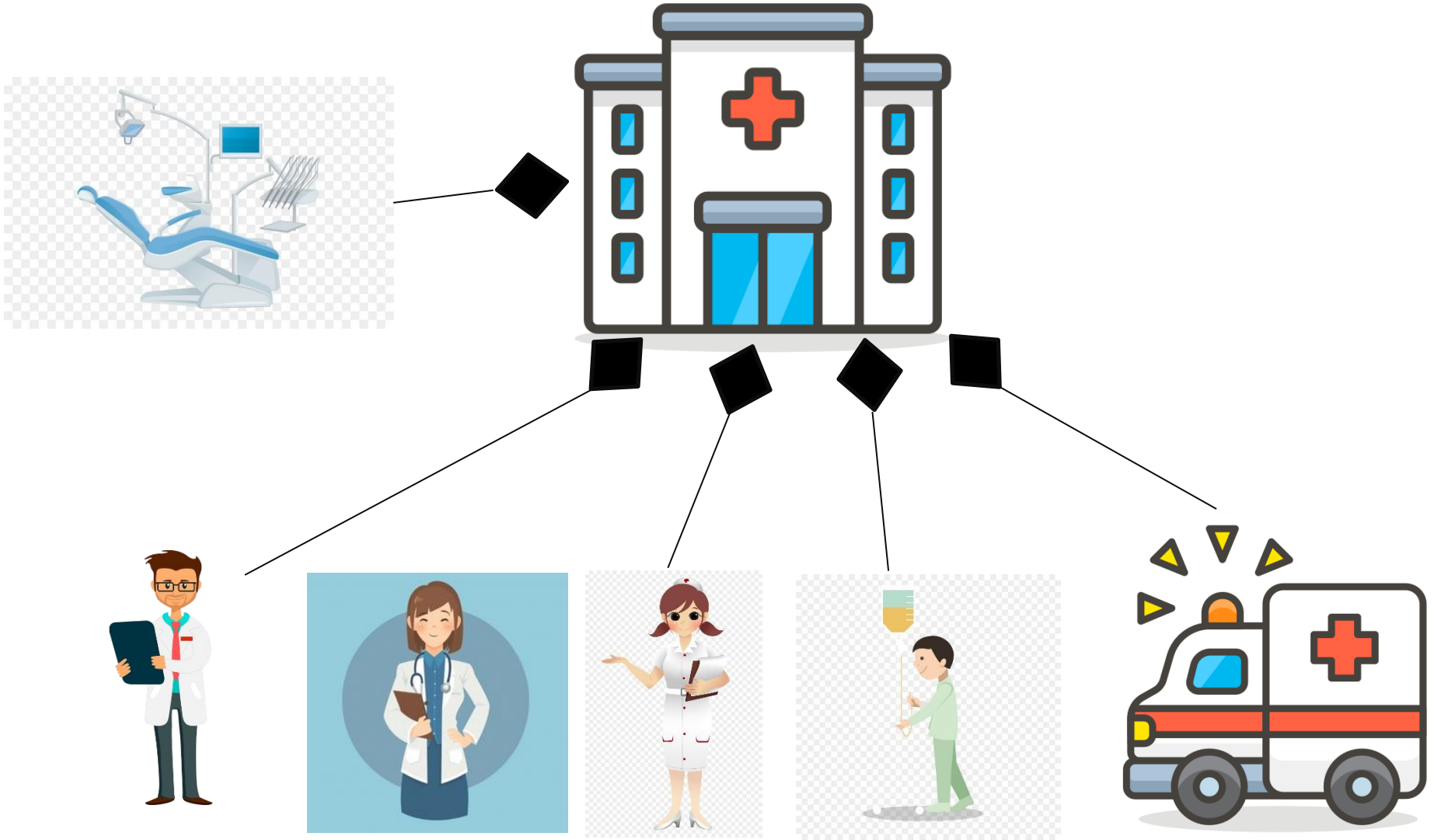
บันได

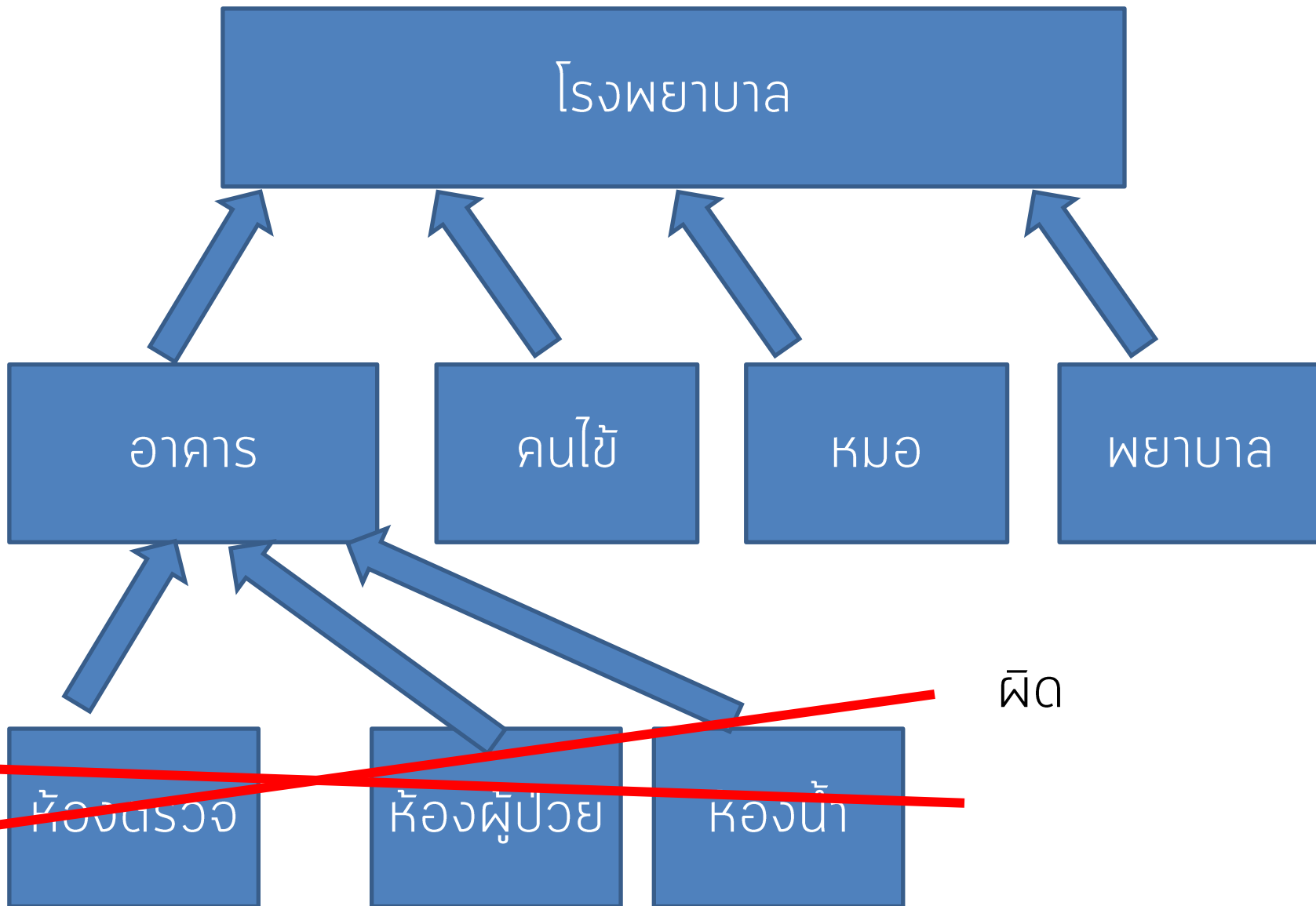
ลิฟท์

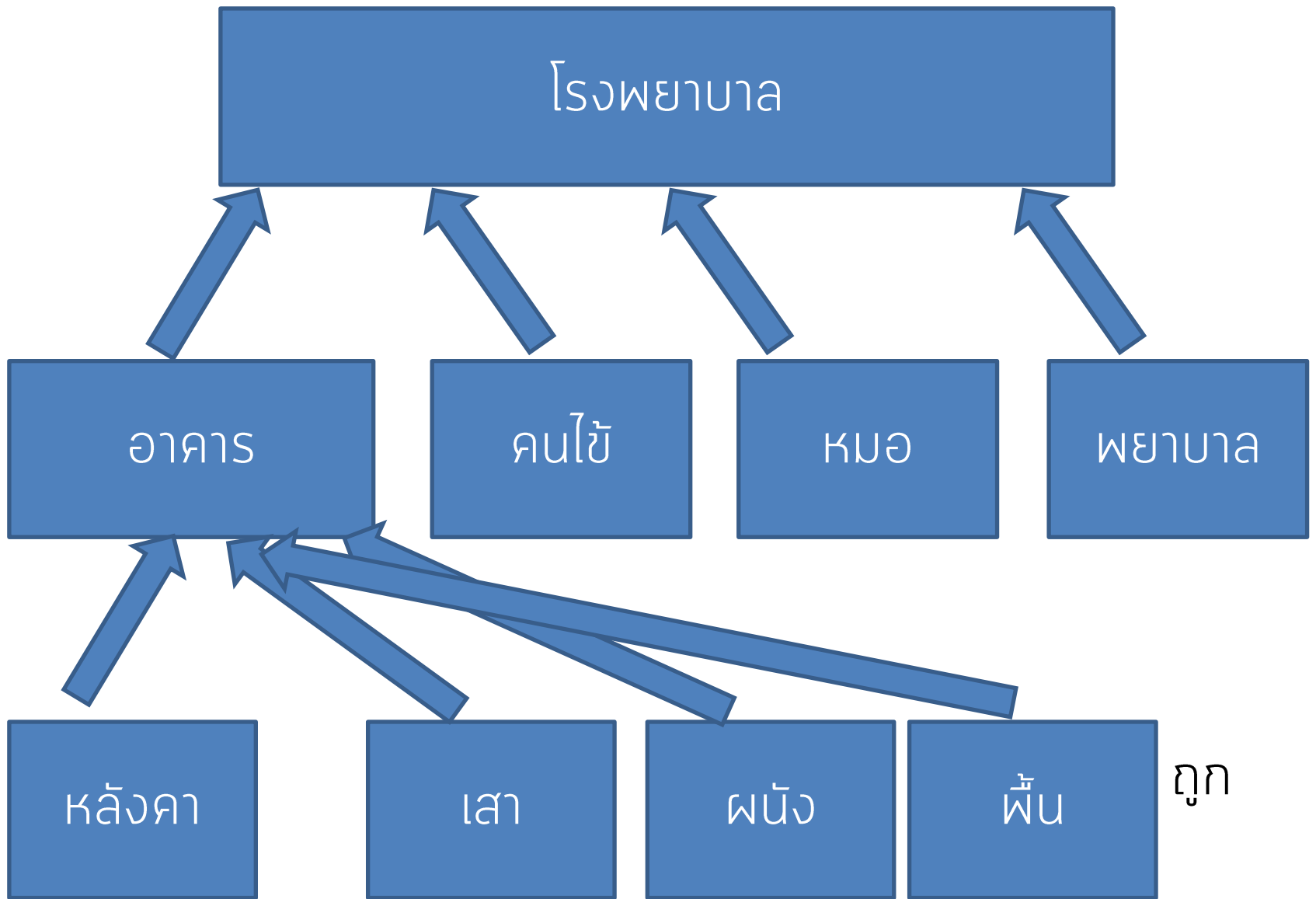
อาจารย์

เจ้าหน้าที่

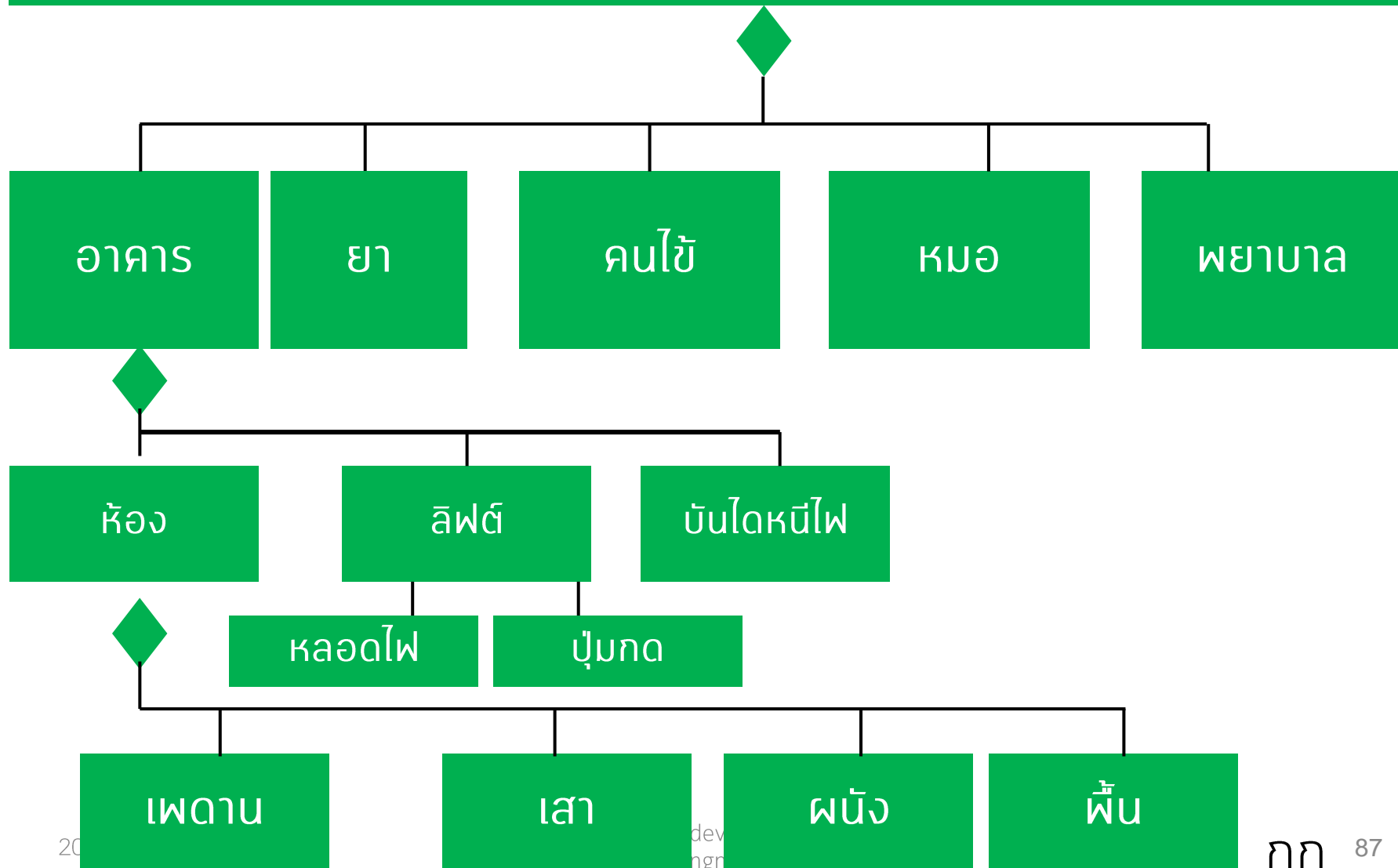
นักศึกษา



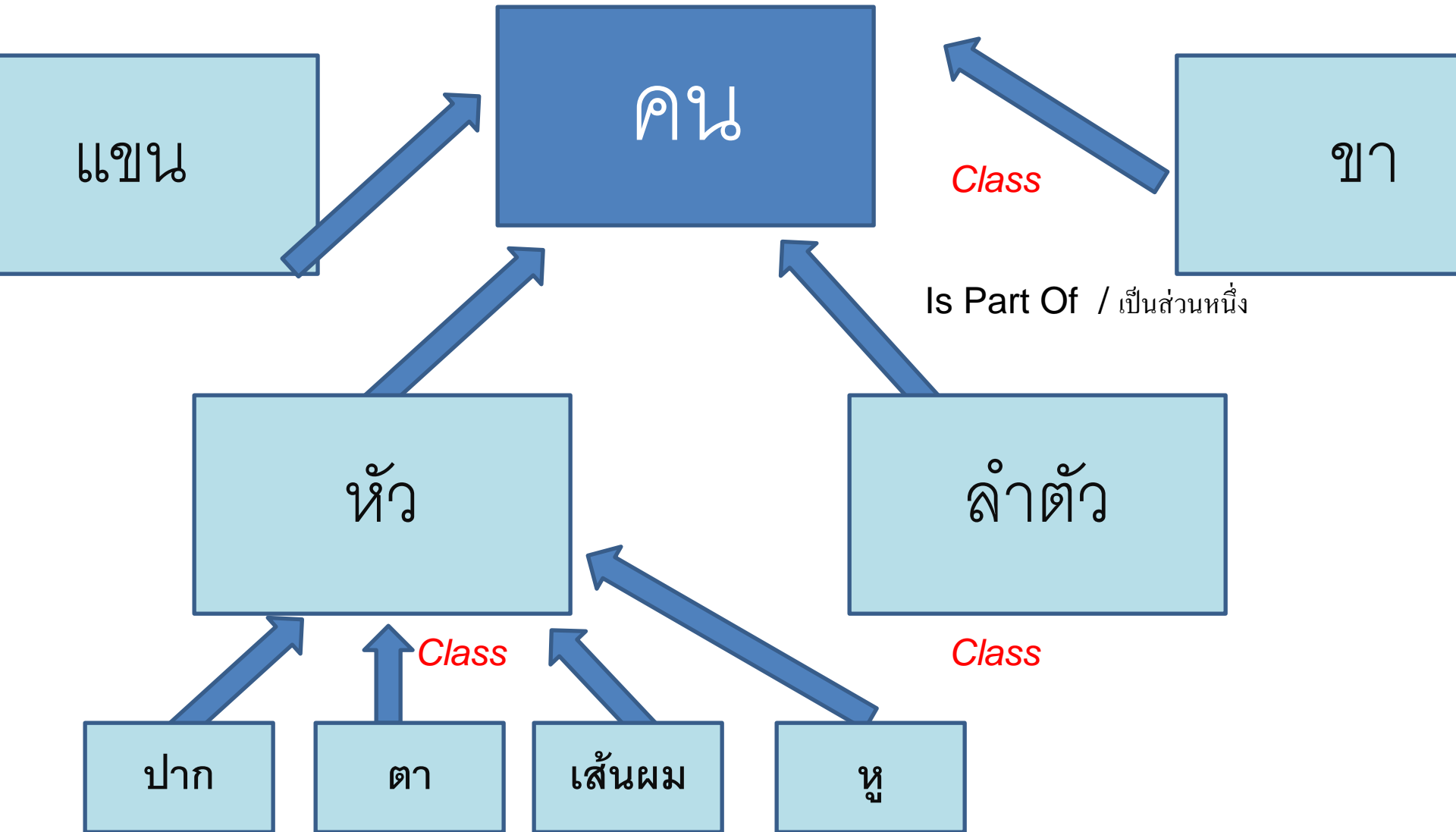




โรงพยาบาล



Aggregation Abstraction



ตัวอย่าง 1

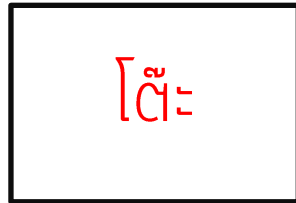
“ห้องเรียนประกอบด้วย กระดานดำ 1 กระดาน มี
เก้าอี้และโต๊ะจำนวนหนึ่ง มีนักศึกษา มีอาจารย์”

จากข้อความข้างต้น สามารถสรุปได้ว่า class
กระดานดำ class โต๊ะ class เก้าอี้ class นักศึกษา
class อาจารย์ เมื่อนำคลาสเหล่านี้มารวมกันจะได้
class ใหม่ หรือทำให้ได้ความคิดรวบยอดใหม่ นั่นก็
คือ **class ห้องเรียน**

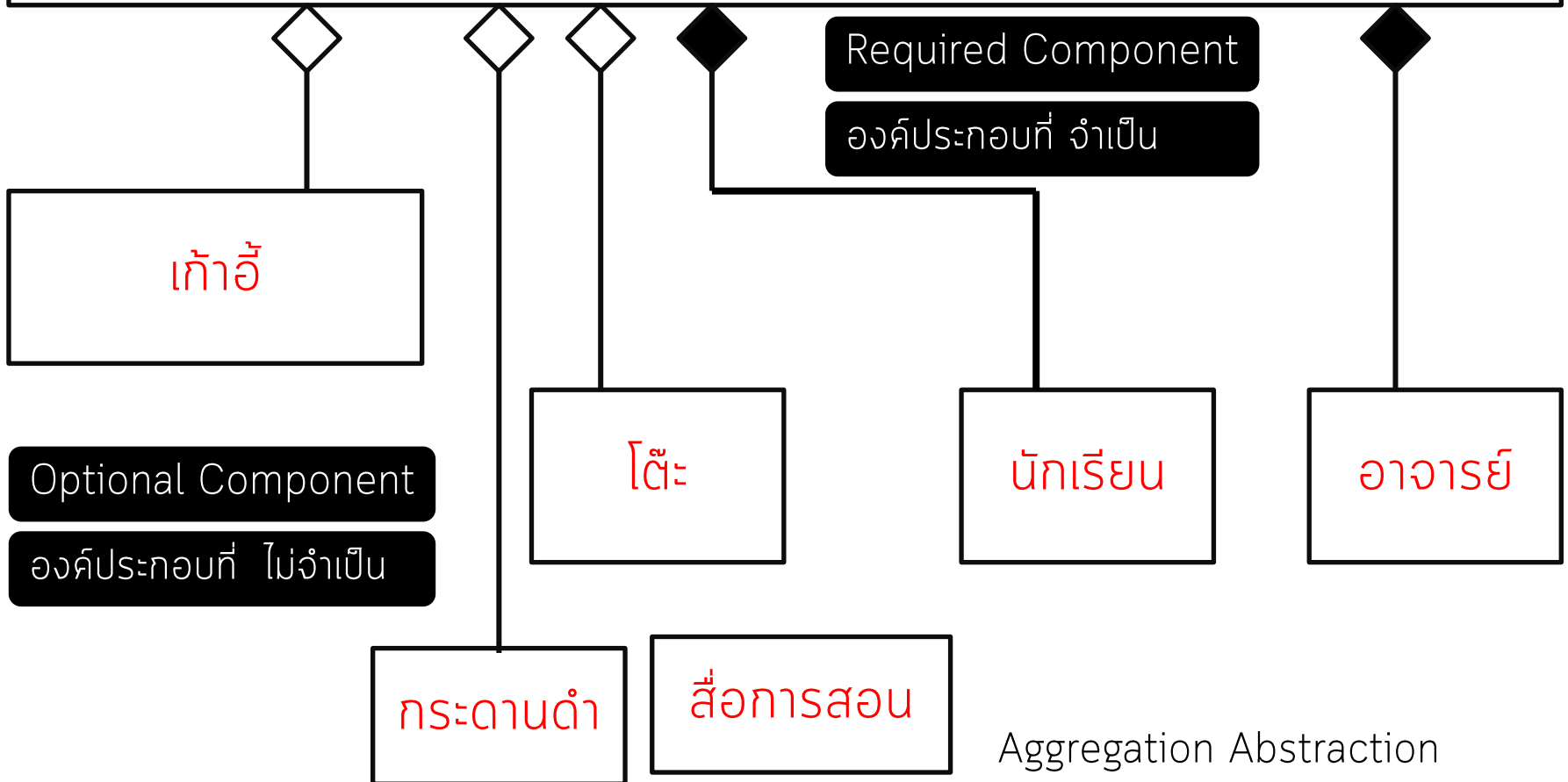


Optional Component
องค์ประกอบที่ ไม่จำเป็น

Required Component
องค์ประกอบที่ จำเป็น

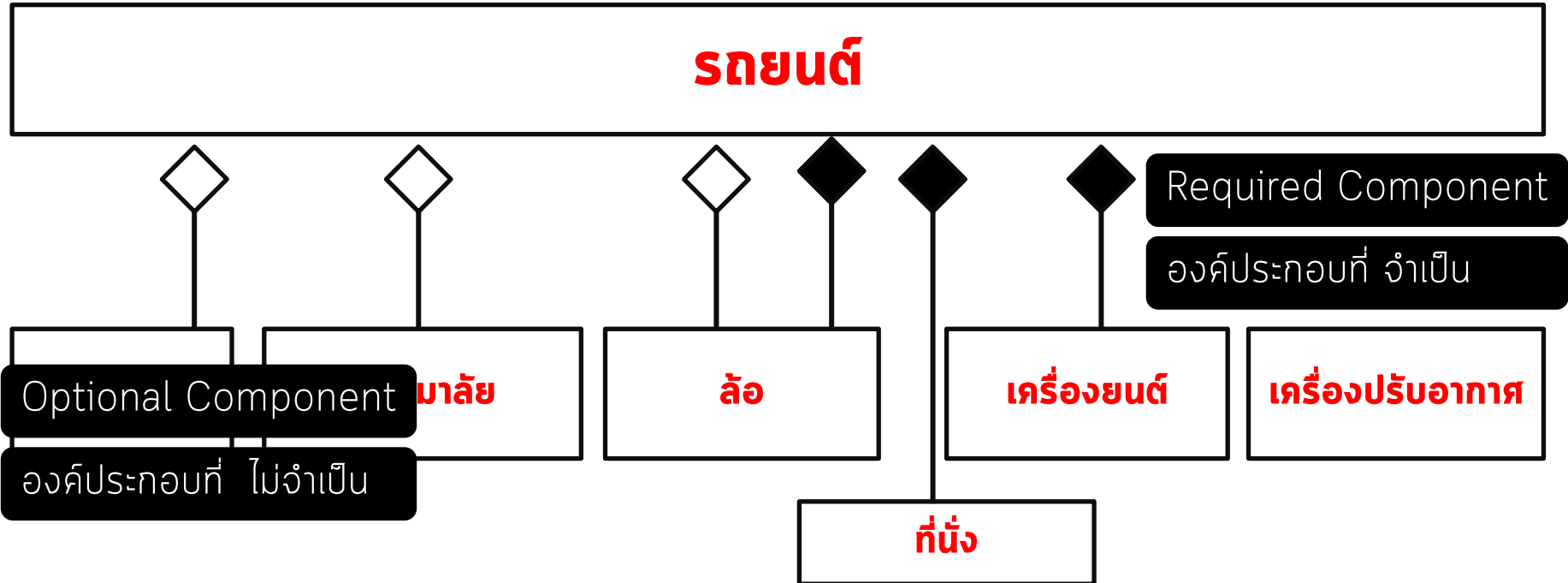


ห้องเรียน (หมายถึงห้องเรียนที่มีการเรียนการสอน)



*** ห้องเรียน ทางกายภาพ(physical) ประกอบไปด้วย เสา หลังคา เพดาน พื้น ผนัง

จเขียนแผนภาพแสดง Aggregation ของรถยนต์



*** พร้อมทั้งระบุส่วนประกอบที่จำเป็นและไม่จำเป็น

*** พร้อมทั้งระบุ ชื่อ - นามสกุลและรหัส capture ส่งทางไลน์

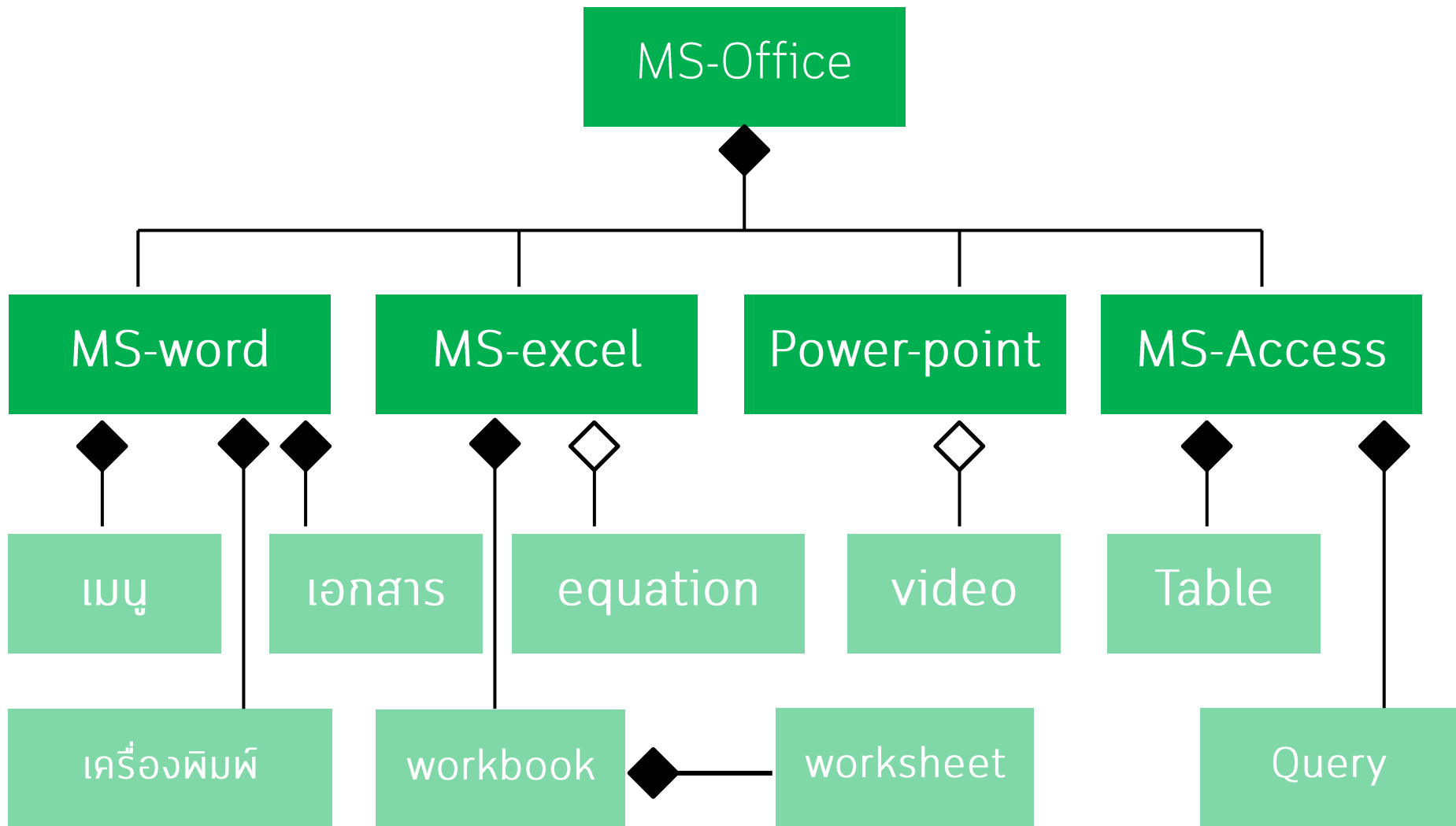
จเขียนแผนภาพแสดง Aggregation ของ โทรศัพท์มือถือ



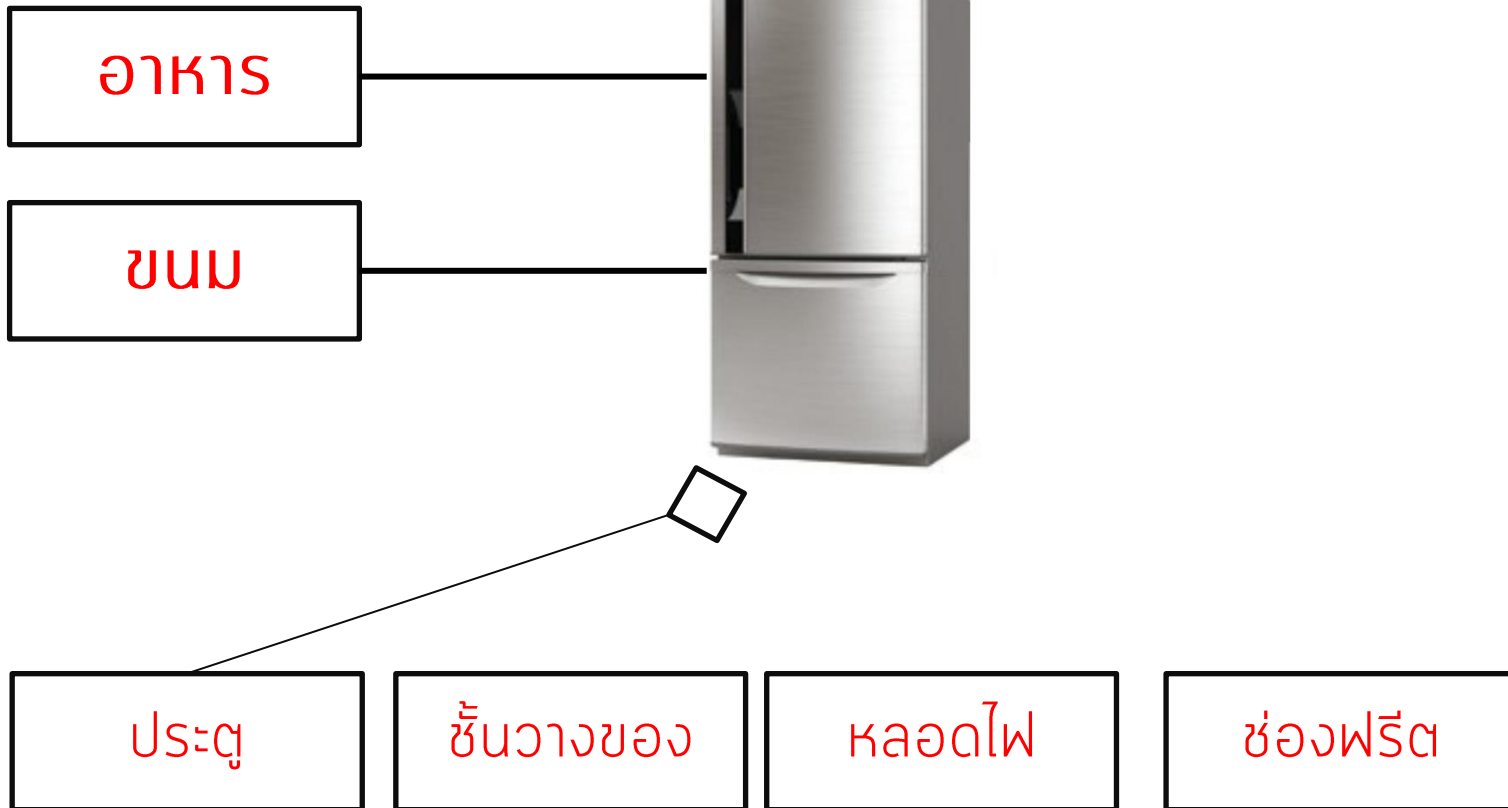
*** พร้อมทั้งระบุส่วนประกอบที่จำเป็นและไม่จำเป็น

*** พร้อมทั้งระบุ ชื่อ - นามสกุลและรหัส capture ส่งทางไลน์

วงเวียนแผนภาพแสดง Aggregation ของ MS-Office



Association relationship



อาหาร

ขนม

อาหาร ขนม นม หรือ ผัก ไม่ใช่ส่วนประกอบของตู้เย็น

class

ห้องคอม

- หมายเลขห้อง
- อาจารย์
- นักศึกษา
- คอมพิวเตอร์
- โต๊ะคอมพิวเตอร์
- แอร์

object

- เปิดได้
- ปิดได้
- ใช้เรียน/สอน

ห้อง 843 เป็นห้องพักอาจารย์
ไม่สามารถใช้สอนคอมพิวเตอร์
ได้

ห้อง 822

ห้อง 844

ห้อง 833

Is member of

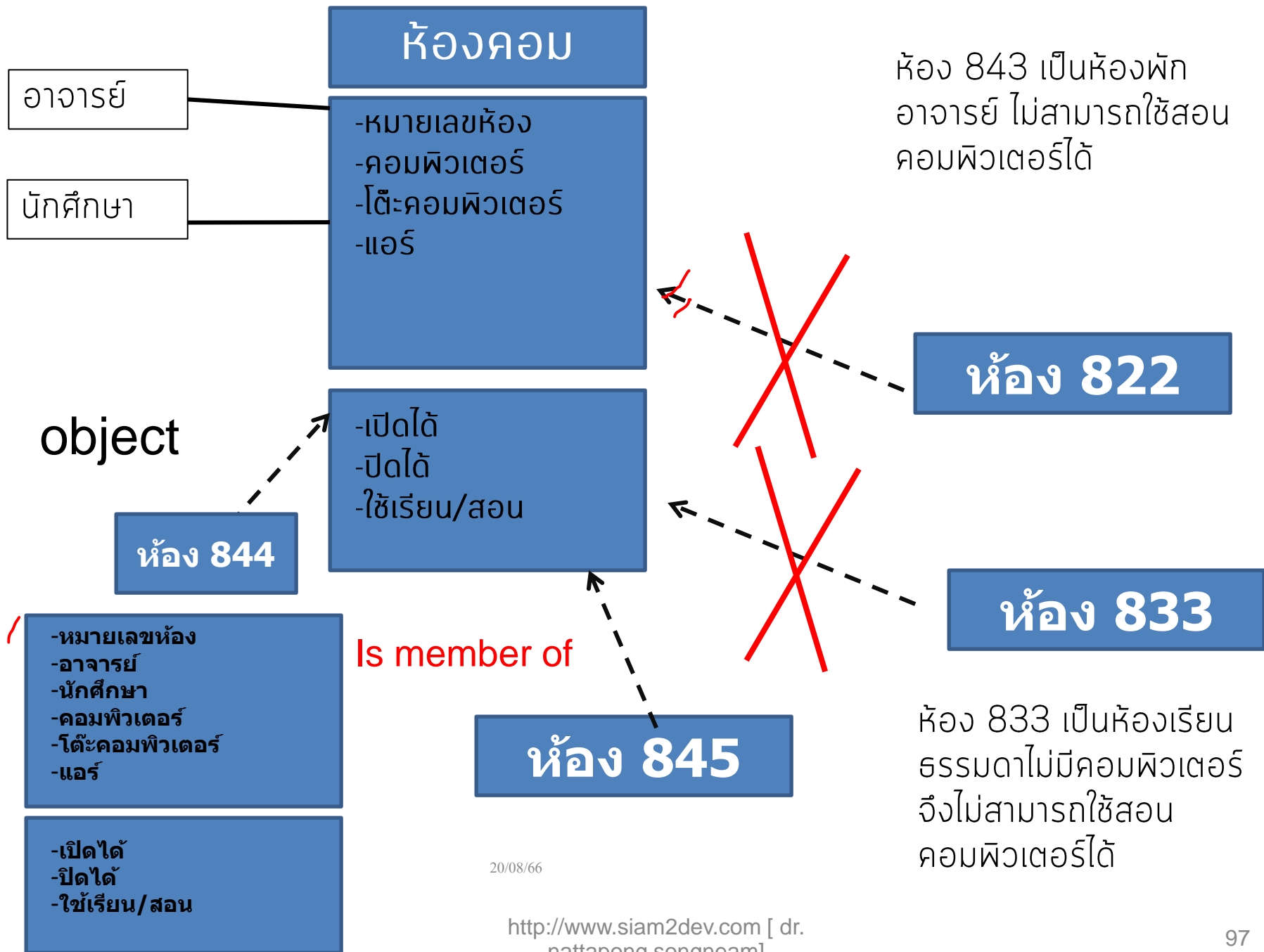
ห้อง 845

- หมายเลขห้อง
- อาจารย์
- นักศึกษา
- คอมพิวเตอร์
- โต๊ะคอมพิวเตอร์
- แอร์

- เปิดได้
- ปิดได้
- ใช้เรียน/สอน

ห้อง 833 เป็นห้องเรียน
ธรรมดาไม่มีคอมพิวเตอร์ จึงไม่
สามารถใช้สอนคอมพิวเตอร์ได้

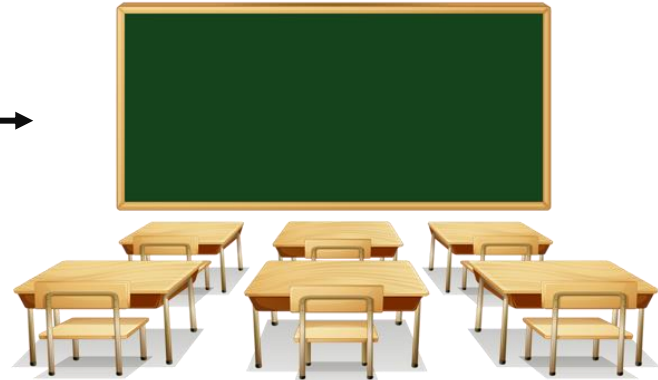
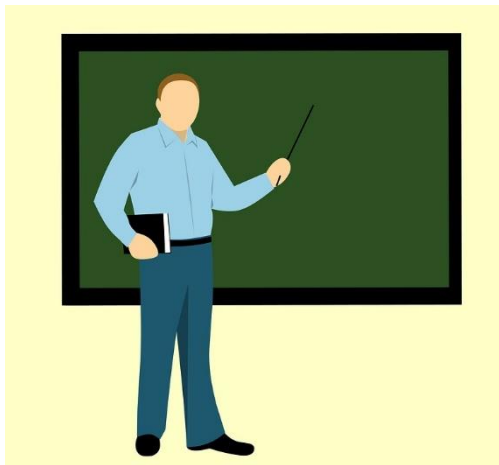
20/08/66



20/08/66

ห้องเรียน

*** ห้องเรียน ทางกายภาพ(physical) ประกอบไปด้วย เสา หลังคา เพดาน พื้น ผนัง



อาจจะมองว่า อาจารย์ ไม่ใช่ส่วนหนึ่งของห้องเรียน อาจารย์เป็นเพียงแค่
ผู้ใช้(association)ห้องเรียน

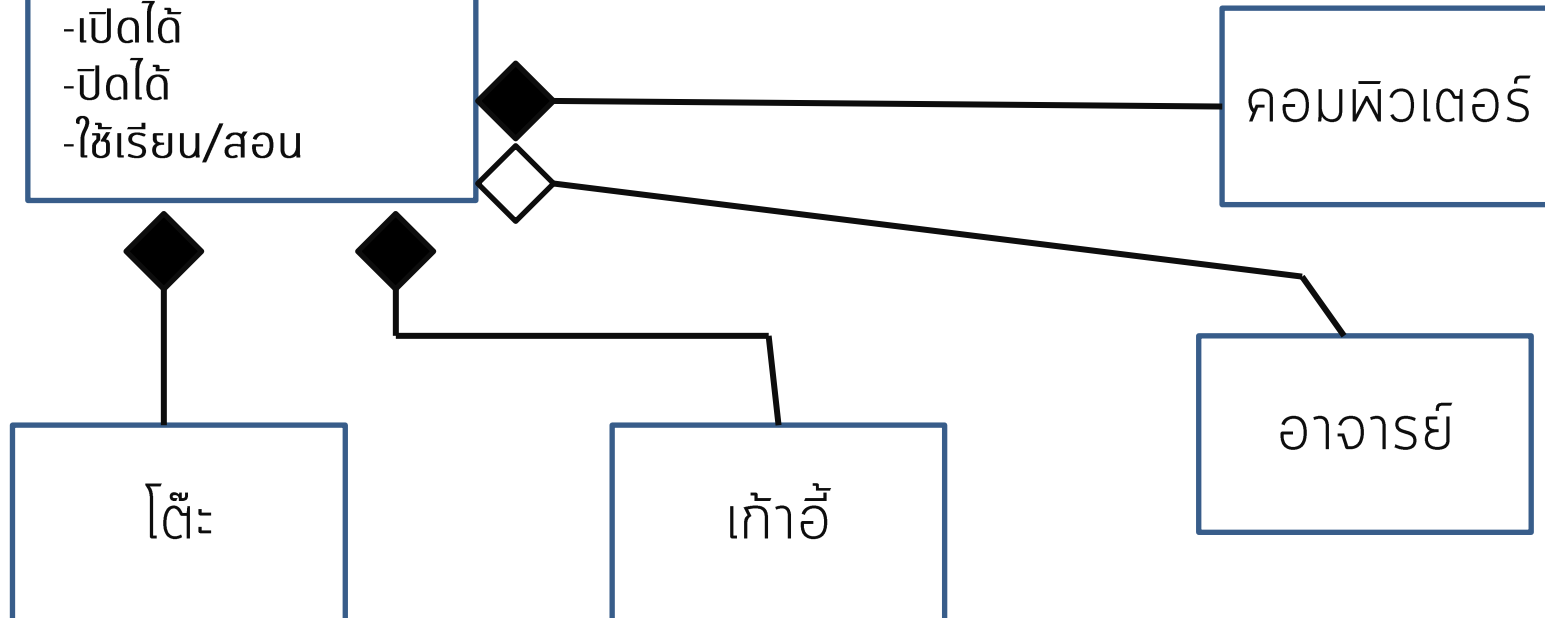
ห้องคอม

- หมายเลขห้อง
- อาจารย์
- นักศึกษา
- คอมพิวเตอร์
- โต๊ะคอมพิวเตอร์
- แอร์

- เปิดได้
- ปิดได้
- ใช้เรียน/สอน

Aggregation Abstraction

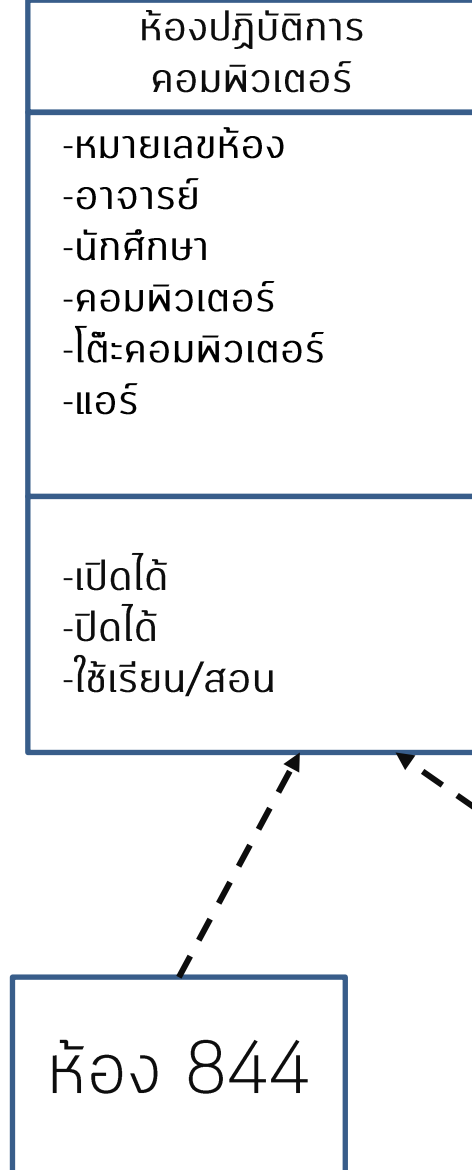
ทั้งโต๊ะ อาจารย์ เก้าอี้ นักเรียน คอมพิวเตอร์
ประกอบกัน ขึ้นเป็นห้องคอมพิวเตอร์ (หมายถึงห้อง
คอมพิวเตอร์ที่มีการเรียนการสอน)



Classification Abstraction

ทั้งโต๊ะ อาจารย์ เก้าอี้ นักเรียน คอมพิวเตอร์
ประกอบกัน ขึ้นเป็นห้องคอม

Class



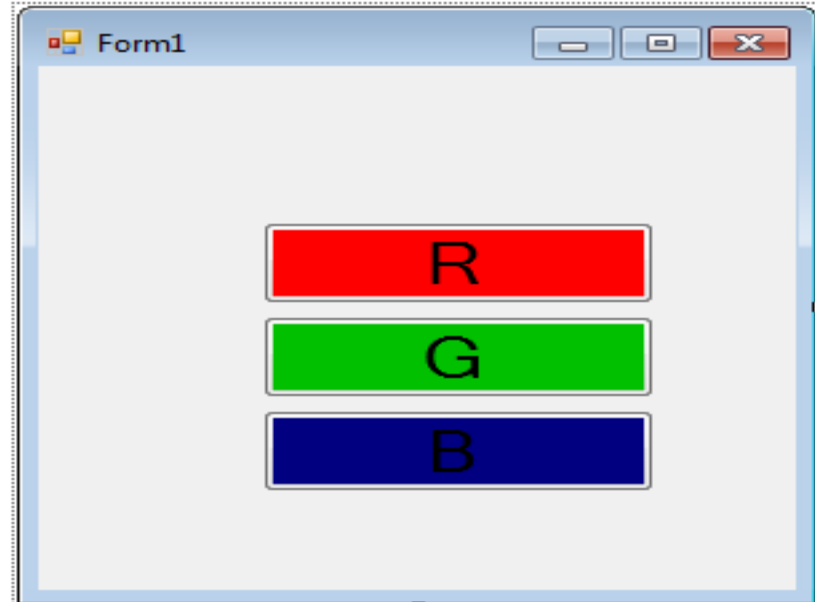
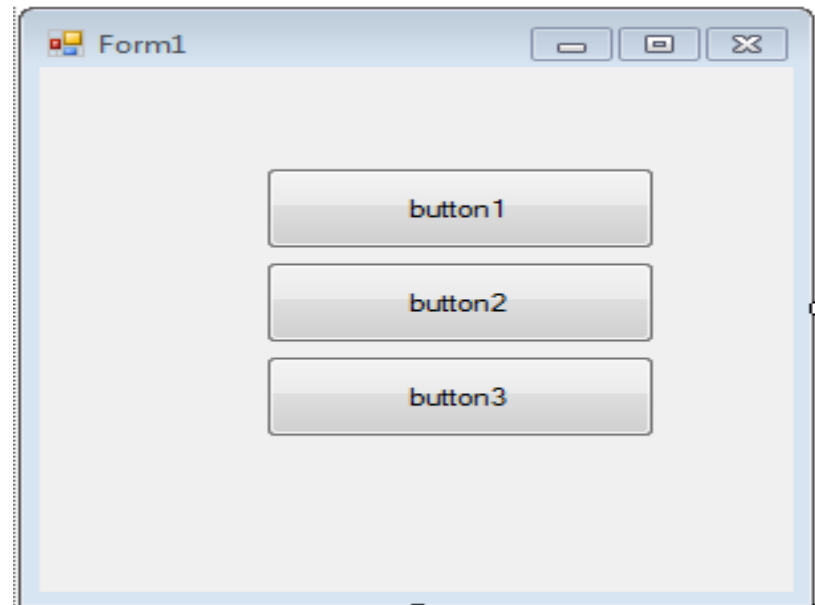
Is member of
เป็นสมาชิกของ

Object

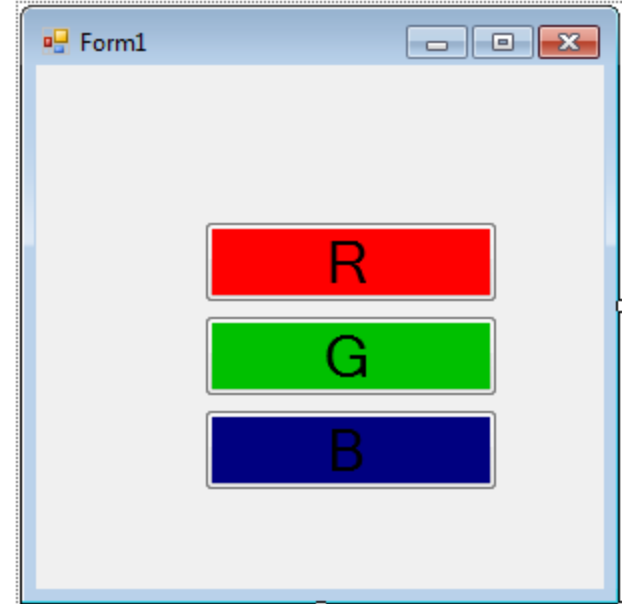
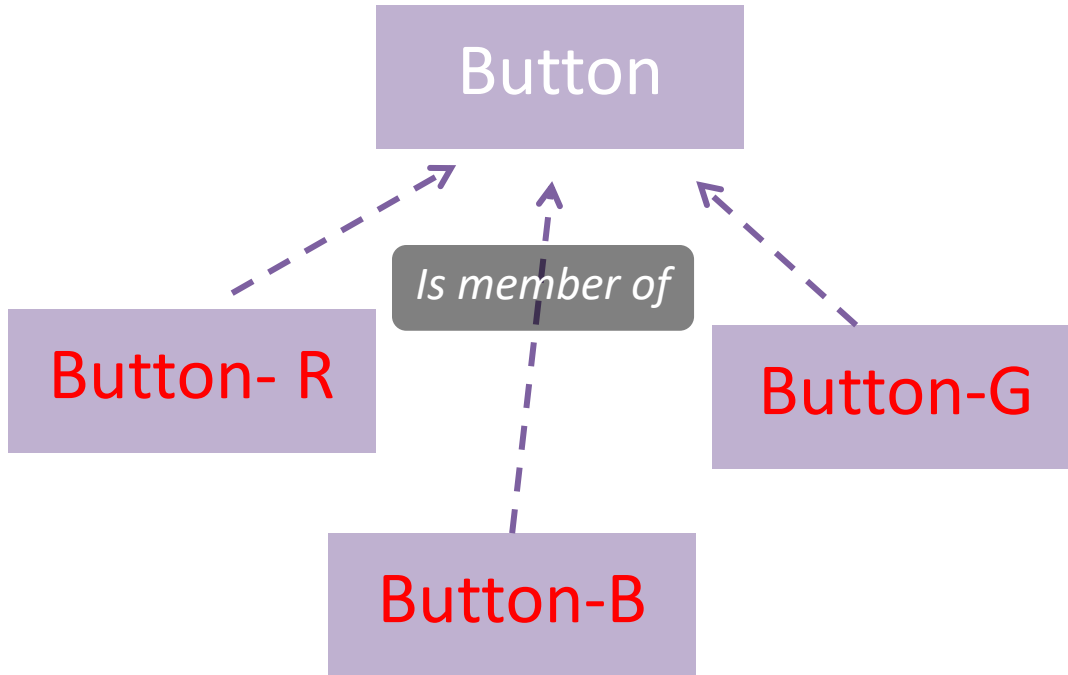
- Class button
 - Button 1
 - Button 2
 - Button 3

ทุกวัตถุจะต้องมี attributes และ function เหมือนกัน

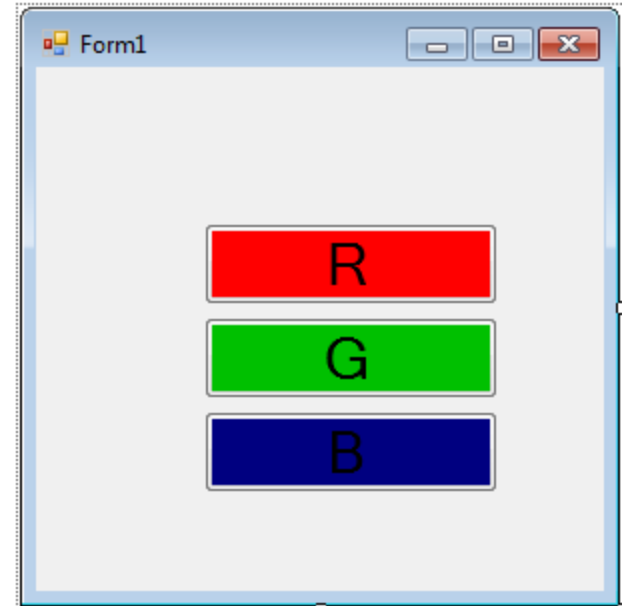
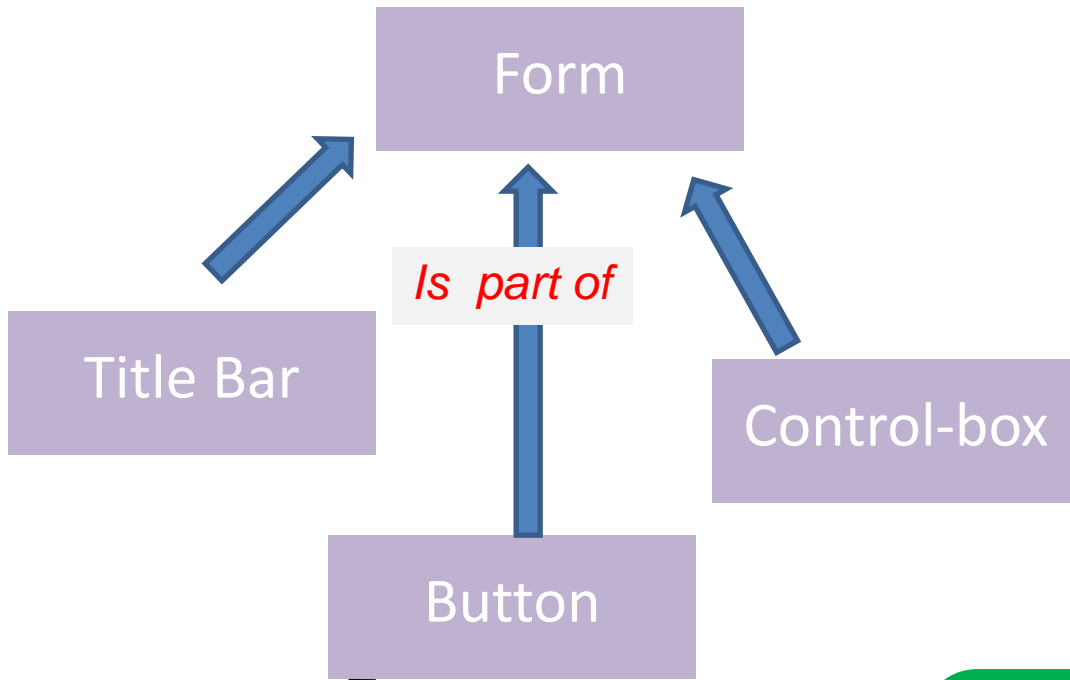
- Text
- Color



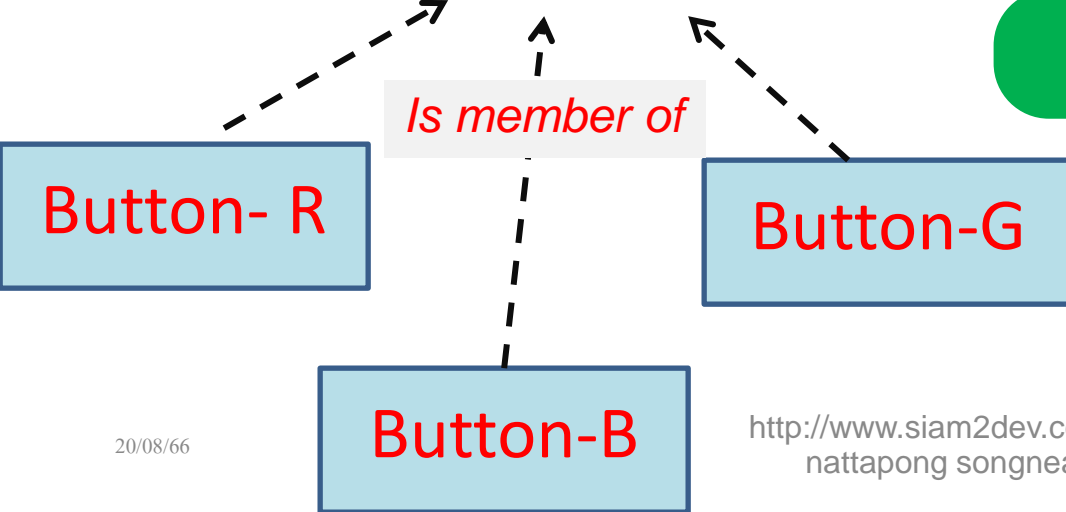
Classification Abstraction

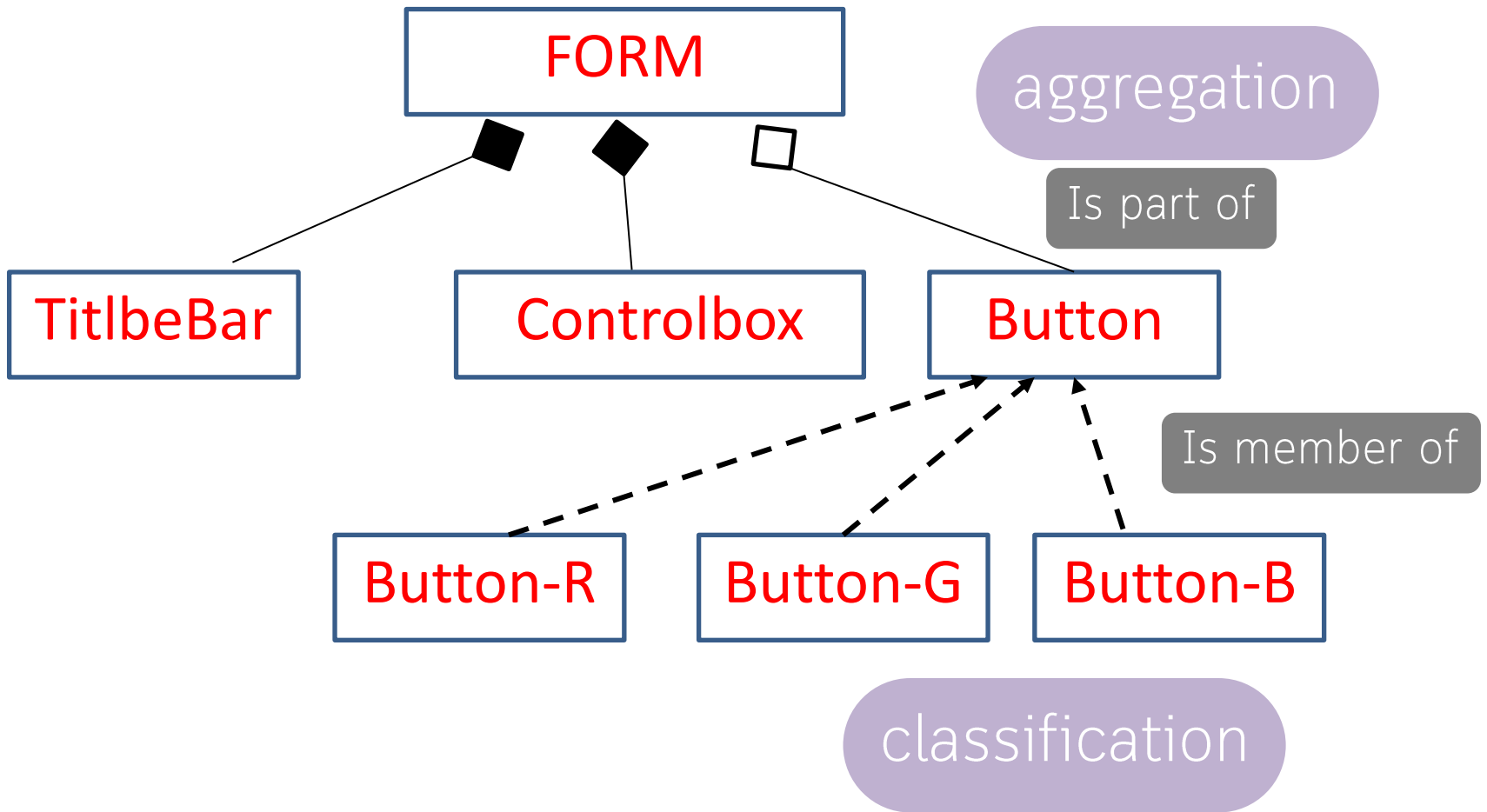


Aggregation Abstraction



Classification Abstraction





BUTTON

-Name
-Text
-Color

-Resizable()
-Enabled()
-Clickable()

FORM

FORM-1

Is member of

Is member of

Is member of

BUTTON-1 : Button

BUTTON-1

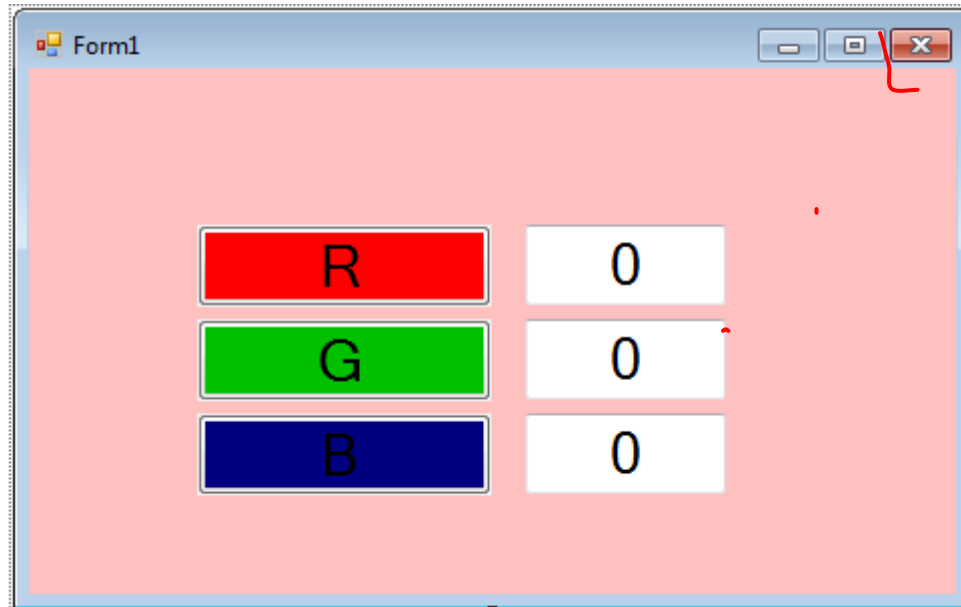
BUTTON-3

-Text : Button-1
- Color : Red
-Name : Button1

-Text : Button-2
- Color : Green
-Name : Button2

-Text : Button-3
- Color : Blue
-Name : Button3

- Class
- Object



โปรแกรมจัดการสินค้า

Button

Label

Textbox

ComboBox

Form

รายละเอียด : สินค้า 1

รายละเอียดของสินค้า

รหัสสินค้า	0001		
ชื่อสินค้า	สินค้า 1		
กลุ่มสินค้า	Harddisk	หน่วยนับ	ตัว
รุ่น/Model	MX2007		
ยี่ห้อ/บริษัท	Kingston		
จำนวน	3	[ปกติผู้ใช้งาน จะไม่สามารถกำหนดได้]	
ราคาขาย	500.00	บาท	
หมายเหตุ:			

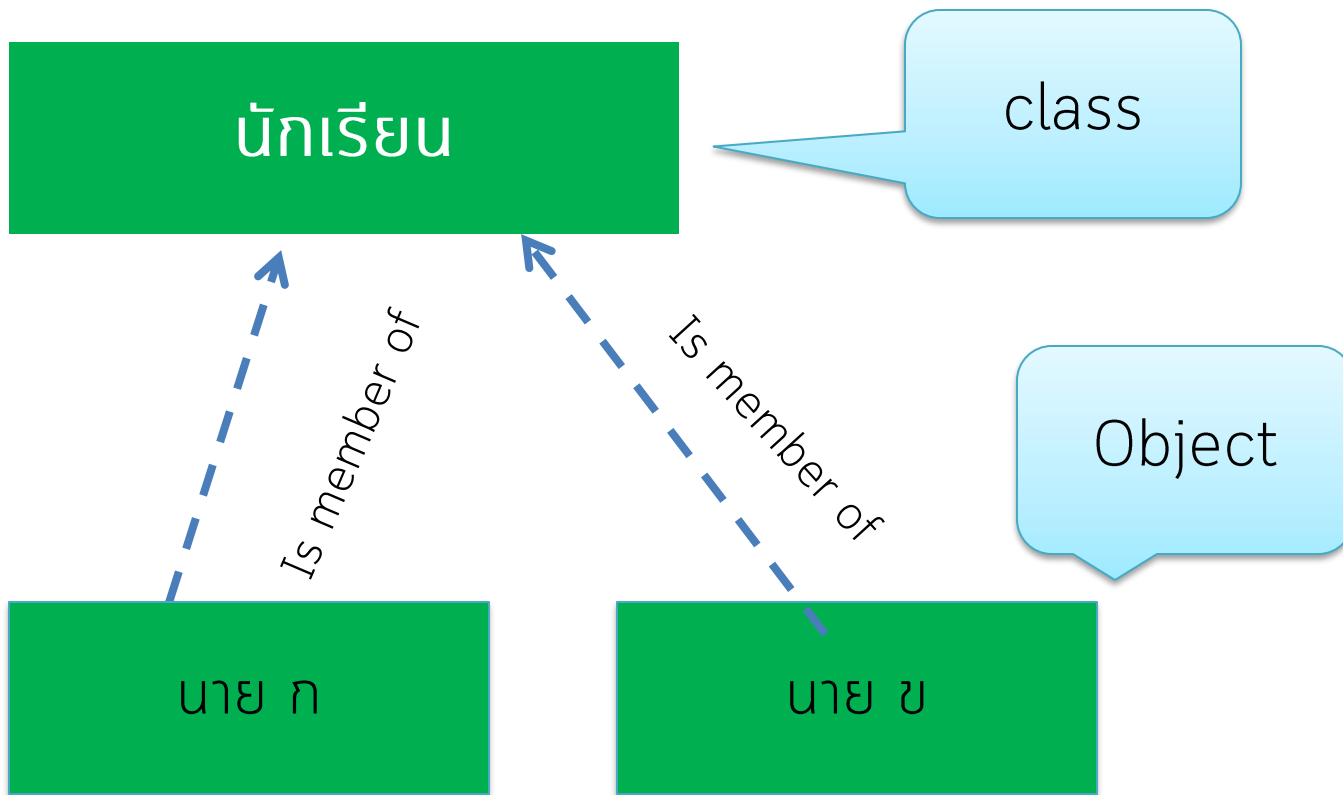
บันทึกข้อมูล F8

ดูรายการ Esc

หลักการ/แนวทาง สำหรับทำ Aggregation abstraction

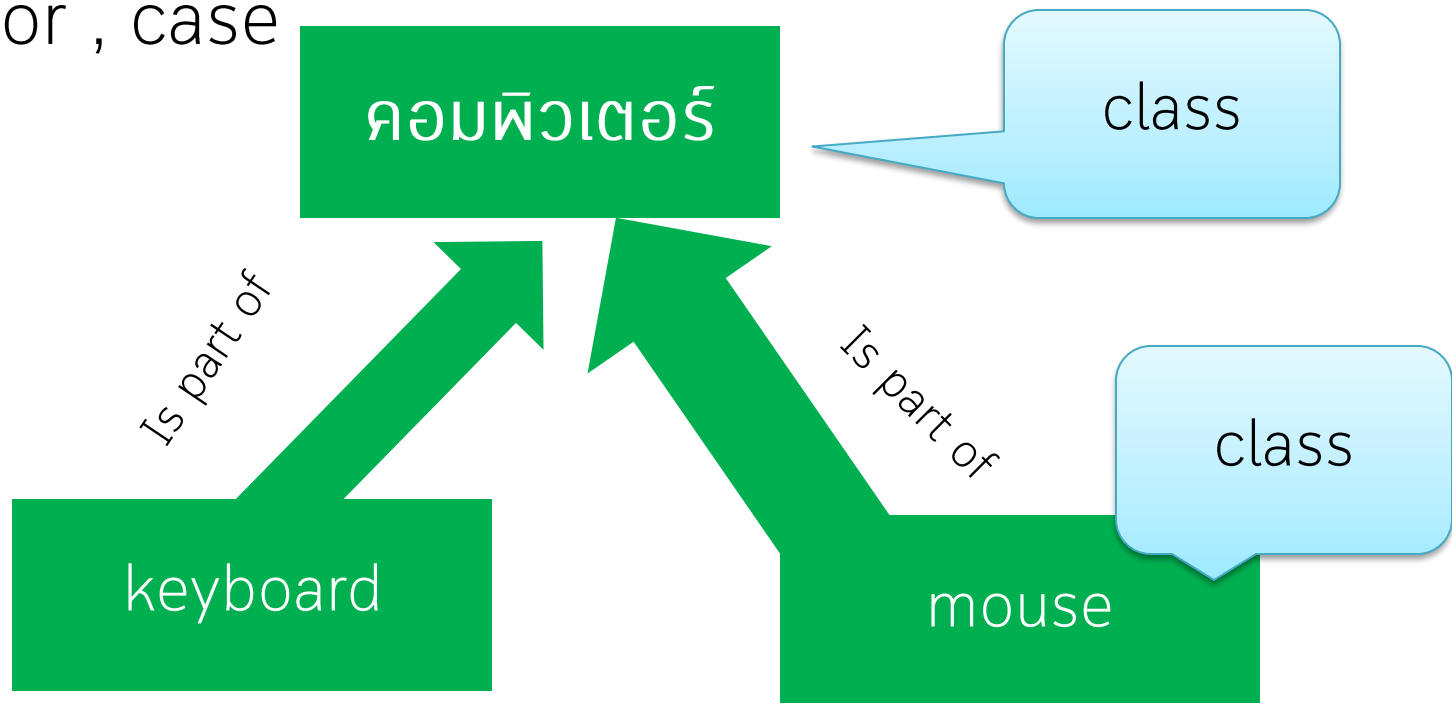
- ถ้าเรารวมคลาสย่อยๆ เข้าด้วยกัน ทำให้ได้ความคิดรวบยอดใหม่ (concept) ได้ เราเรียกกระบวนการนี้ว่า Aggregation Abstraction
- หากรวมแล้วไม่เกิด concept ใหม่หรือไม่ได้คลาสใหม่ที่ต่างไปจากเดิม ก็ไม่ต้องรวม
- นาย ก เป็นนักเรียน (classification abstraction)
- นาย ข ก็เป็นนักเรียน (classification abstraction)
- เครื่องยนต์ เมาะ พวงมาลัย ตัวถัง รวมกันเป็นรถยนต์ (aggregation abstraction)

นายก เป็นนักเรียน (classification abstraction)



ใช้ abstraction ในอธิบาย

- คอมพิวเตอร์ประกอบด้วย mouse , keyboard , monitor , case

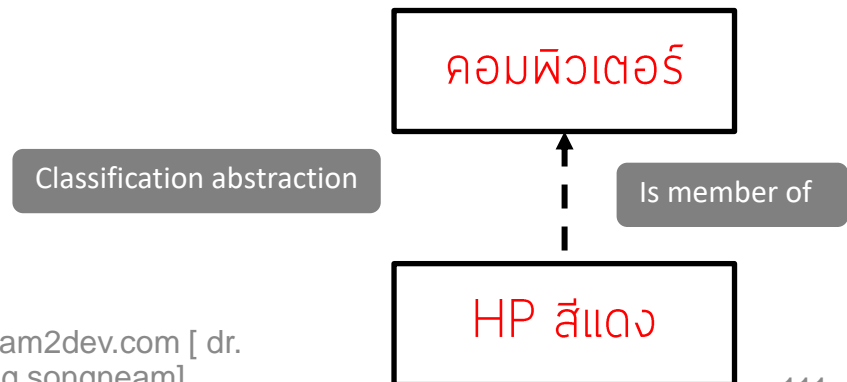


Aggregation Abstraction

แบบฝึกหัด 2.2

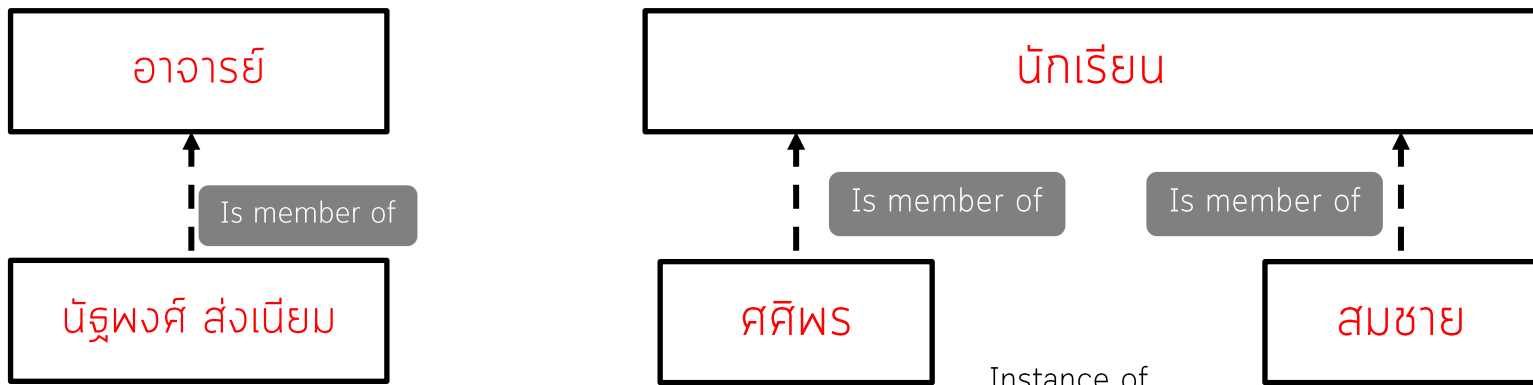
จงตอบคำถามต่อไปนี้ว่าจะใช้ abstraction ใดอธิบายข้อความต่อไปนี้

1. เครื่องคอมพิวเตอร์ HP สีแดงของสมชาย เป็นคอมพิวเตอร์
2. นั้ฐพงศ์ ส่งเนียม เป็นอาจารย์ และ ศศิพร เป็นนักเรียน
3. ปากกาสีแดงเป็น เครื่องเขียน
4. รถสปอร์ตสีแดง ป้ายทะเบียน 4กจ 5432 ของ นายกวมัย
5. ธนาคาร ประกอบไปด้วย เจ้าหน้าที่ เคา์นเตอร์ สปก ลูกค้้า
6. สมุดบัญชี ประกอบไปด้วย ปก หน้า ชื่อบัญชี รายการเดินบัญชี
7. ปรานีเป็นพยาบาล
8. สมควร เป็น สปก.
9. มหาวิทยาลัย ประกอบไปด้วย อาคารเรียน นักศึกษา เจ้าหน้าที่ อาจารย์ ห้องสมุด



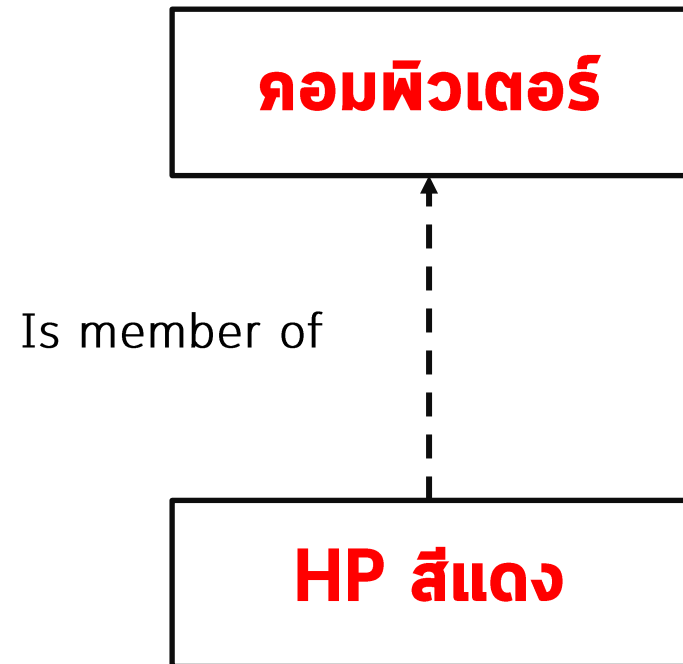
classification abstraction

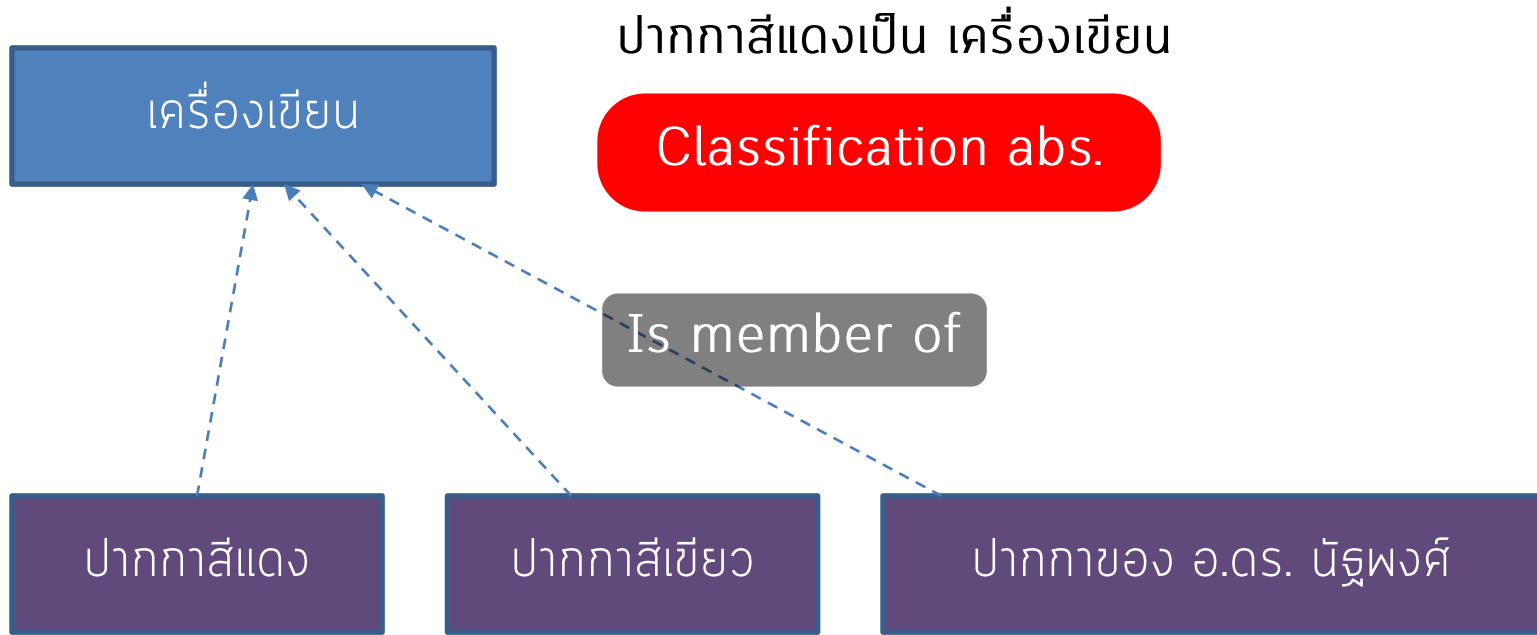
- นั้ฐพงค์ ส่องเนียม เป็นอาจารย์ และ ศศึพร เป็นนักเรียน
classification abs เพราะเป็นการบอกว่่าวัตถุจัดอยู่ในคลาสิใด



Classification abstraction

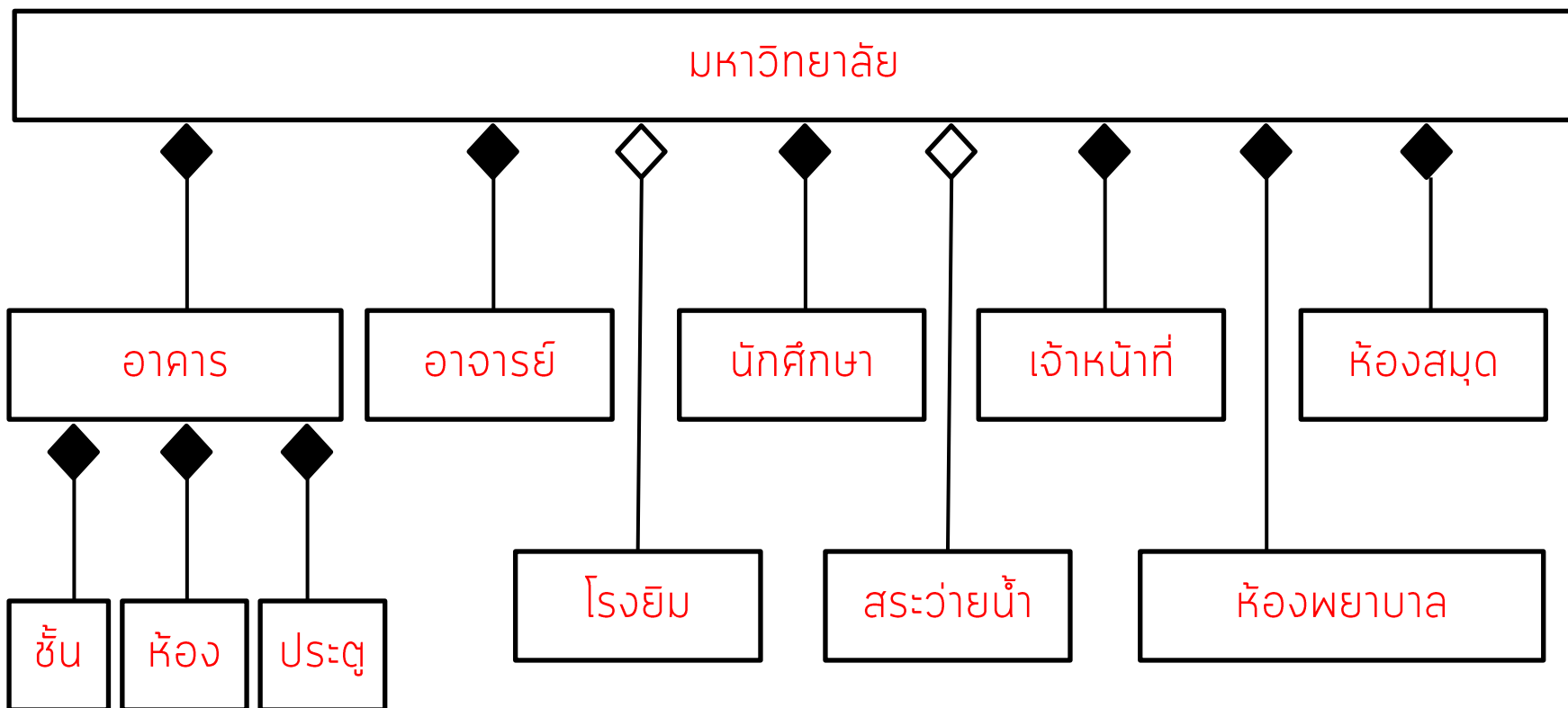
1. เครื่องคอมพิวเตอร์ HP สีแดงของสมชาย เป็นคอมพิวเตอร์



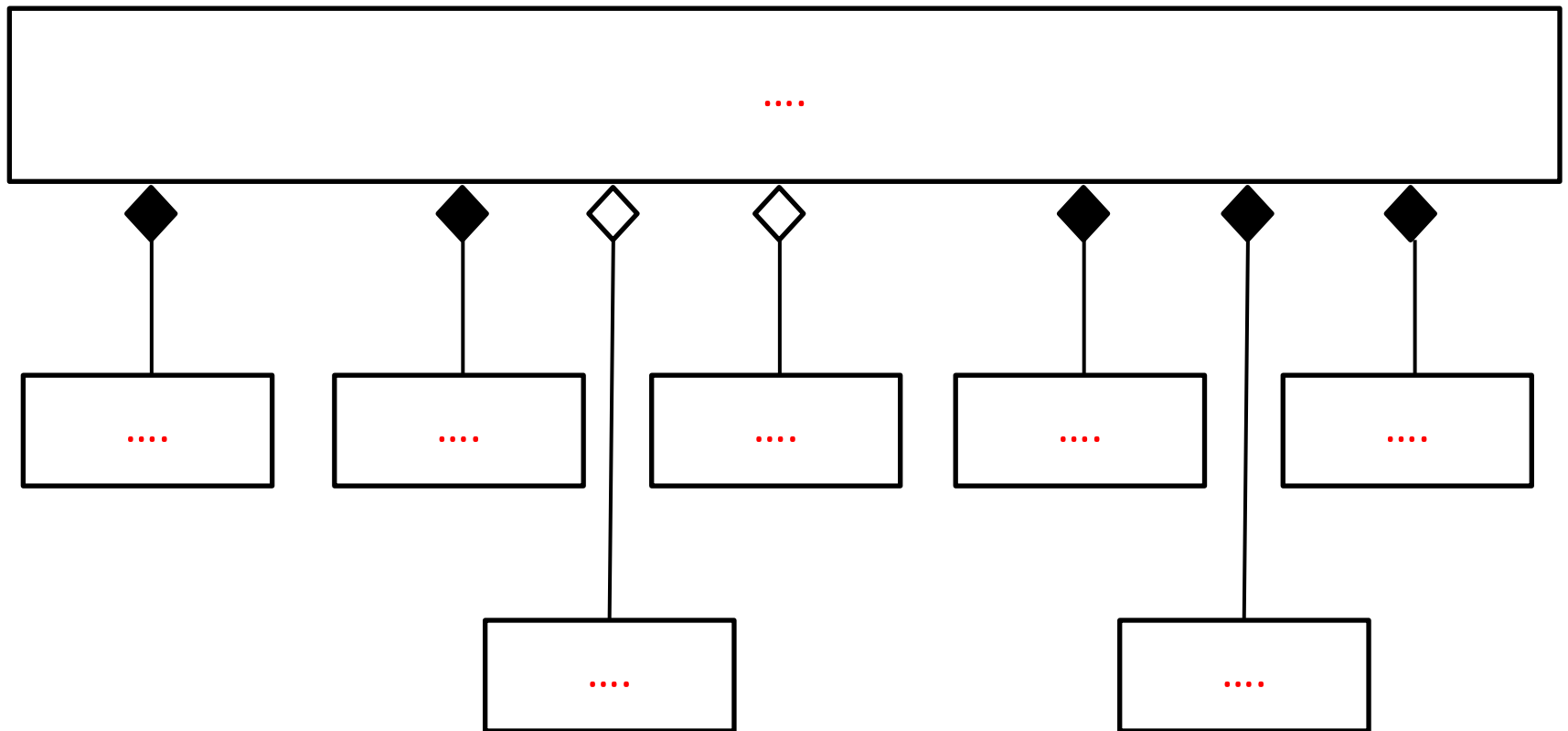


เหตุผล การบอกว่าวัตถุอยู่ในคลาสใด ก็ใช้ classification abs. อธิบาย
ดังนั้น ปากกาสีแดง เป็นสมาชิกของคลาส เครื่องเขียน

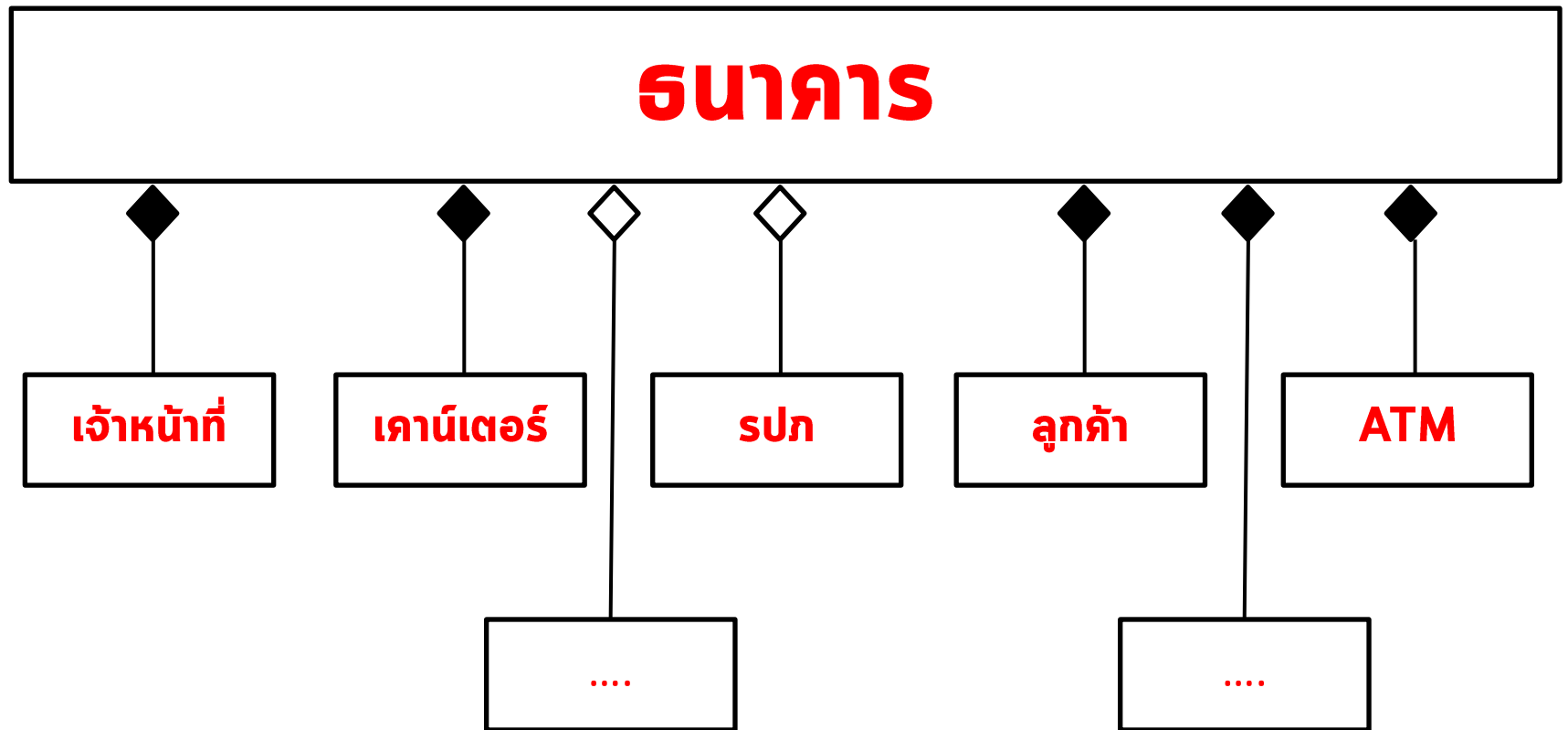
- มหาวิทยาลัย ประกอบไปด้วย อาคารเรียน นักศึกษา เจ้าหน้าที่ อาจารย์ ห้องสมุด ใช้ **Aggregation Abstraction** อธิบาย เพราะเป็นการบอกว่าคลาสมีคลาสย่อย หรือประกอบไปด้วยคลาสย่อย อะไรบ้าง



ธนาคาร ประกอบไปด้วย เจ้าหน้าที่ เคา์เตอร์ รปภ ลูกค้า และ ตู้ ATM บัญชีธนาคาร ตู้เซฟ



ธนาคาร ประกอบไปด้วย เจ้าหน้าที่ เคา์เตอร์ สปก ลูกค้า และ ตู้ ATM บัญชีธนาคาร ตู้เซฟ



ปราณีเป็นพยาบาล

พยาบาล

Is member of

ปราณี

สมศรี

สมควร

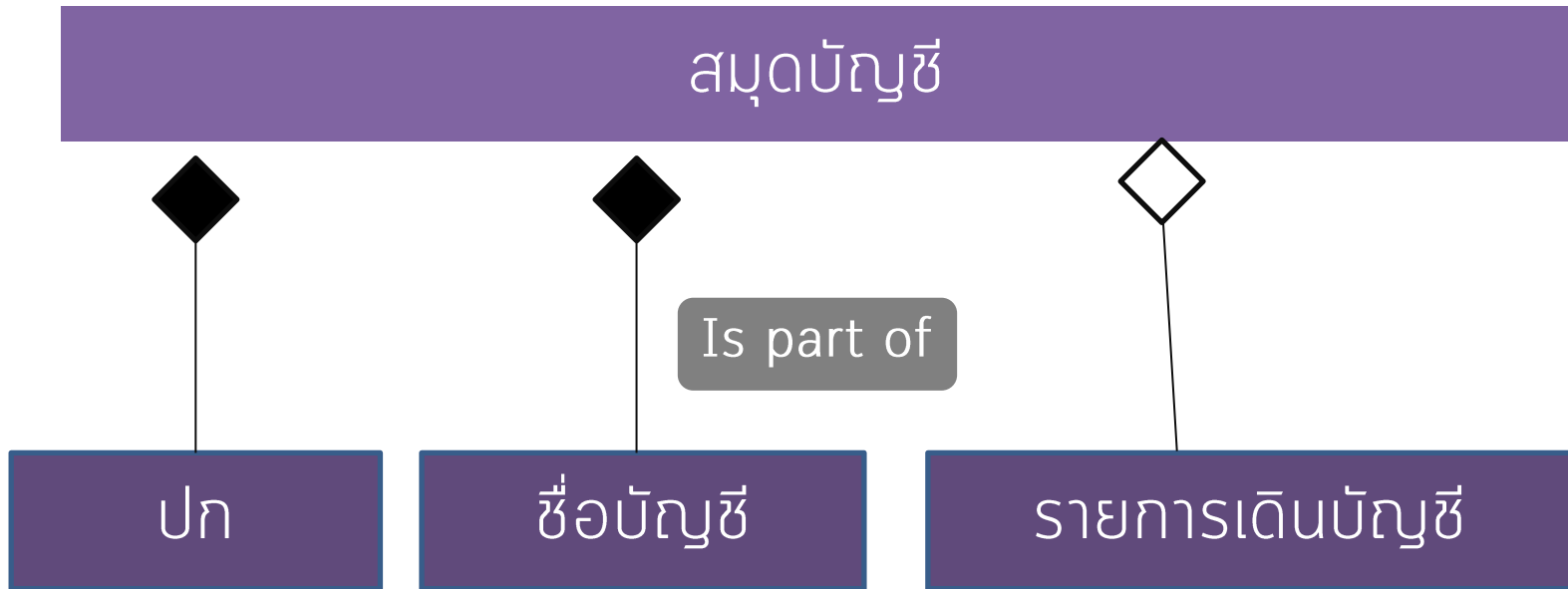
ปราณีเป็นพยาบาล

Classification abstraction

*** หมายเหตุ : สี ของแผนไม่มีความเกี่ยวข้องกับแผนภาพ
ปกติ จะไม่ใส่สีพื้นหลังหรือสีตัวอักษร

สมุดบัญชี ประกอบด้วย ปก หน้า ชื่อบัญชี รายการ
เดินบัญชี

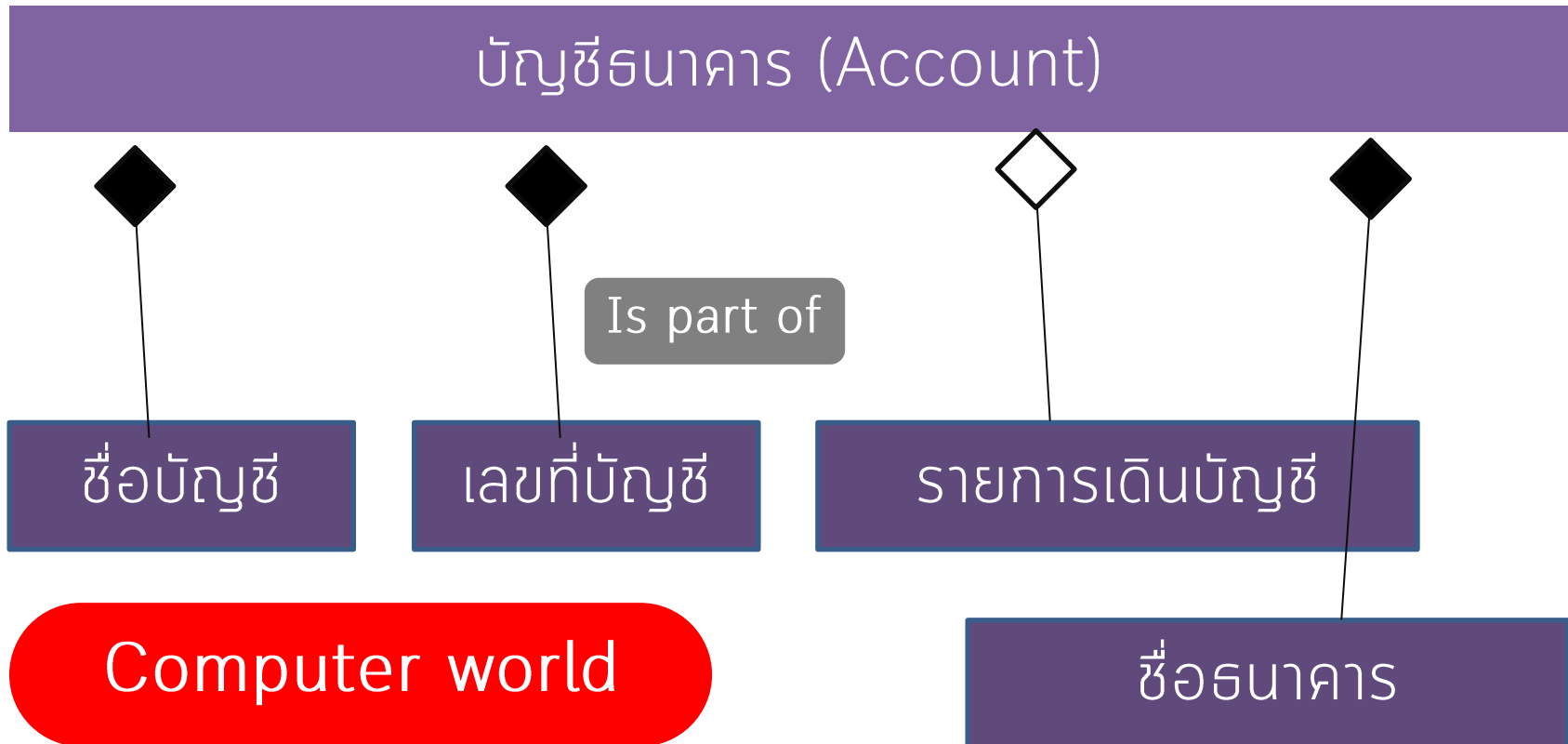
Aggregation abstraction



Real world

สมุดบัญชี ประกอบด้วย ปก หน้า ชื่อบัญชี รายการเดินบัญชี

Aggregation abs.

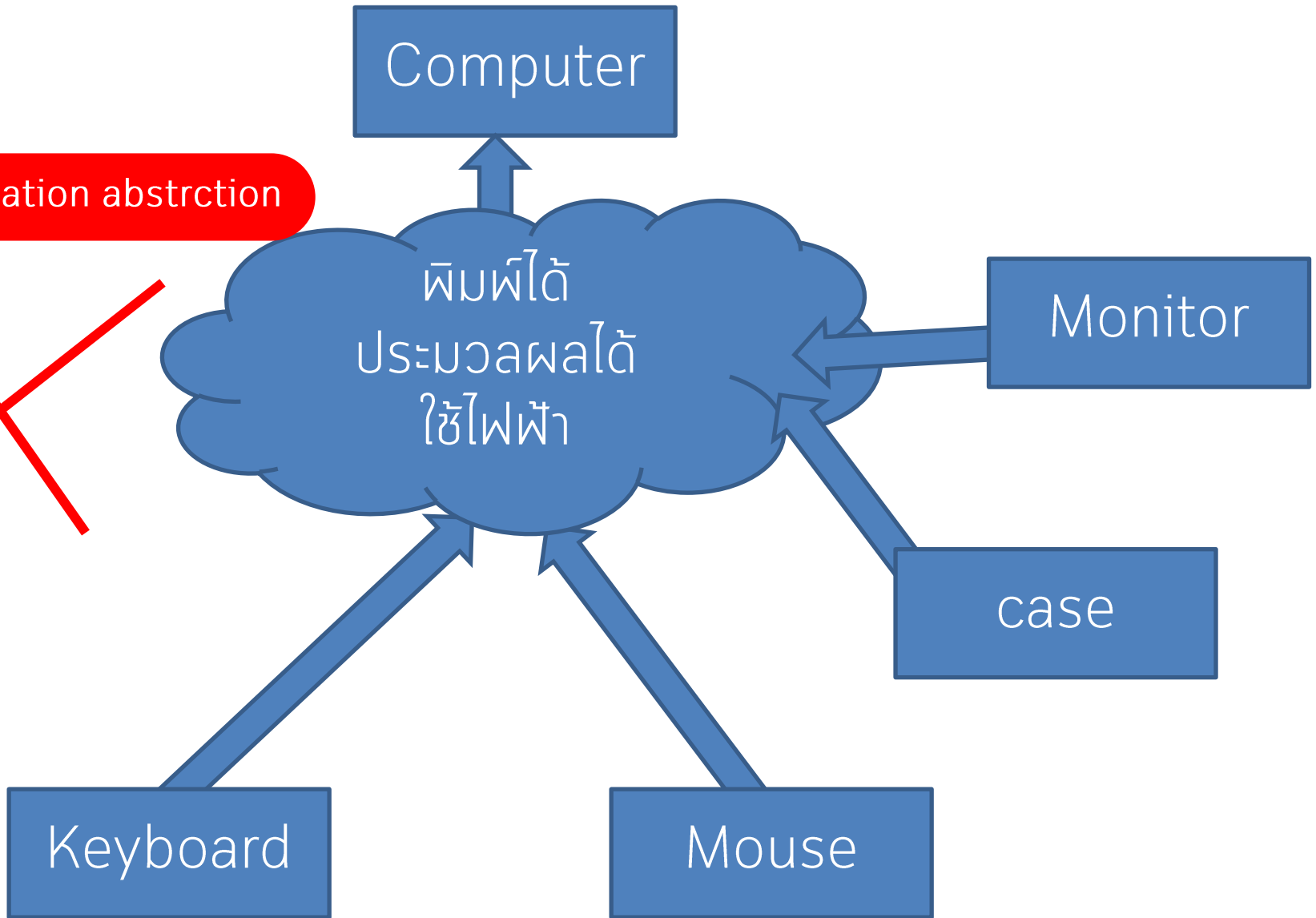
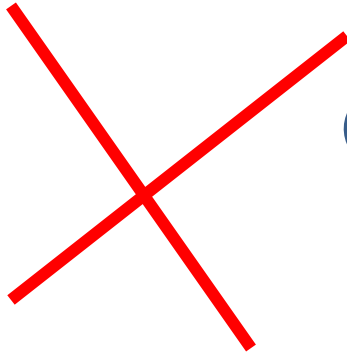


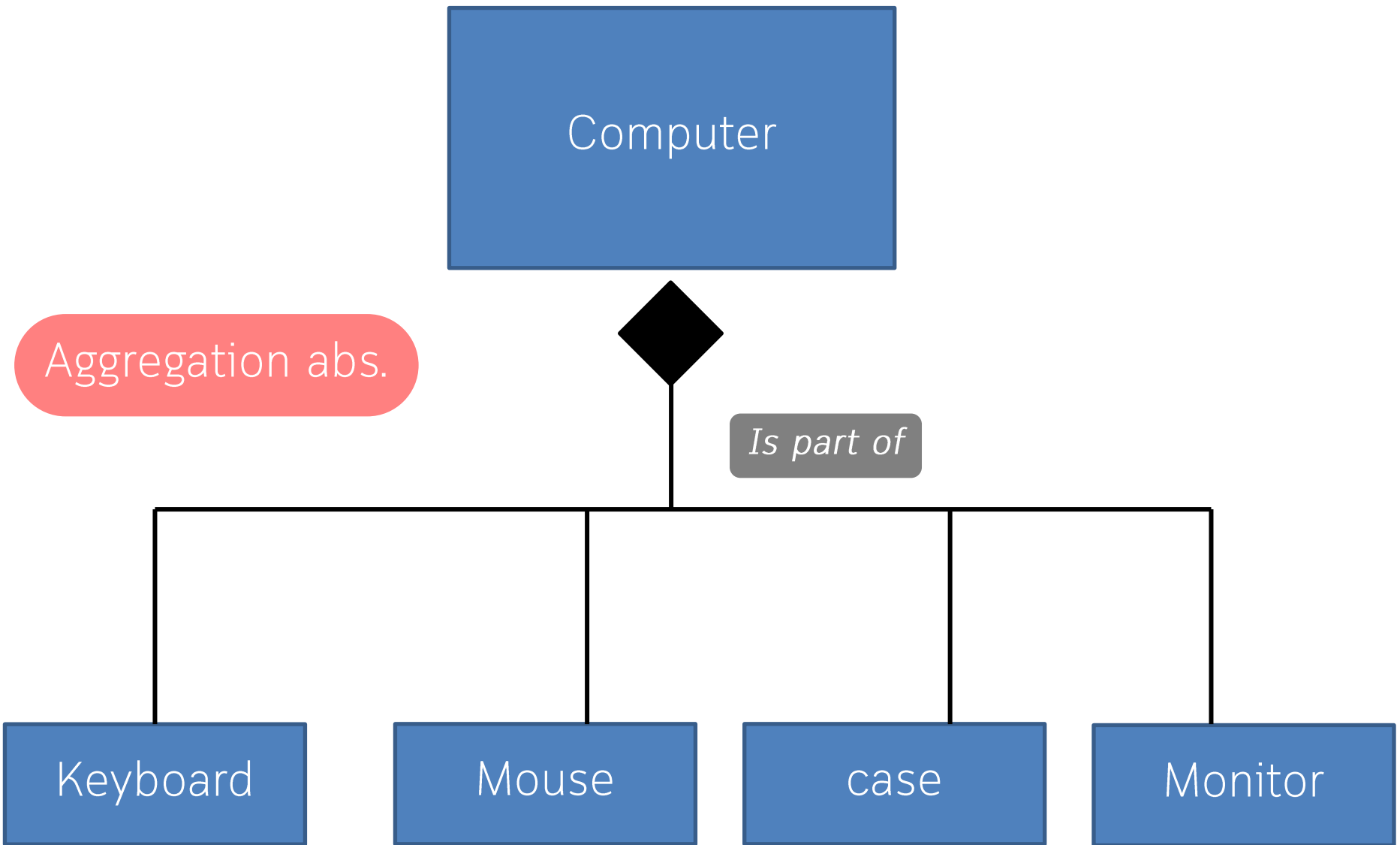
Classification abs

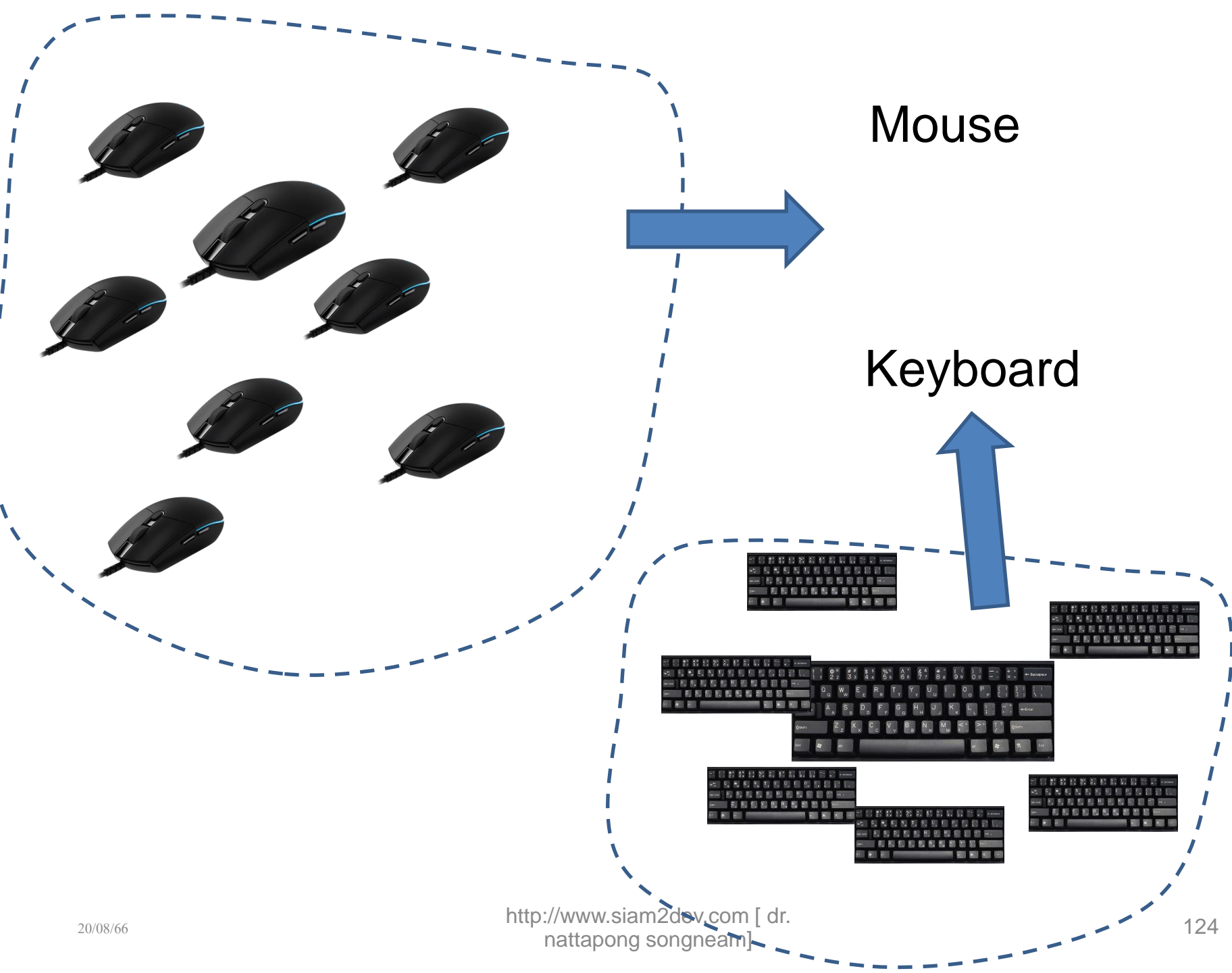
- ถ้าเรา บอกว่า สิ่งนั้น พิมพ์ได้ เขียนโปรแกรมได้ เล่นเน็ตได้ ใช้ไฟฟ้า ประมวลผลได้ เรียกว่าคอมพิวเตอร์ ...

Classification abs

Classification abstrction







Mouse

Keyboard

ส่วนประกอบของห้องเรียน



ห้องเรียน

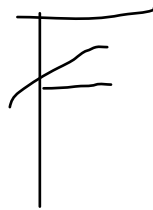
- Attribute

- หมายเลขห้อง
- โต๊ะ
- เก้าอี้
- อาจารย์
- นักเรียน

A handwritten capital letter 'A' in black ink, positioned to the right of the 'Attribute' list.

- Function

- เปิด
- ปิด
- สอน
- เรียน
- บำรุงรักษา

A handwritten capital letter 'F' in black ink, positioned to the right of the 'Function' list.

Attribute

PDT :: Primitive Data Type / ชนิดข้อมูลพื้นฐาน

- Integer
- String
- Character
- Date/Time
- boolean
- ETC.

**ADT :: Abstract Data Type / คลาส / ออบเจกต์ /
Reference type**

ห้องเรียน

• Attribute

– หมายเลขห้อง

– โต๊ะ

– เก้าอี้

– อาจารย์

– นักเรียน

PDT

ADT

ADT

• Function

– เปิด

– ปิด

– สอน

– เรียน

– บำรุงรักษา

ตัวอย่างภาษา C#

```
{
    class ROOM
    {
        private string room_no;
        private Teacher Tc;
        private Student St;
        private Tables Tb;
        public void OpenRoom() {
            // do something
        }
        public void CloseRoom() {
            // do something
        }
    }
    // สร้าง คลาส นักเรียน
    class Student {}
}

```

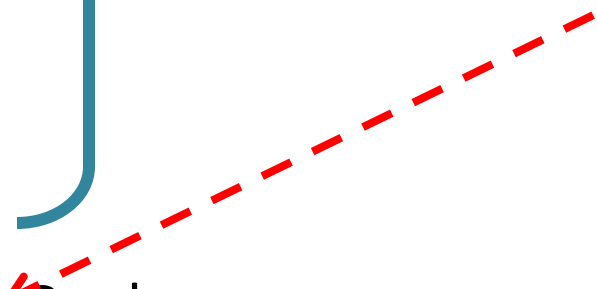
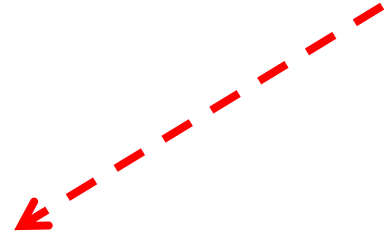
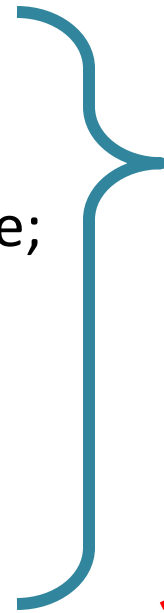
PDT

ADT

C#

PDT

```
Class Student {  
    private string ST_ID;  
    private string ST_FName;  
    protected string ST_Lname;  
    private string ST_TEL;  
    private string ST_Major;  
    private string ST_Faculty;  
    private string Gender;  
    private Student_CARD ST_Card;  
  
}
```



ADT

C#

```
class Student_CARD
{
    private string University;
    private string Fname;
    private date BOD;
    private string ST_ID;
    private image IMG;
    private University university;
}
```

PDT

ADT

University

- Attribute

- Name

- Address

- Building

- Person

- Faculty

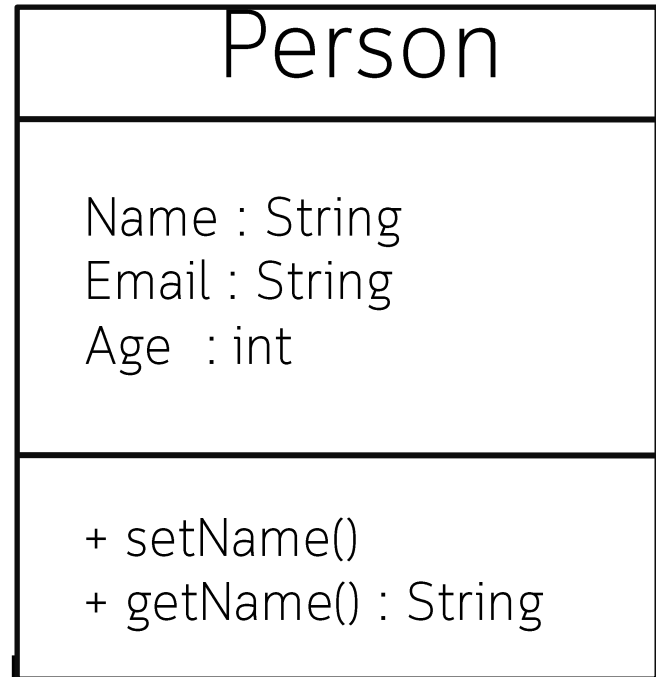
PDT

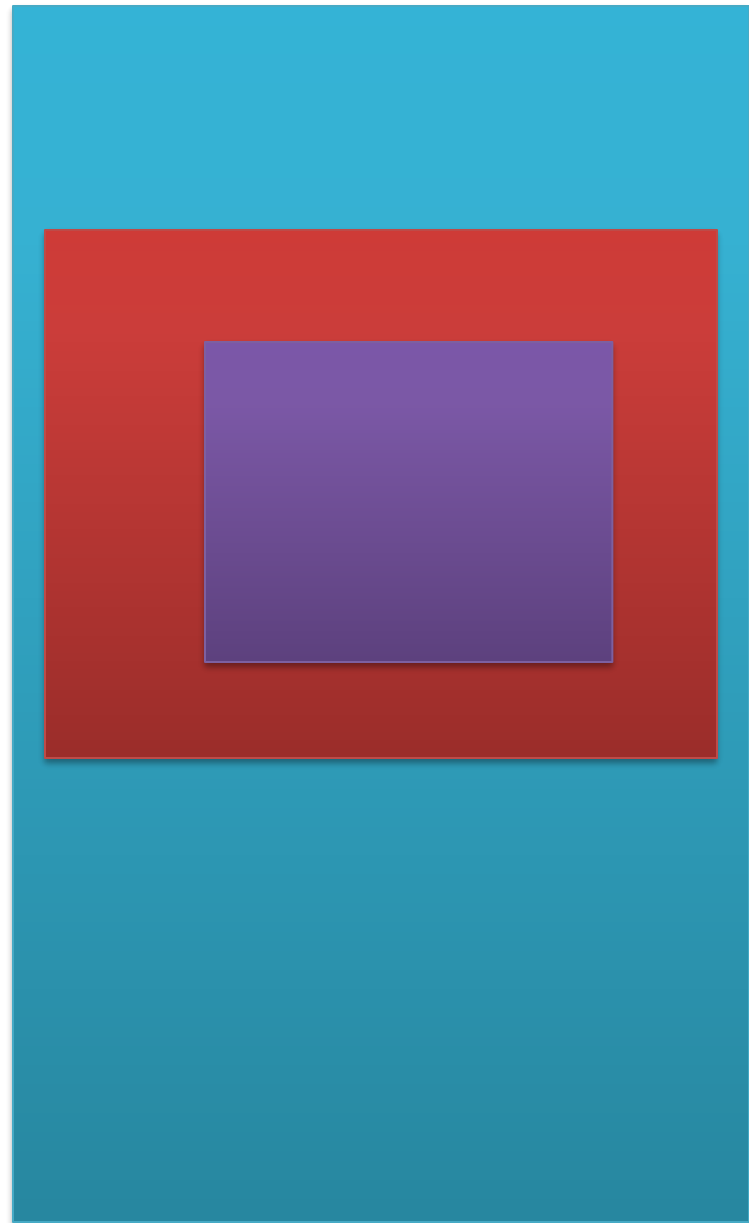
ADT

- Function

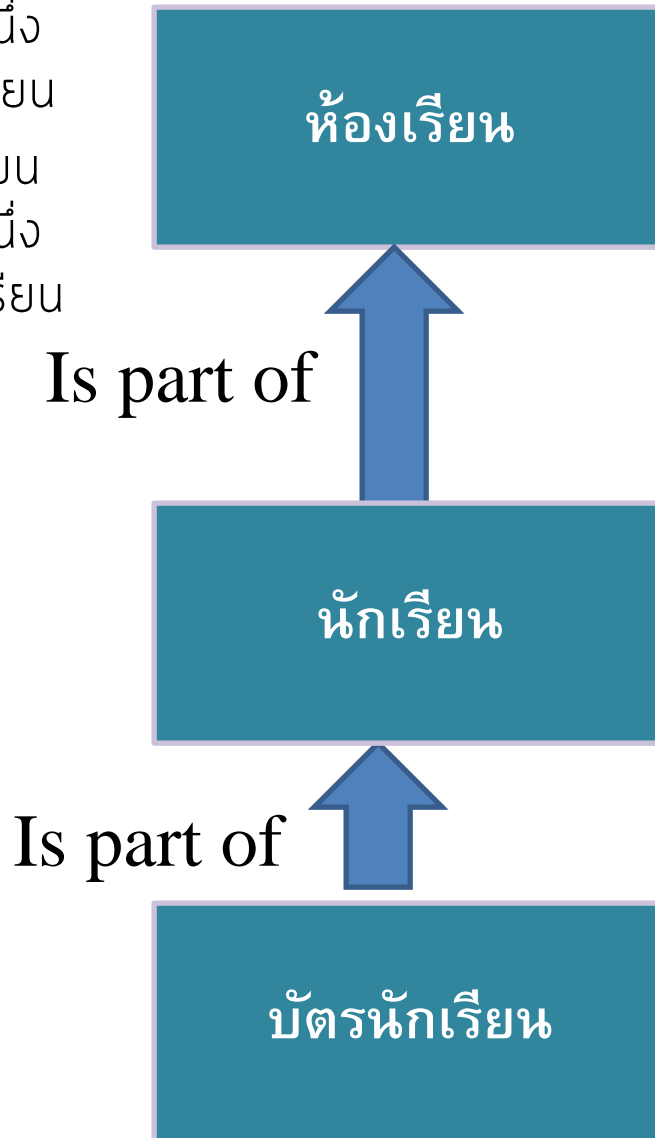
ตัวอย่างภาษา java

```
class person {  
    String name;  
    String email;  
    int age;  
  
    public String getName() { return name; }  
    public void setName(String n) { name = n; }  
}
```

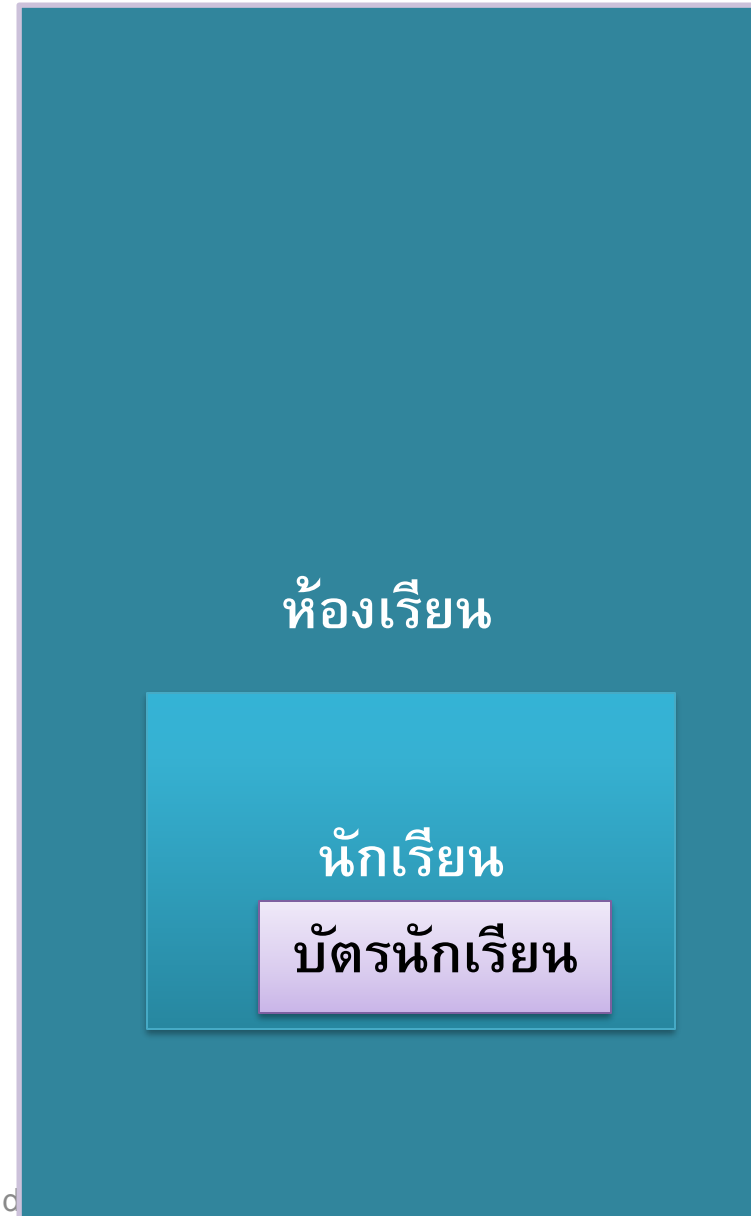


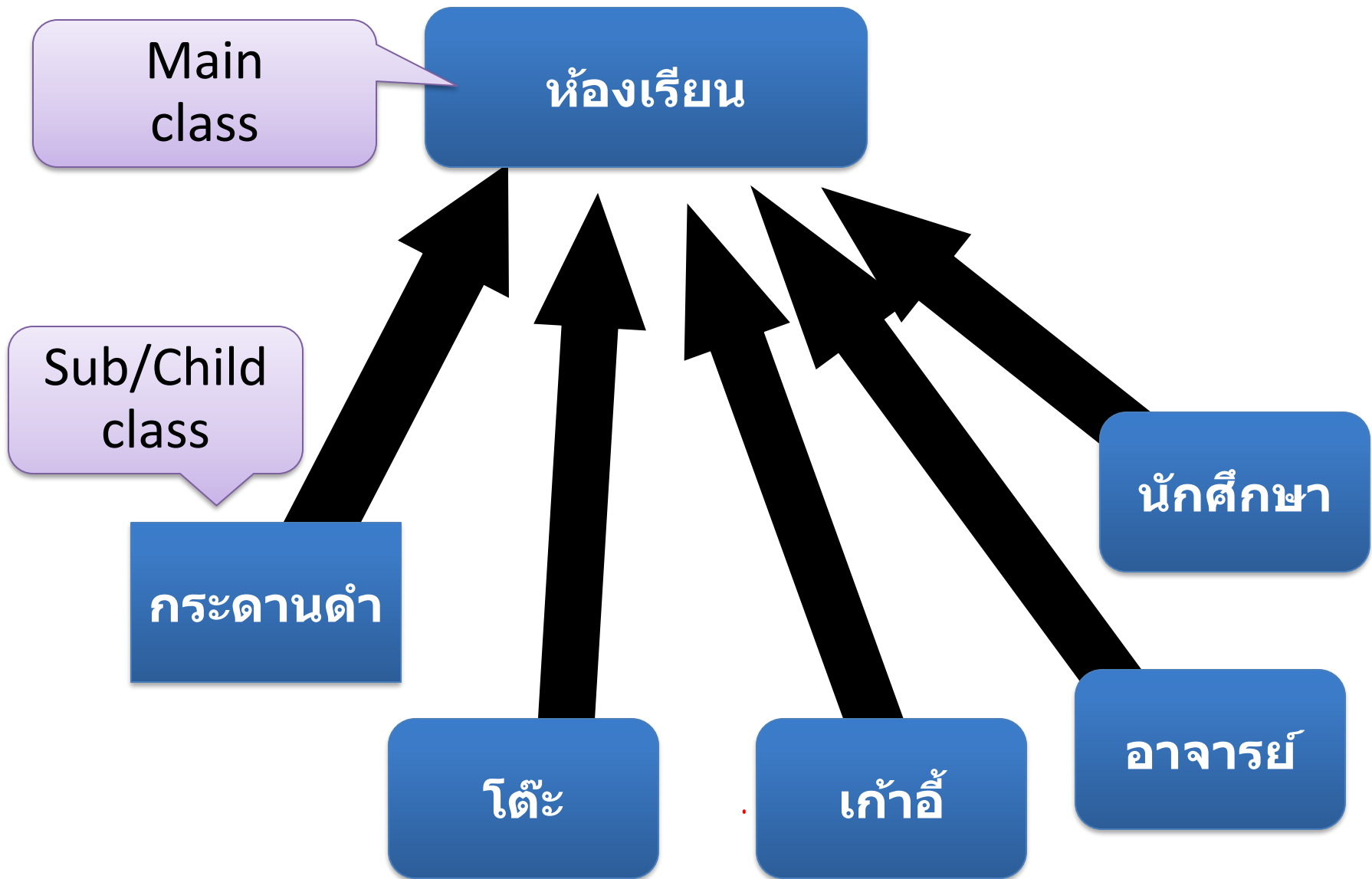


- บัตรนักเรียน เป็นส่วนหนึ่งของ นักเรียน
- และ นักเรียน เป็นส่วนหนึ่งของ ห้องเรียน



Inner classes

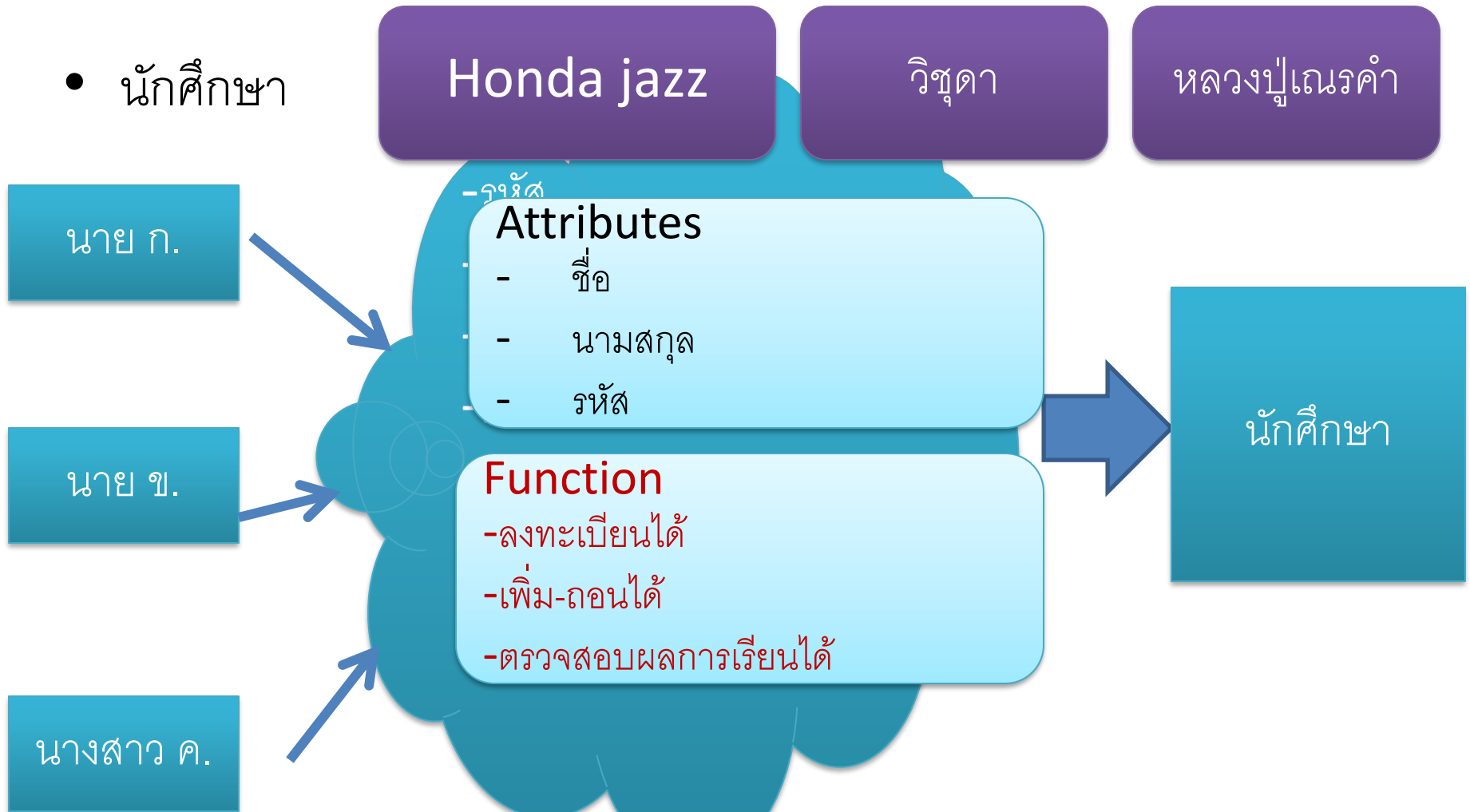


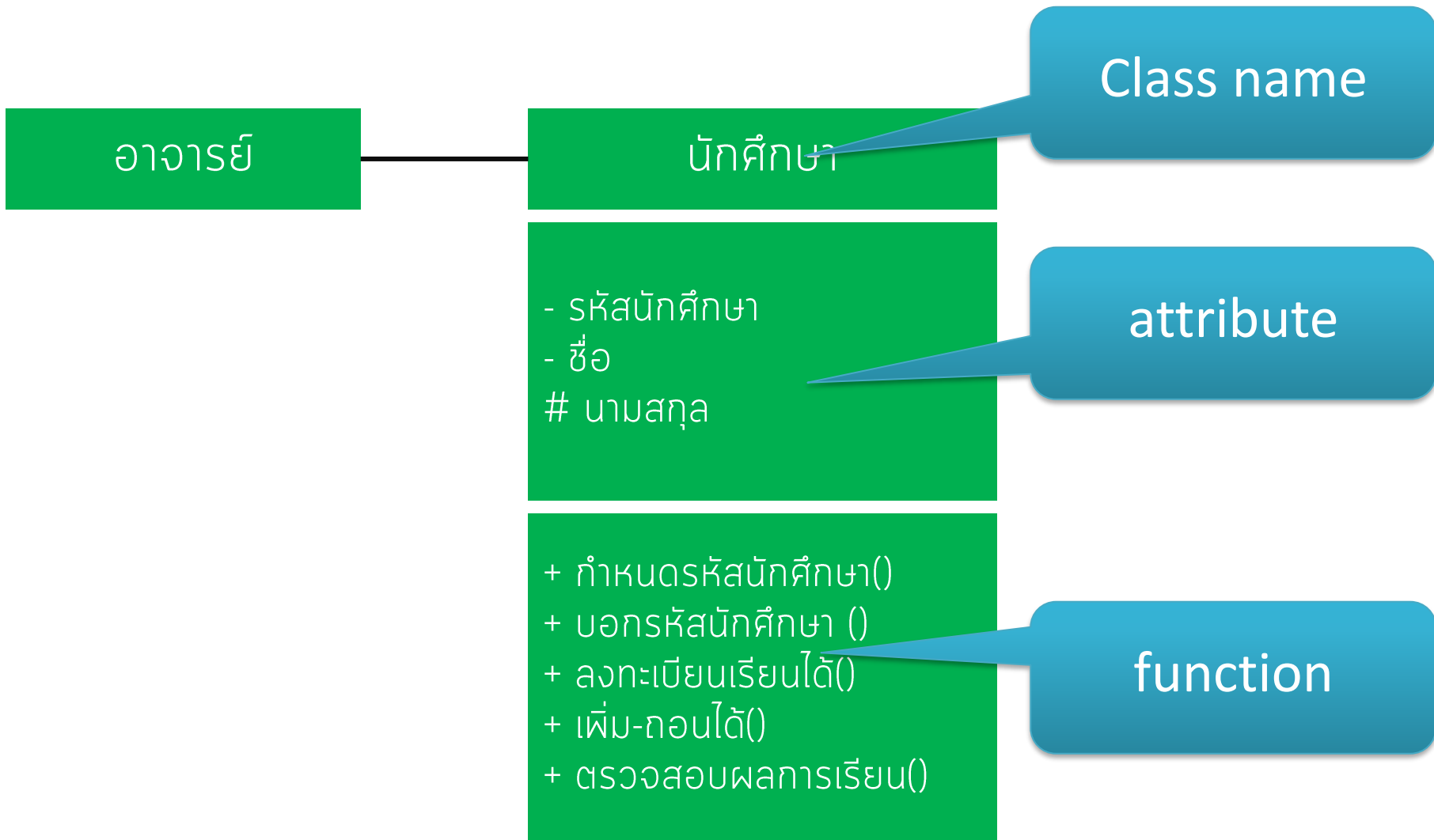


เราไม่ได้หมายถึง ณ ตอน นี้จะมีอาจารย์ หรือนักศึกษาหรือไม่

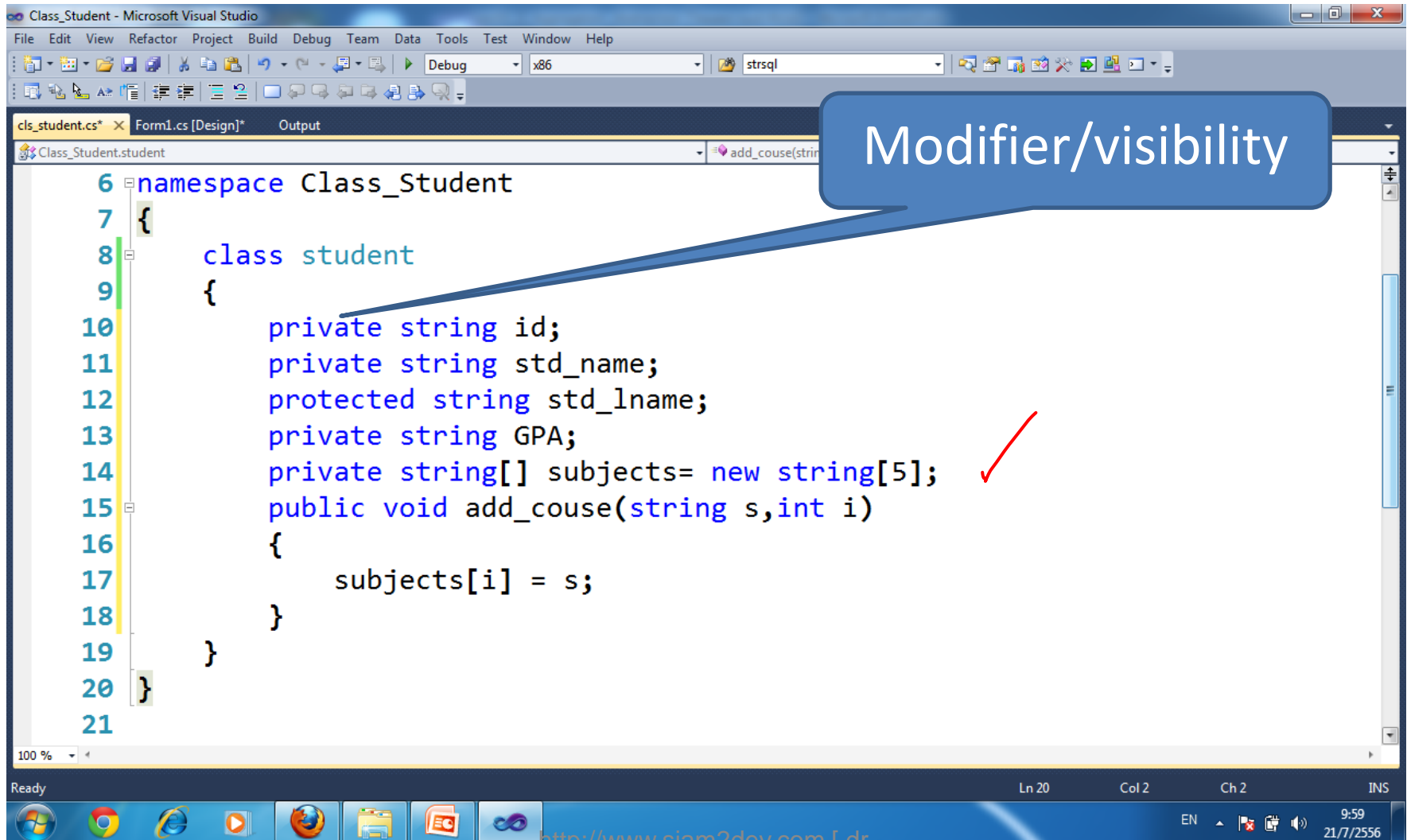
จงให้ concept กับวัตถุต่อไปนี้

- นักศึกษา





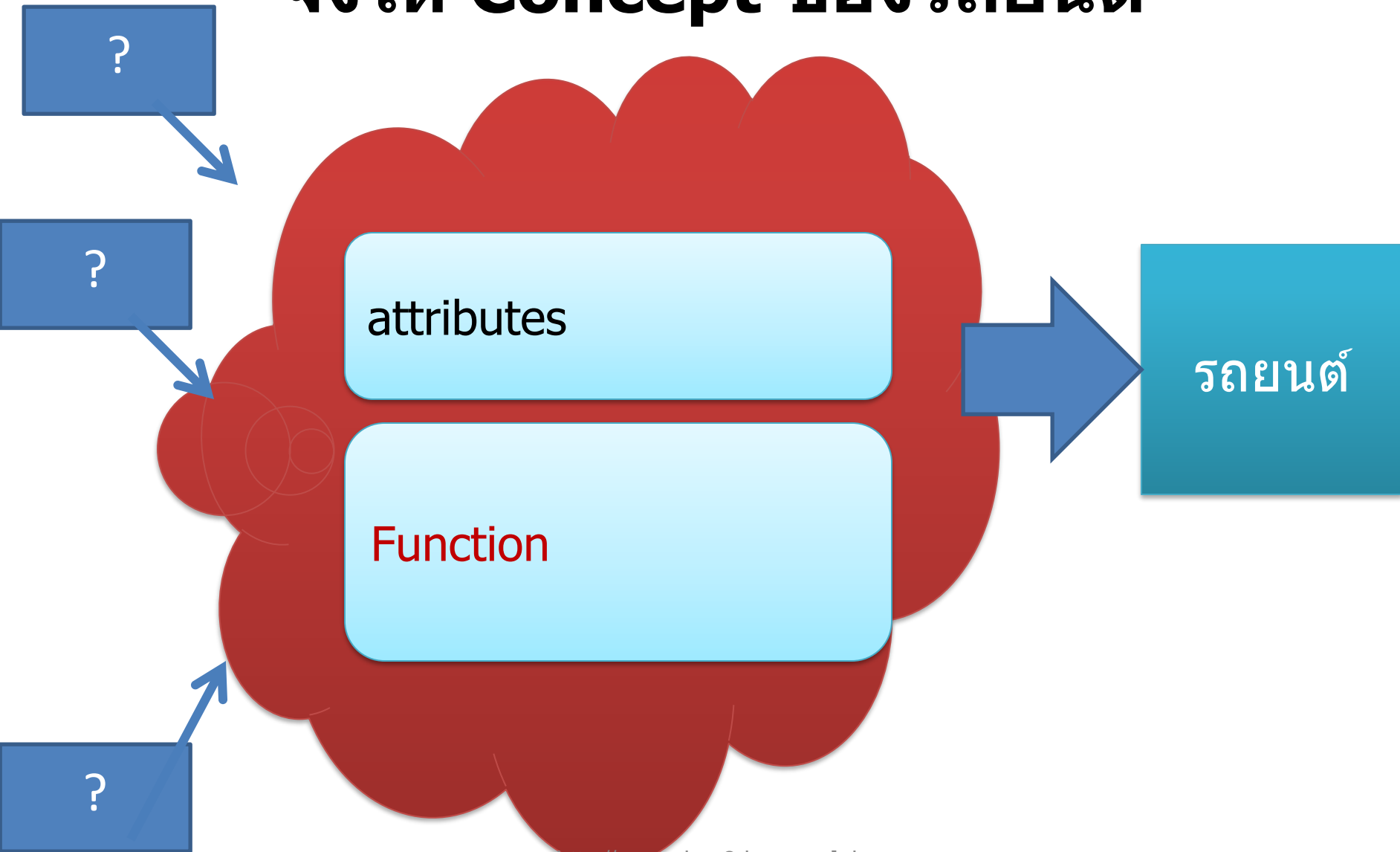
Class student with C#.net



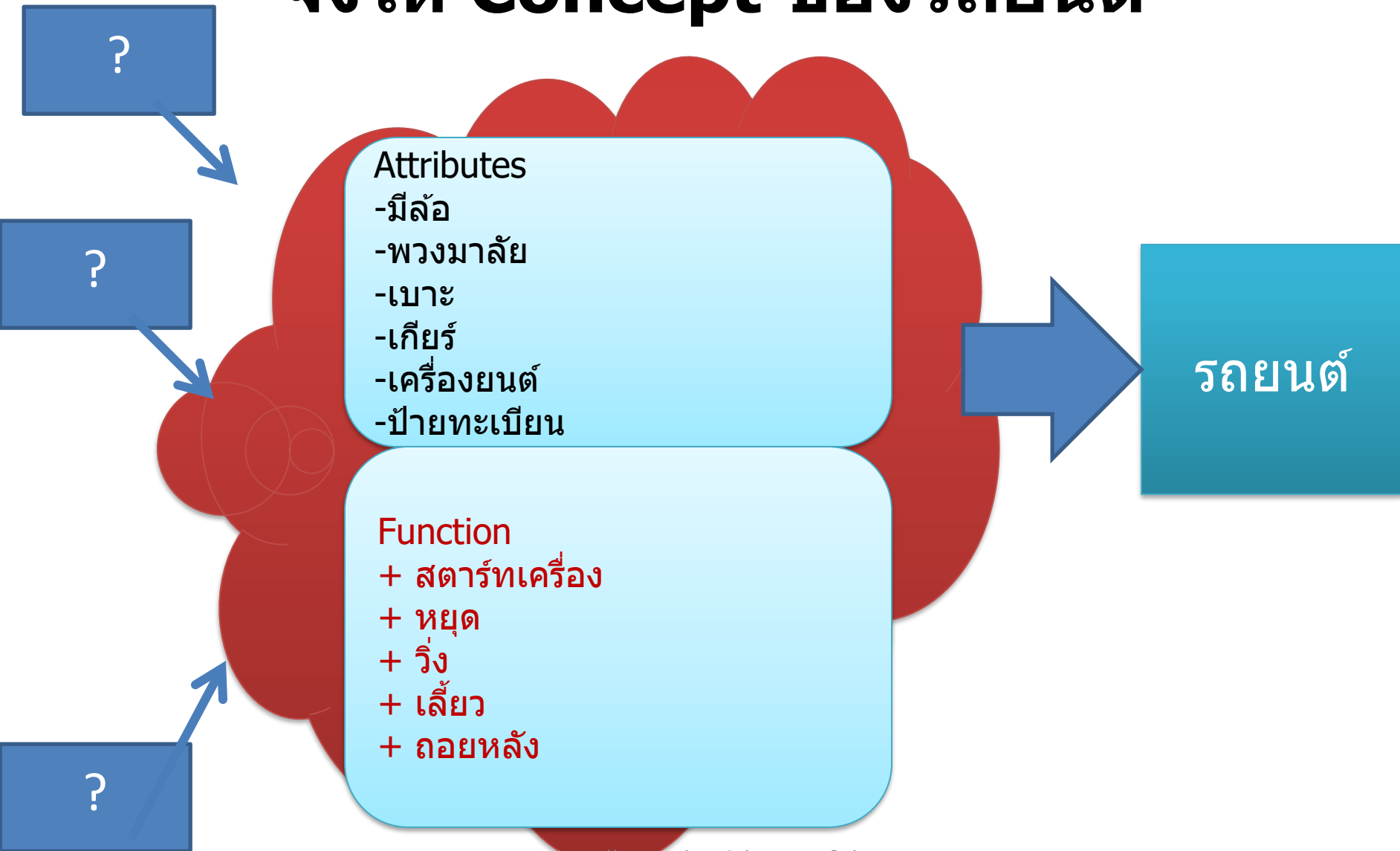
```
6 namespace Class_Student
7 {
8     class student
9     {
10         private string id;
11         private string std_name;
12         protected string std_lname;
13         private string GPA;
14         private string[] subjects= new string[5];
15         public void add_course(string s,int i)
16         {
17             subjects[i] = s;
18         }
19     }
20 }
21
```

Modifier/visibility

จงให้ Concept ของรถยนต์



จงให้ Concept ของรถยนต์

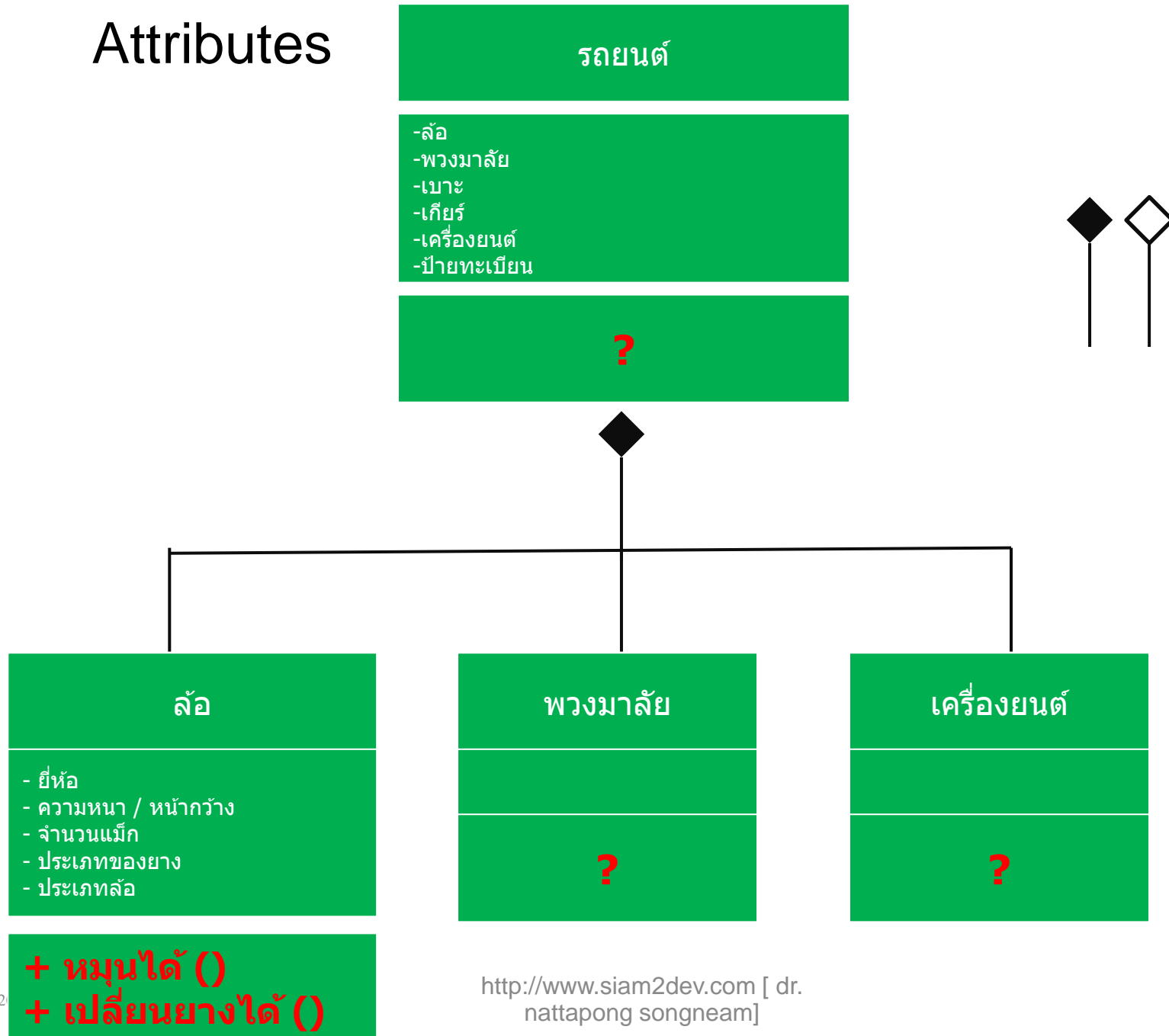


รถยนต์

?

?

Attributes



Attributes

รถยนต์

- ล้อ
- พวงมาลัย
- เบาะ
- เกียร์
- เครื่องยนต์
- ป้ายทะเบียน

?

Function

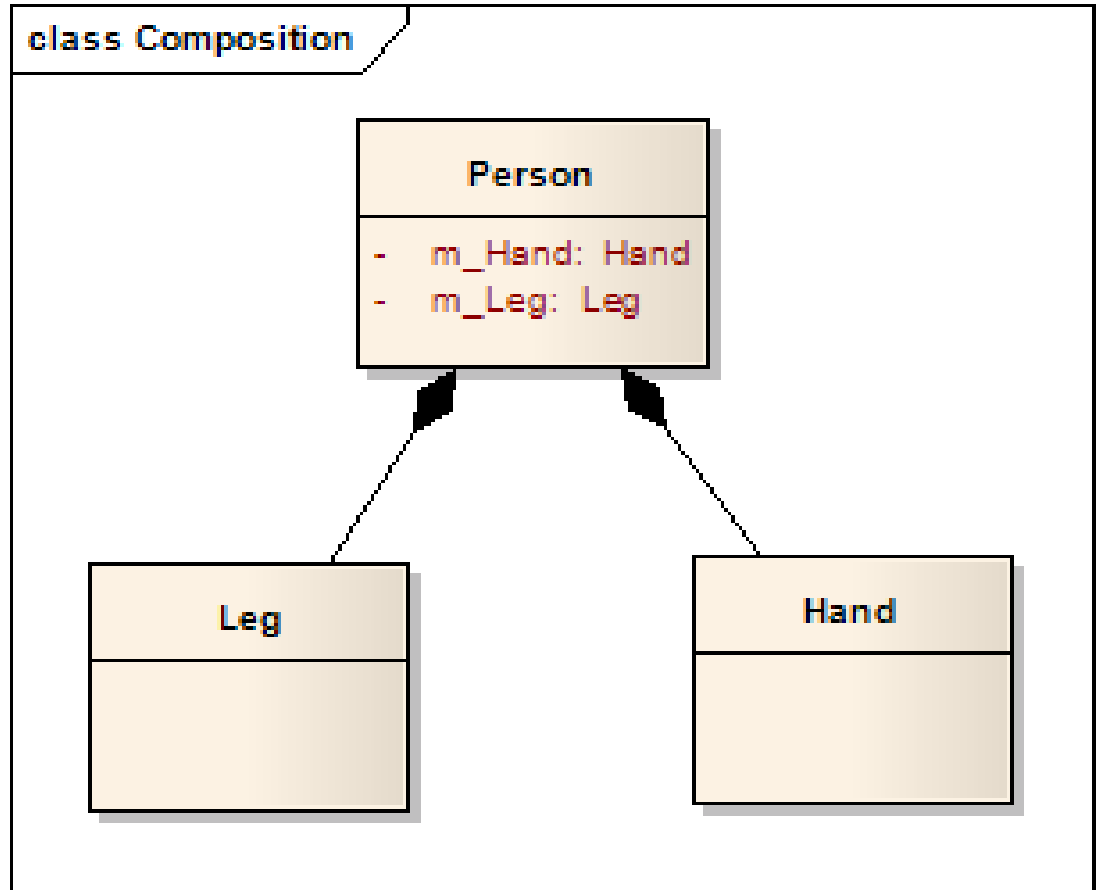
รถยนต์

- ล้อ
- พวงมาลัย
- เบาะ
- เกียร์
- เครื่องยนต์
- ป้ายทะเบียน

- + สตาร์ทเครื่อง()
- + หยุด()
- + วิ่ง()
- + เลี้ยว()
- + ถอยหลัง()

Person

- จากรูปนี้ถ้าไม่มีแขน ไม่ขา เป็น คนหรือไม่



มองในมุมมองของ composition

มองในมุมมองของ Decomposition

ห้องเรียน

ถ้ามองในทางกลับกันจะได้ว่า
Class ห้องเรียนประกอบไปด้วย
Class กระดานดำ
Class โต๊ะ
Class เก้าอี้
Class นักศึกษา
Class อาจารย์

นักศึกษา

กระดานดำ

โต๊ะ

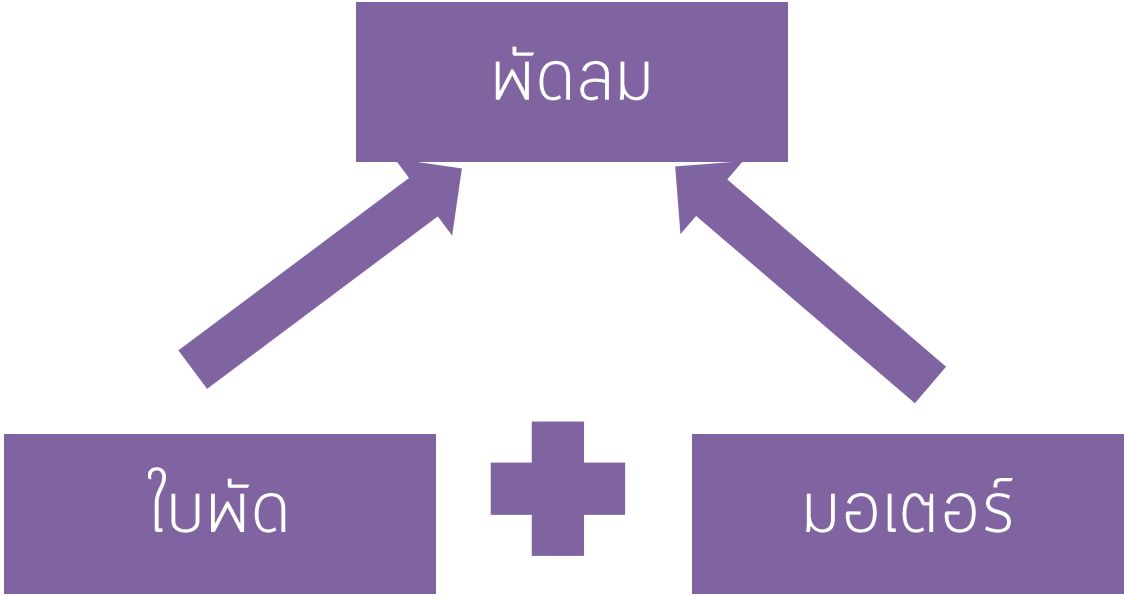
เก้าอี้

อาจารย์

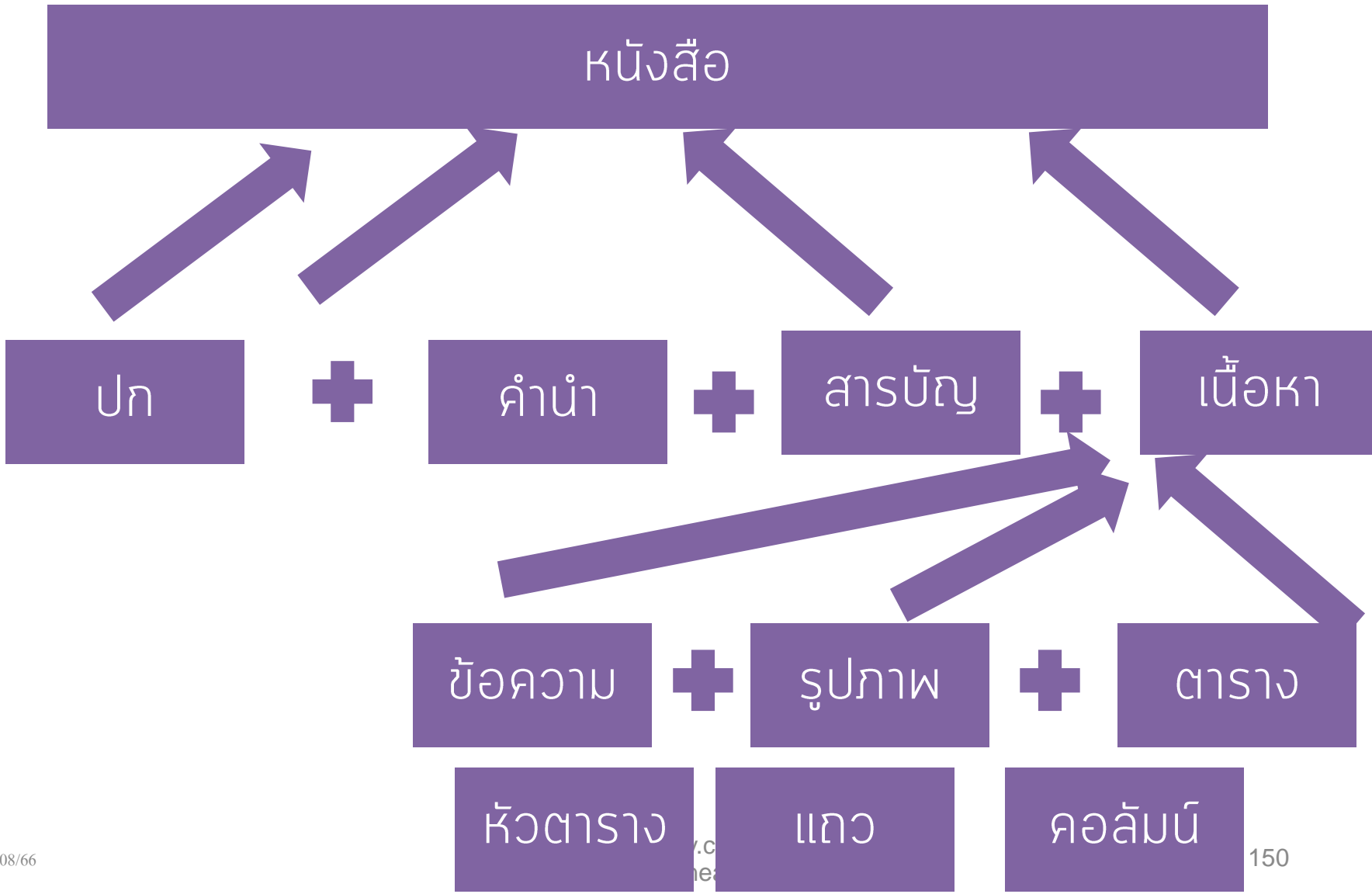
การรวมเป็นคลาสอาจมองในลักษณะ component ก็ได้

- มุมมองแบบ Composition
- มุมมองแบบ Decomposition

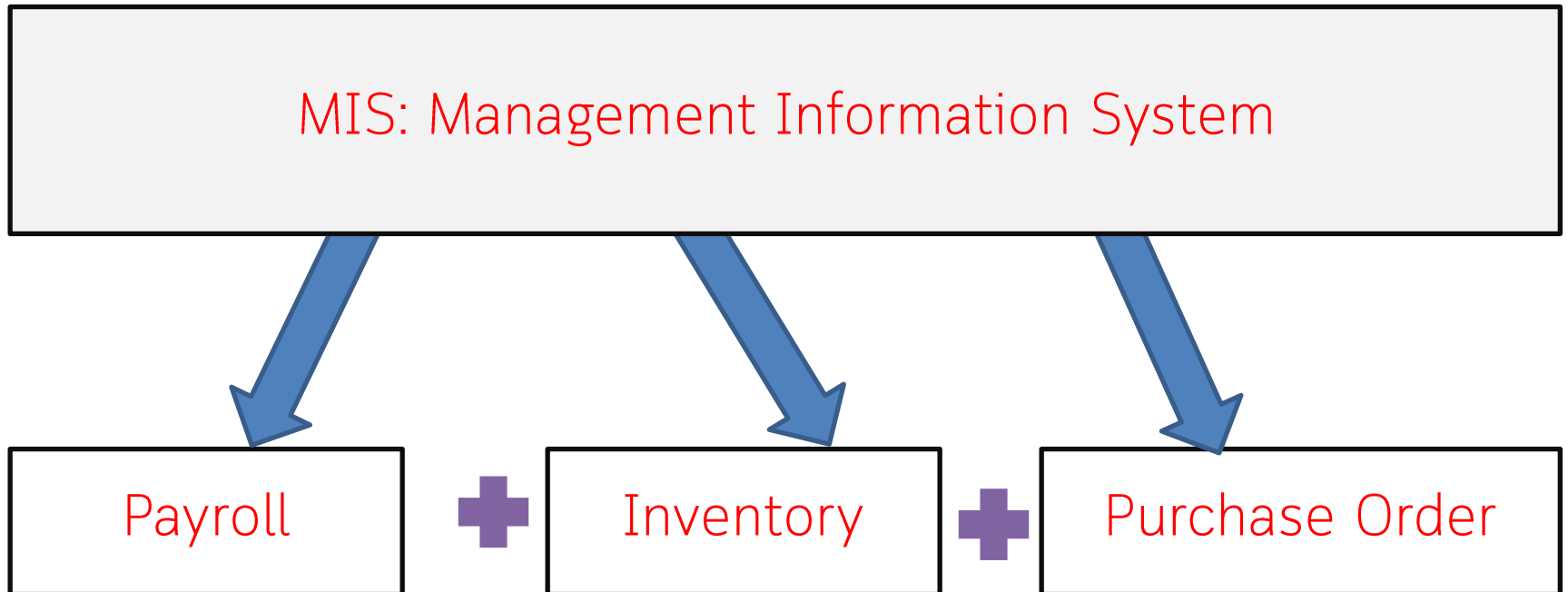




มองในมุมมองของ composition คลาสย่อยรวมกันเป็นคลาสใหญ่



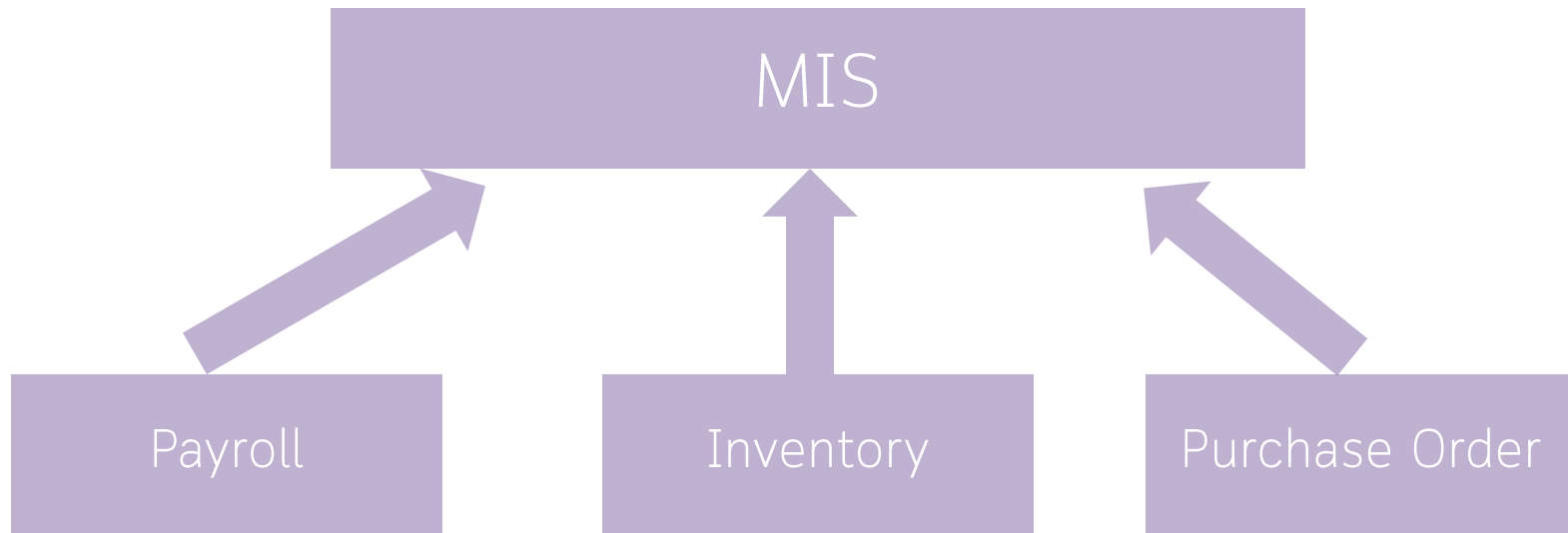
Decomposition



ถ้ามองในเชิง composition จากรูปนี้จะอ่านว่าอย่างไร

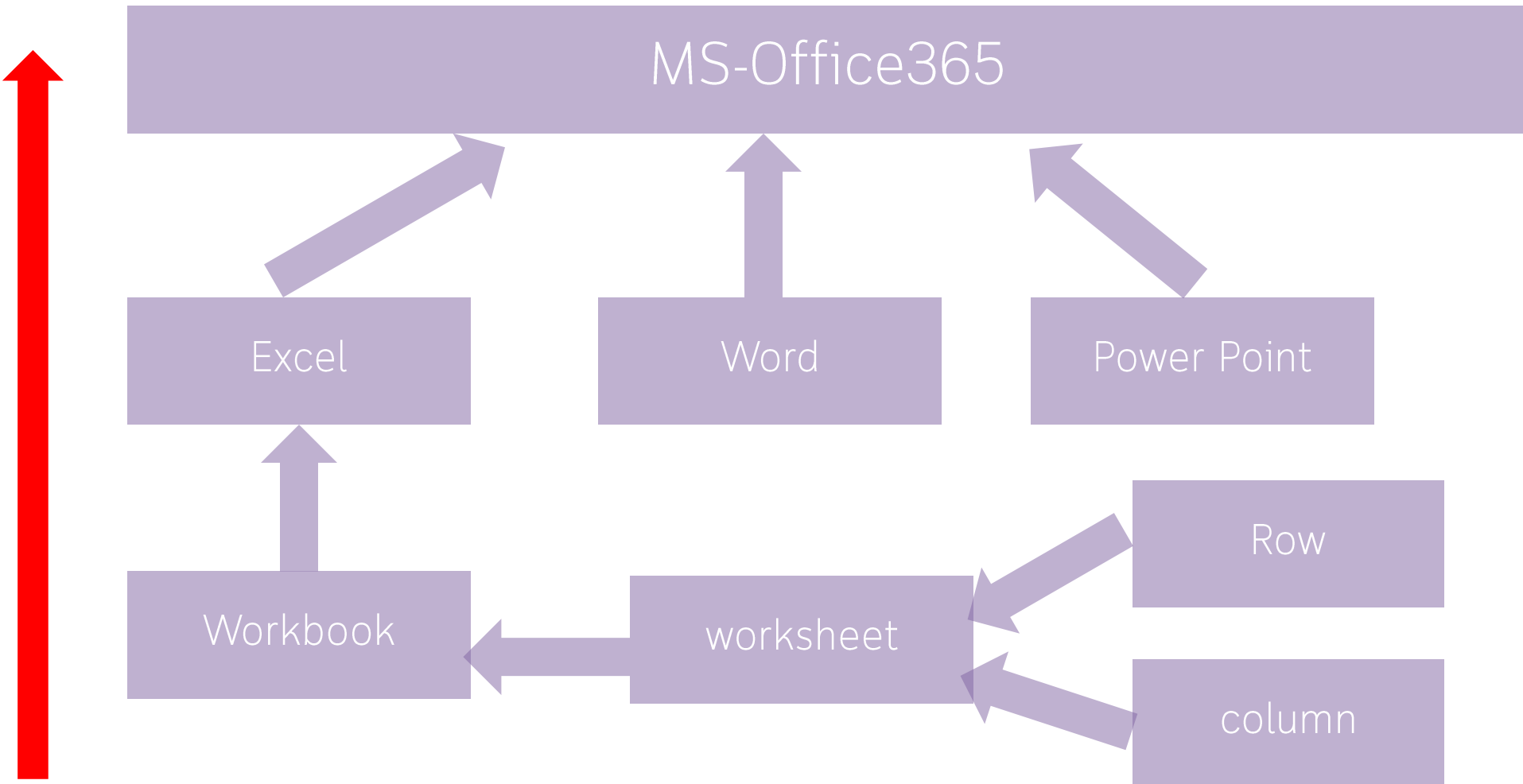
Composition

ระบบเงินเดือน รวมกับ ระบบคลังสินค้า และระบบสั่งซื้อกลายเป็นระบบ MIS



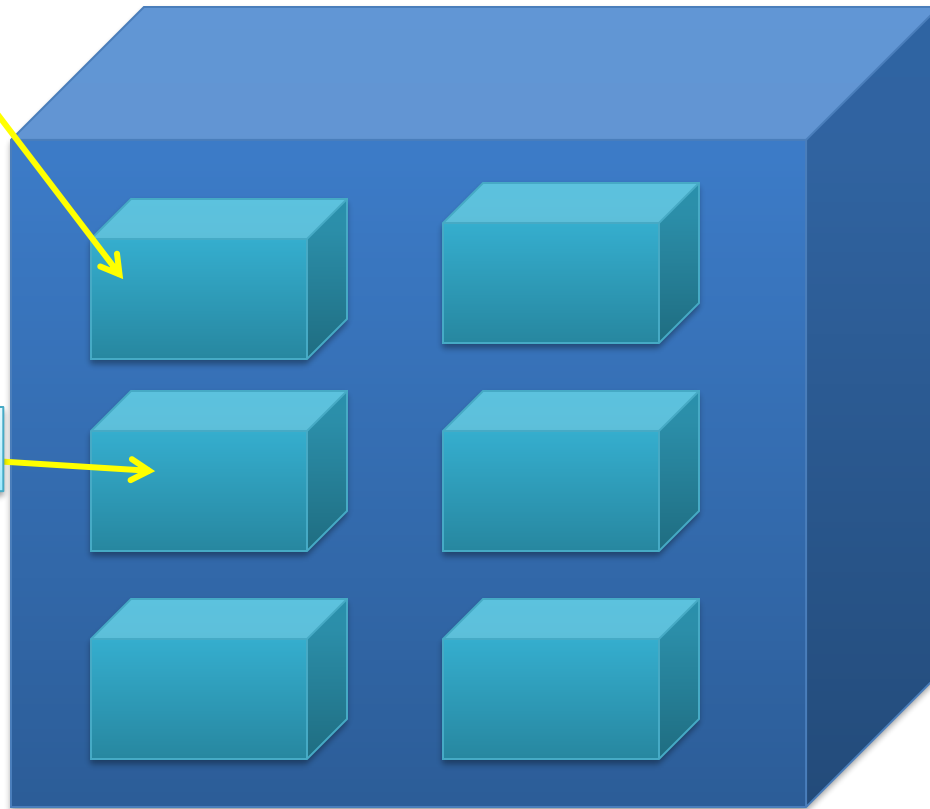
Composition

ระบบเงินเดือน รวมกับ ระบบคลังสินค้า และระบบสั่งซื้อกลายเป็นระบบ MIS



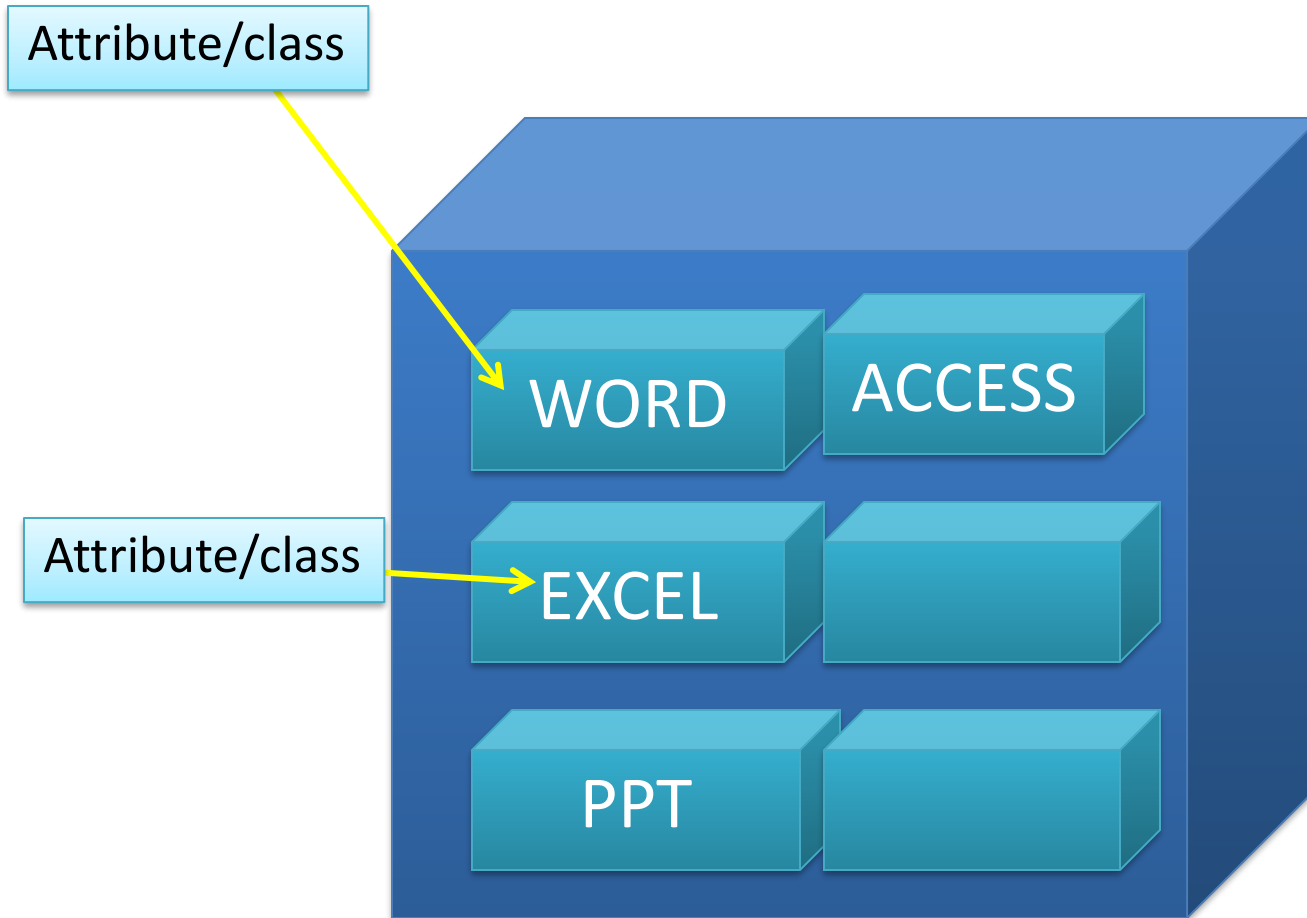
components

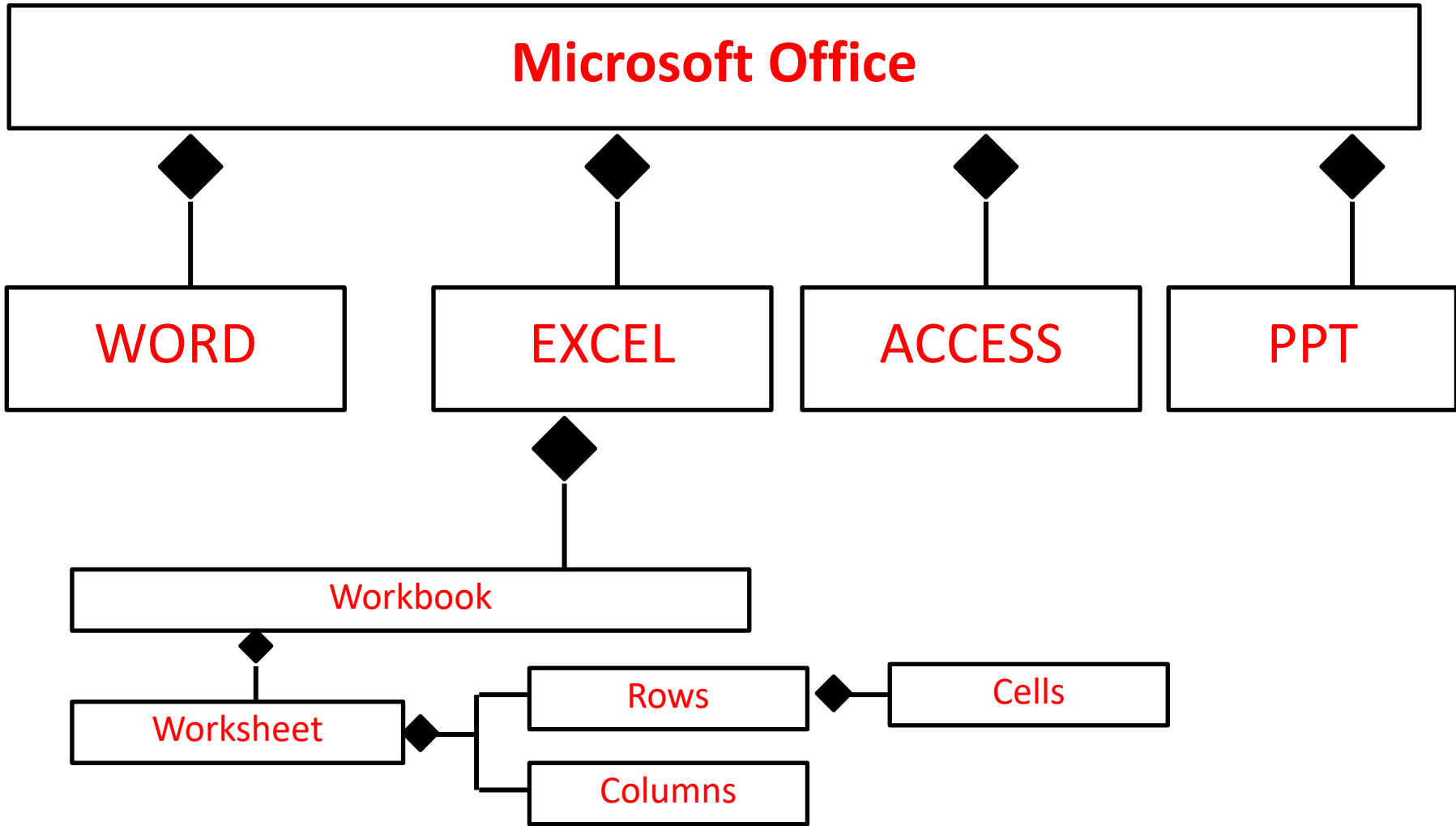
Attribute/class

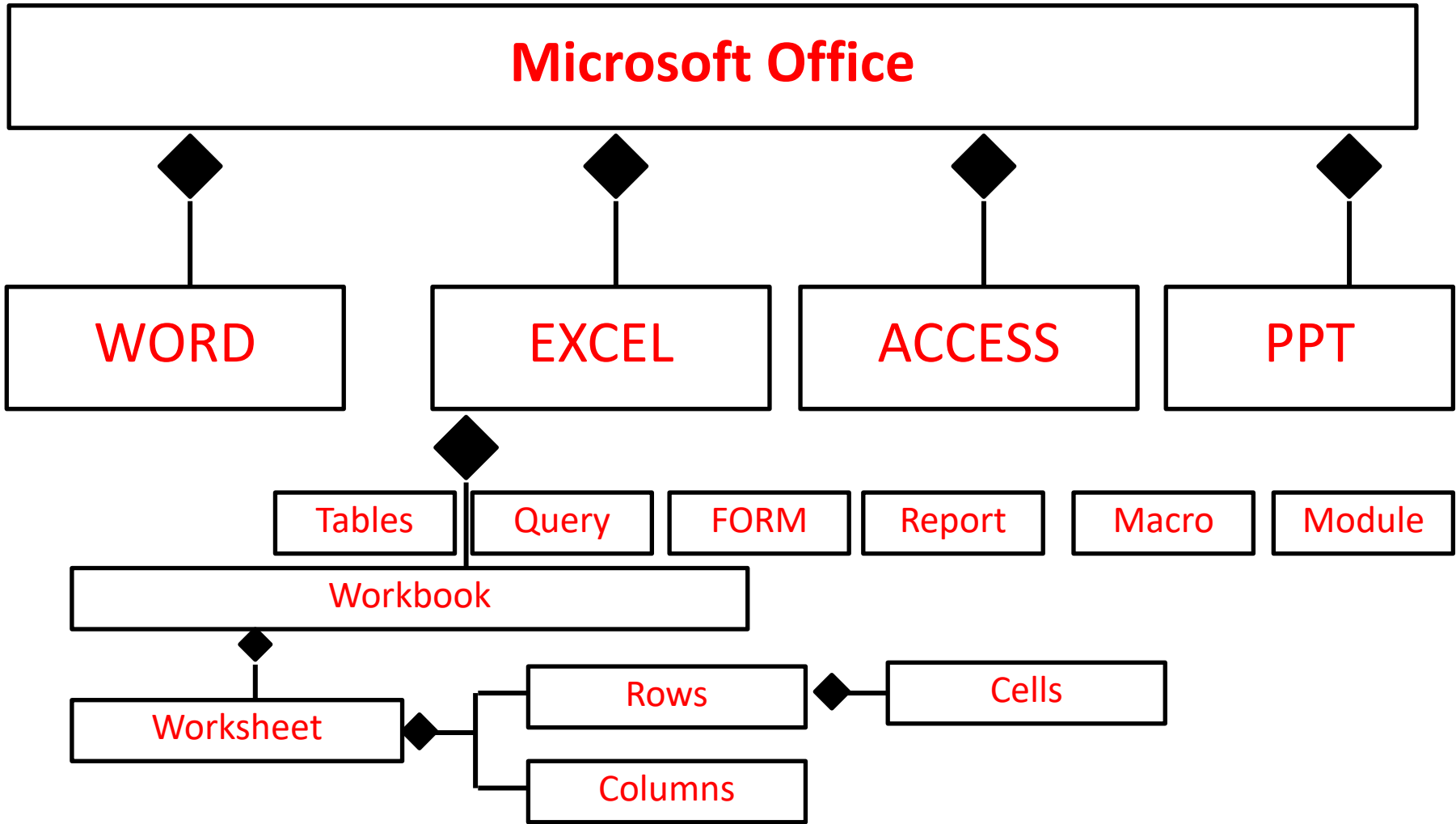


Attribute/class

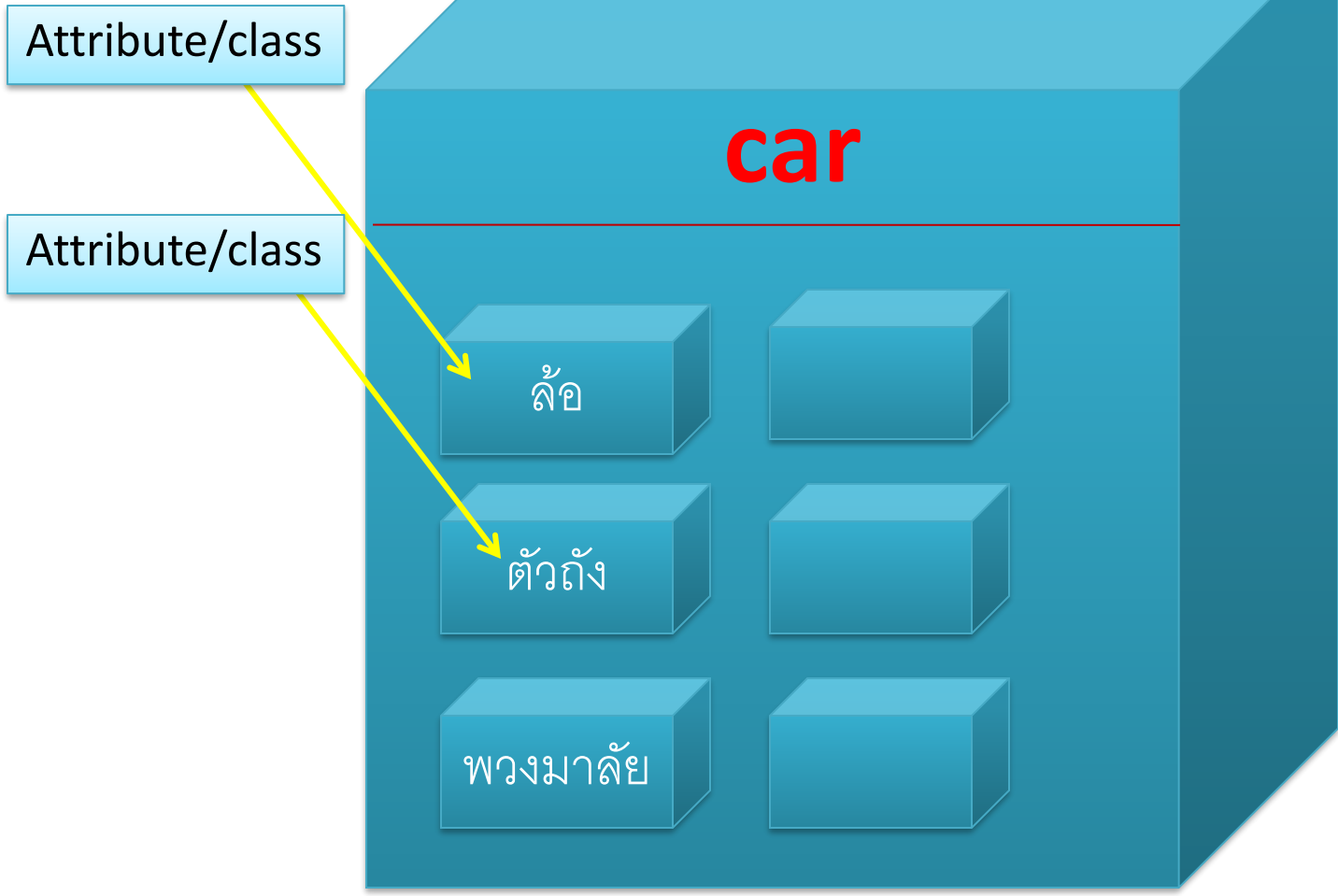
MS_OFFICE



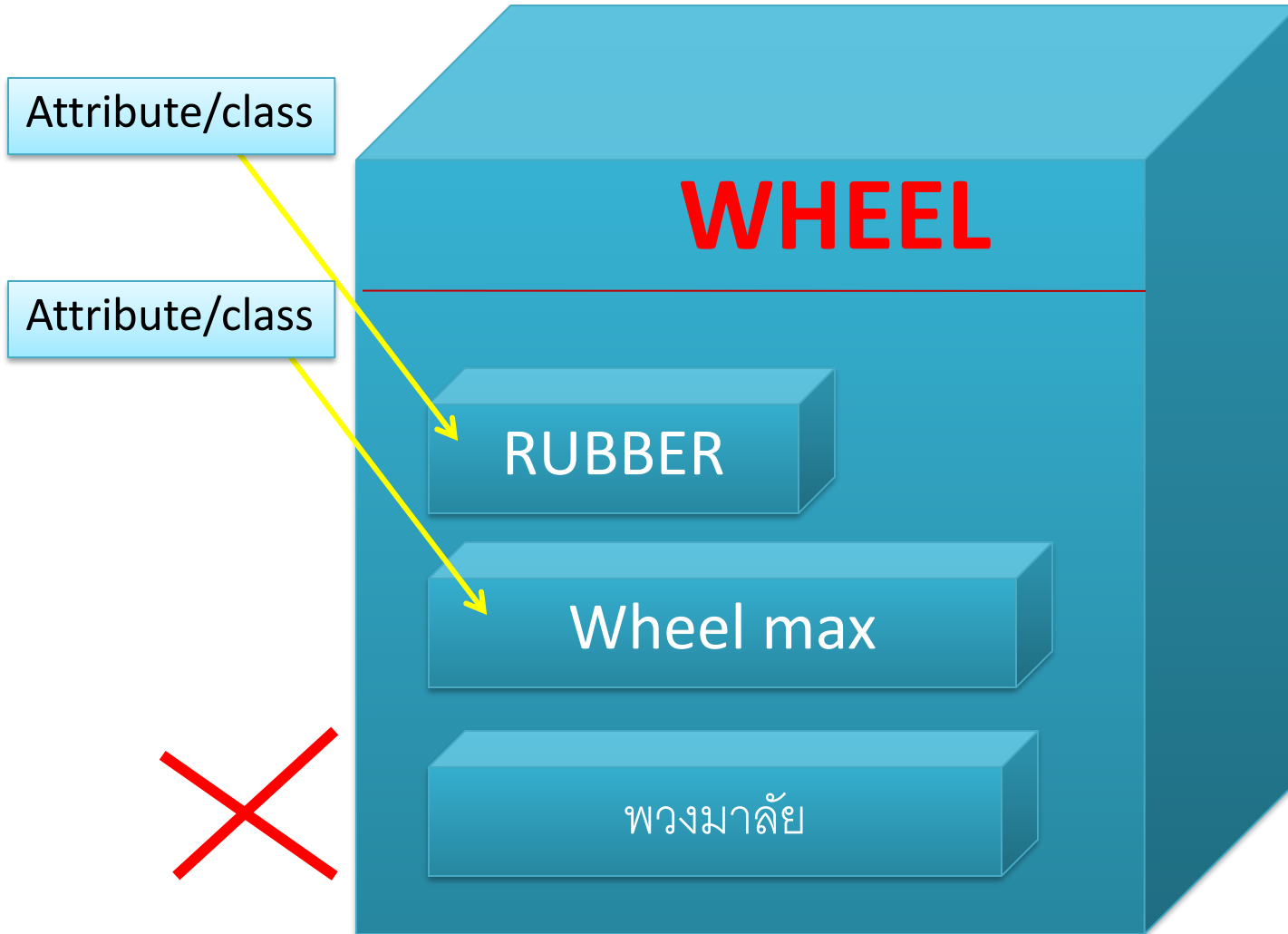




components



components





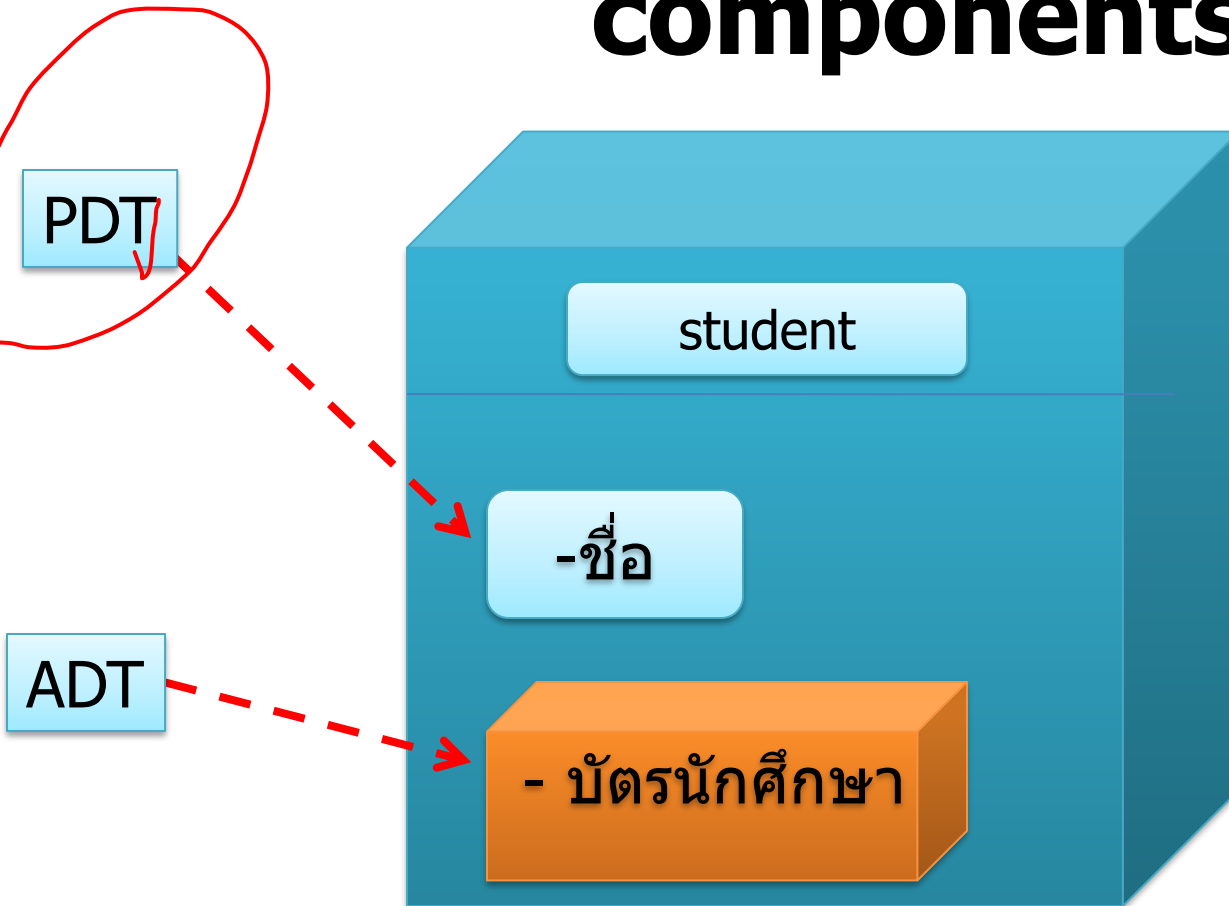
ตัวอย่างของ Component ใน C#.net

```
7 {
8   class cls_Car
9   {
10      wheel w;
11      engine e;
12   }
13   class wheel
14   {
15      string rubber;
16      string max;
17   }
18   class engine
19   {
20      string brandname;
21      string model;
22      public void start_engine() { }
23      public void stop_engine() { }
24   }
25 }
```

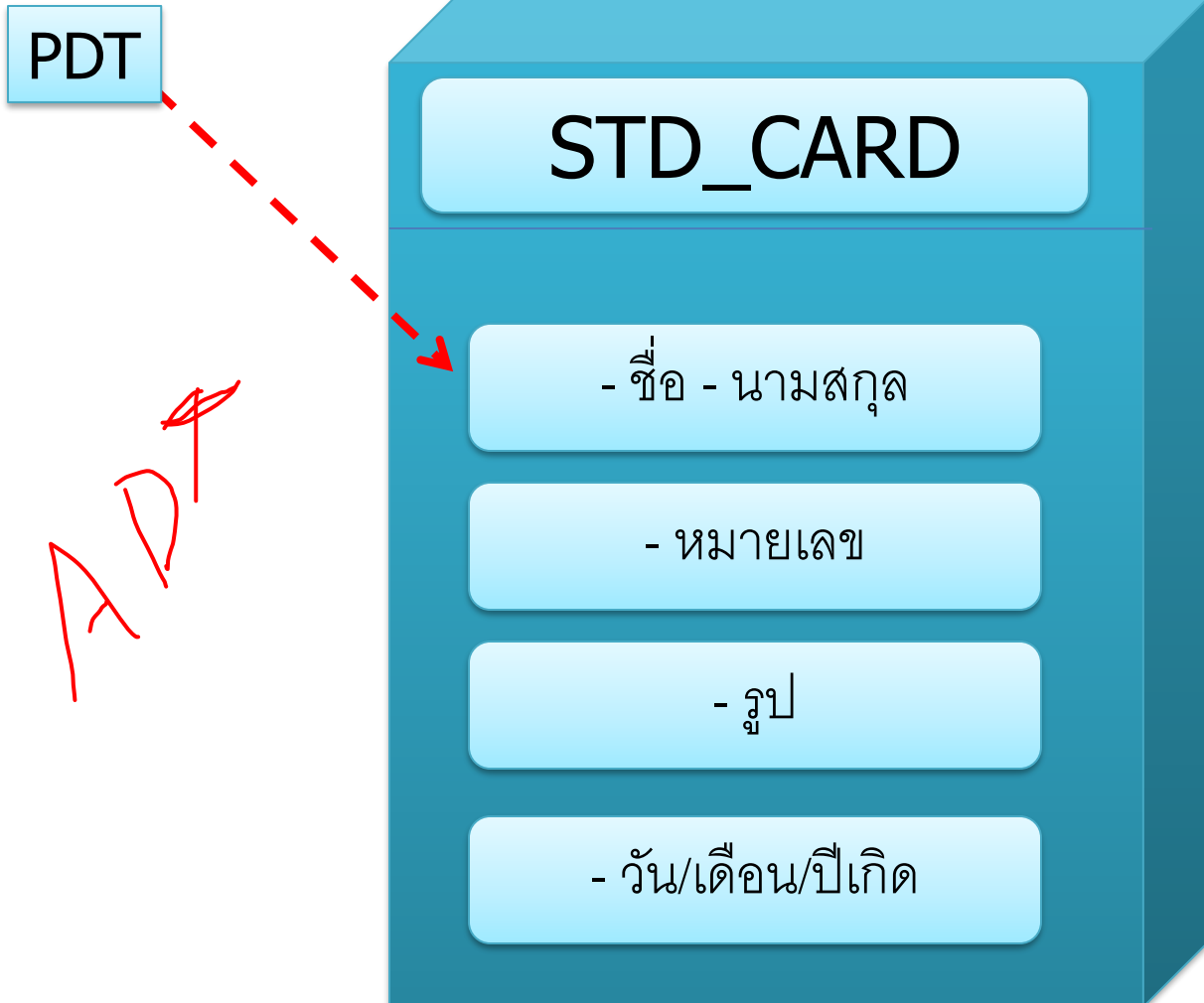
Handwritten red annotations: A large red bracket on the right side of the code block, underlining the `class wheel` and `class engine` definitions, and the text "PDT" written in red across the `engine` class.

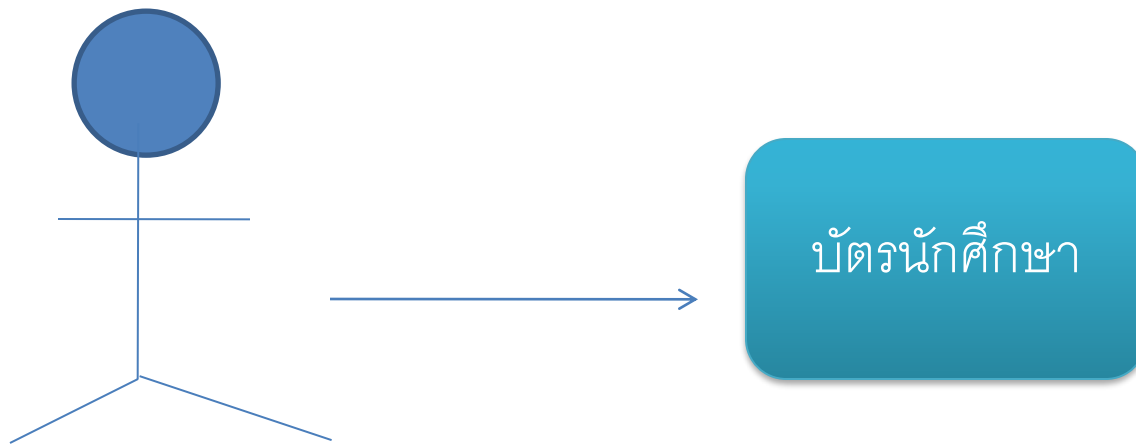


components



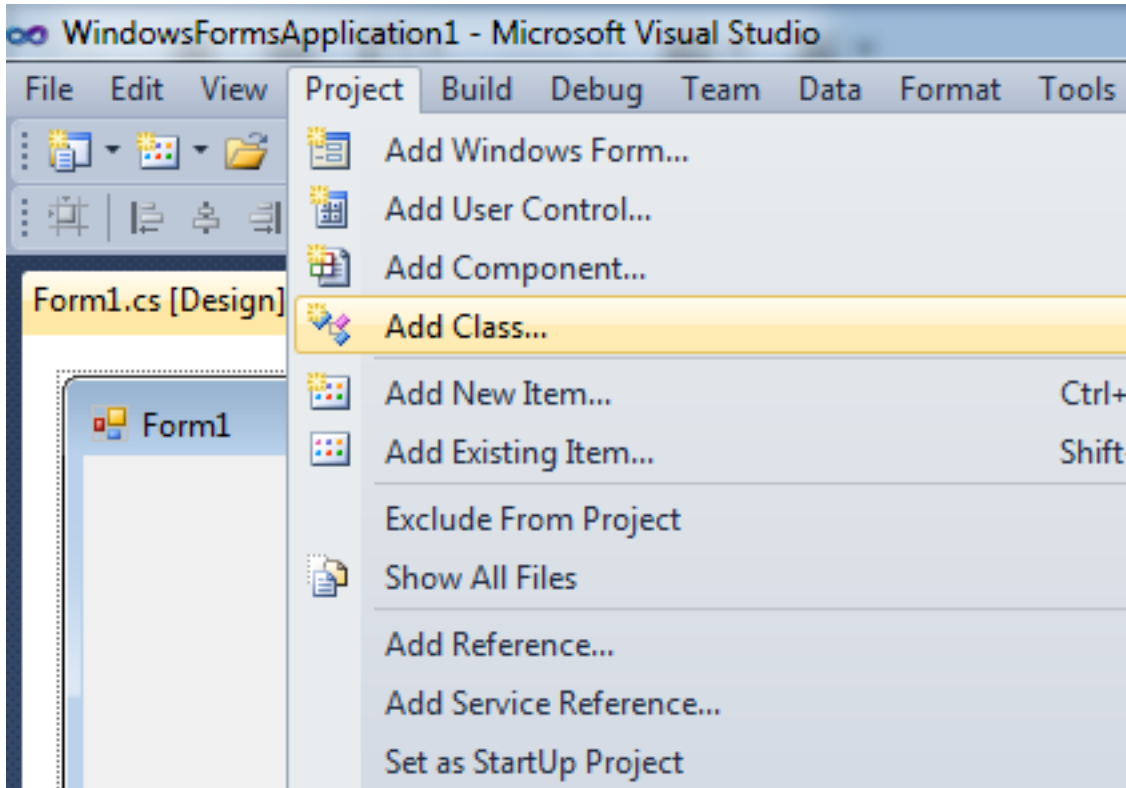
components



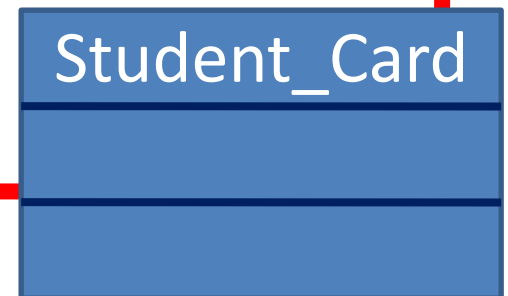
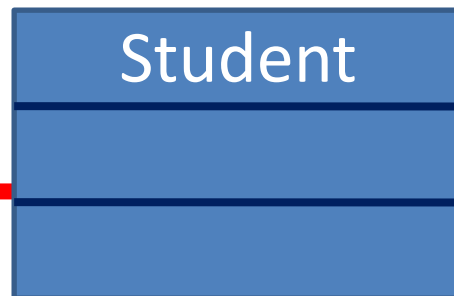


Student

Add class in C#



```
3 using System.Linq;
4 using System.Text;
5
6 namespace WindowsFormsApplication1
7 {
8
9     class Student { }
10    class Student_Card { }
11
12 }
13
```



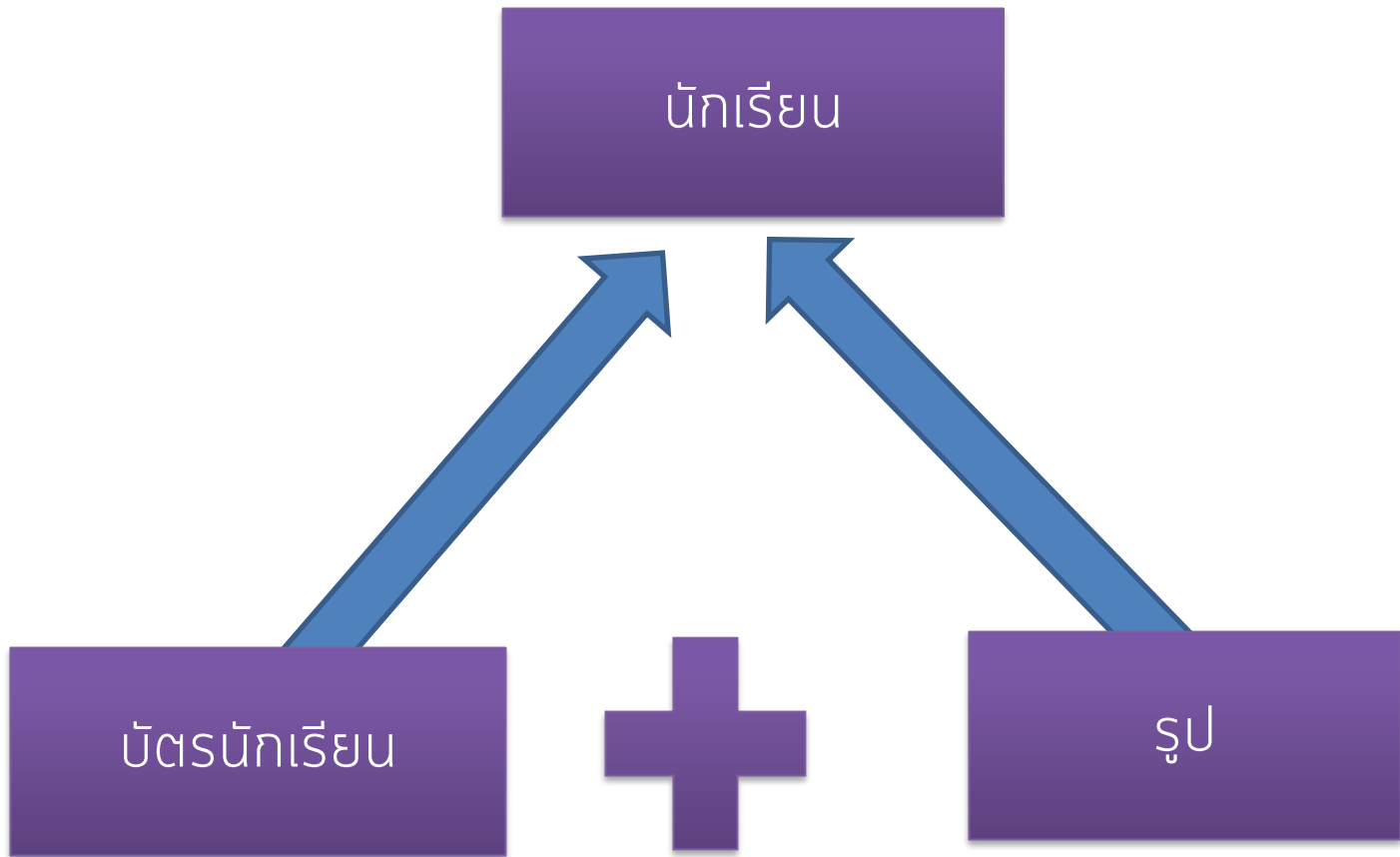
```
6 namespace WindowsFormsApplication1
7 {
8
9     class Student {
10         private string name;
11         private string ID;
12         private Student_Card st_card;
13
14     }
15     class Student_Card {
16         private string ID;
17         private string Image;
18         private string address;
19     }
20
21 }
```

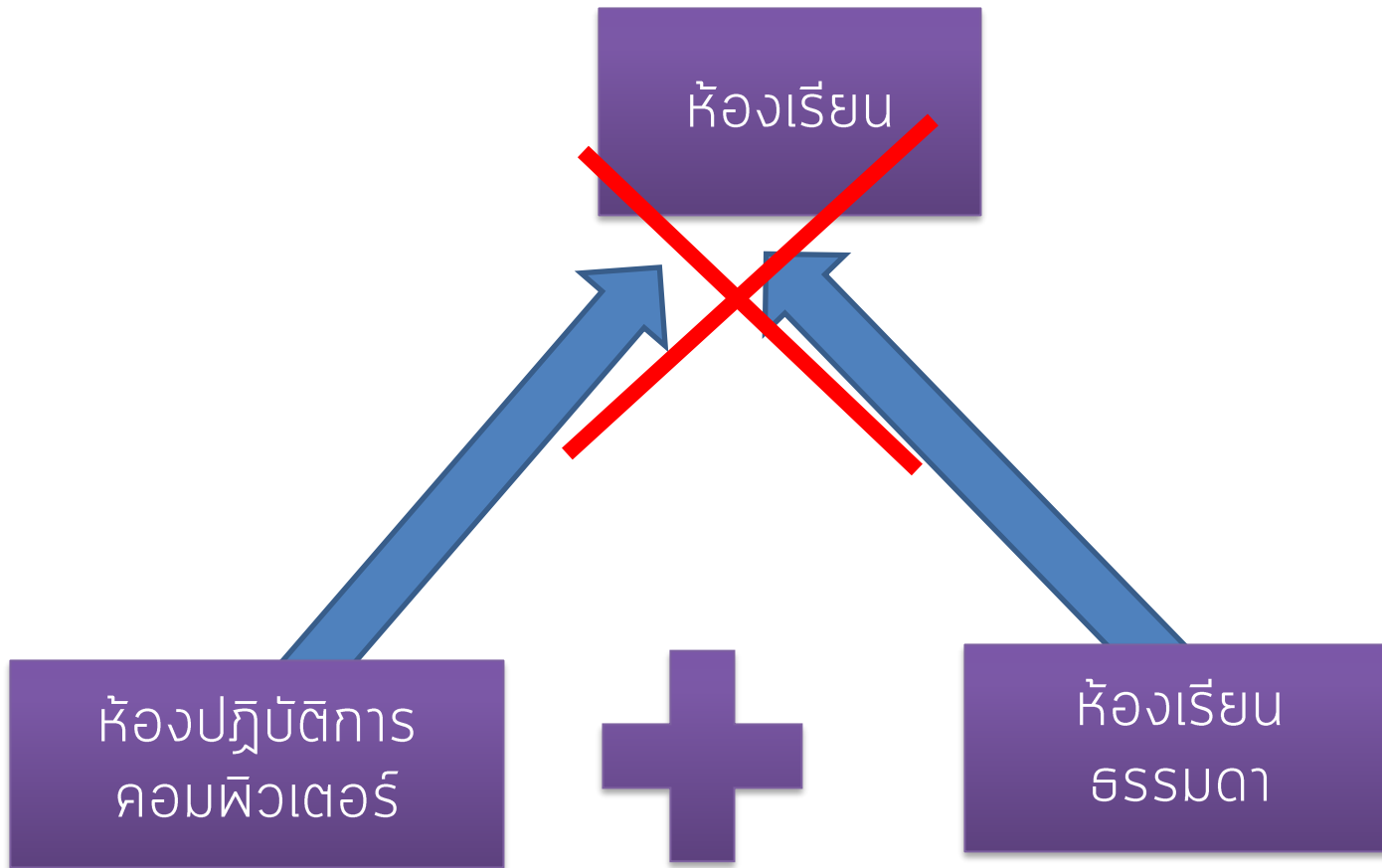
Data member

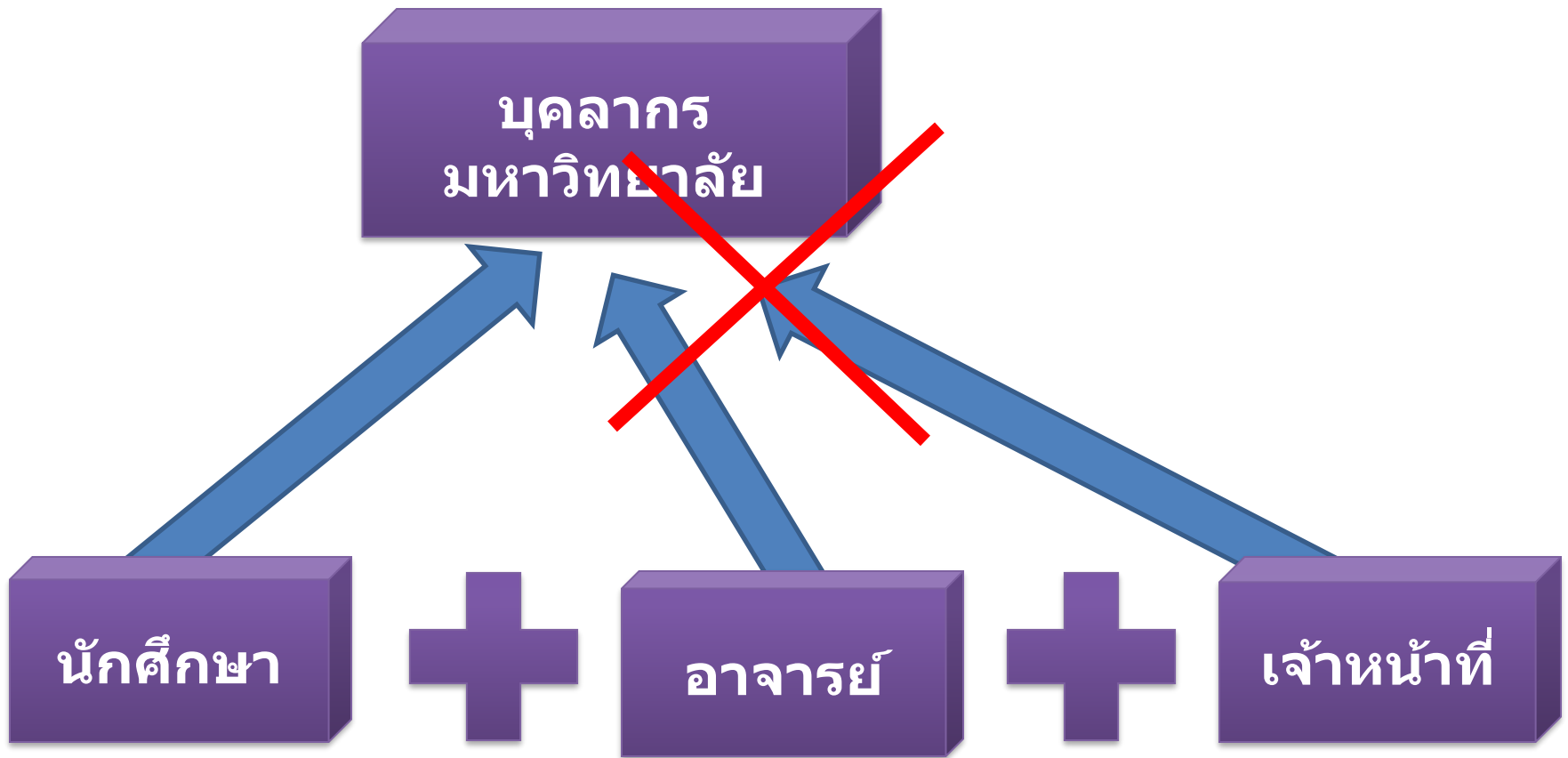
class member

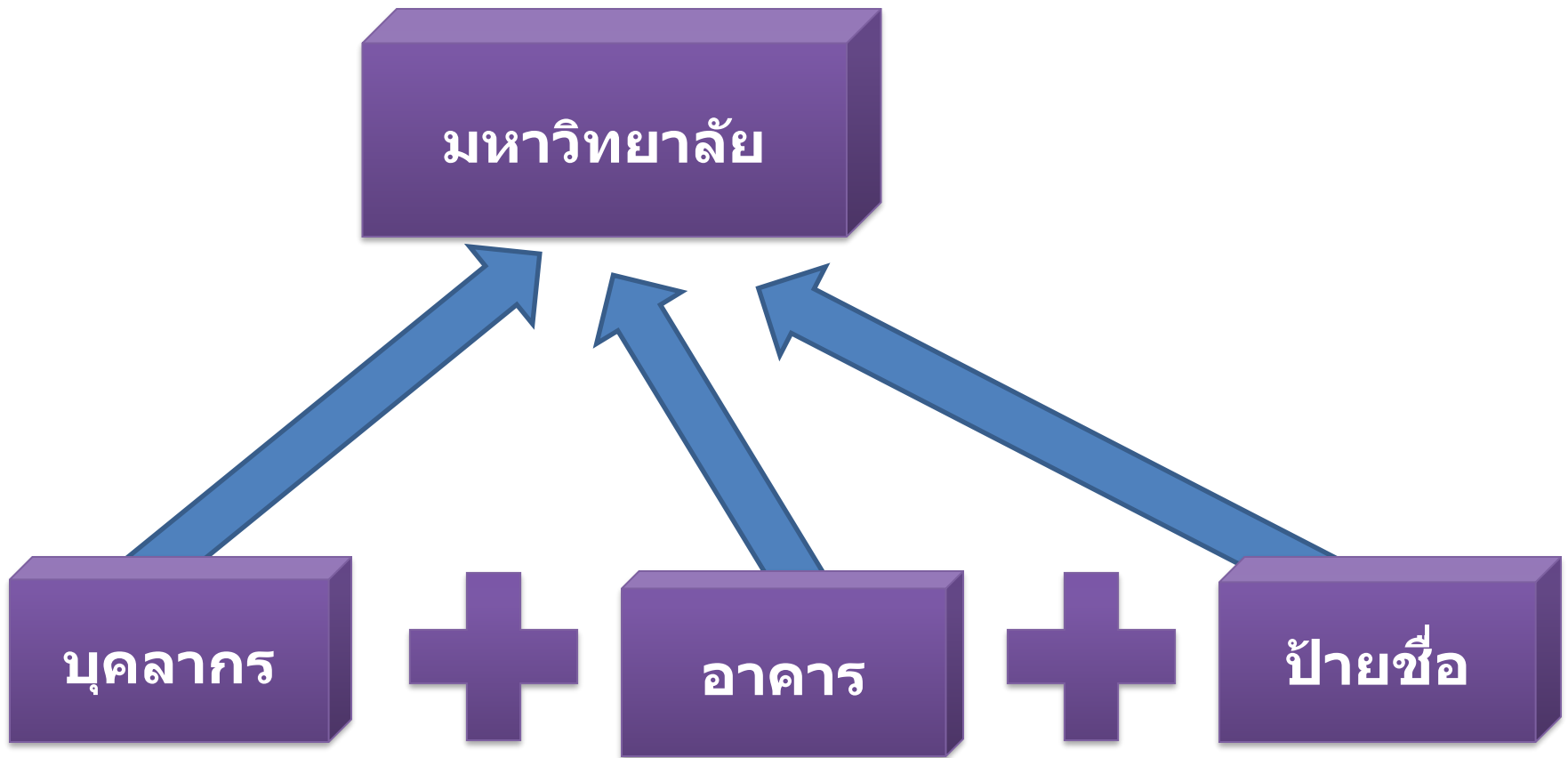
Is part of

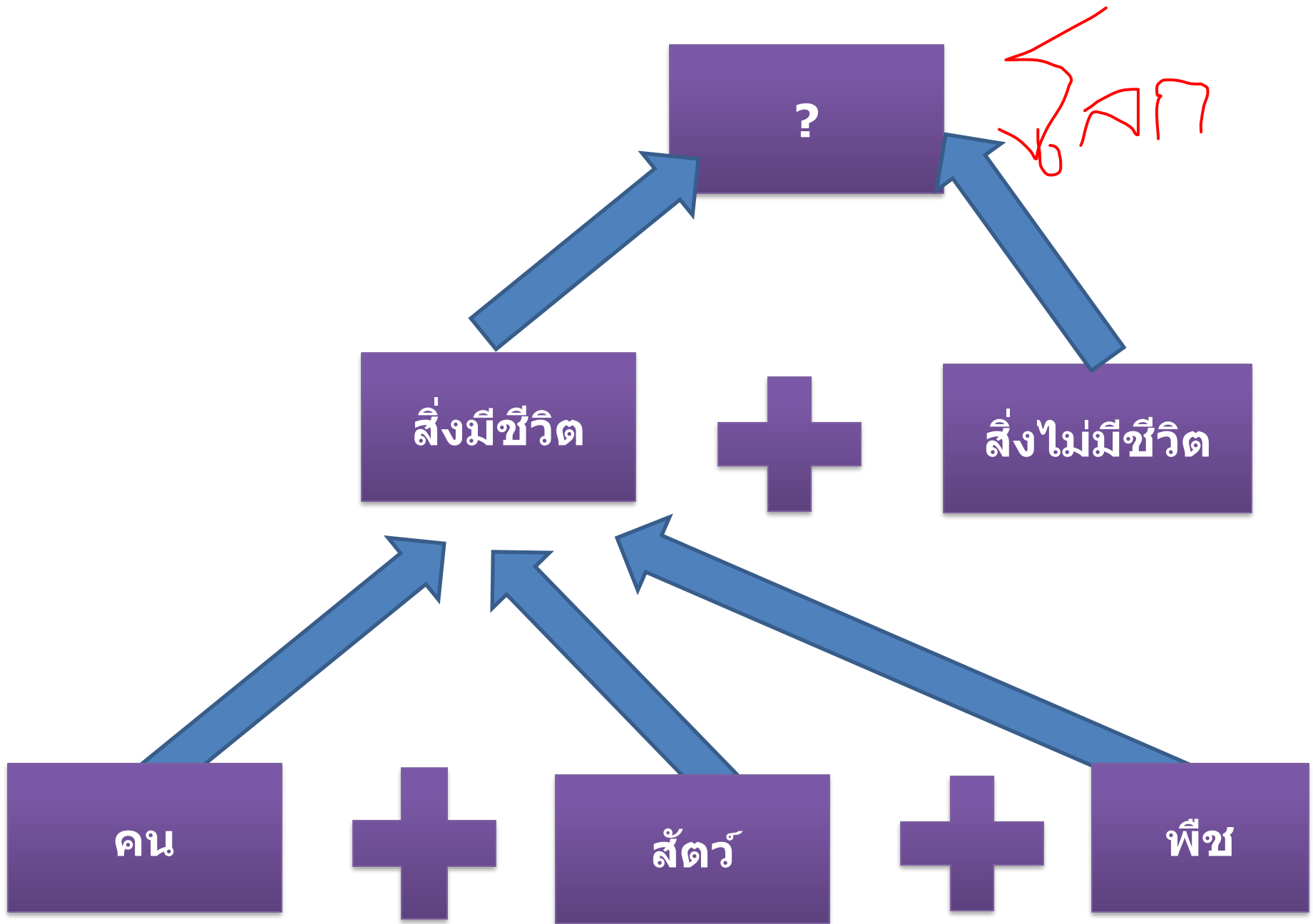






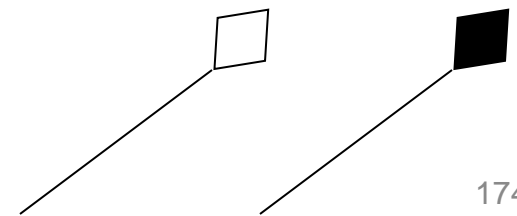






Aggregation Abstraction

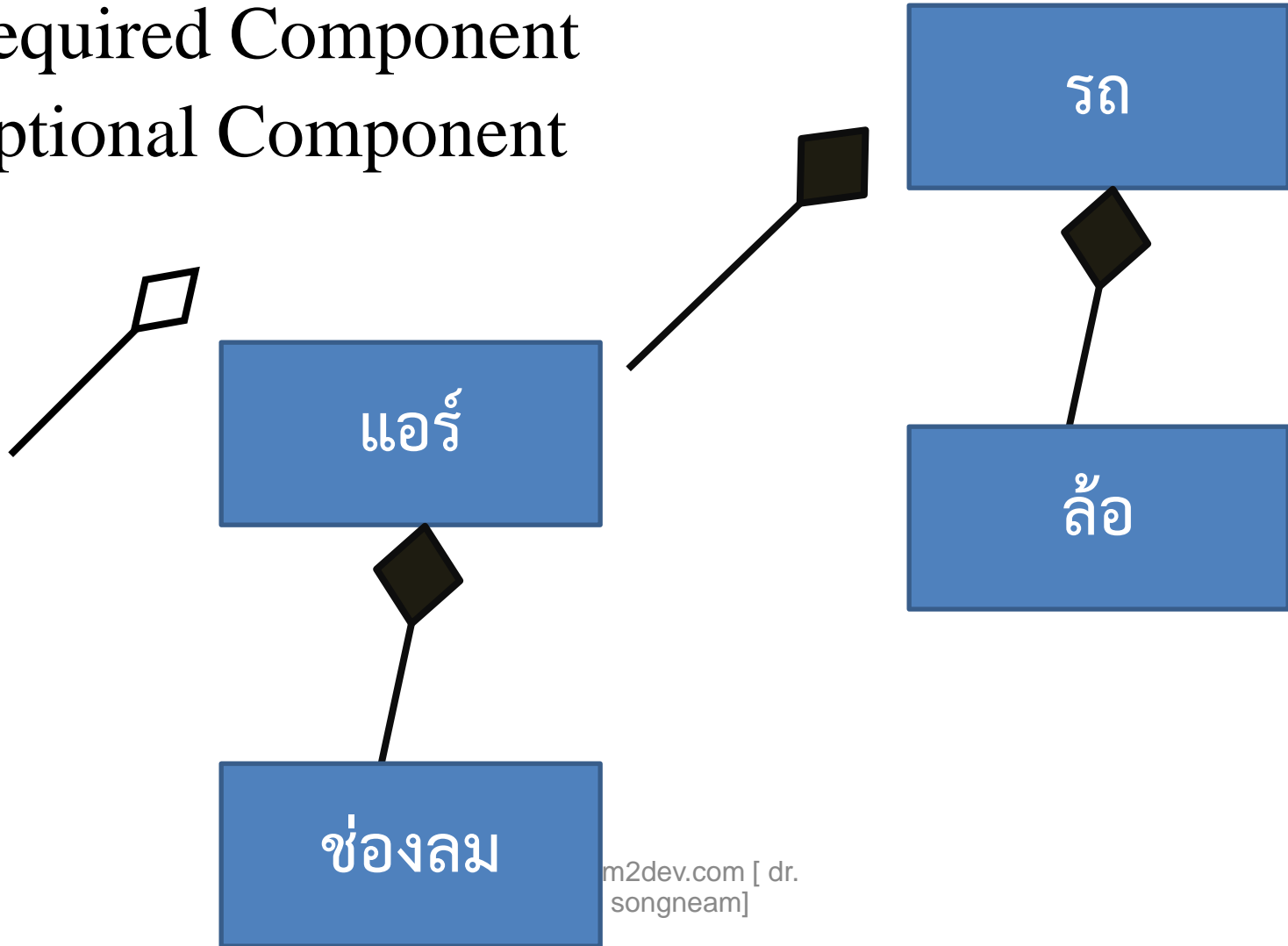
- คือ การ พยายามตอบคำถามที่ว่า มี class ใดเป็นส่วนประกอบ (Is part of) ของ class อื่นหรือไม่ และที่สำคัญ *“การประกอบกันของ class ต้องทำให้เกิด class ใหม่ ซึ่งมี concept ใหม่ด้วย”*
- ในทาง object orientation นั้น การแสดงสัญลักษณ์เพื่อแสดง Aggregation Abstraction ของ class นั้น ทำได้โดยการโยงลูกศร เป็นสี่เหลี่ยมขนมเปียกปูน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



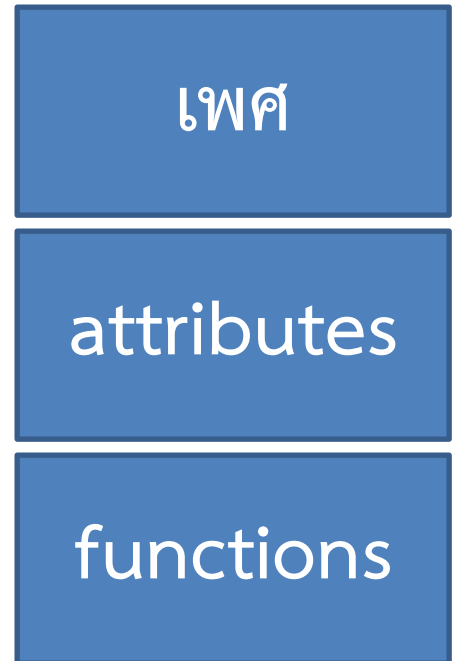
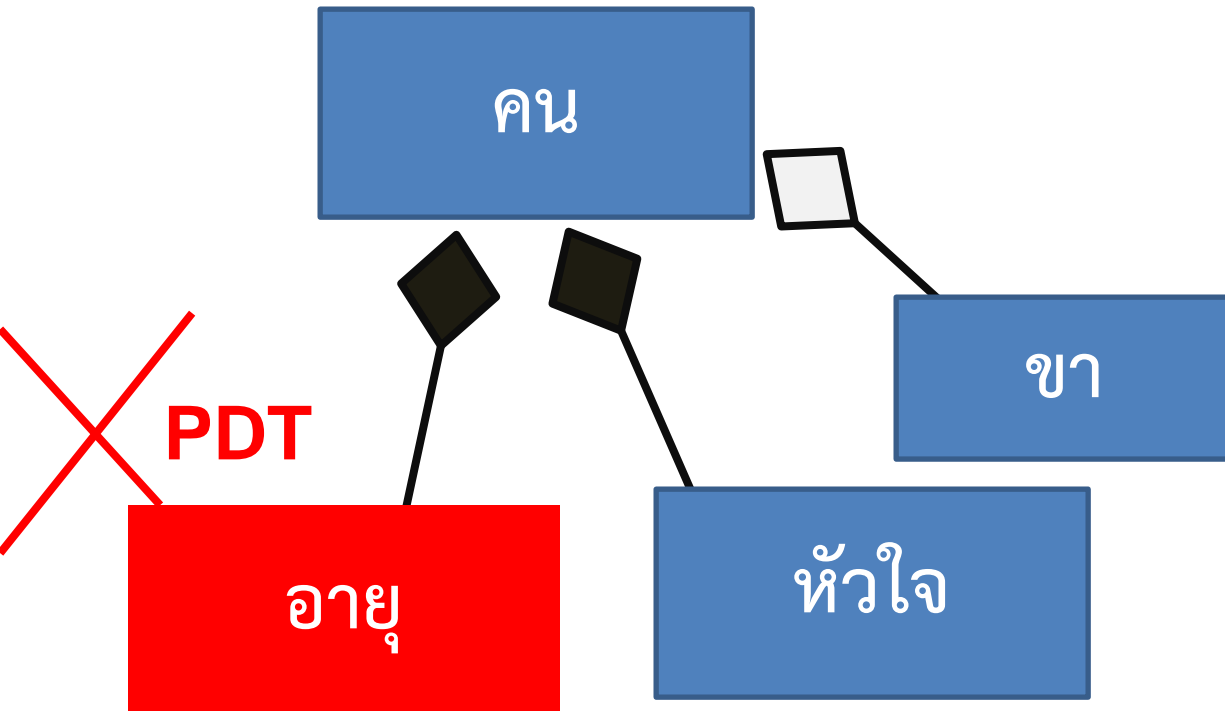
ตัวอย่างเช่น
รถจำเป็นต้องมีล้อ

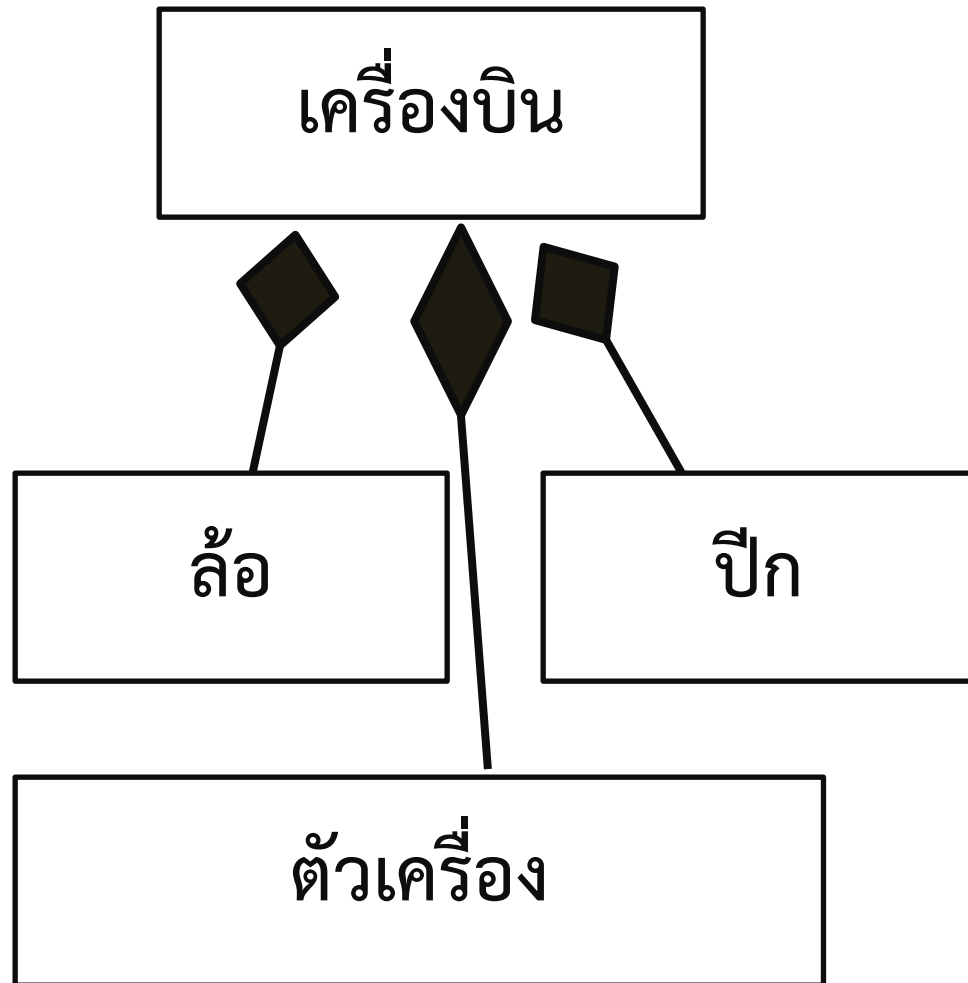
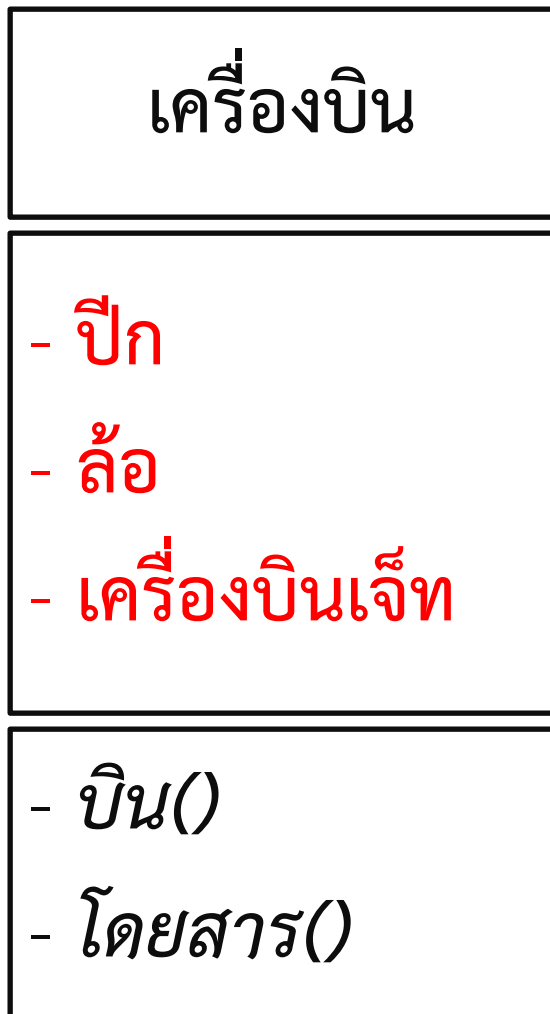
Aggregate

- Required Component
- Optional Component



Aggregation Abs.





Aggregation Abs.

ขั้นตอนการทำแอกกรีเกชัน

ขั้นตอนการทำแอกกรีเกชัน ในการทำแอกกรีเกชันไม่ได้มีการกำหนดขั้นตอนที่แน่นอนไว้อย่างชัดเจนแต่จากประสบการณ์ของผู้เขียนสามารถสรุปออกมาเป็นขั้นตอนคร่าว ๆ ได้ดังนี้

ขั้นตอนที่ 1. หาคำนาม หรือคลาส (ได้คลาส classification abstraction)

ขั้นตอนที่ 2. ระบุส่วนประกอบย่อยโดยวาดเป็นแผนภาพเพื่อแสดงส่วนประกอบย่อย

ขั้นตอนที่ 3. กำหนดส่วนประกอบที่จำเป็น และไม่จำเป็น

ขั้นตอนที่ 4. กำหนดคาร์ดินัลลิตี้ (การบ่งบอกจำนวนสมาชิก)

ขั้นตอนที่ 5. กำหนดรายละเอียดของคลาส

ตัวอย่างที่ 2.2.10

- **ตัวอย่างที่ 2.2.10** คอนโดมีเนียมโดยทั่วไปจะประกอบไปด้วยห้องพักหลาย ๆ ห้อง โดยแต่ละห้องนั้นจะมีส่วนประกอบหลักๆ ดังนี้ 1. ห้องนอนจำนวน 1 ห้อง 2. ห้องน้ำจำนวน 1 ห้อง ซึ่งจะต้องมีทุกห้อง 3. เครื่องปรับอากาศซึ่งอาจจะมีหรือไม่มีก็ได้ หรือบางห้องอาจจะมีได้มากกว่า 1 เครื่อง โดยภายในห้องนอนจะมีเตียงนอนอย่างน้อย 1 หลังแต่บางห้องอาจจะมีมากกว่านี้ก็ได้ และมีโทรทัศน์จำนวน 1 เครื่อง

ขั้นตอนที่ 1. หาคำนาม

ขั้นตอนที่ 1 หาคำนาม หรือคลาส โดยใช้คลาสสิฟิเคชันแอบแทรกชัน

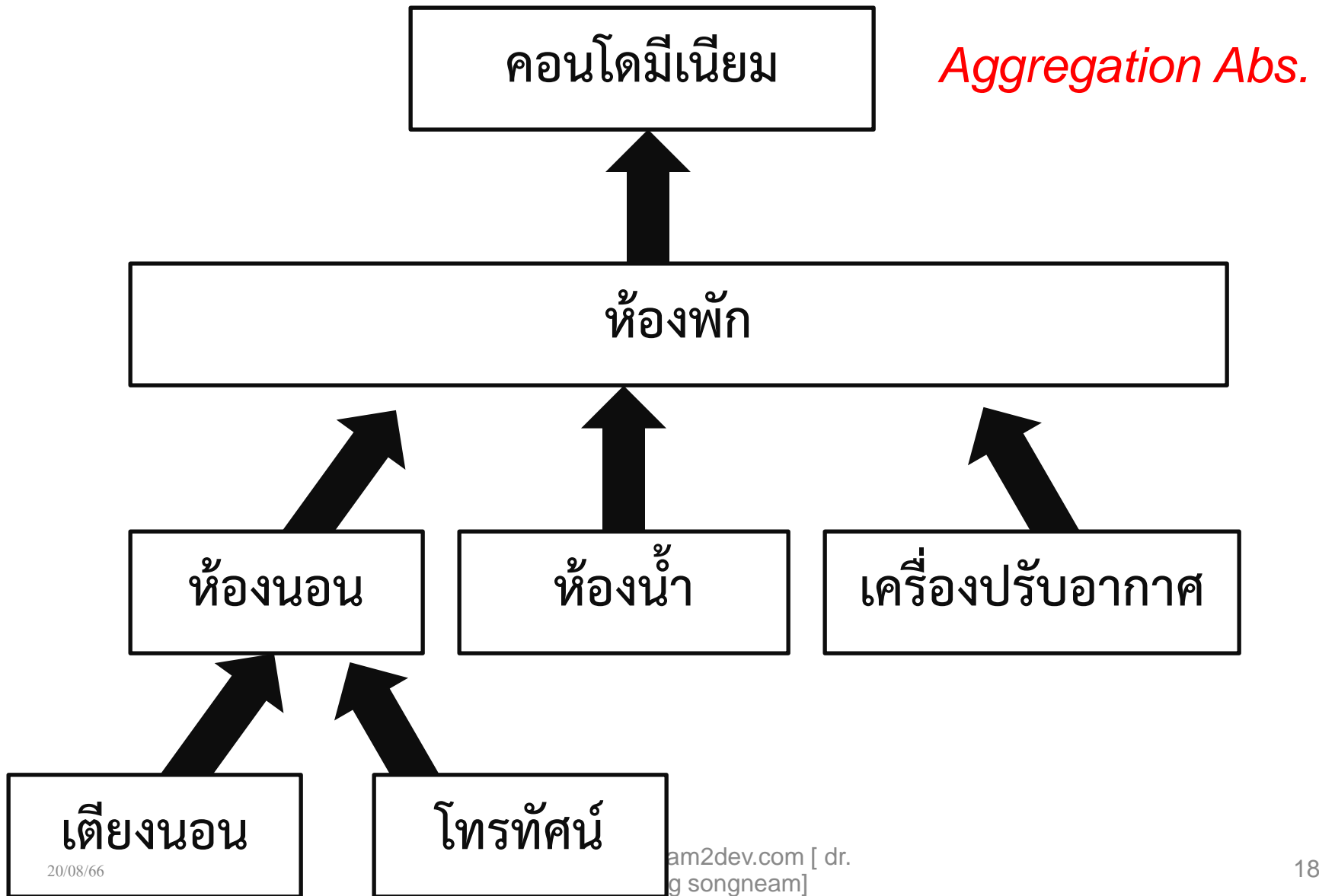
จากข้อความข้างต้น จะระบุคลาสต่างๆ ได้ดังนี้

- 1) คลาสคอนโดมีเนียม
- 2) คลาสห้อง
- 3) คลาสห้องนอน
- 4) คลาสห้องน้ำ
- 5) คลาสเครื่องปรับอากาศ
- 6) คลาสเตียงนอน
- 7) คลาสโทรทัศน์

Fundamental classes
กลุ่มคลาสที่ได้จาก classification abs.

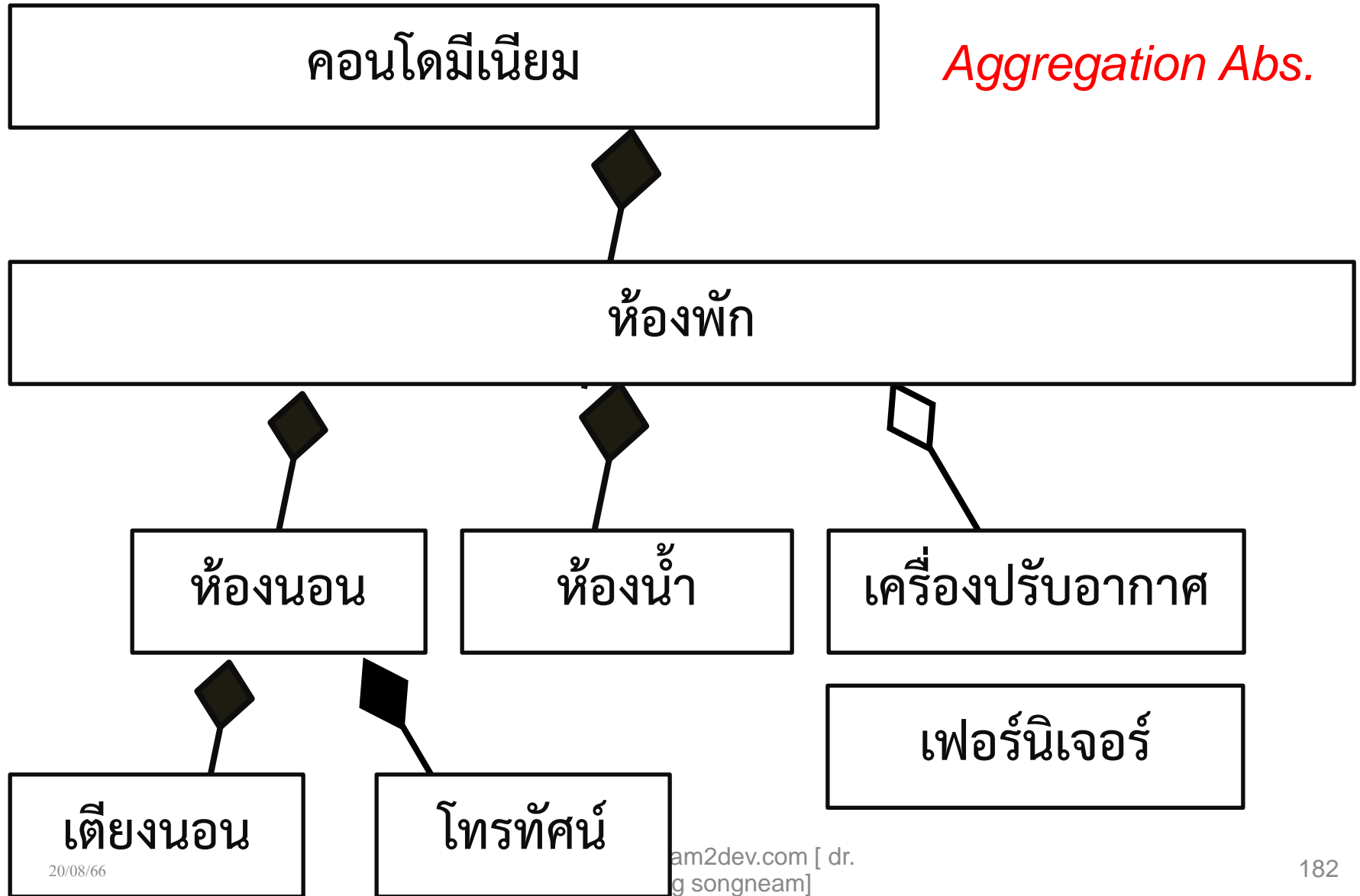
ขั้นตอนที่ 2.

ระบุส่วนประกอบย่อยโดยวาดเป็นแผนภาพแสดงส่วนประกอบ ดังแสดงในภาพ



ขั้นตอนที่ 3. กำหนดส่วนประกอบที่จำเป็น(Required)

และไม่จำเป็น (Optional) ดังแสดงในภาพ

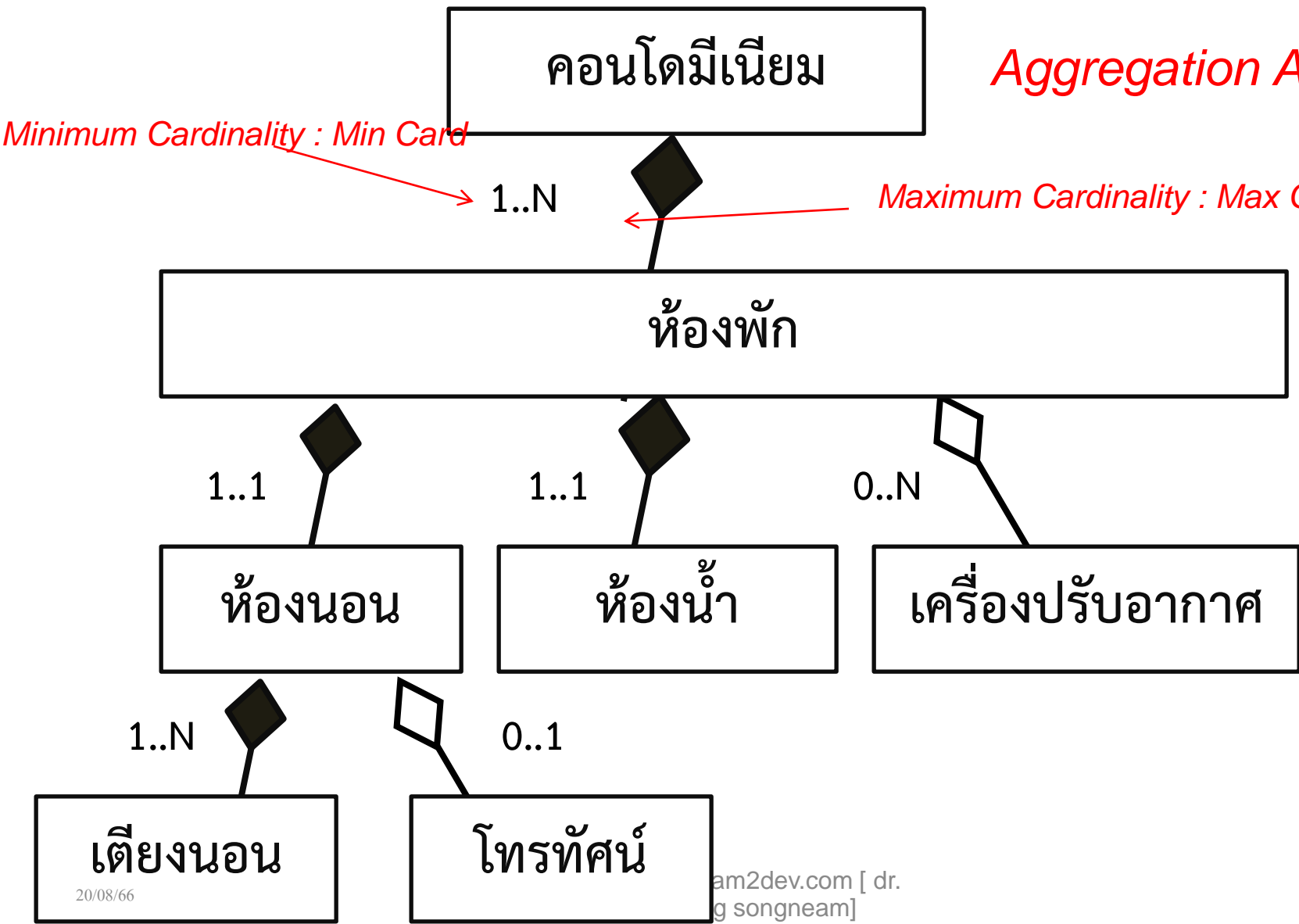


ขั้นตอนที่ 4. กำหนดคาร์ดินัลลิตี ดังแสดงในภาพ

Aggregation Abs.

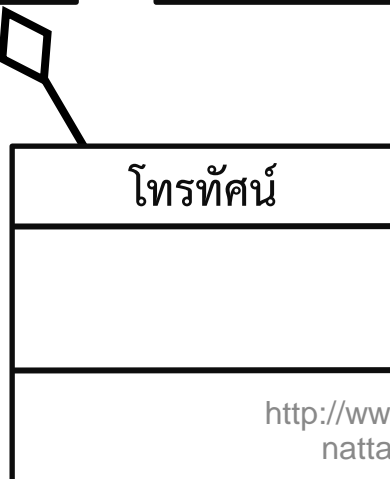
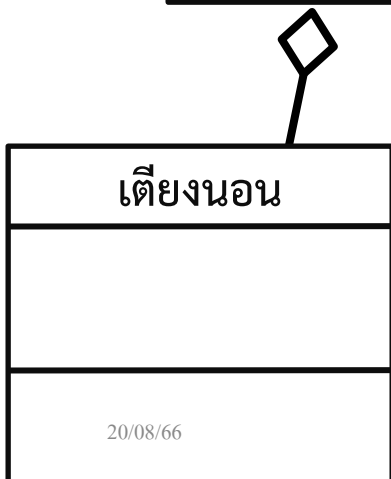
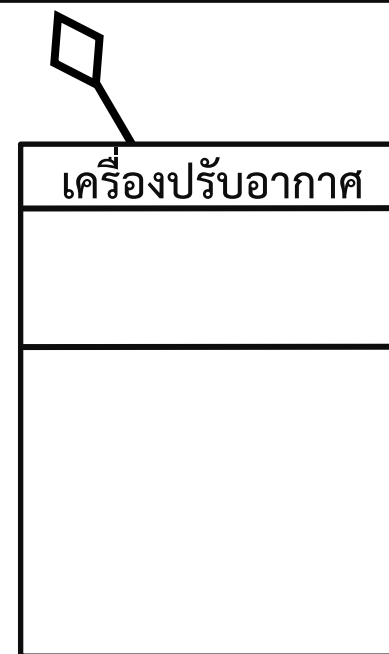
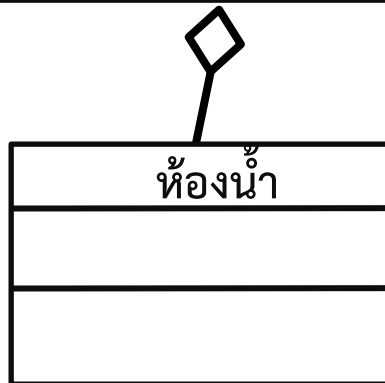
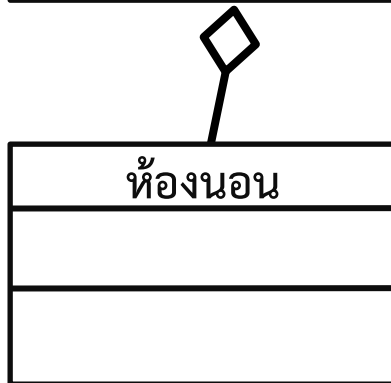
Minimum Cardinality : Min Card

Maximum Cardinality : Max Card





Aggregation Abs.





Aggregation Abs.



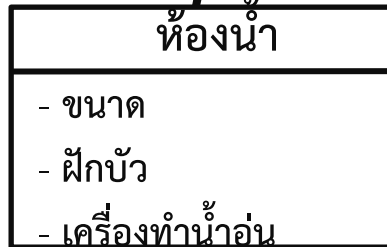
1..N



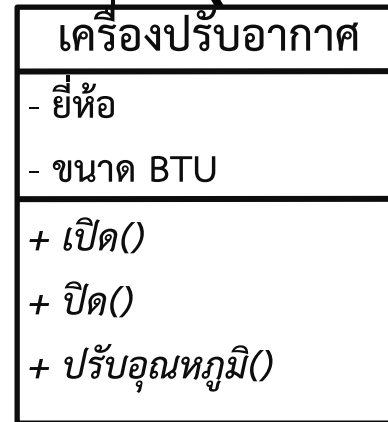
1..1



1..1



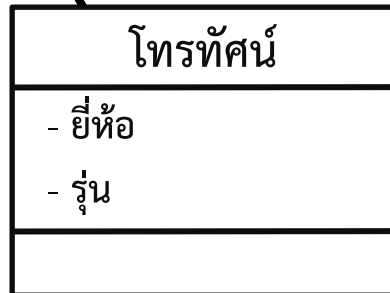
0..N



1..1



0..1



ขั้นตอนที่ 5. กำหนดรายละเอียดของคลาส

ขั้นตอนที่ 5. กำหนดรายละเอียดของคลาส

คอนโดมีเนียม
- ชื่อคอนโดมีเนียม - ที่ตั้ง
+ เช่า() + ขาย()

คอนโดมีเนียม
- ชื่อคอนโดมีเนียม - ที่ตั้ง
+ เช่า() + ขาย() + เชื้อคราคา() : floating-Point

ห้องพัก
- หมายเลขห้อง - เจ้าของ - ห้องนอน - ห้องน้ำ - เครื่องปรับอากาศ
+ เช่า() + ขาย() + เชื้อคราคา() : floating-Point + โอน() + เข้าพัก()

Aggregation Abs.



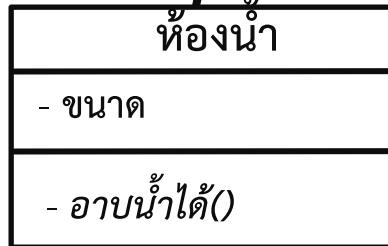
1..N



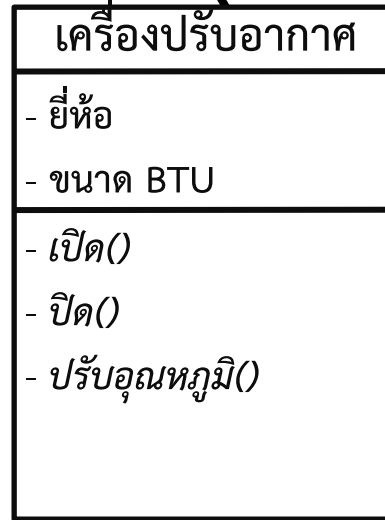
1..1



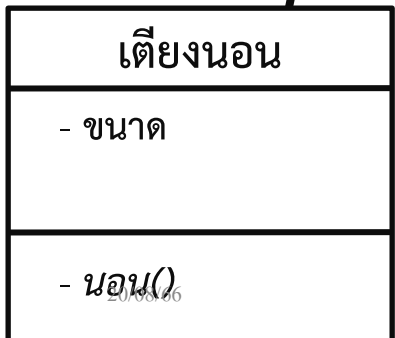
1..1



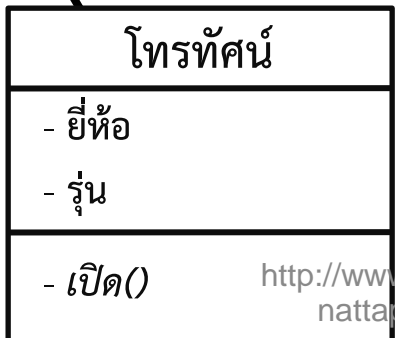
0..N



1..N

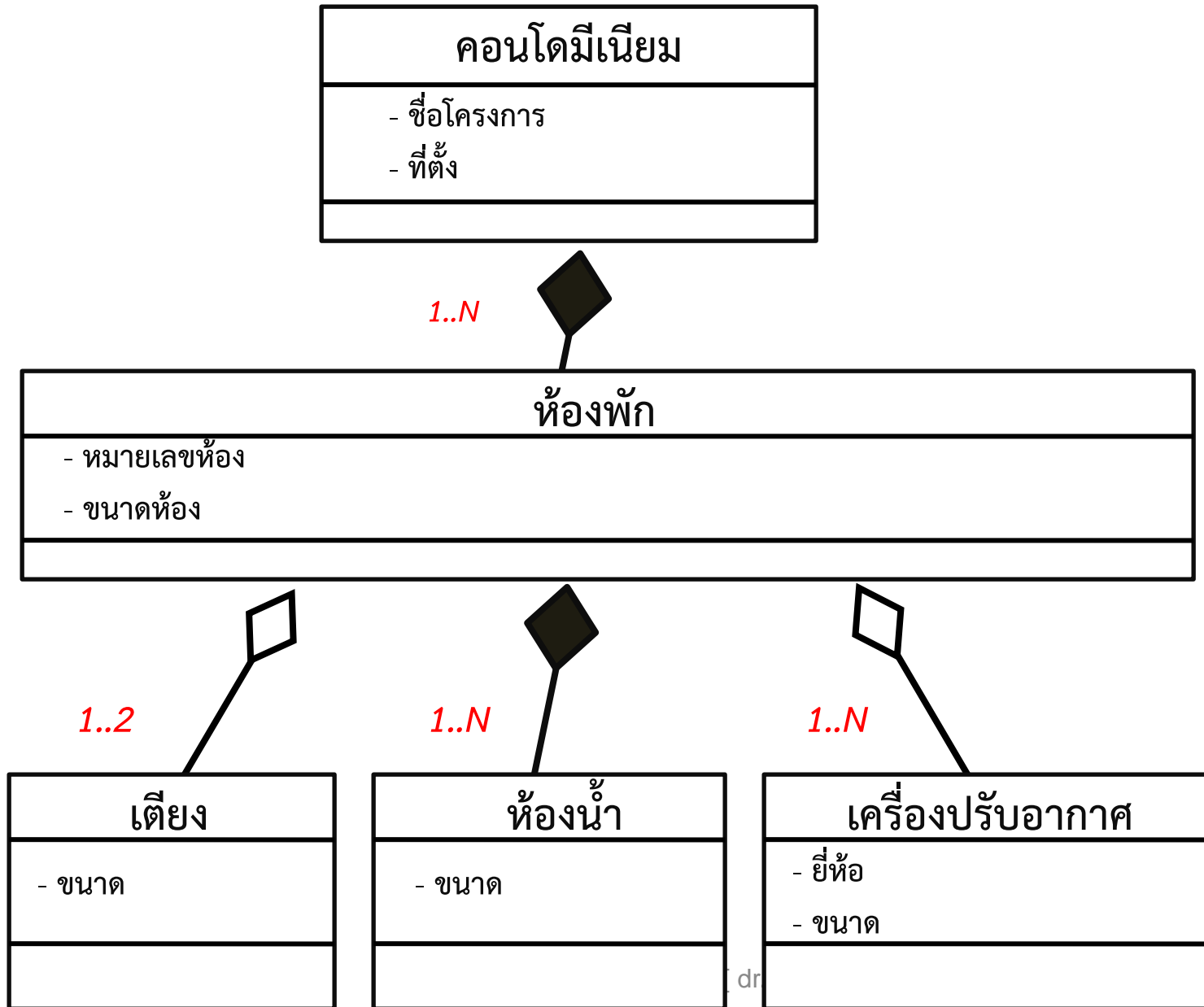


0..1

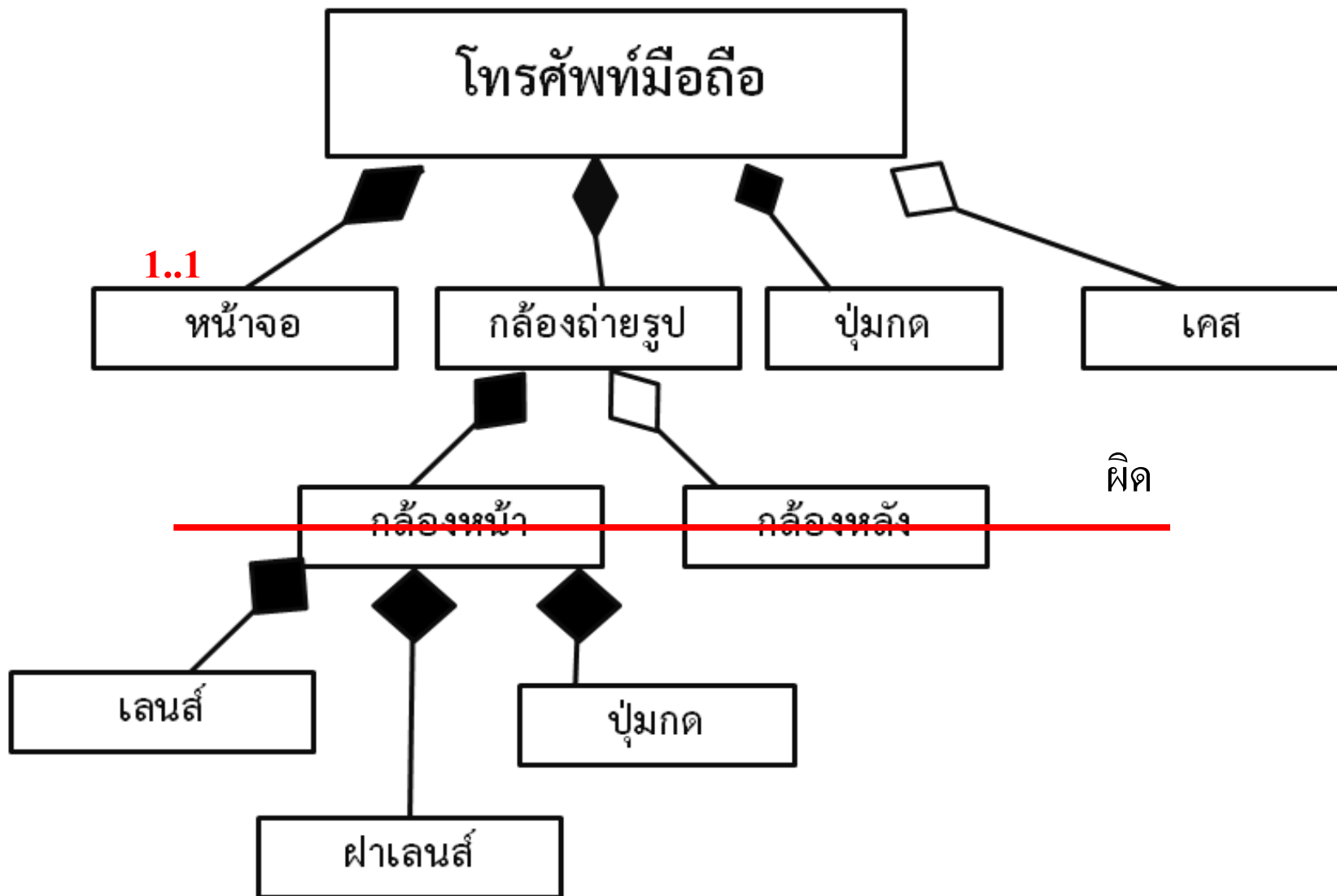


ขั้นตอนที่ 5. กำหนดรายละเอียดของคลาส

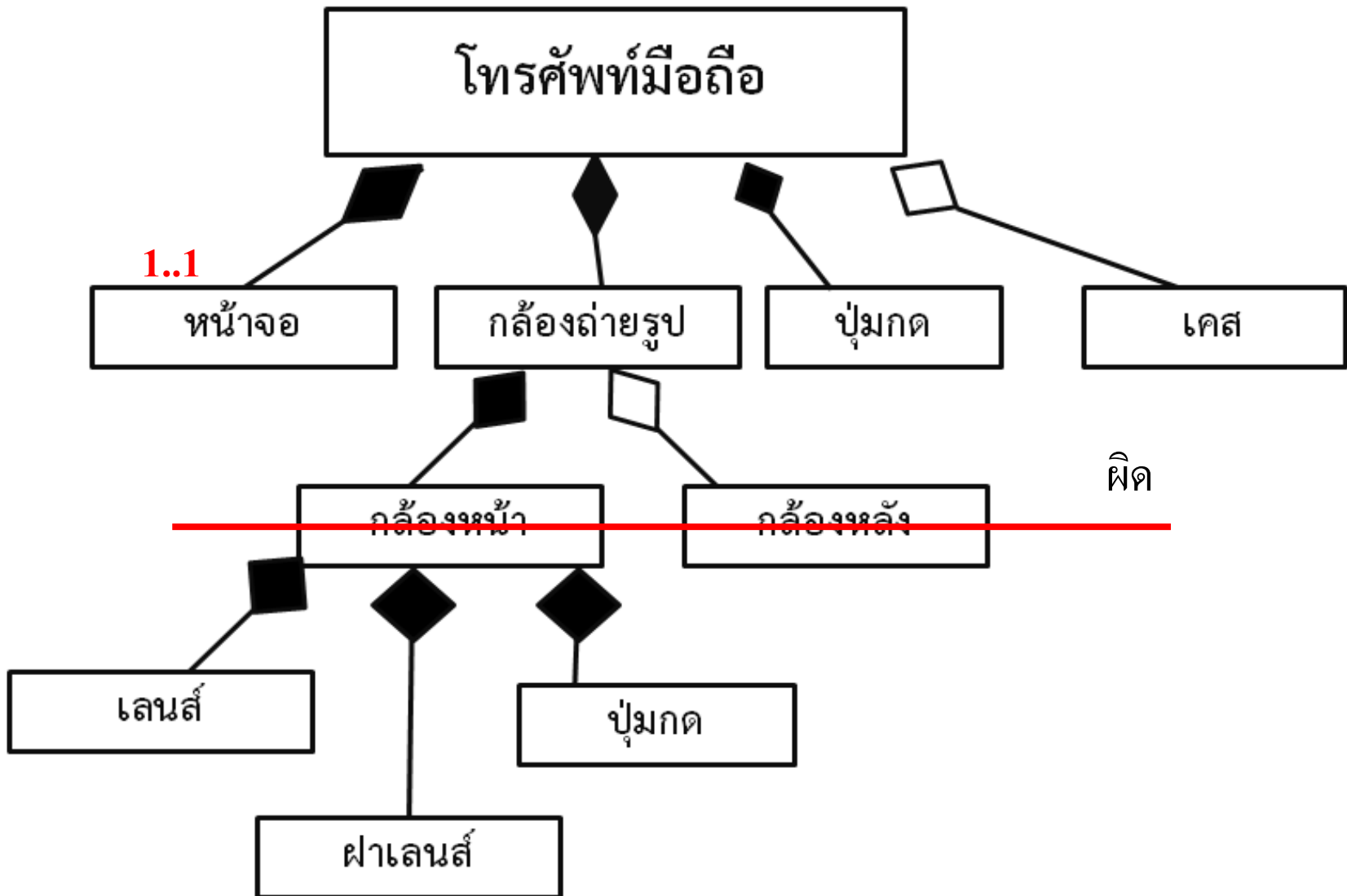
Aggregation Abs.



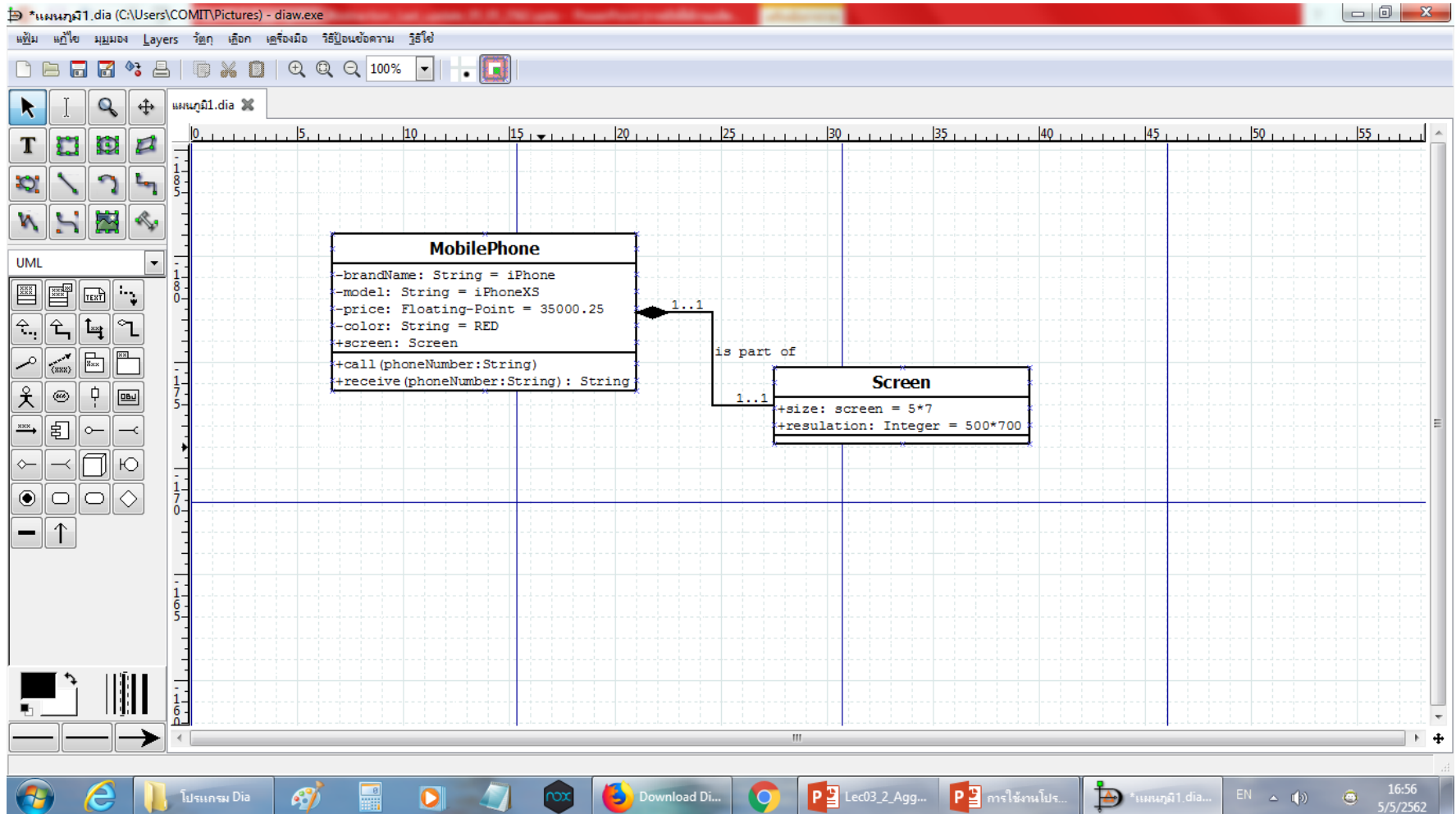
ให้เขียน attribute, function, cardinality ให้ครบ

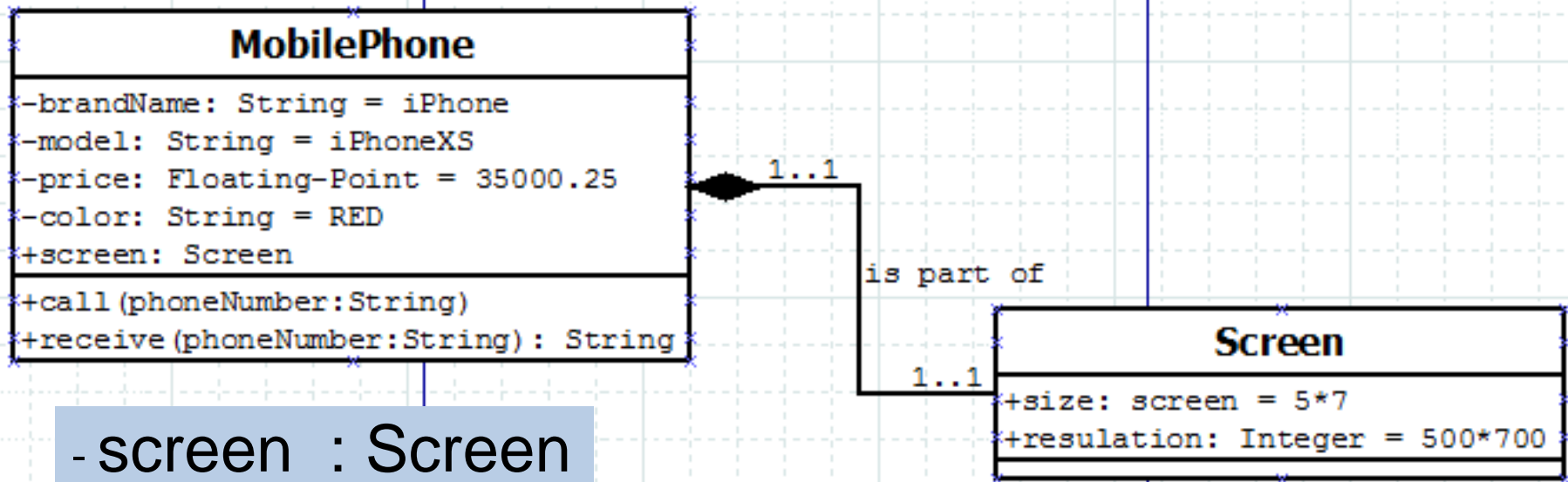
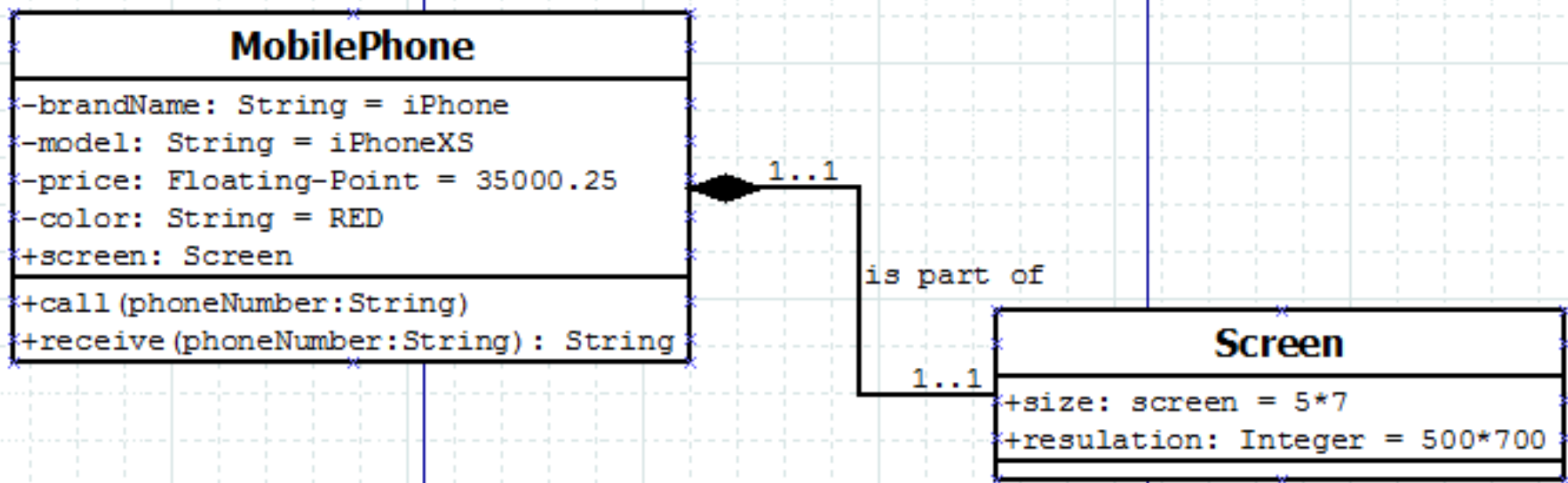


ให้เขียน attribute, function, cardinality ให้ครบ



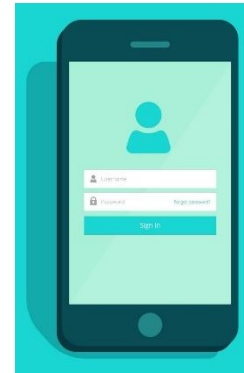
Dia





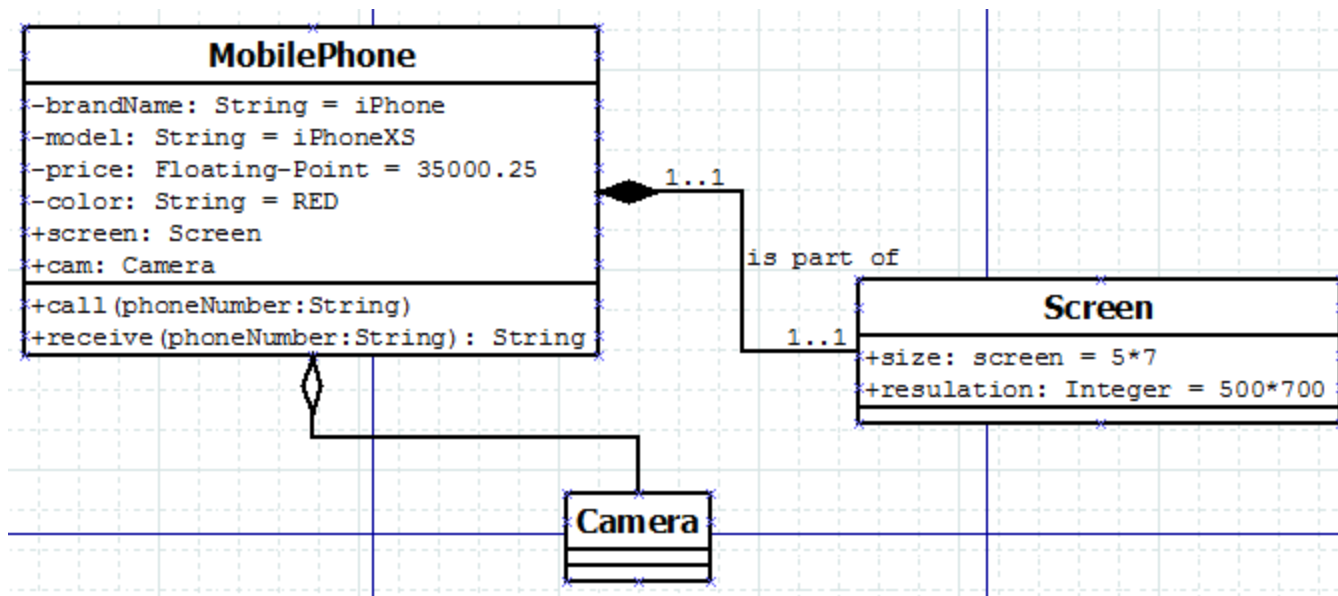
คน สามารถมองเห็น จอภาพของโทรศัพท์ จากมุมมอง outside view

พิมพ์ภาพทางแป้นพิมพ์ keyboard

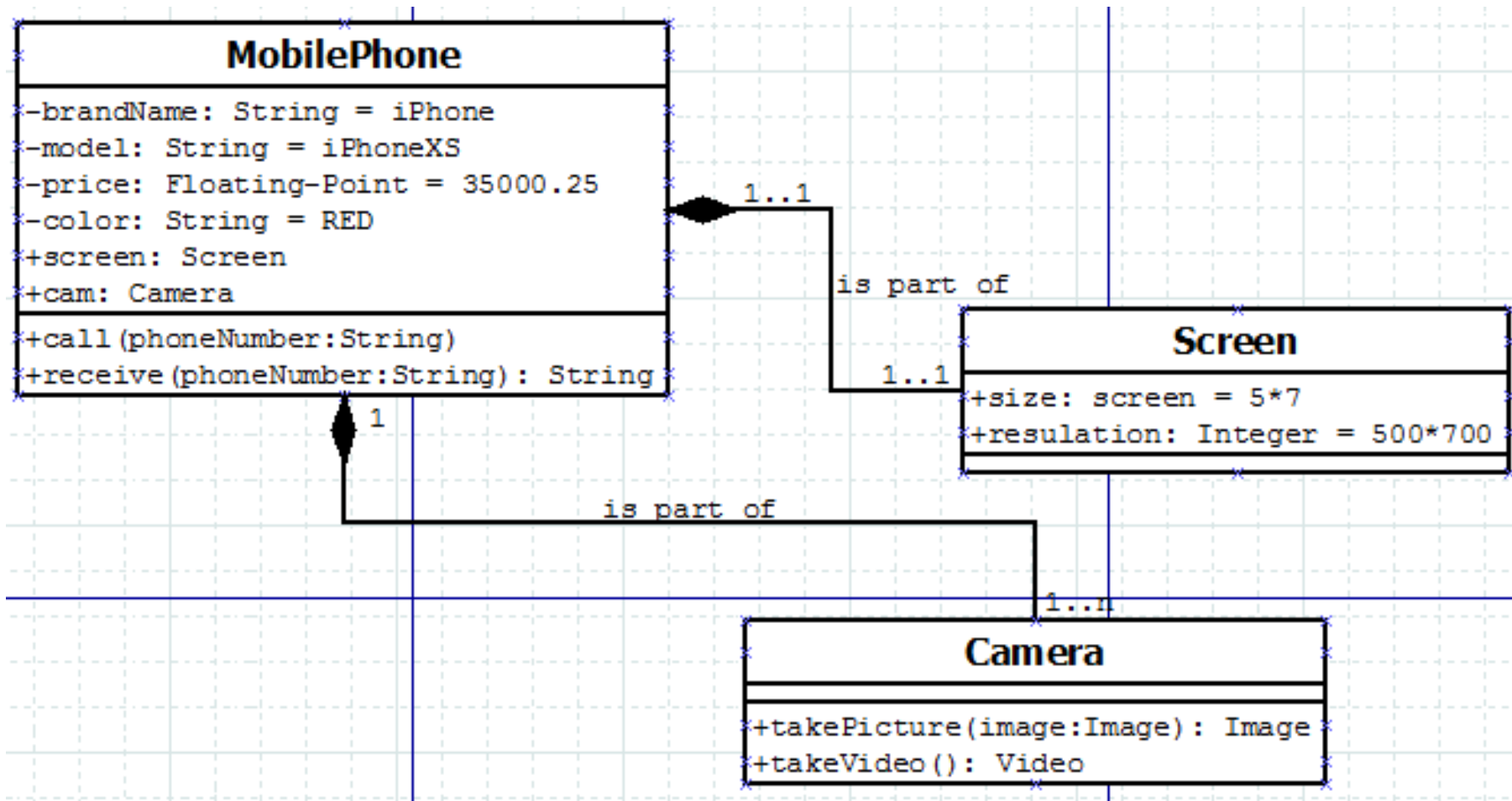


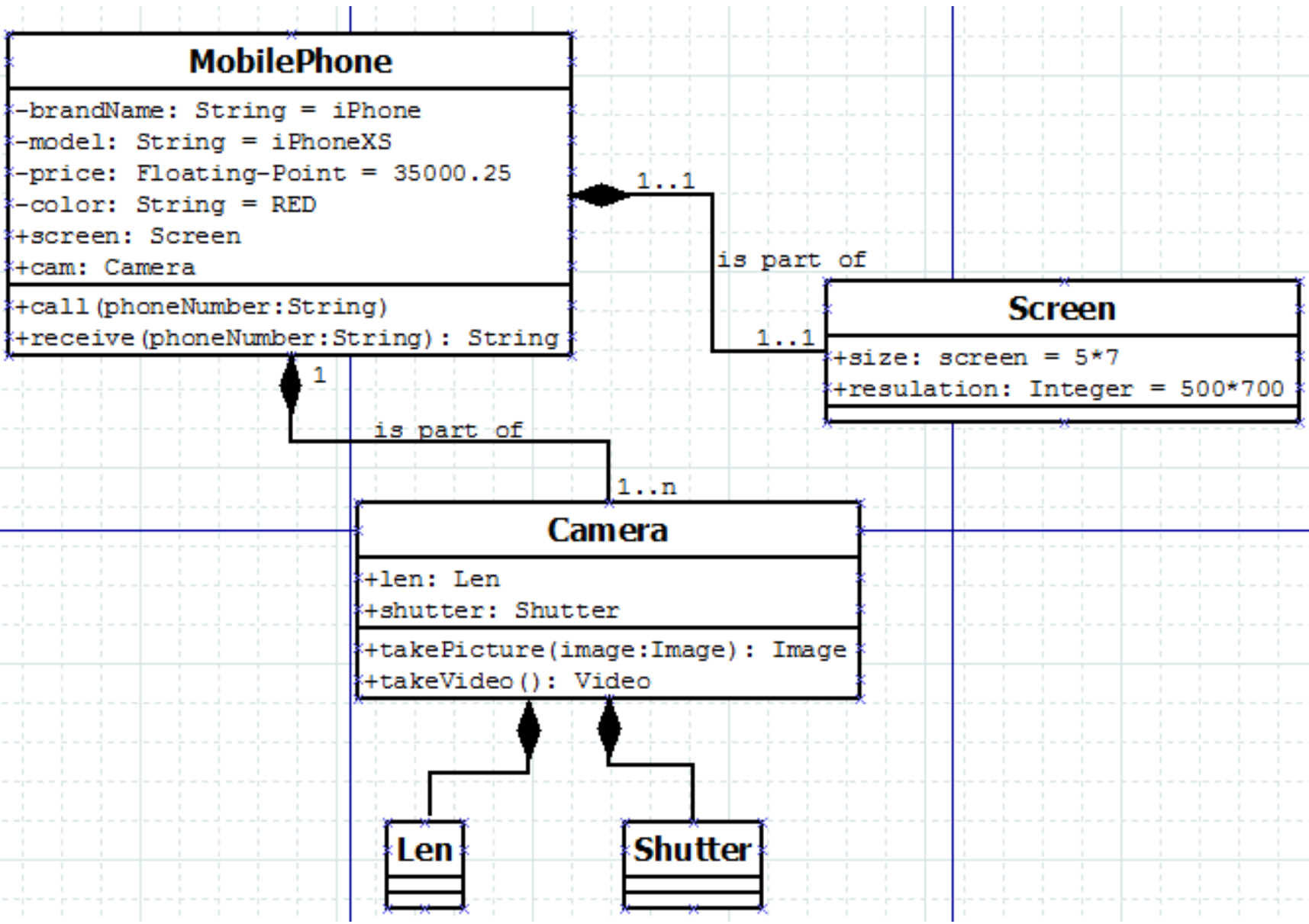
เลื่อน / slide จอภาพได้





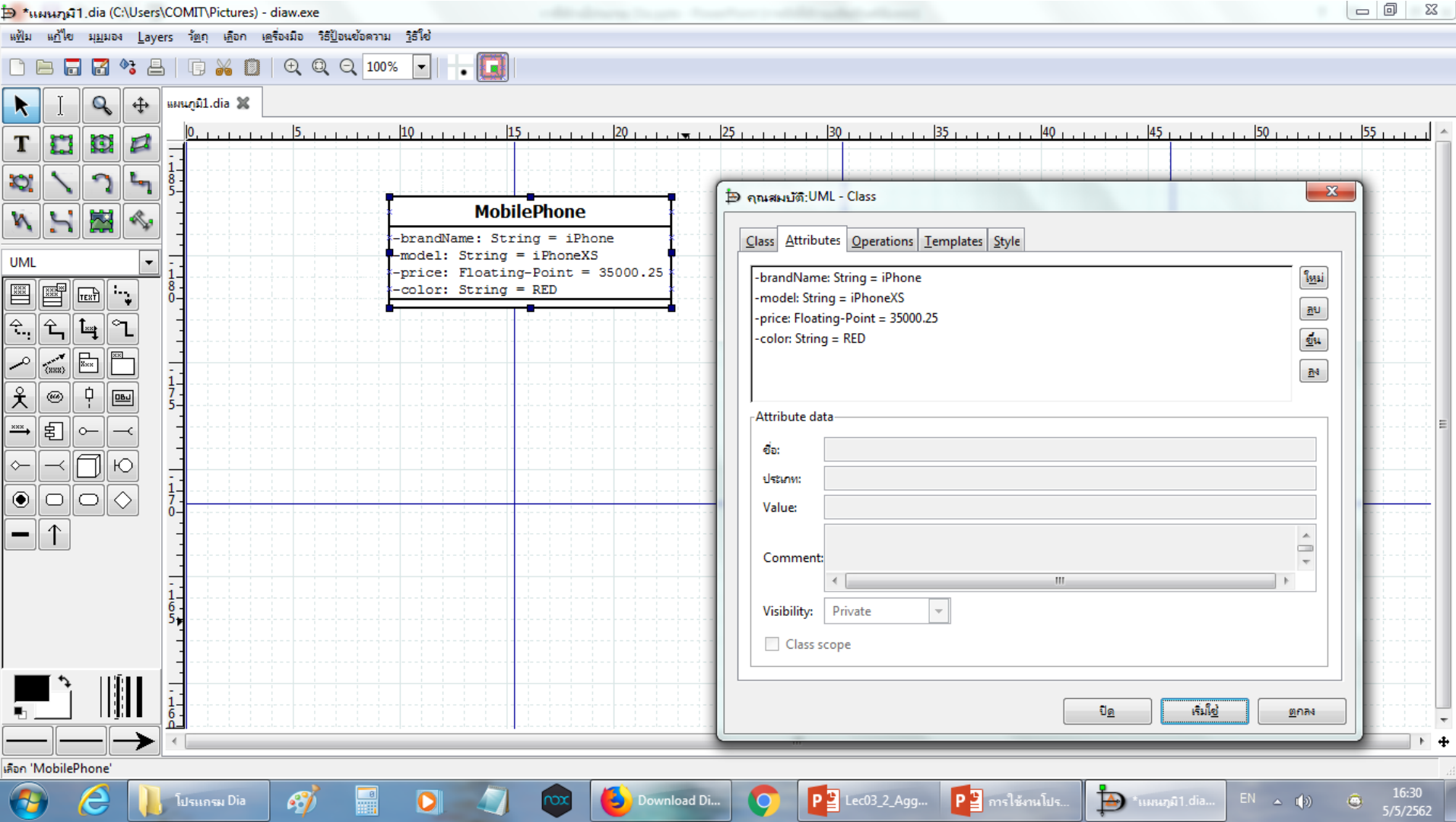
SmartPhone/MobilePhone





โทรศัพท์มือถือ

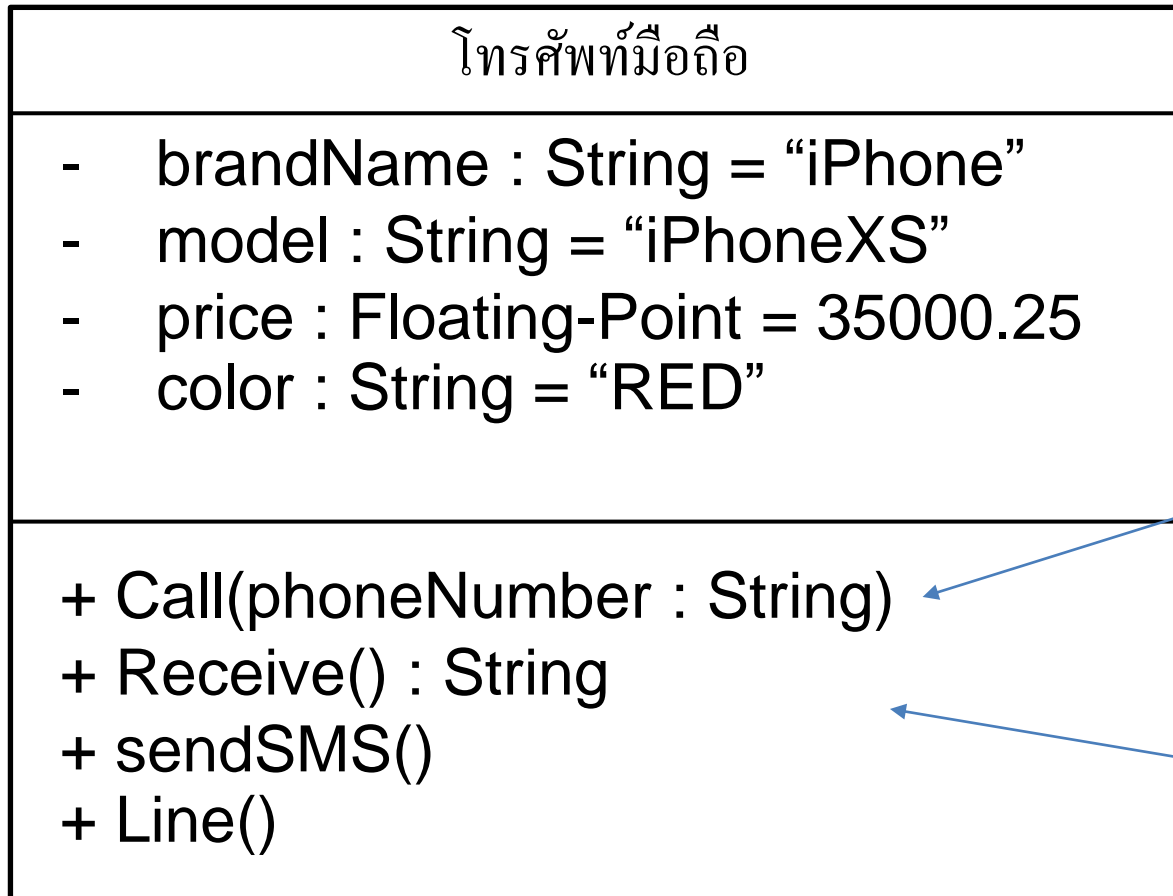
- brandName : String = "iPhone"
- model : String = "iPhoneXS"
- price : Floating-Point = 35000.25
- color : String = "RED"



Function/Behavior/Method/Sub/Procedure/Operation

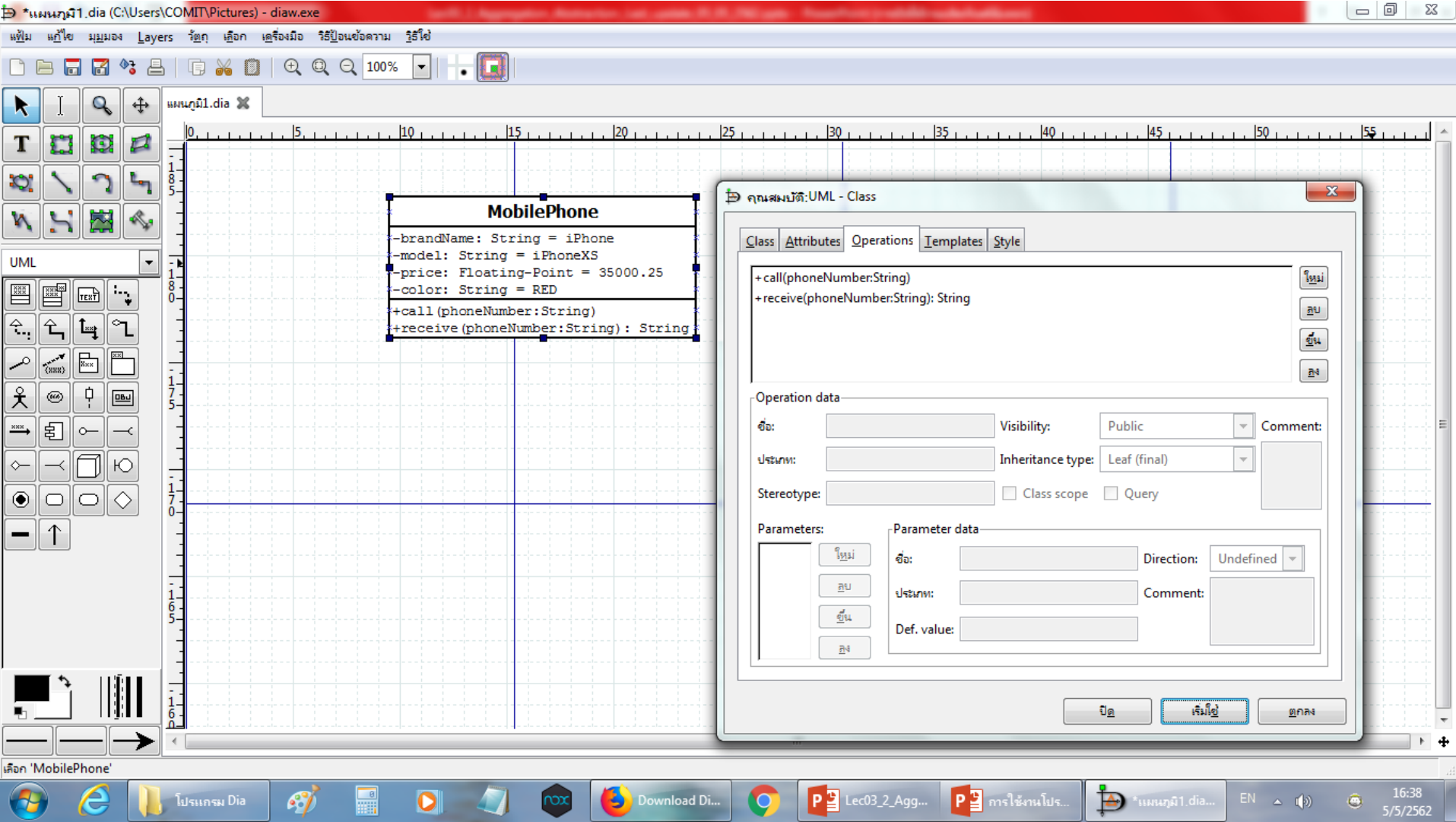
โทรศัพท์มือถือ
<ul style="list-style-type: none">- brandName : String = "iPhone"- model : String = "iPhoneXS"- price : Floating-Point = 35000.25- color : String = "RED"
<ul style="list-style-type: none">+ Call()+ Receive()+ sendSMS()+ Line()

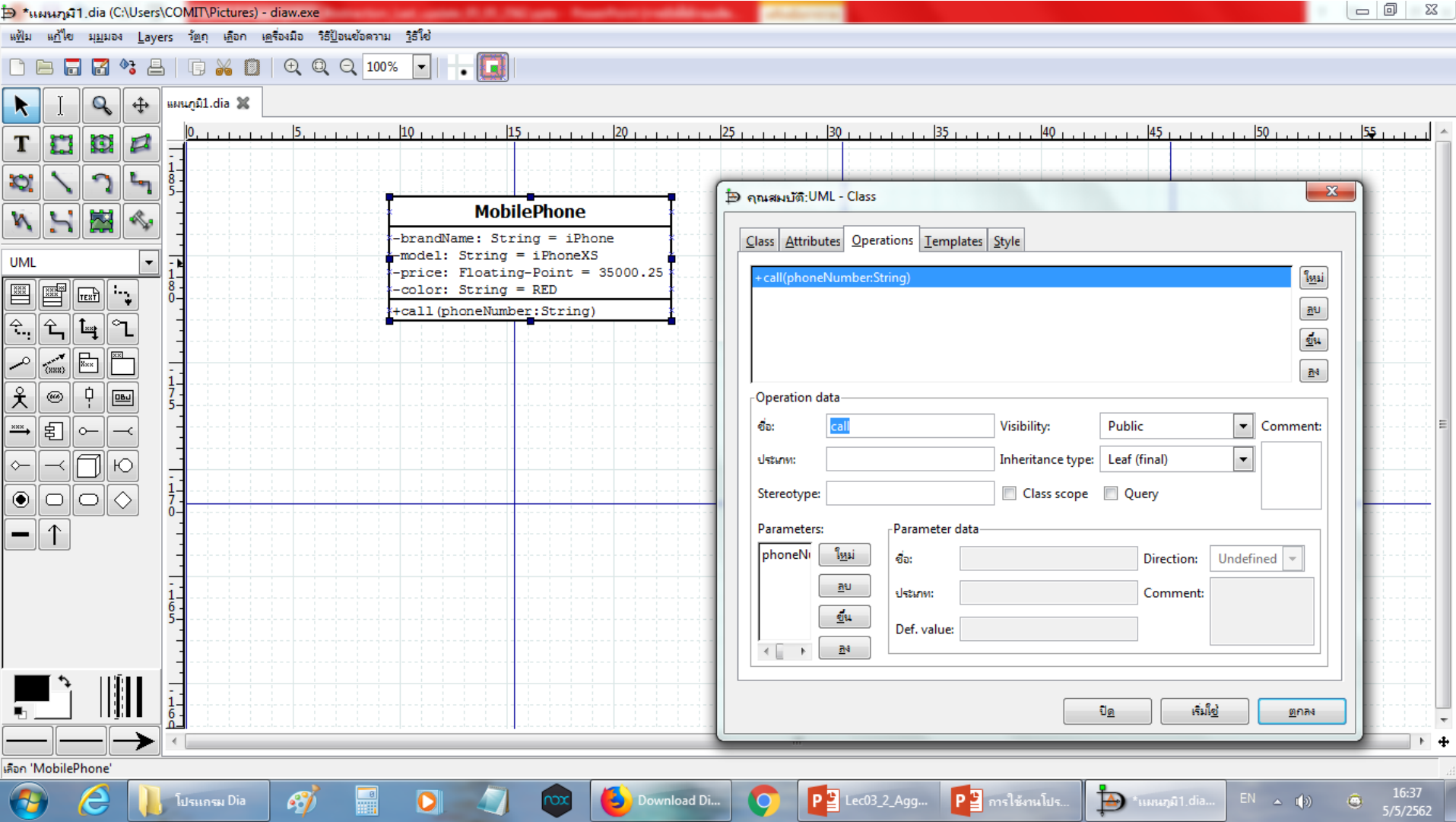
Function/Behavior/Method/Sub/Procedure/Operation

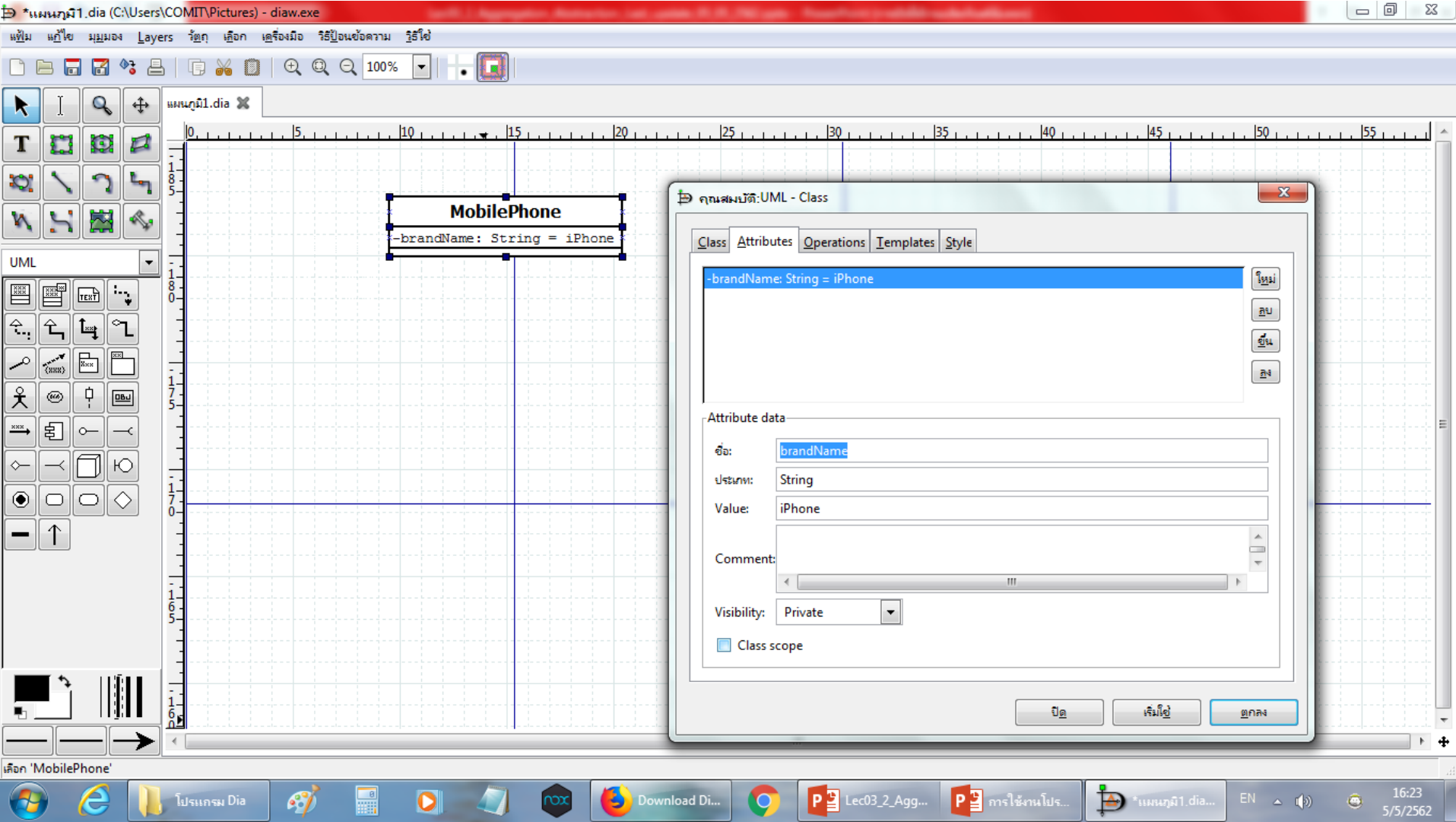


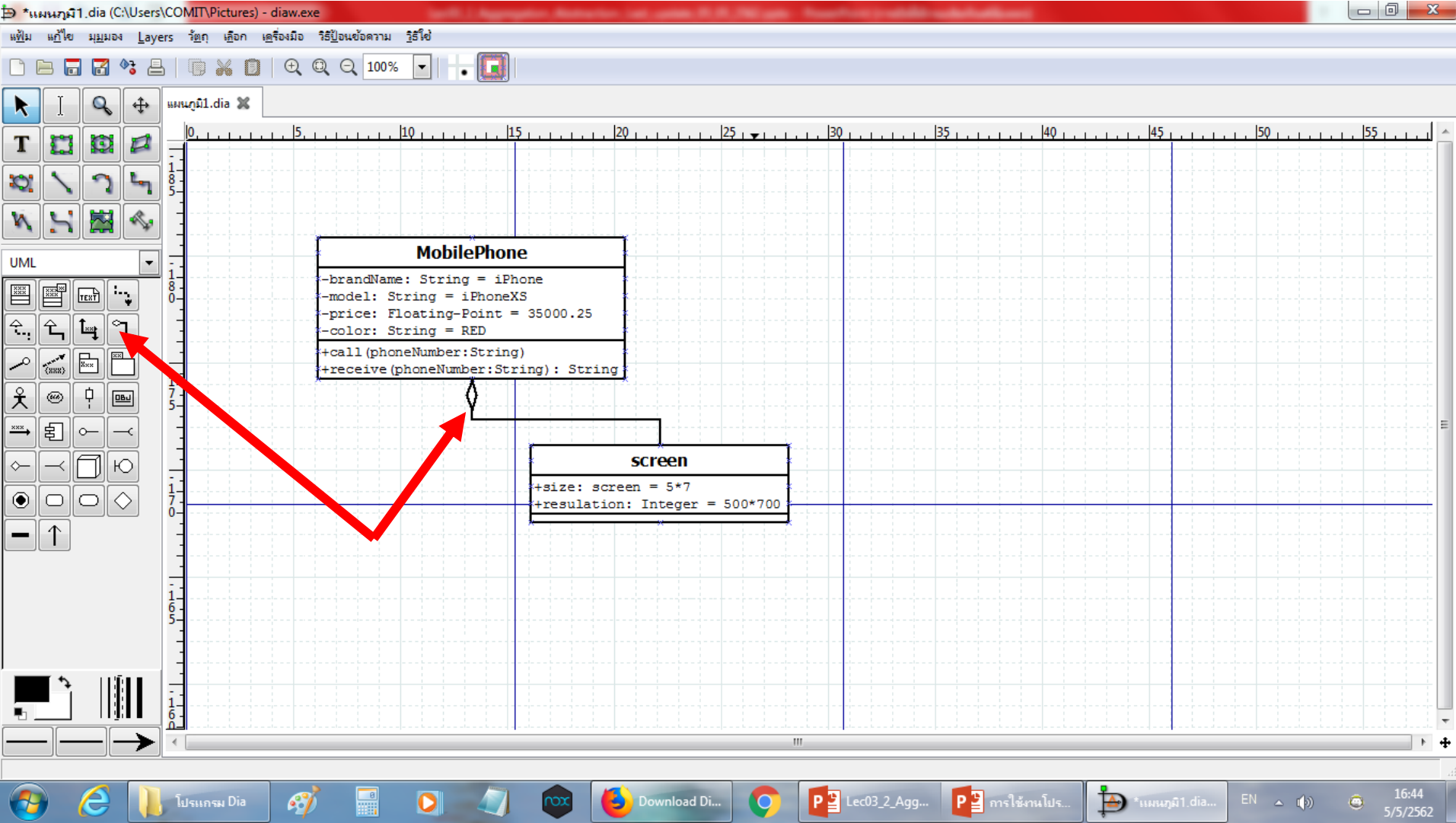
void

Return









UML Diagram Editor (diaw.exe) showing a class diagram with two classes: **MobilePhone** and **screen**.

```

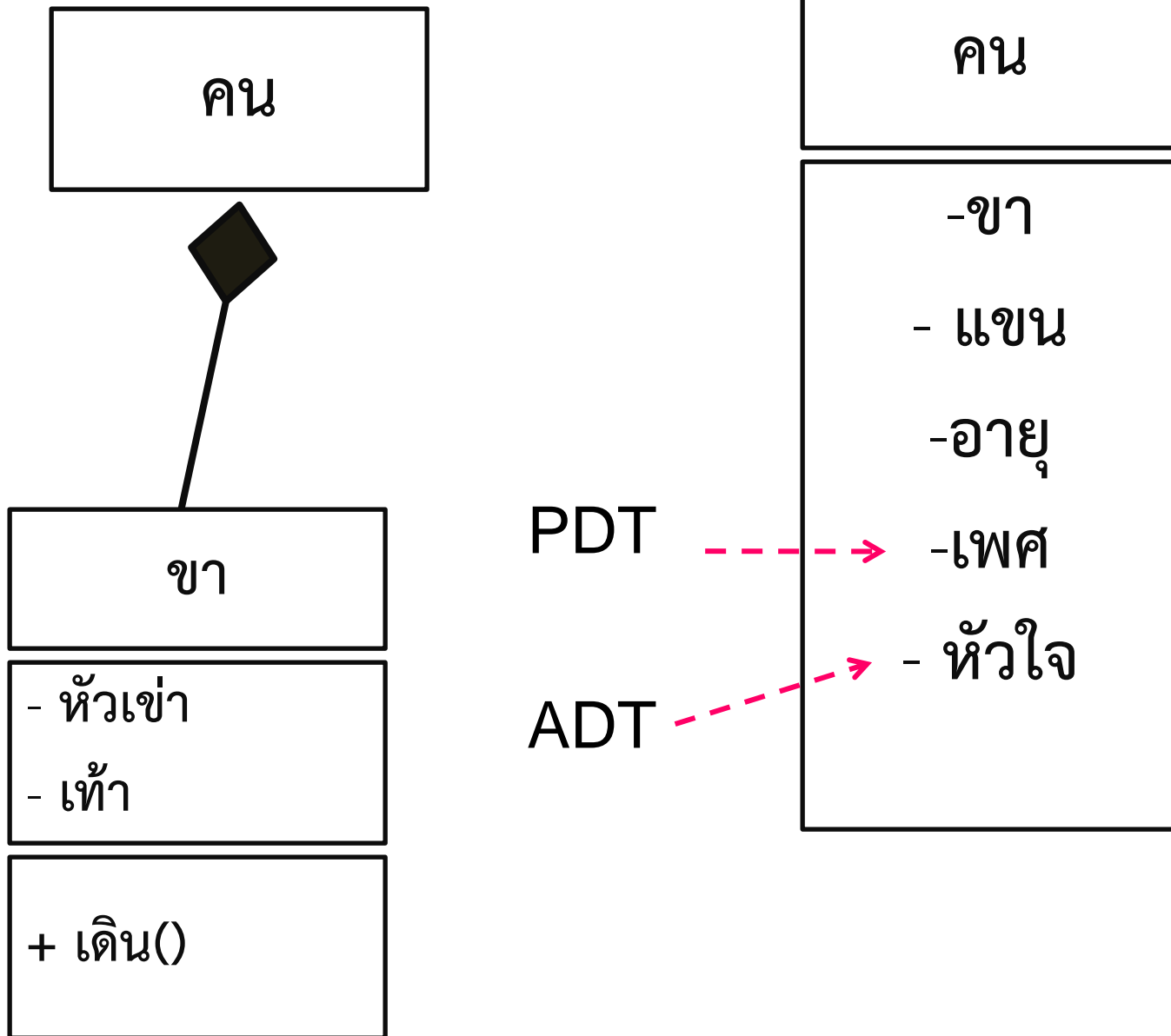
classDiagram
    class MobilePhone {
        -brandName: String = iPhone
        -model: String = iPhoneXS
        -price: Floating-Point = 35000.25
        -color: String = RED
        +call(phoneNumber: String)
        +receive(phoneNumber: String): String
    }
    class screen {
        +size: screen = 5*7
        +resulation: Integer = 500*700
    }
    MobilePhone "1..1" -- "1..1" screen : is part of
  
```

The **screen** class has attributes: `+size: screen = 5*7` and `+resulation: Integer = 500*700`. The **MobilePhone** class has attributes: `-brandName: String = iPhone`, `-model: String = iPhoneXS`, `-price: Floating-Point = 35000.25`, and `-color: String = RED`. It has methods: `+call(phoneNumber: String)` and `+receive(phoneNumber: String): String`.

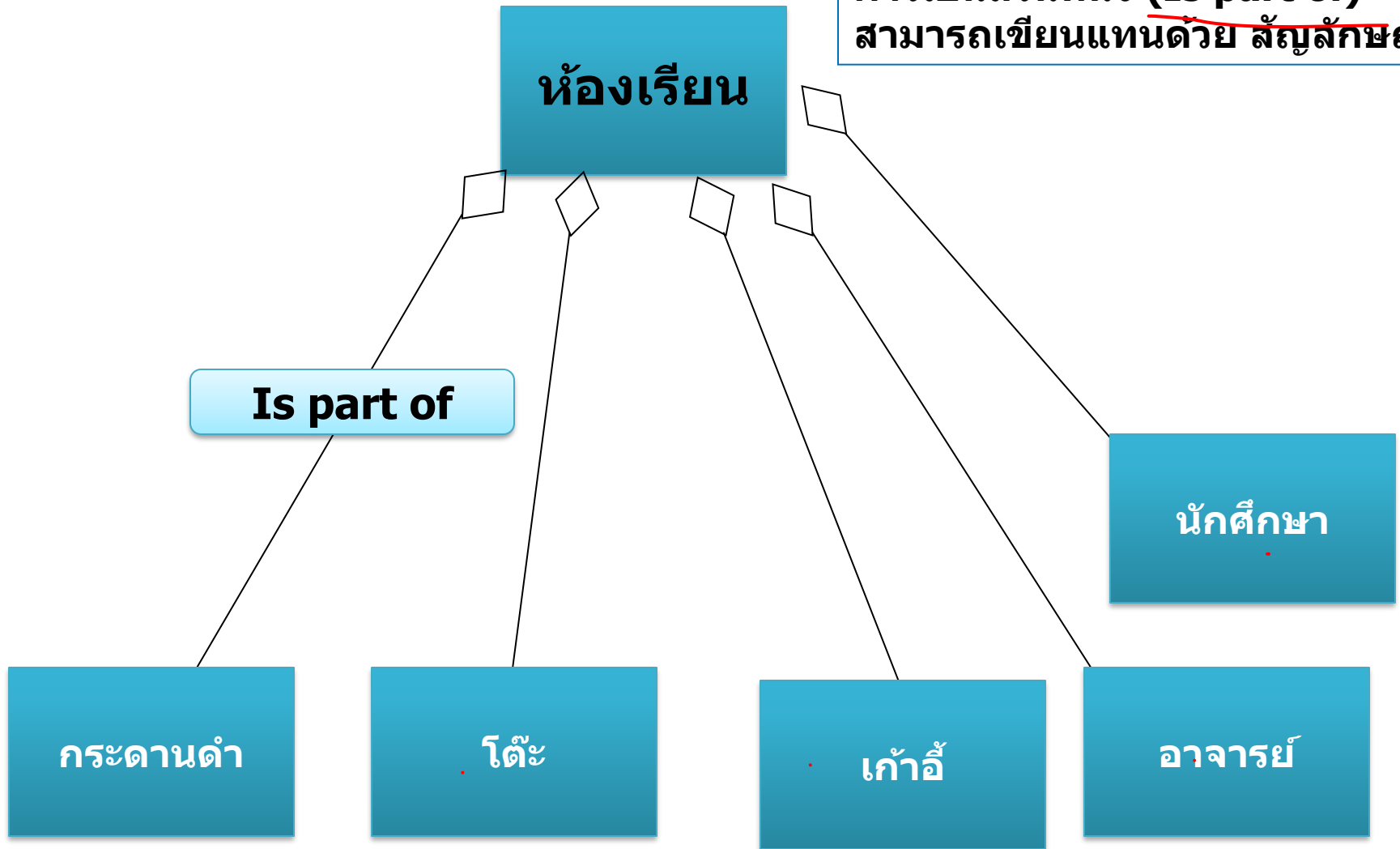
An association relationship labeled "is part of" exists between **MobilePhone** and **screen**, with multiplicity `1..1` on both ends.

The "คุณสมบัติ:UML - Association" dialog box is open, showing the following configuration:

- ชื่อ: is part of
- Direction: From A to B
- Show direction: ไม่ใช้
- ประเภท: Composition
- Role: Side A, Side B
- Multiplicity: 1..1, 1..1
- Visibility: Implementation, Implementation
- Show arrow: ไม่ใช้, ไม่ใช้
- Autoroute: ใช่
- สีอักษร: #000000
- สีเส้น: #000000
- Buttons: ปิด, เพิ่มชื่อ, ลบคง

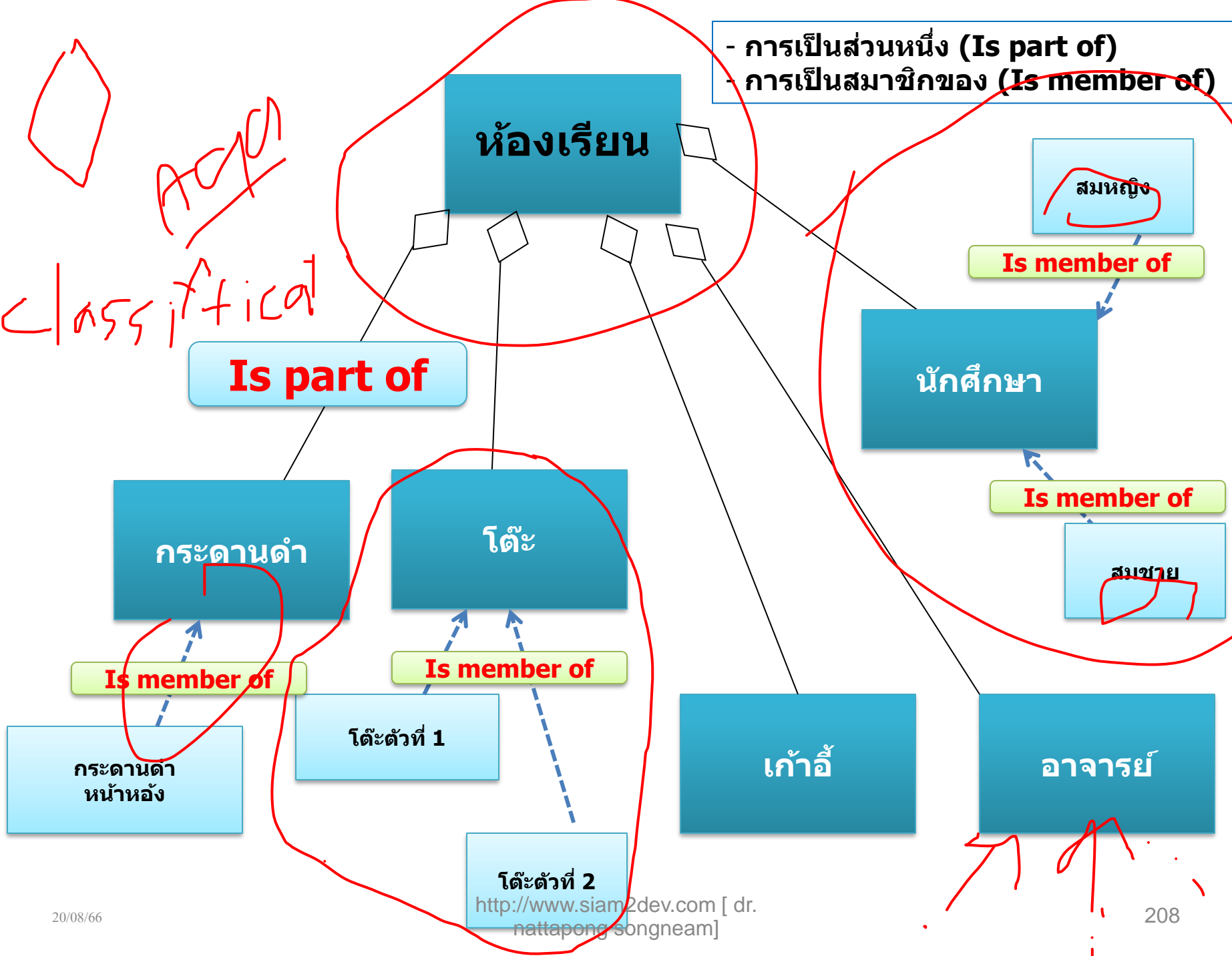


การเป็นส่วนหนึ่ง (Is part of)
สามารถเขียนแทนด้วย ~~สัญลักษณ์~~



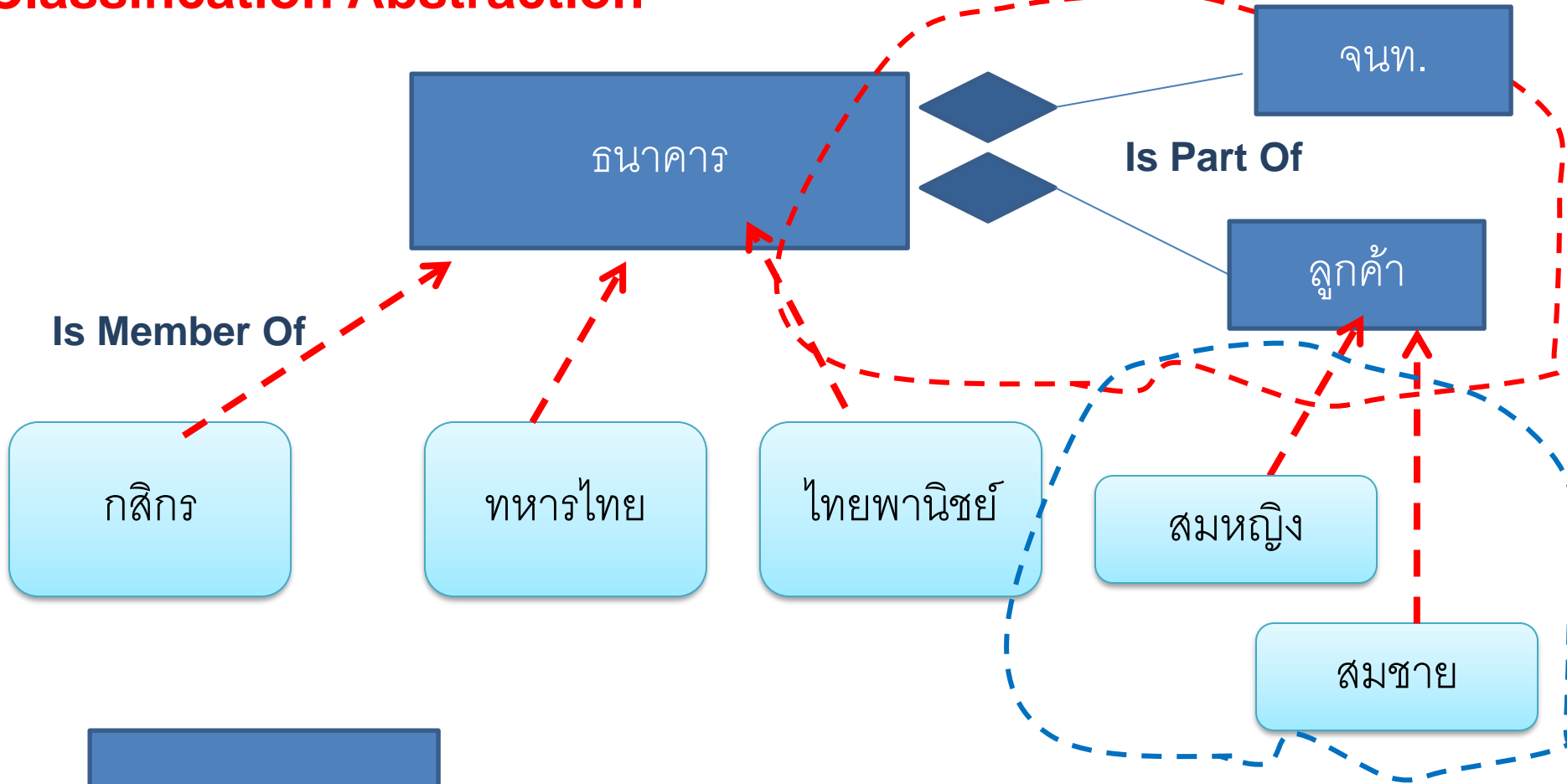
- การเป็นส่วนหนึ่ง (Is part of)
- การเป็นสมาชิกของ (Is member of)

Handwritten:
 classifical



Aggregation Abstraction

Classification Abstraction



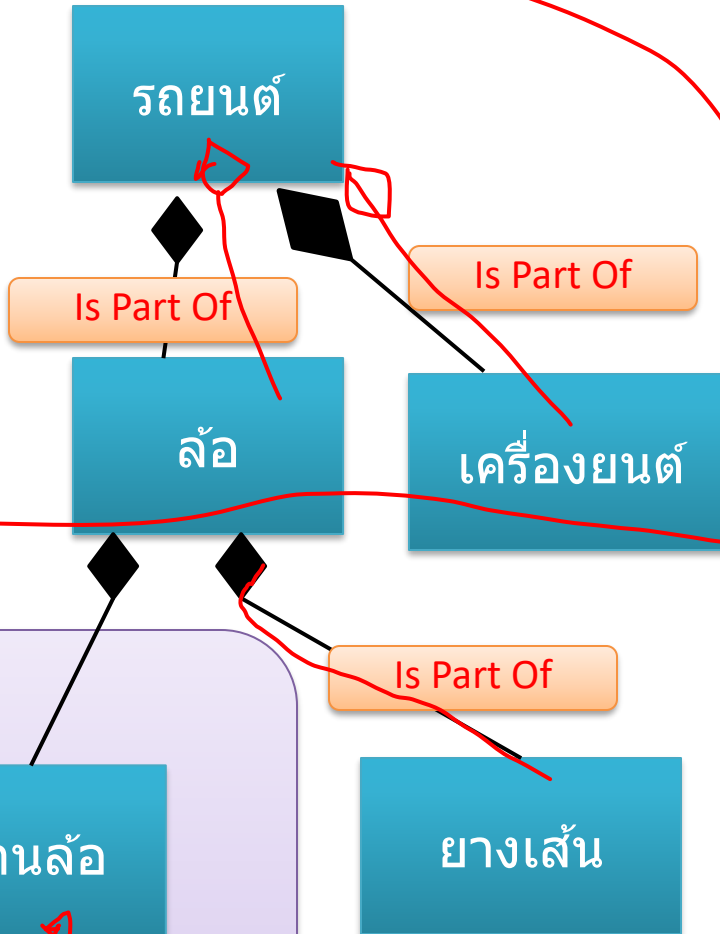
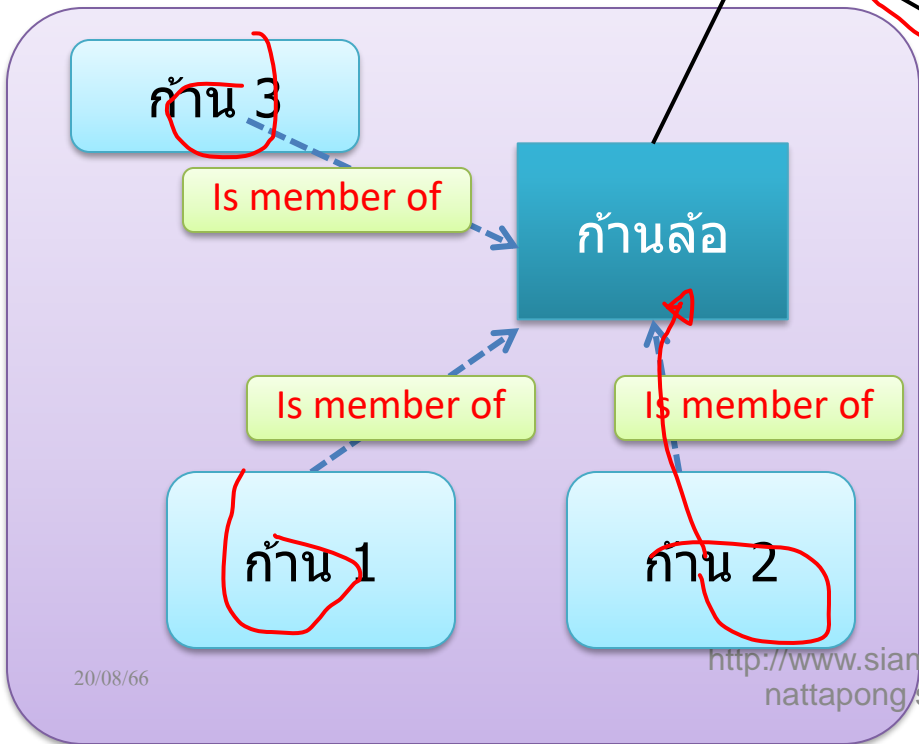
คลาส(class)

วัตถุ(Object)

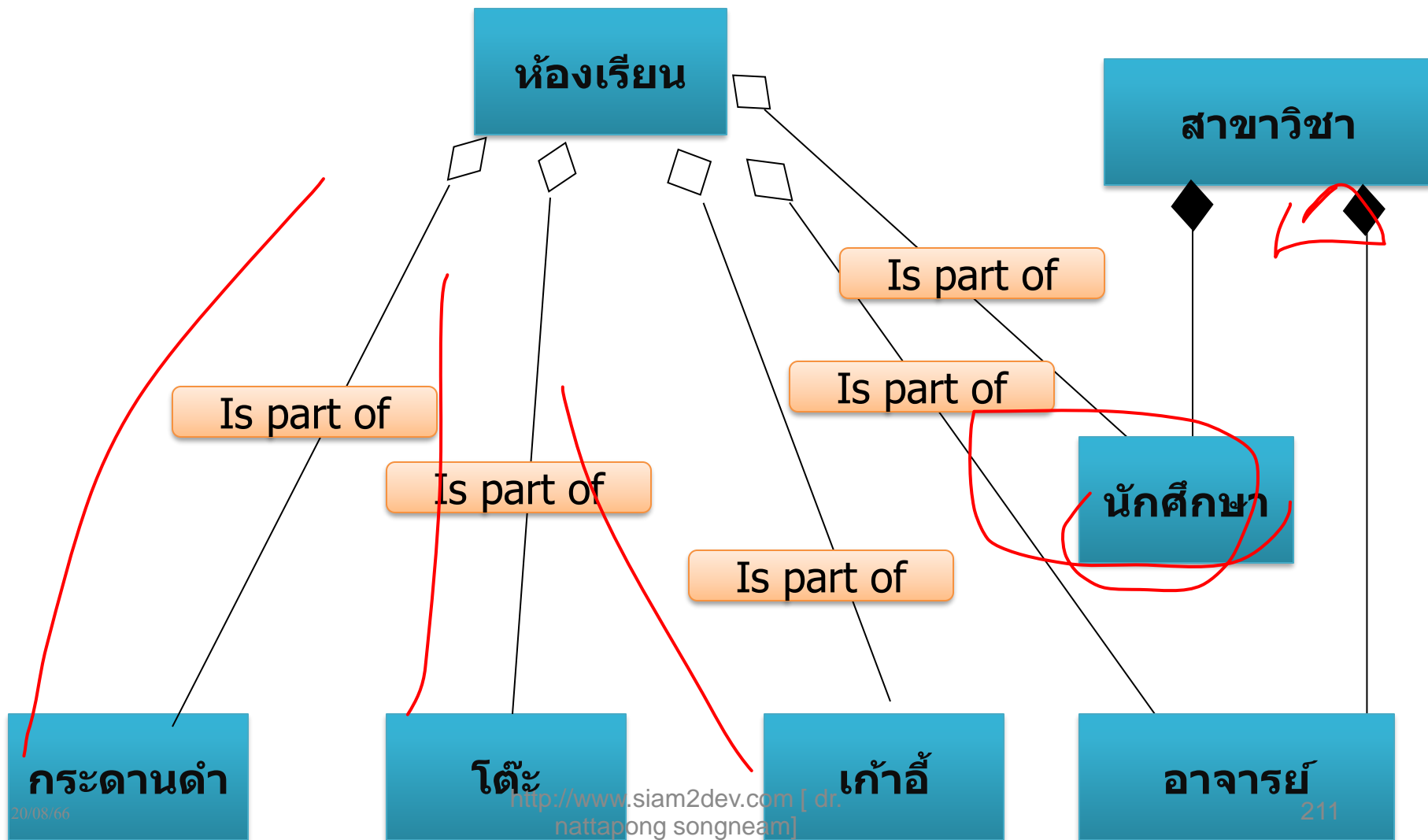
Classification Abstraction

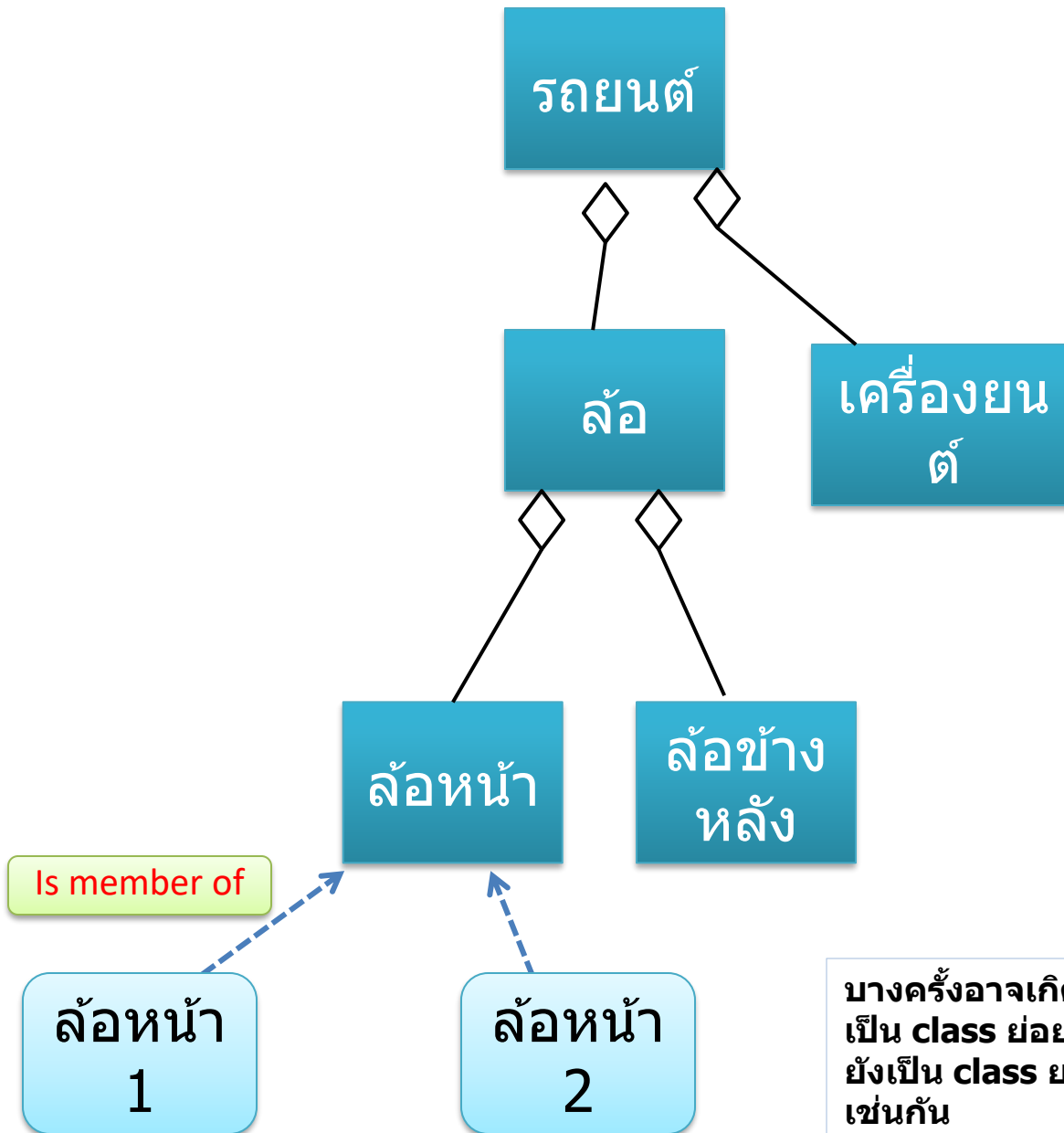
classifi

A



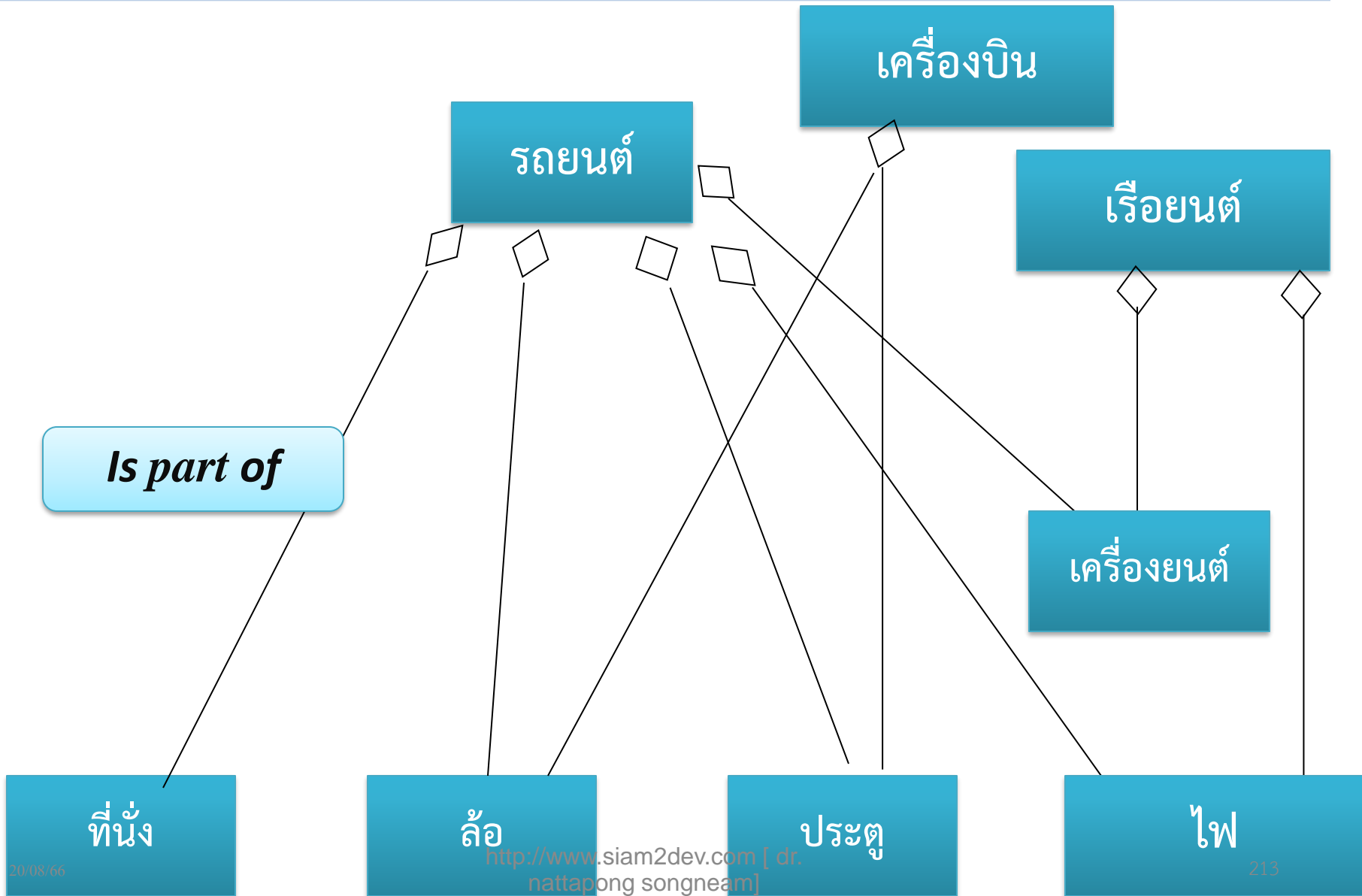
บางครั้งอาจเกิดกรณีที่ class หนึ่งสามารถเป็น class ย่อยของ class หนึ่งได้
พร้อมทั้งยังเป็น class ย่อยของอีก class หนึ่งได้เช่นกัน



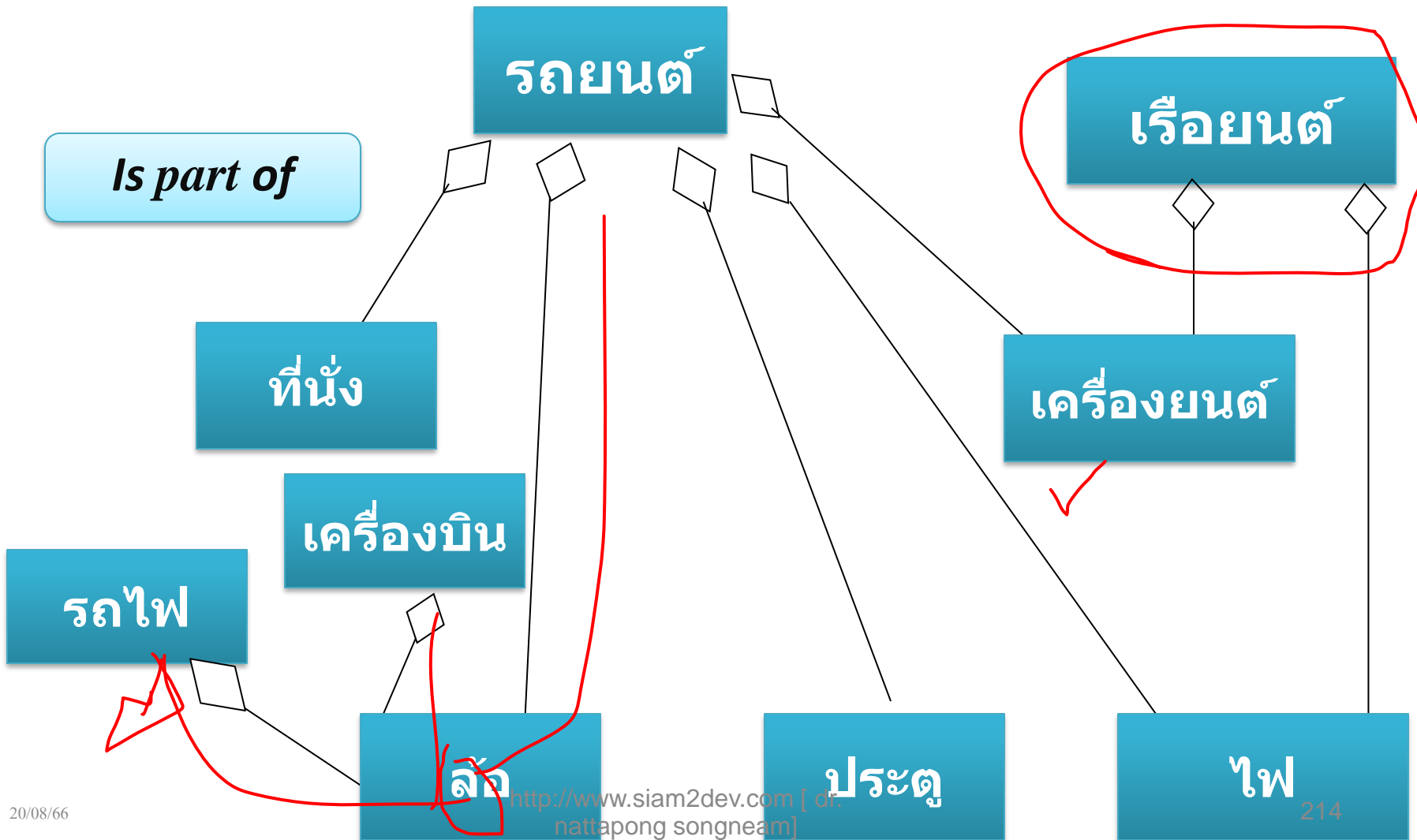


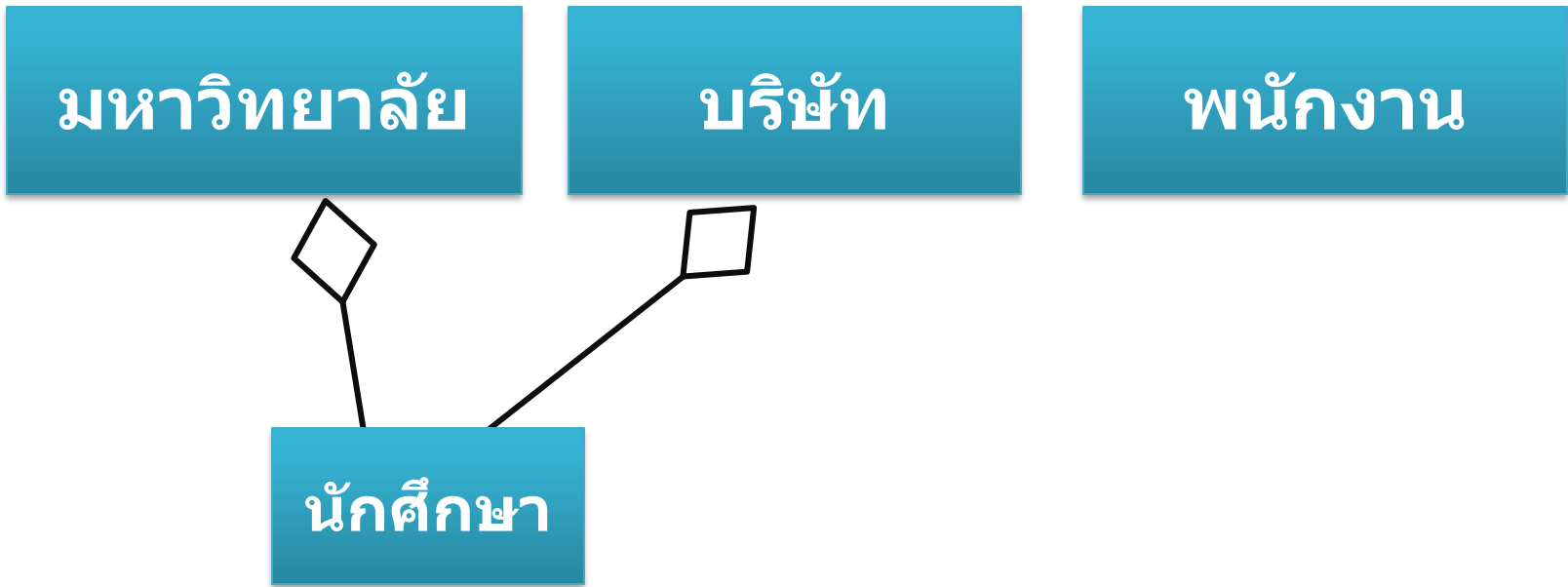
บางครั้งอาจเกิดกรณีที่ class หนึ่งสามารถเป็น class ย่อยของ class หนึ่งได้พร้อมทั้งยังเป็น class ย่อยของอีก class หนึ่งได้เช่นกัน

บางครั้งอาจเกิดกรณีที่ class หนึ่งสามารถเป็น class ย่อยของ class หนึ่งได้พร้อมทั้งยังเป็น class ย่อยของอีก class หนึ่งได้เช่นกัน

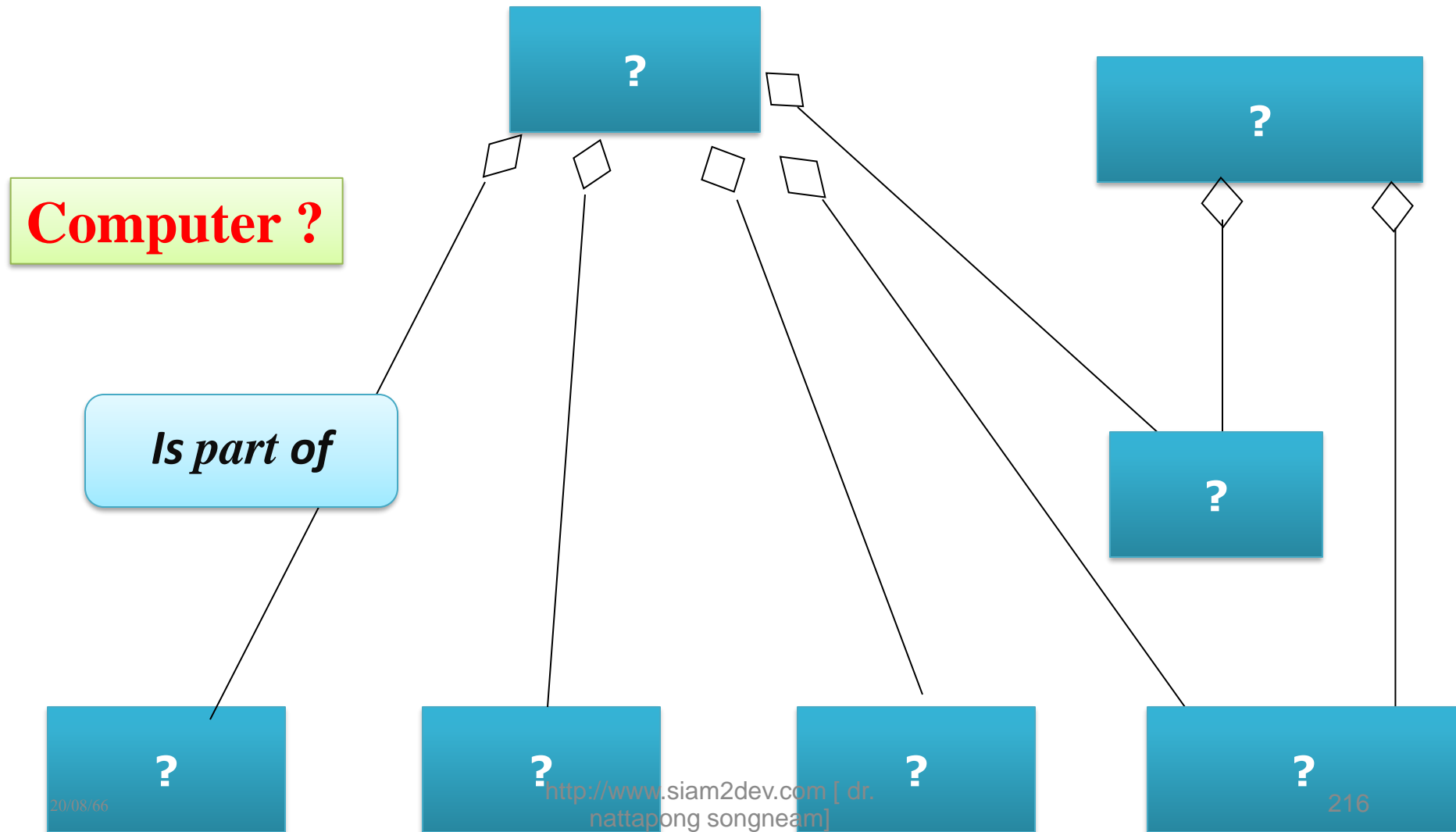


บางครั้งอาจเกิดกรณีที่ class หนึ่งสามารถเป็น class ย่อยของ class หนึ่งได้พร้อมทั้งยังเป็น class ย่อยของอีก class หนึ่งได้เช่นกัน

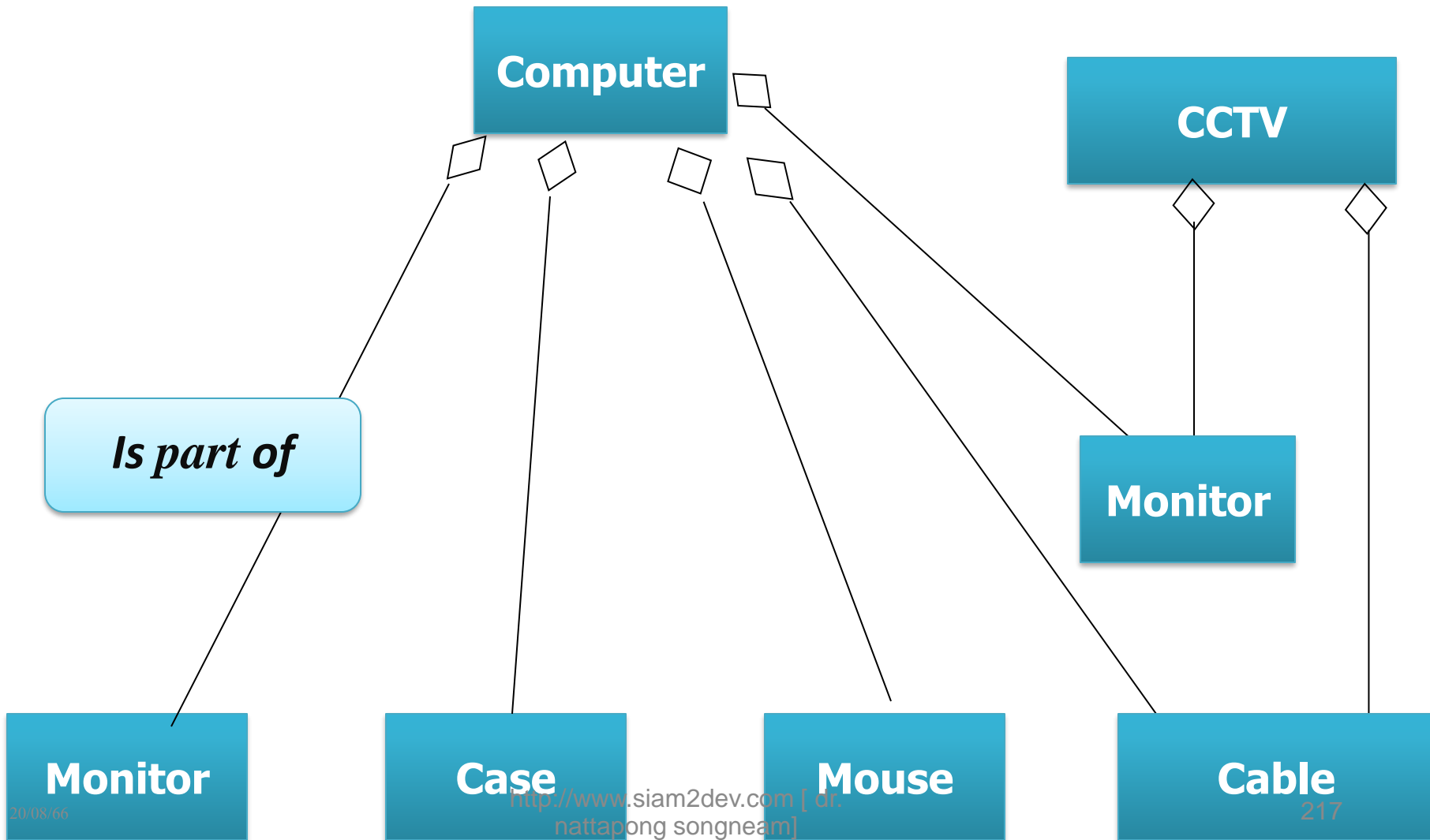




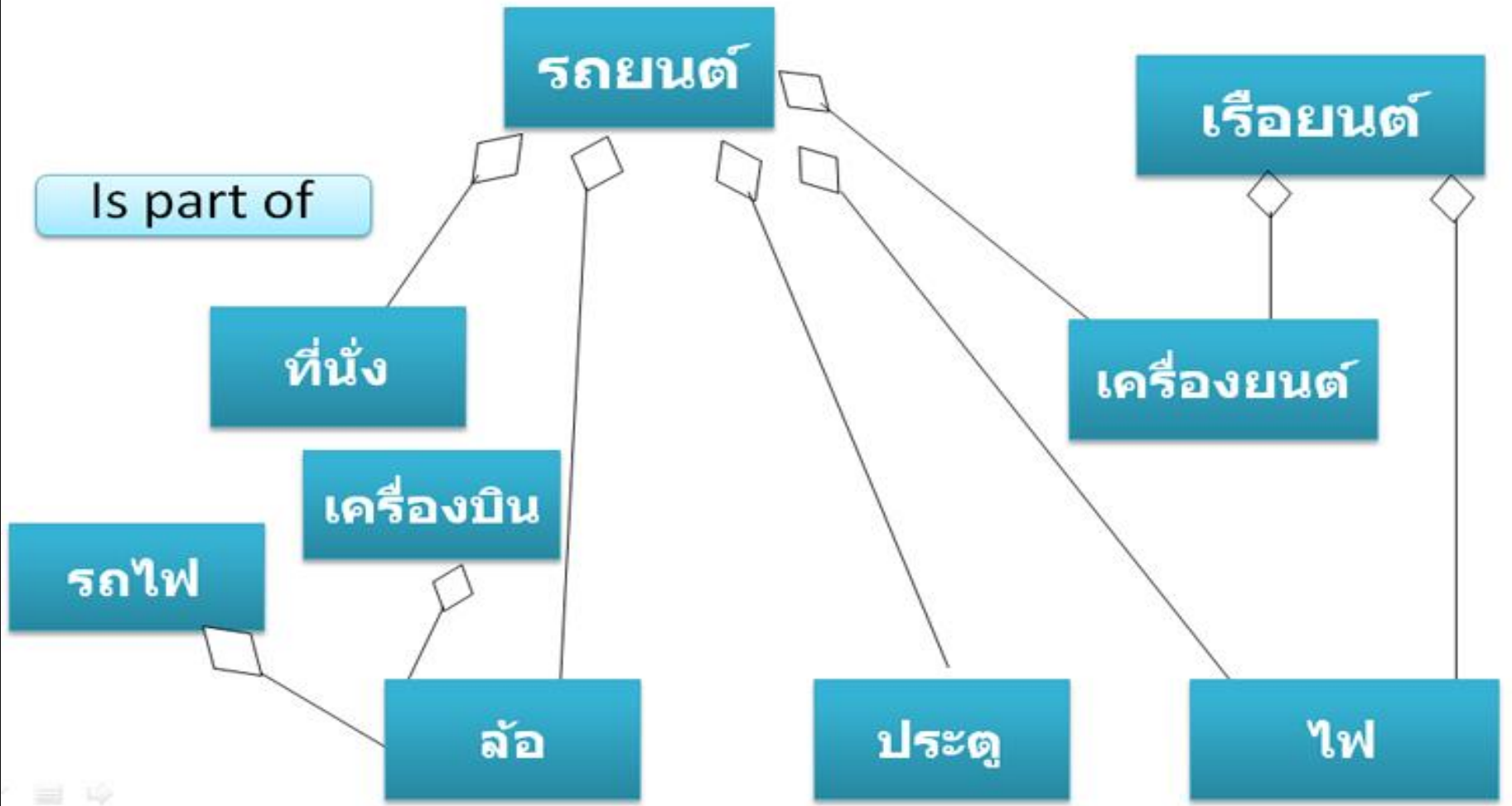
บางครั้งอาจเกิดกรณีที่ class หนึ่งสามารถเป็น class ย่อยของ class หนึ่งได้พร้อมทั้งยัง
เป็น class ย่อยของอีก class หนึ่งได้เช่นกัน



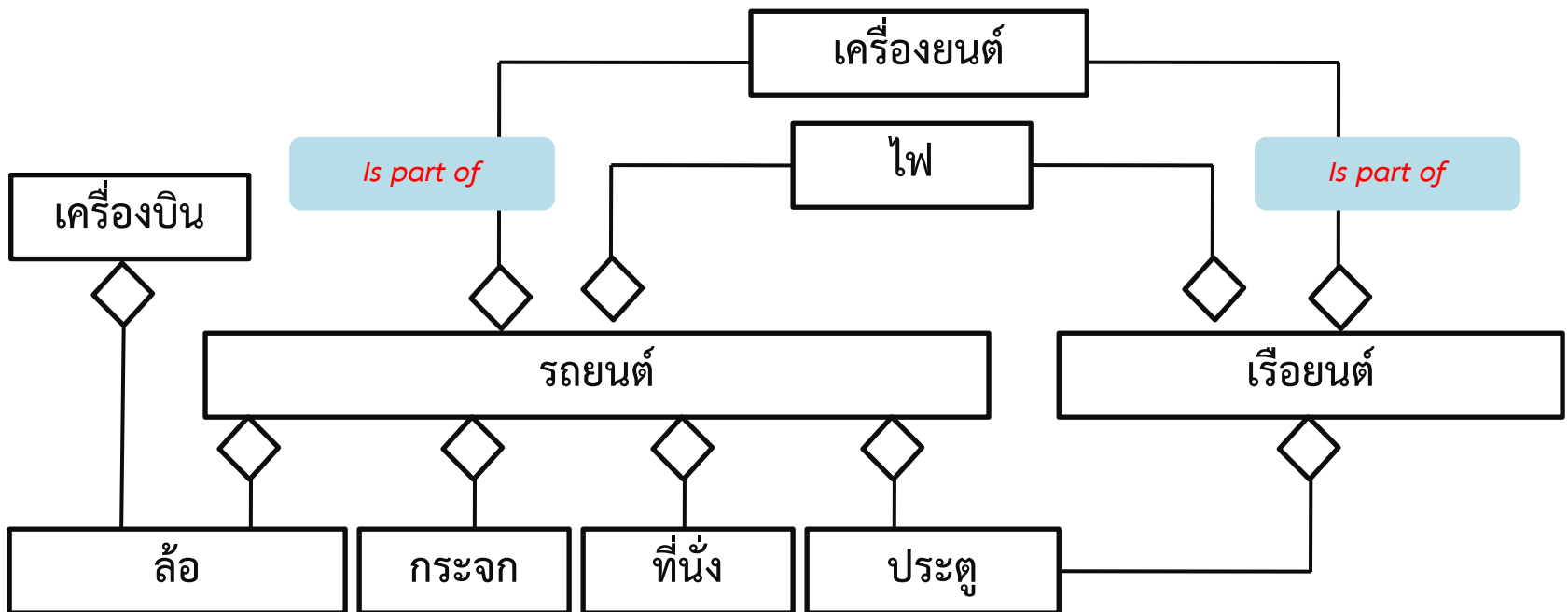
บางครั้งอาจเกิดกรณีที่ class หนึ่งสามารถเป็น class ย่อยของ class หนึ่งได้
พร้อมทั้งยังเป็น class ย่อยของอีก class หนึ่งได้เช่นกัน



บางครั้งอาจเกิดกรณีที่ class หนึ่งสามารถเป็น class ย่อยของ class หนึ่งได้พร้อมทั้งยังเป็น class ย่อยของอีก class หนึ่งได้เช่นกัน



ปรับเปลี่ยนภาพคลาสใหม่



ยานพาหนะ

ล้อ

เครื่องยนต์

ที่นั่ง

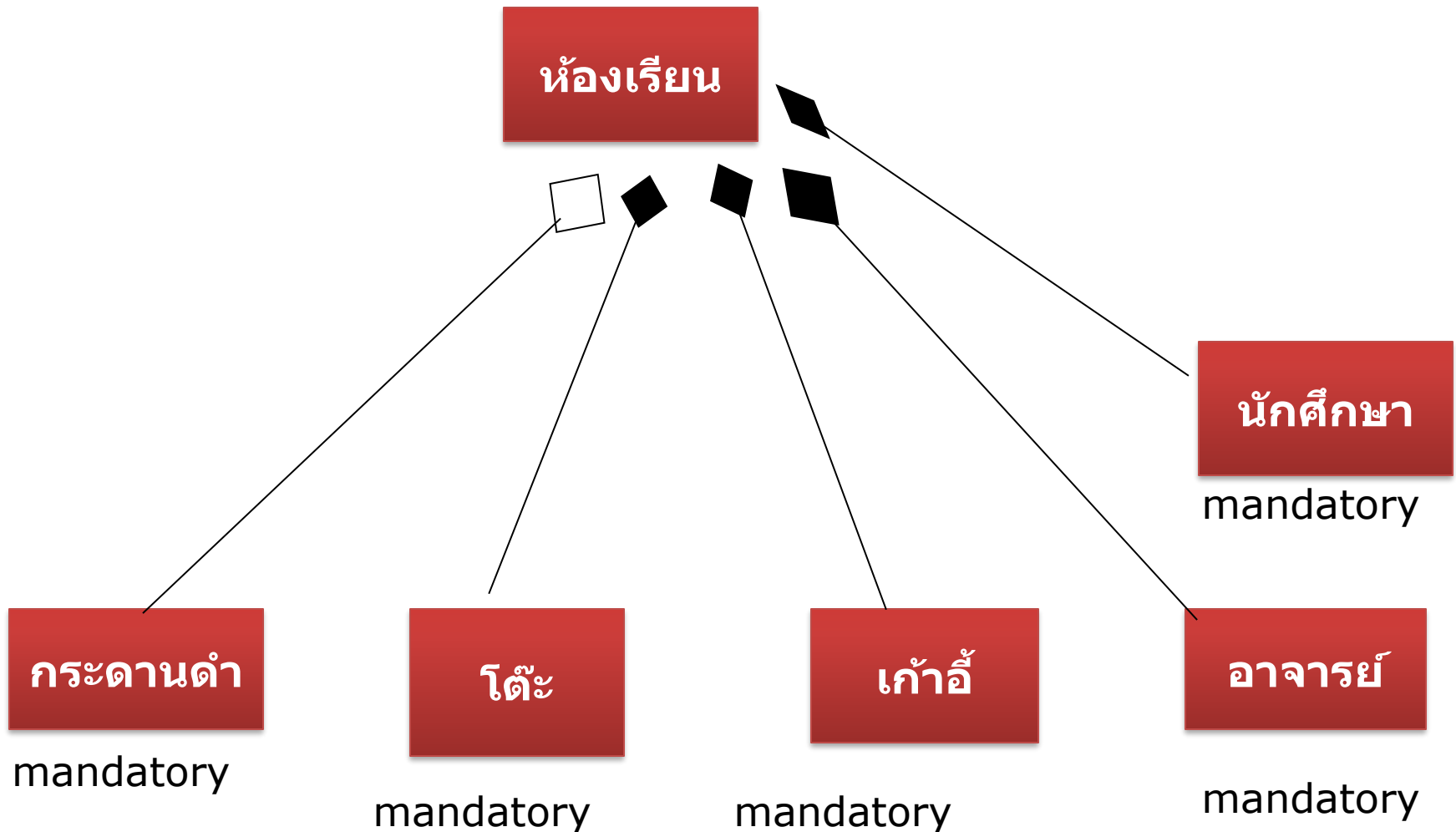
Cardinality , Required Component ,Optional Component

สำหรับ *Aggregation Abstraction* การประกอบกันของ class หรือความสัมพันธ์เชิง is part of นั้นอาจจะเกิดกรณี ที่ class หลัก (main class) ประกอบไปด้วย class ย่อย (Composite class) ชนิดที่หรือ เพียงชั้นเดียว แต่ประกอบด้วย class ชนิดที่สอง จำนวน 4 ชั้นขึ้นไป และอาจจะประกอบไปด้วยชนิดที่สาม ไม่จำกัดจำนวน หรืออาจไม่มีเลยก็ได้ ซึ่งสิ่งที่ใช้ในการแสดงความสัมพันธ์ ดังกล่าวนี้เรียกว่า *Cardinality*

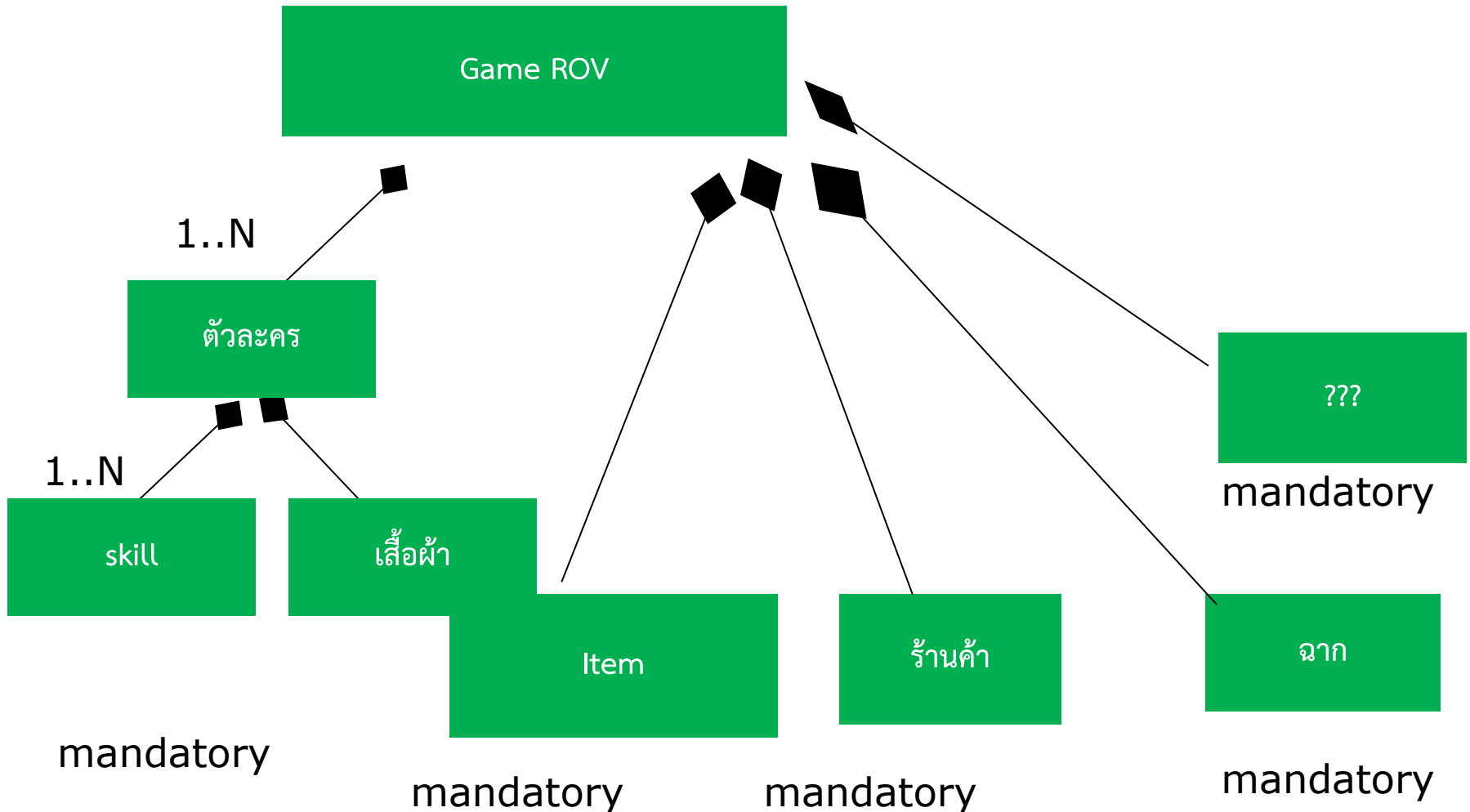
Cardinality , Required Component ,Optional Component

ในทาง OO เราเรียก Class ย่อย ว่า Component นั่นเองโดย
เราเรียกส่วนประกอบที่**จำเป็น**ต้องมีว่า Require Component เช่น
รถยนต์ต้องมีเครื่องยนต์ ถ้าไม่มีเครื่องยนต์ รถยนต์ก็ ไม่สามารถวิ่งได้
นั่นเอง เราเรียกส่วนประกอบที่**ไม่จำเป็น**ต้องมีว่า Optional
Component เช่น เครื่องปรับอากาศในรถยนต์ไม่จำเป็นต้องมีก็ได้
หมายถึงไม่มีเครื่องปรับอากาศรถยนต์ก็ยังสามารถวิ่งได้ รถยนต์ไม่
จำเป็นต้องมีกล้องมองข้างหลังดังนั้นจึงเป็น Optional Component

Cardinality , Required Component ,Optional Component



Cardinality , Required Component ,Optional Component

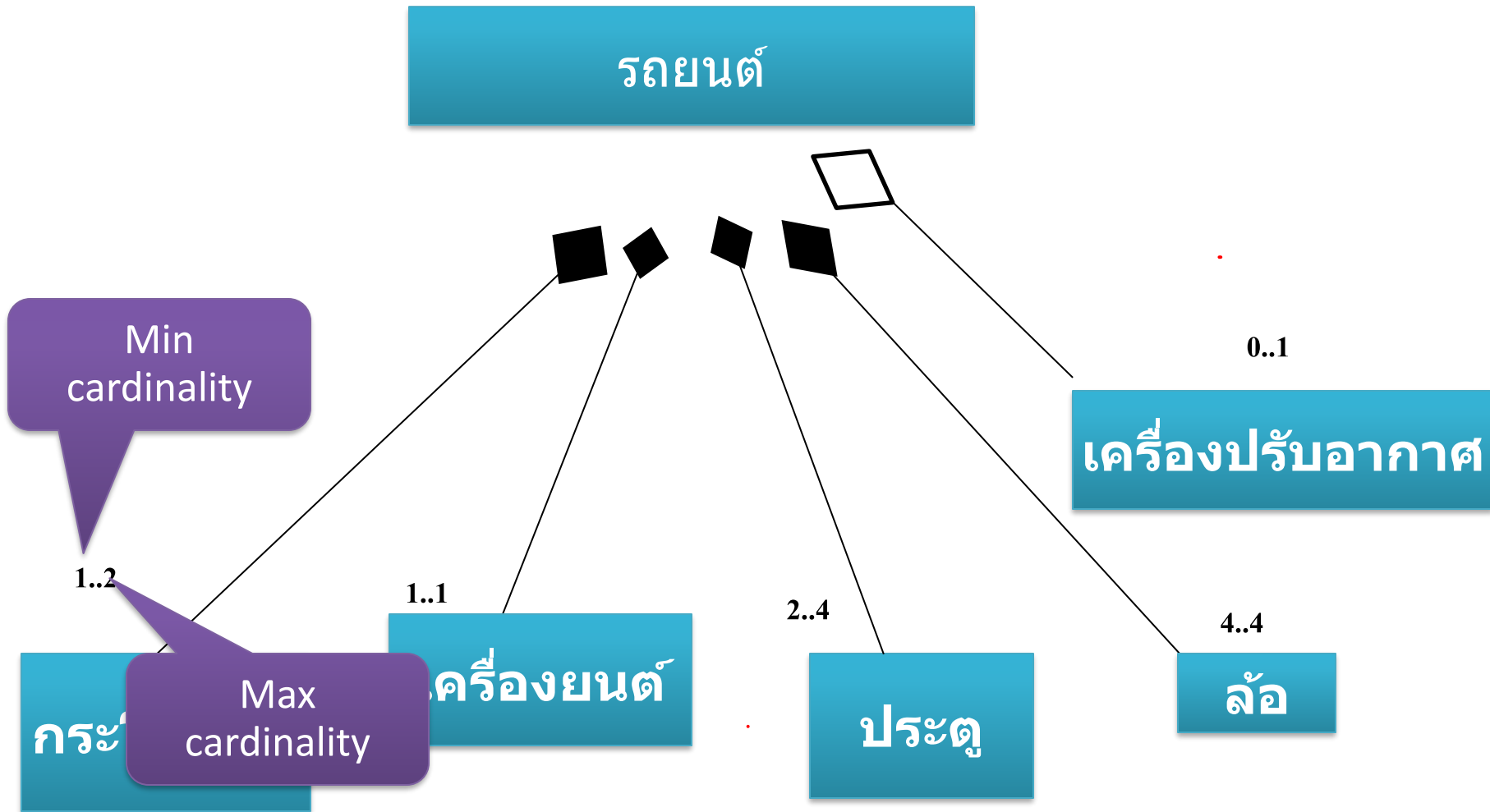


ส่วนประกอบของรถยนต์

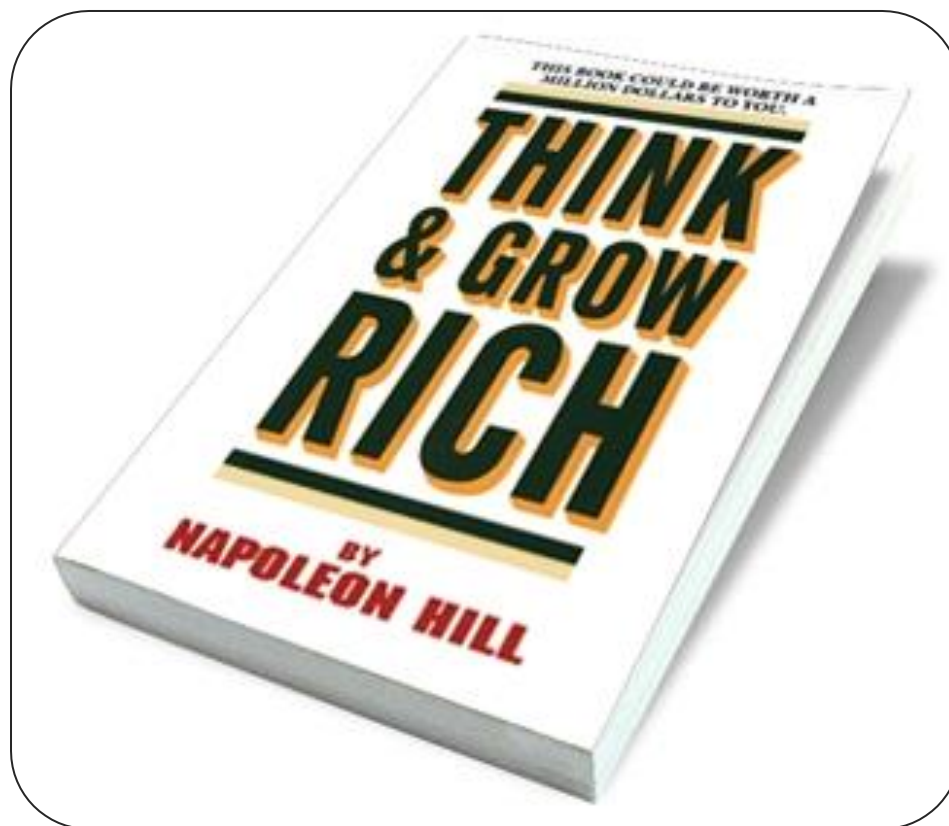


รูปจาก : <http://www.thaissecondhand.com/product/9951092/>

Cardinality, Required Component, Optional Component

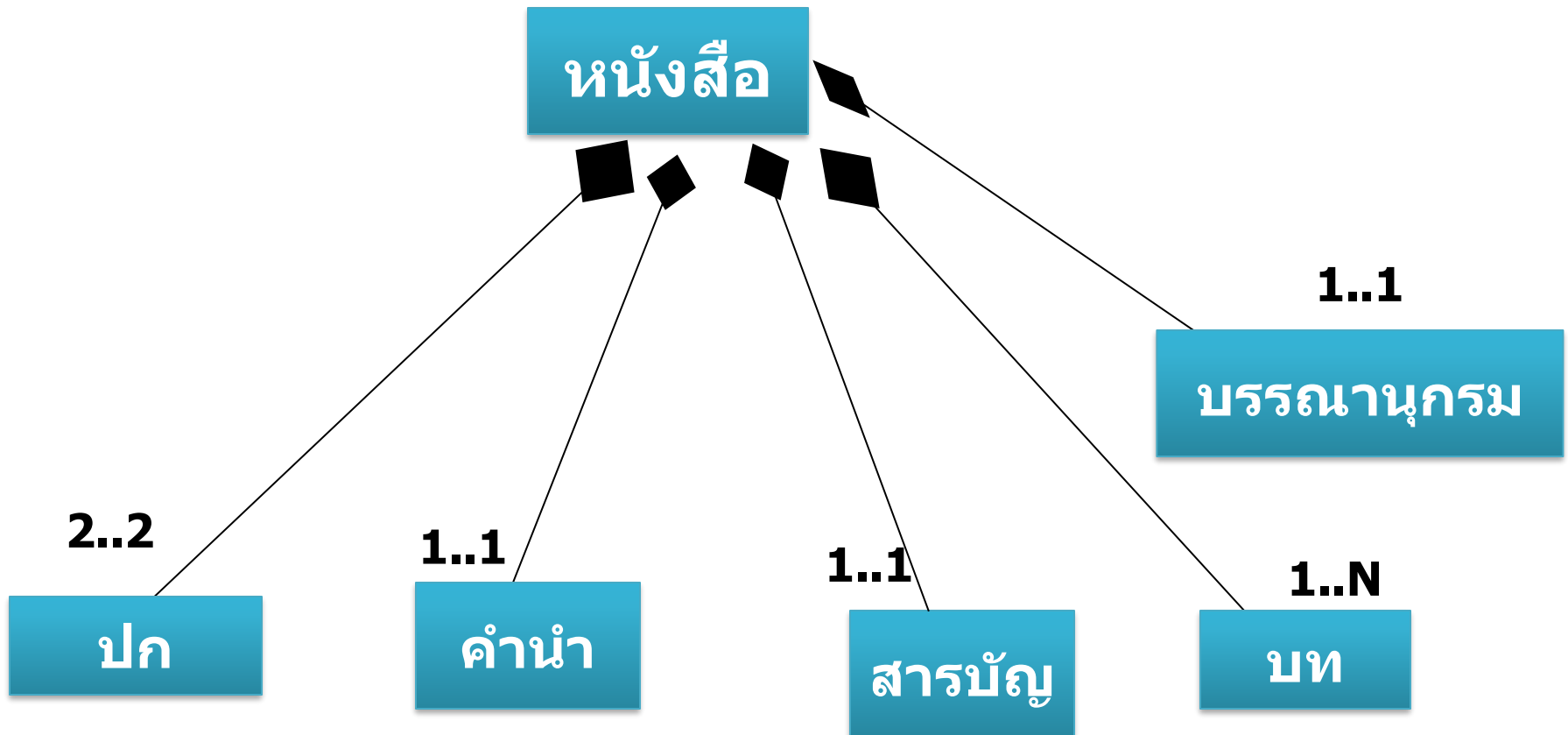


ส่วนประกอบของหนังสือ



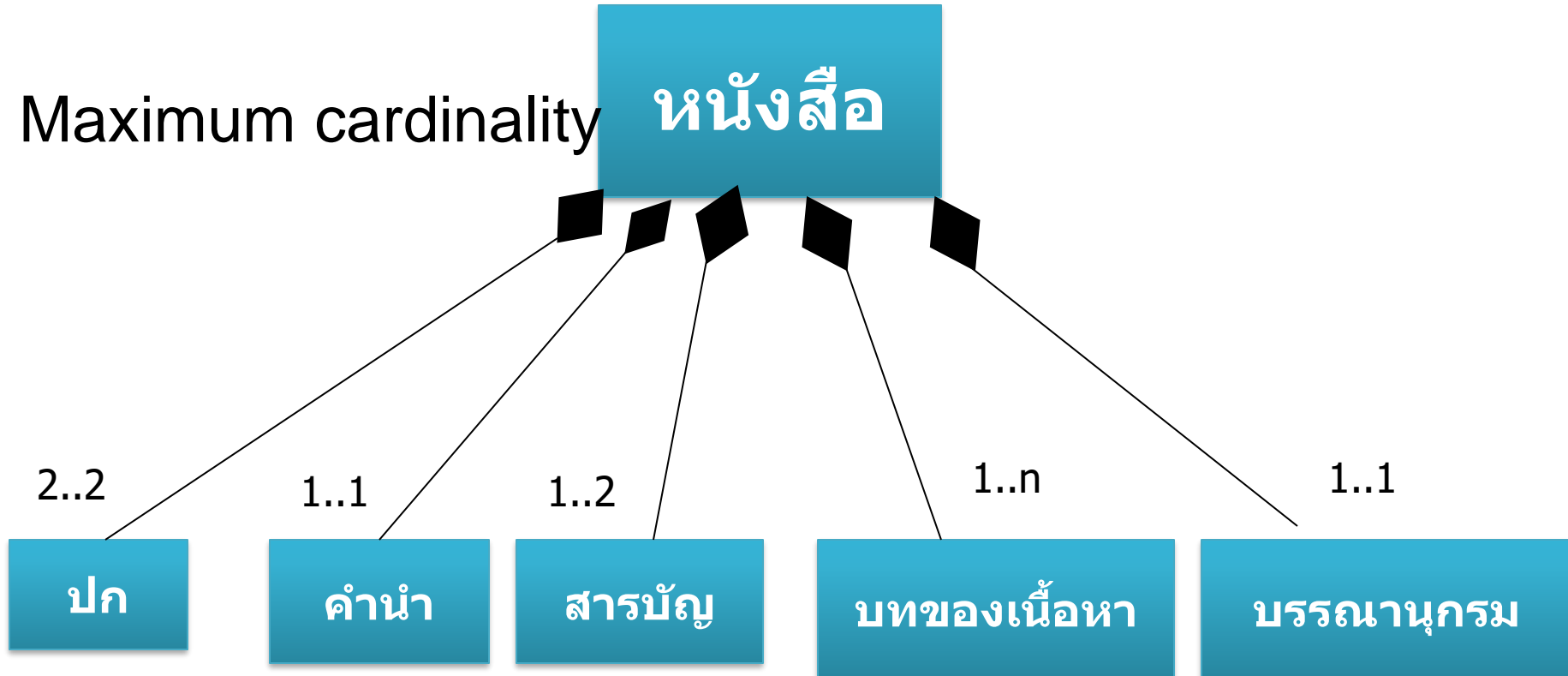
รูปจาก : <http://board.postjung.com/641579.html>

Cardinality , Required Component ,Optional Component



Minimum cardinality

Maximum cardinality

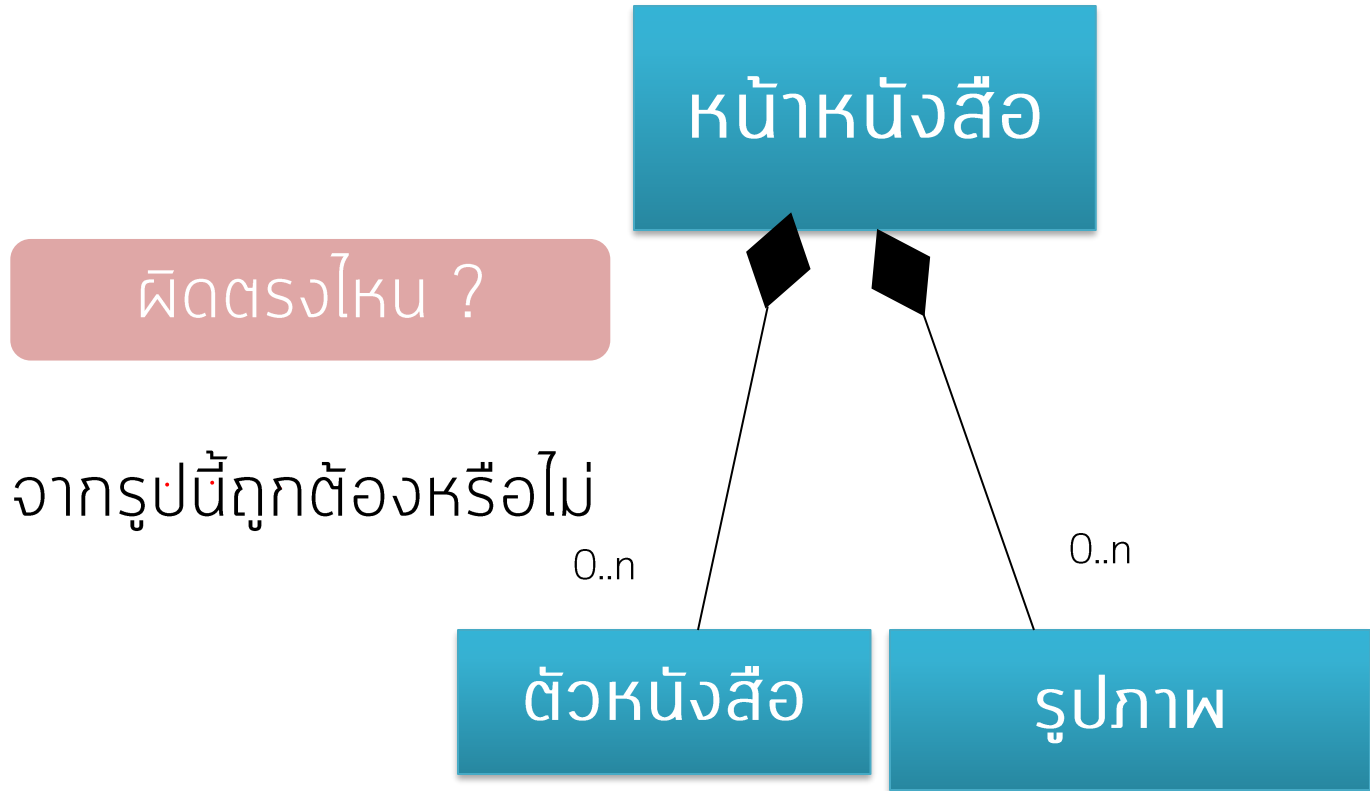


ต.ย. การเขียน cardinality ของ class หนังสือ

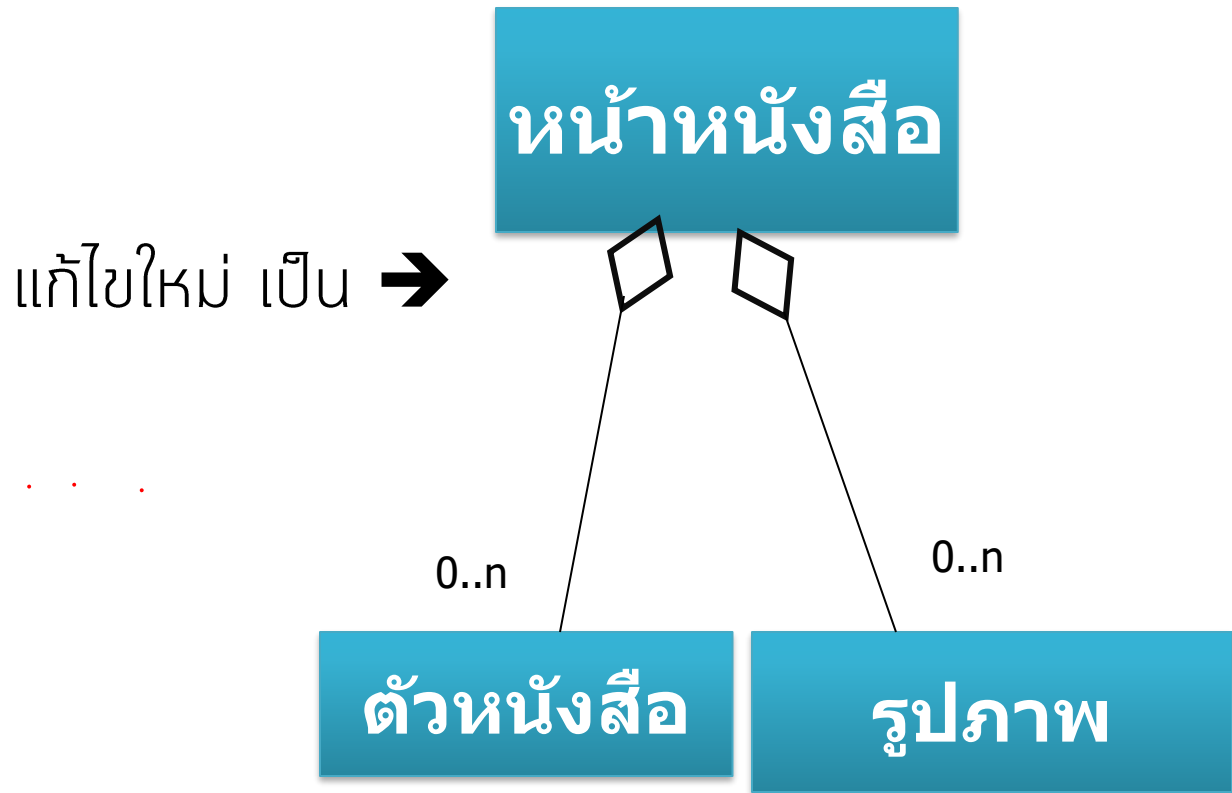
บทความเนื้อหา

1..n

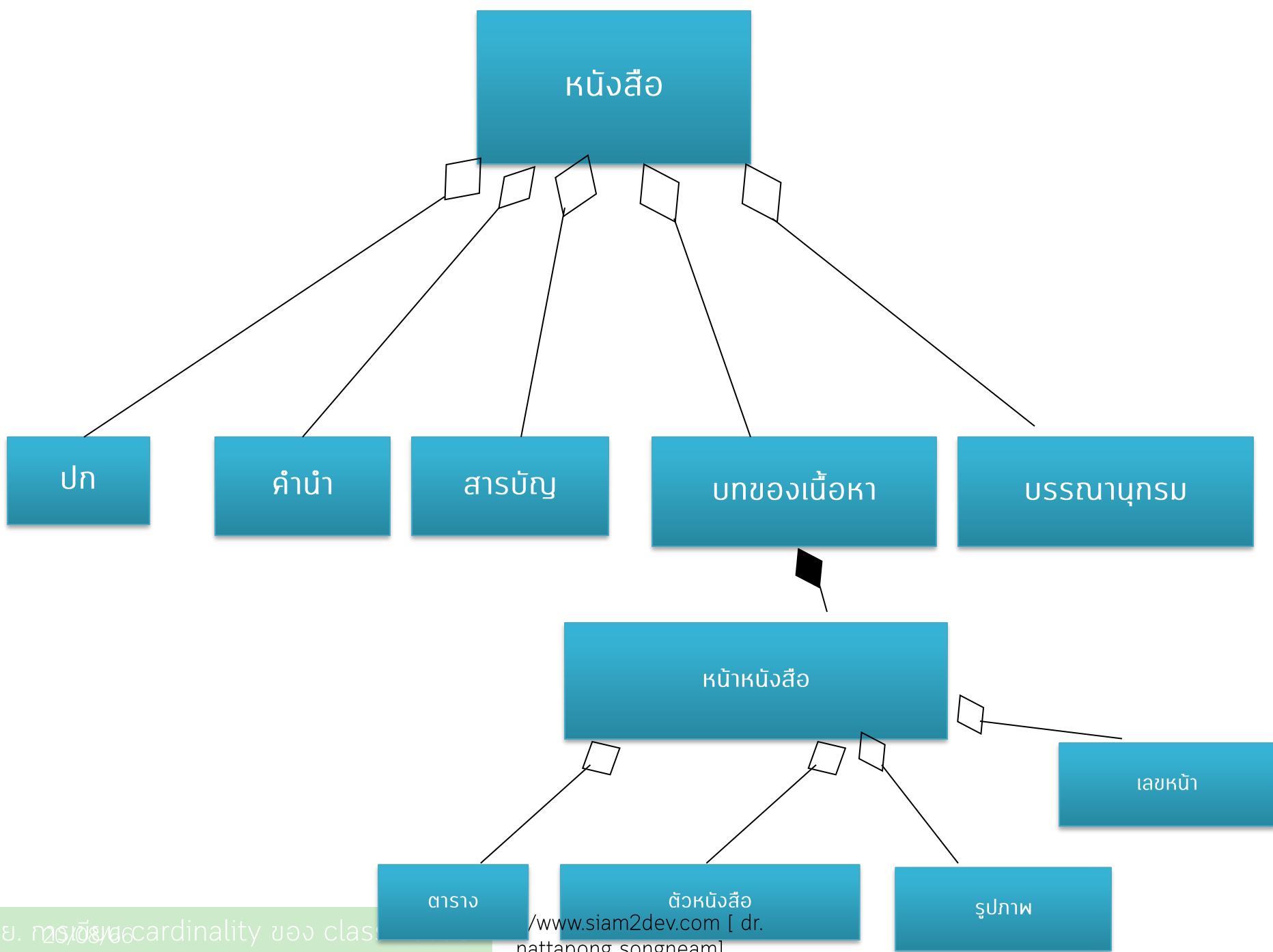
หน้าหนังสือ



ต.ย. การเขียน cardinality ของ class หน้าหนังสือ



ด.ย. การเขียน cardinality ของ class หน้าหนังสือ



หนังสือ

ปก

คำนำ

สารบัญ

บทของเนื้อหา

บรรณานุกรม

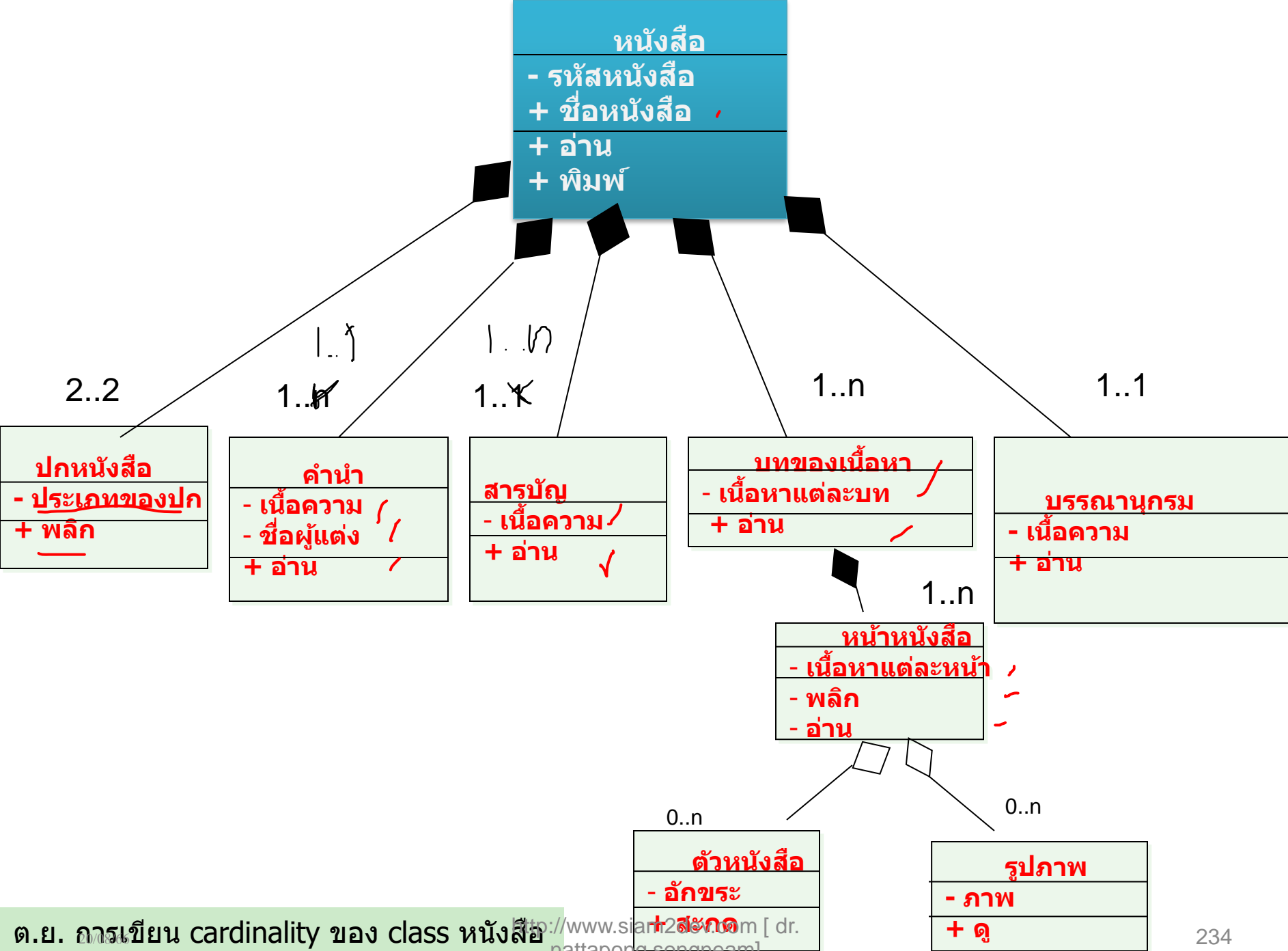
หน้าหนังสือ

เลขหน้า

ตาราง

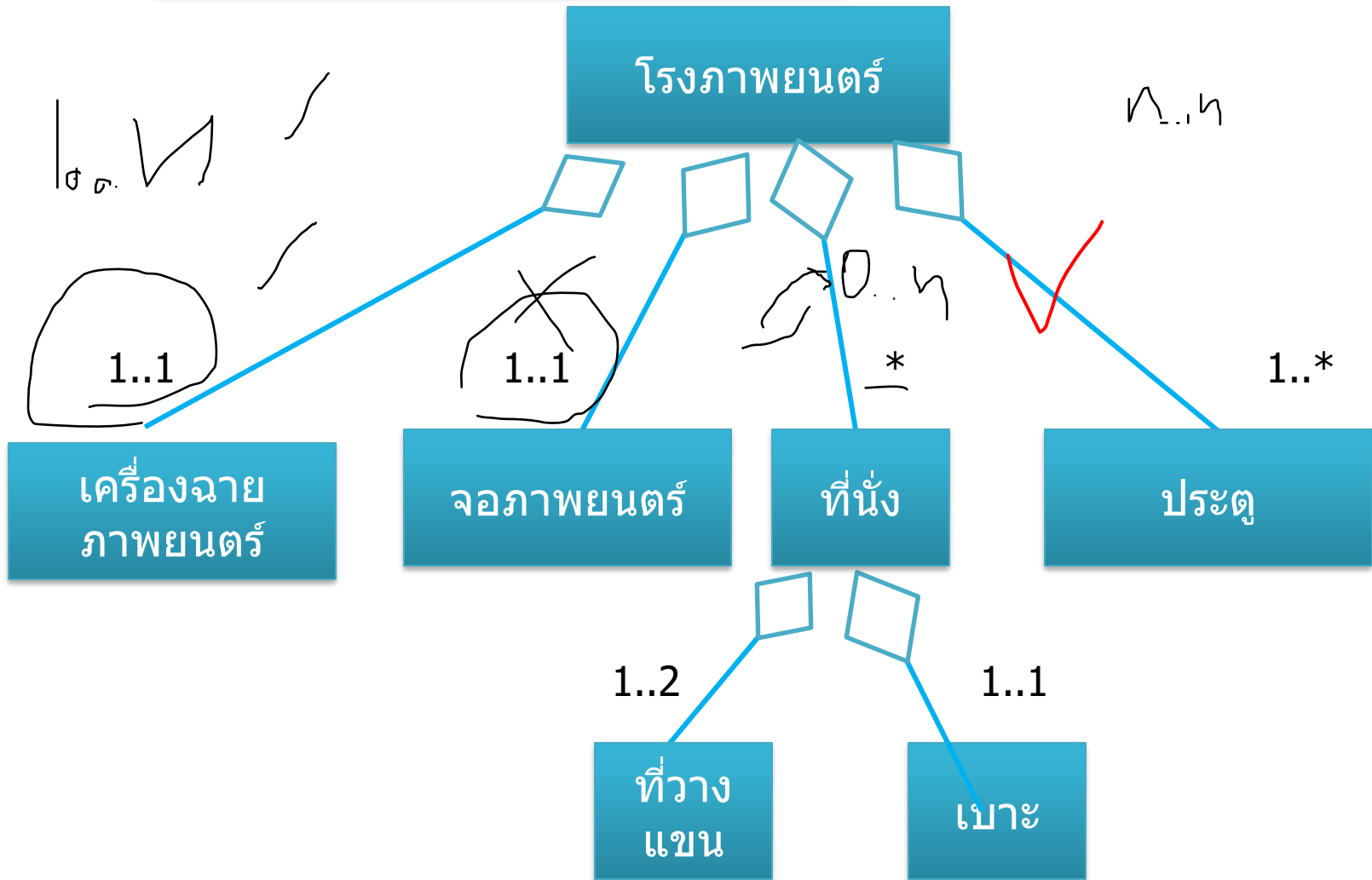
ตัวหนังสือ

รูปภาพ



ด.ย. การเขียน cardinality ของ class หนังสือ

Problem Domain : โรงพยาบาลนคร



จงให้รายละเอียดของ *โรงพยาบาลนคร*

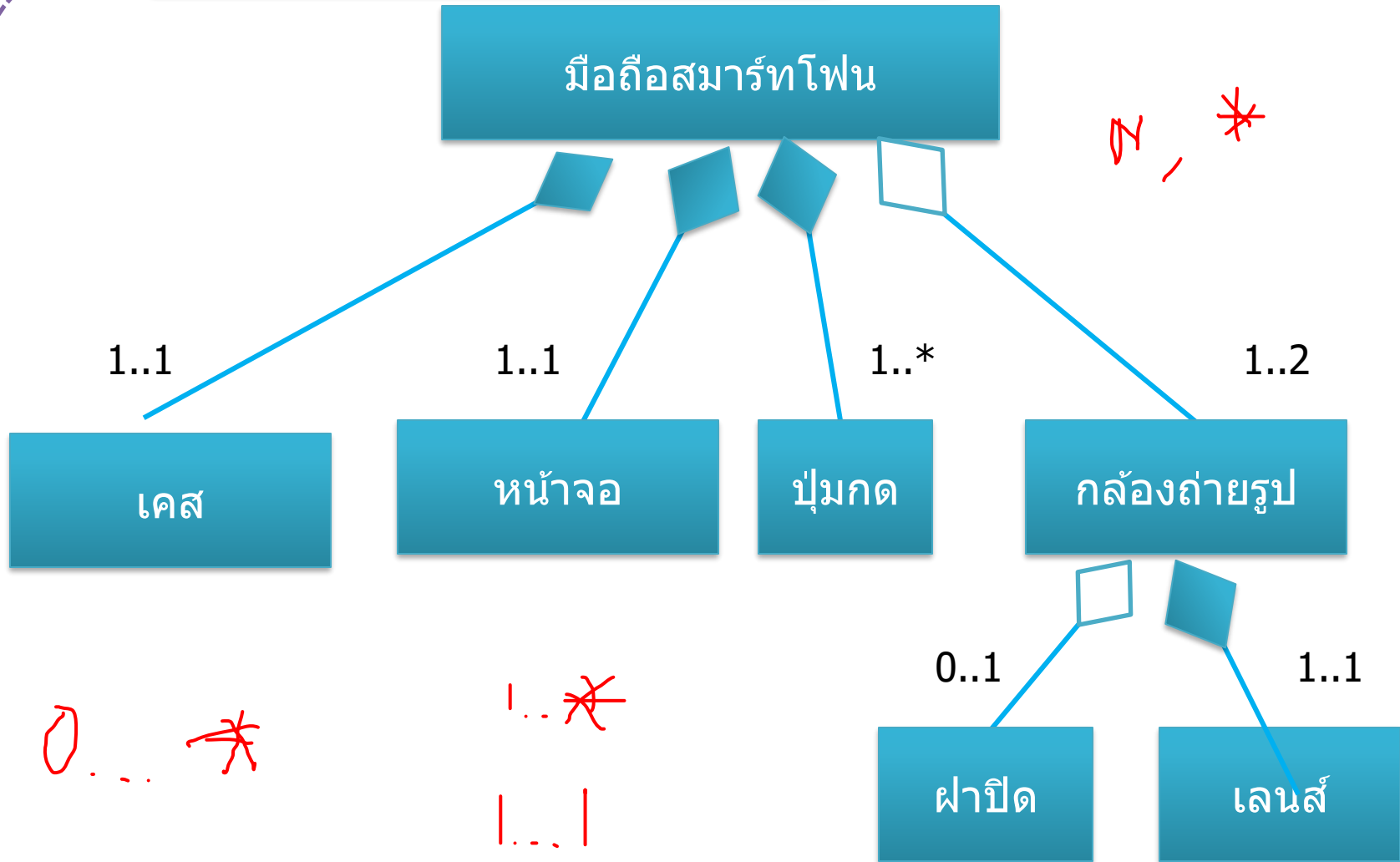
ส่วนประกอบของโทรศัพท์มือถือ



รูปจาก : <http://www.techycite.com/topic/10867.html>

Problem Domain : โทรศัพท์มือถือ

มือถือสมาร์ทโฟน

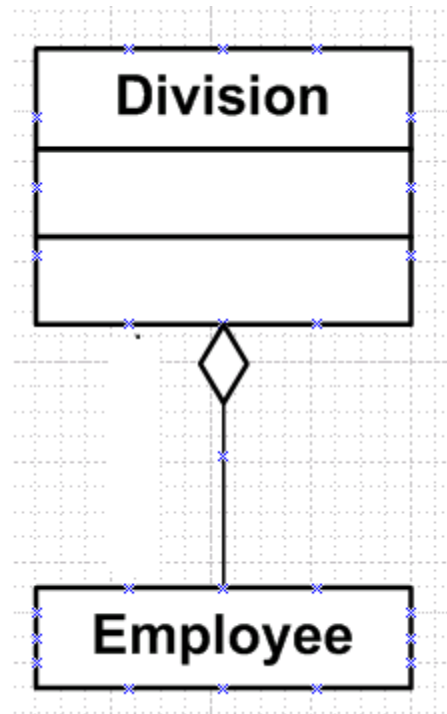
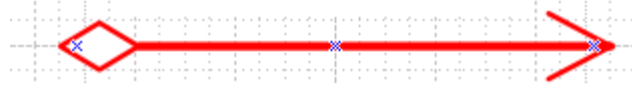


จงให้รายละเอียดของ

โทรศัพท์มือถือ

- ??

Aggregation

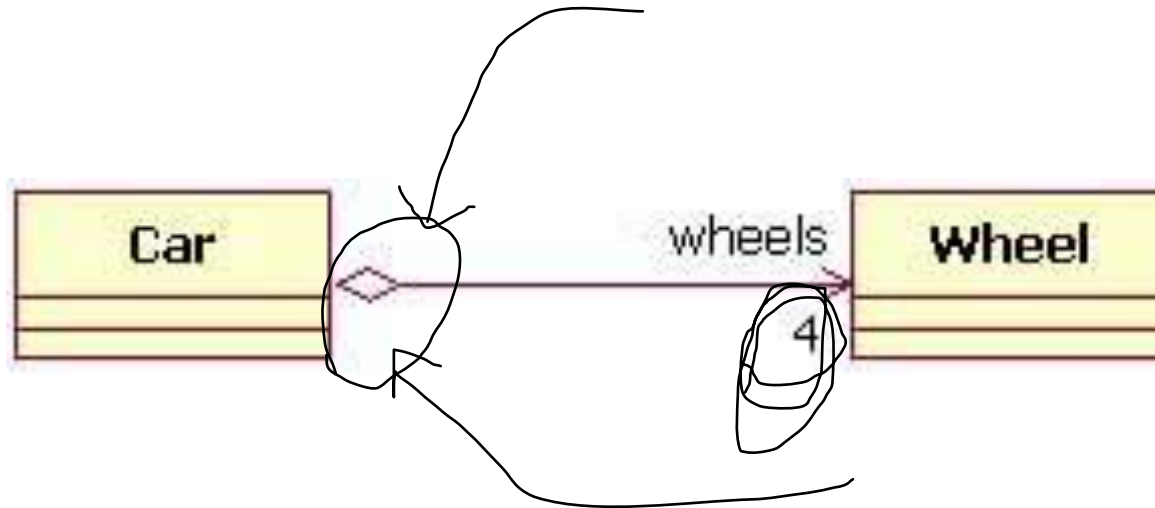


A unfilled diamond shows aggregation.
This relationship indicates that one is
PART of the other.

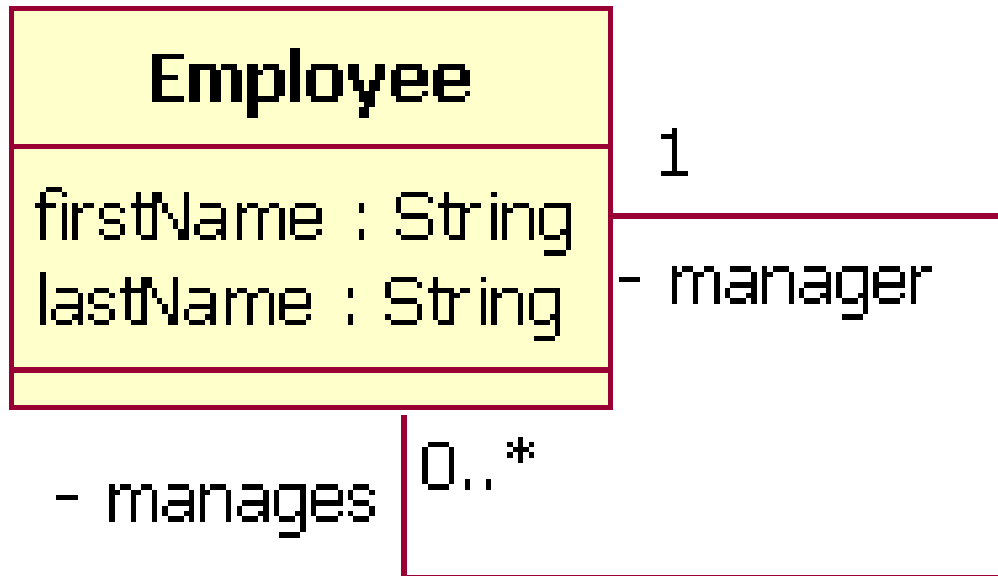
An Employee is PART of a division.

But a division can stand on its own
without Employees.

Aggregation



Example of a reflexive association relationship



The Person Class : an example in VB.NET

```
Class Person
    Private name As String
    Private age As Integer

    Public Sub New(ByVal nm As String, ByVal ag As Integer)
        name = nm
        age = ag
    End Sub

    Public Function getJob() As String
        Return "You got the Job"
    End Function

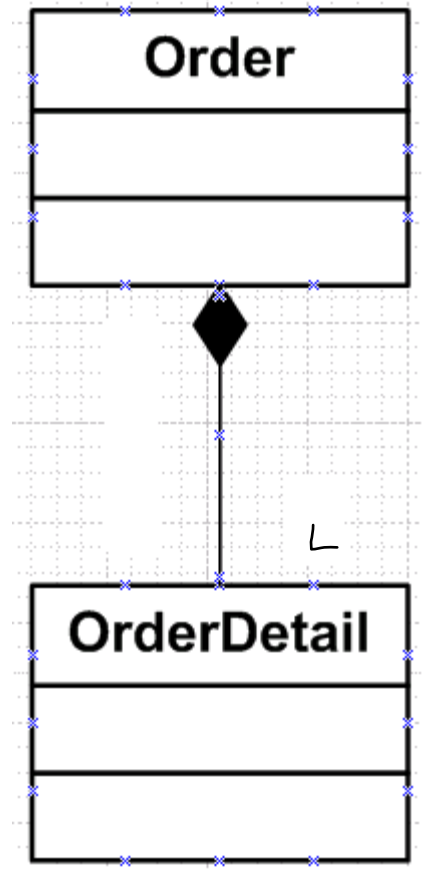
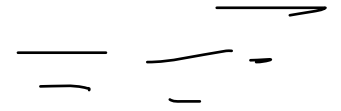
    Public Function getAge() As Integer
        Return age
    End Function

    Public Sub divideName()
    End Sub

End Class
```



Composite

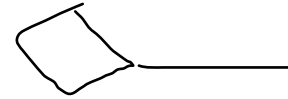


A filled diamond shows Composite Aggregation. This relationship indicates that one is PART of the other.

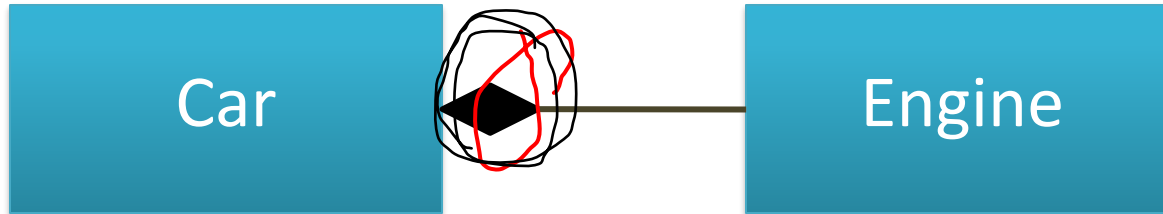
An OrderDetail is PART of an order.

But an Order CANNOT stand on its own without Order Details.

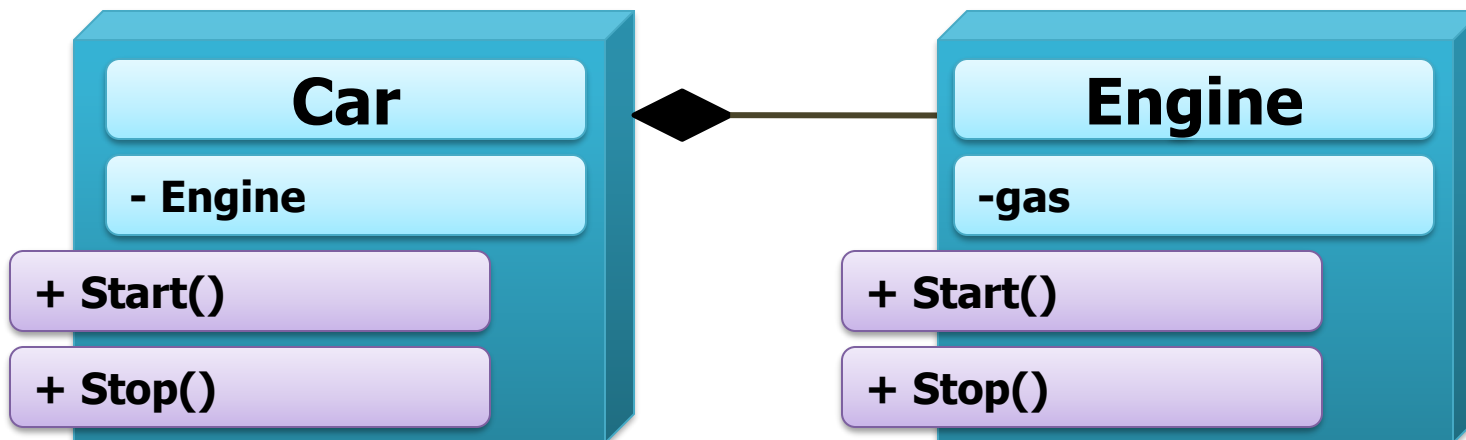
Composite



- An Engine **is PART of** a car.
- But a Car CANNOT stand on its own without engine.



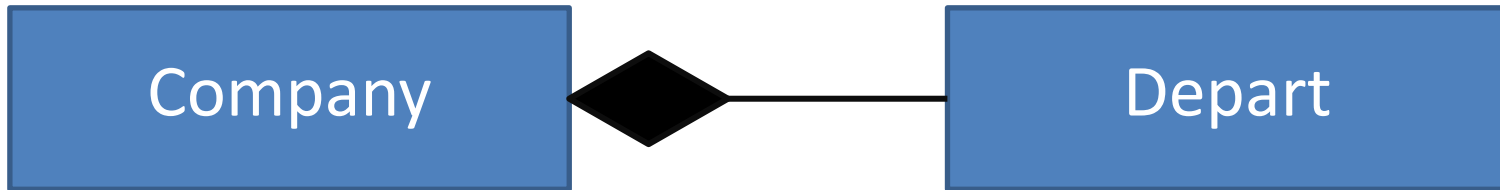
ทำให้ชัดเจนโดยการใส่รายละเอียดลงไป



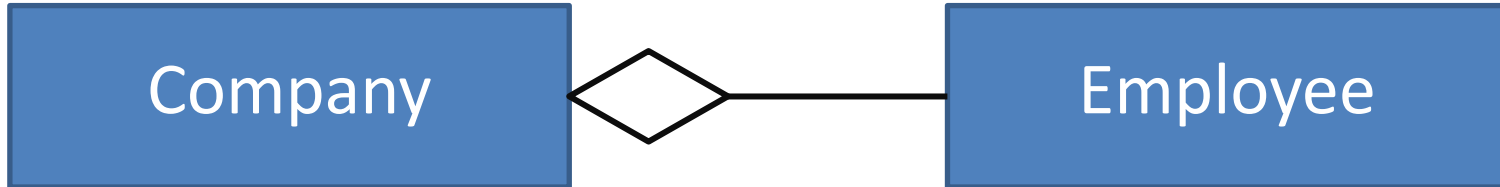
Composite

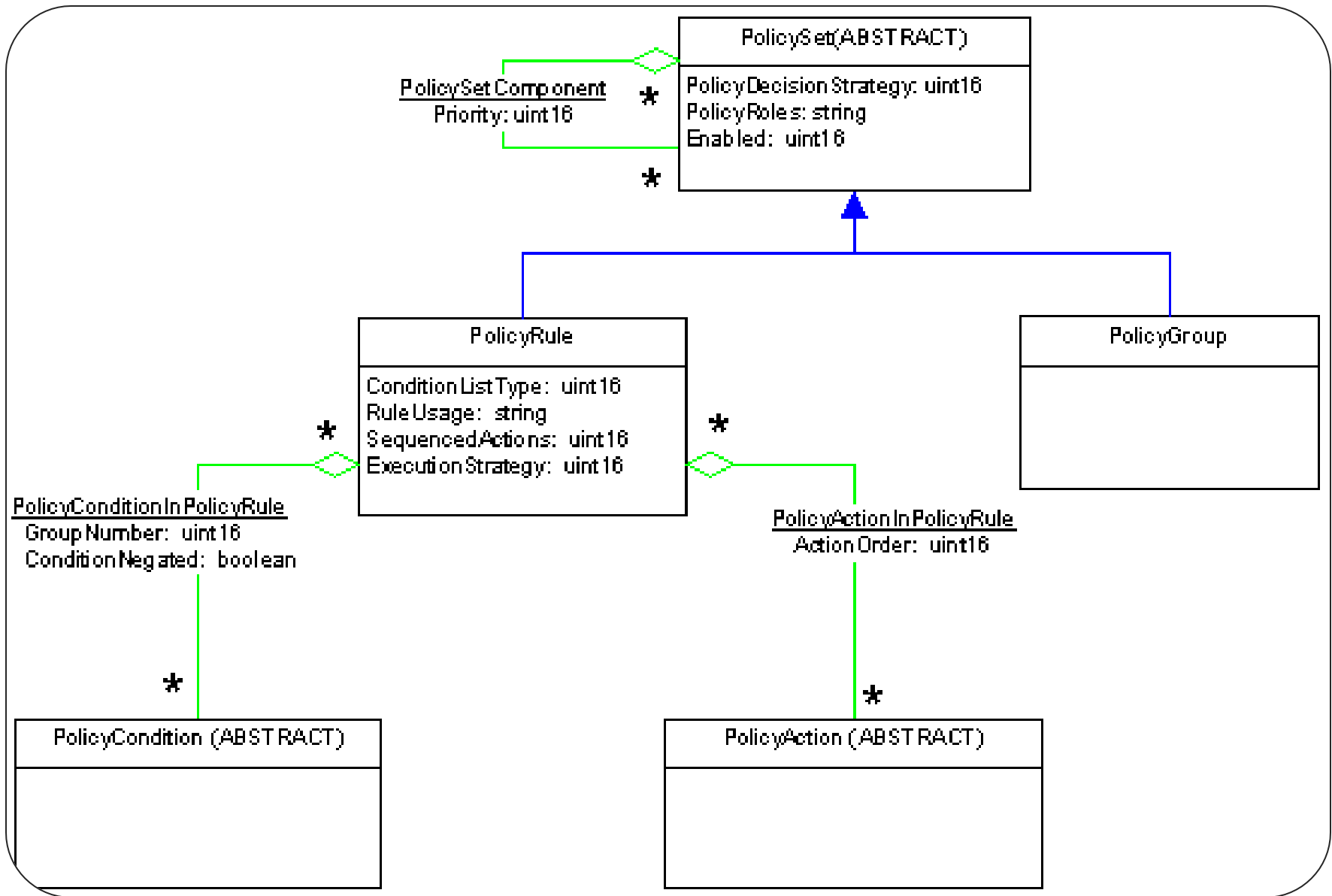


Composite



Aggregate





ASSIGNMENT 3_02

1. จงอธิบายความหมายของ Aggregation Abstraction
2. Cardinality คืออะไร
3. จงบอกความแตกต่างของ Aggregate กับ Composite
4. จงอธิบายความแตกต่างระหว่างมุมมองของ composition กับ decomposition

ASSIGNMENT 3_02(ต่อ)

5. จงเขียนแผนภาพที่แสดง Aggregation Abstraction ของ Computer PC
6. จงเขียนแผนภาพที่แสดง Aggregation Abstraction ของ โทรศัพท์มือถือ (Mobile)
7. จงเขียนแผนภาพที่แสดง Aggregation Abstraction ของ สมุดธนาคาร (Bank Account)
8. จงเขียนแผนภาพที่แสดง Aggregation Abstraction ของ

แบบฝึกหัด 3.3

3.3.1 คลินิกรักษาโรคทั่วไปแห่งหนึ่งจะประกอบไปด้วยแพทย์ประจำ 1 นาย พยาบาลอย่างน้อย 1 คน แต่อาจจะมากกว่านี้ได้ โดยจะมีห้องสำหรับตรวจรักษาโรคทั่วไป 1 ห้อง และห้องสำหรับรักษาโรคเฉพาะทางอีก 1 ห้อง และมียารักษาโรคต่างๆ เป็นจำนวนมาก

3.3.2 ในห้องสมุดของมหาวิทยาลัยราชภัฏพระนคร จะประกอบไปด้วยเจ้าหน้าที่ดูแลห้องสมุด อาจารย์ เจ้าหน้าที่ ผู้บริหาร และนักศึกษา โดยเจ้าหน้าที่ดูแลห้องสมุดจะมีจำนวน 3 ท่าน ผู้อำนวยการห้องสมุดเป็นผู้บริหารดูแล 1 ท่าน และมีนักศึกษาและอาจารย์ ทั้งมหาวิทยาลัยเข้ามาใช้ยืม-คืนหนังสือต่างๆ ในห้องสมุด

เฉลย

3.1 คลินิกรักษาโรคทั่วไปแห่งหนึ่งจะประกอบไปด้วยแพทย์ประจำ 1 นาย พยาบาลอย่างน้อย 1 คน แต่อาจจะมากกว่านี้ได้ โดยจะมีห้องสำหรับตรวจรักษาโรคทั่วไป 1 ห้อง และห้องสำหรับรักษาโรคเฉพาะทางอีก 1 ห้อง และมียารักษาโรคต่างๆ เป็นจำนวนมาก

ขั้นตอนที่ 1 แยกคลาส

1.1 คลาส แพทย์/ เภสัชกร

1.2 คลาส คลินิกรักษาโรค

1.3 คลาส พยาบาล

1.4 คลาสห้องรักษาโรคทั่วไป

1.5 คลาสห้องรักษาโรคเฉพาะทาง

1.6 คลาสยา