

Classification Abstraction

วัตถุประสงค์

- อธิบายหลักการในการกำหนด Problem Domain ได้
- สามารถหา Object ใน Domain
- สามารถใช้หลักการของ Classification Abstraction ในการสร้าง Class จาก Object ที่กำหนดให้ได้
- สามารถบอกหลักการ Encapsulation และ Information Hiding ของ Class ได้

การกำหนด Problem Domain

- Problem Domain คือ ขอบเขตของสิ่งที่กำลังจะพิจารณา
- Problem Domain สามารถกำหนดได้จากการสอบถามความต้องการ (Requirement) จากผู้ใช้ระบบงานนั้น ๆ
- ถึงแม้ว่าในขั้นตอนการวิเคราะห์ระบบจะยังไม่สามารถกำหนด Problem Domain ที่ชัดเจน ก็ขอให้กำหนด Big Picture ของ Problem Domain ให้ได้ออกมาก่อน
- ถ้ากำหนดภาพรวมของ Problem Domain ไม่ได้ ก็จะมี class จำนวนมากมายมหาศาลเกิดขึ้น จนยากที่จะออกแบบ software ที่ดีได้

การค้นหา Object ใน Domain

- ทำได้โดยการค้นหาคำนามทั้งหมดที่มีใน Problem Domain
- แยกแยะว่าสิ่งใดคือ Object และสิ่งใดคือ Attribute
- ระวัง!! คำนามบางคำก็เป็น Object แต่บางคำเป็น Attribute

ความยากในการทำ Classification

- โดยทั่วไป เราแยกแยะวัตถุโดยใช้เงื่อนไขบางอย่างมา
กำหนดเป็นกรอบ
 - แล้ว ขอบเขตของแขน ขา อยู่ตรงไหน
 - เราแยกเสียงเครื่องดนตรีในเพลงได้ไหม
 - ในเอกสารหนึ่งชิ้น เราใช้อะไรแยกแยะประเภทของเอกสาร
 -

ตัวอย่างการทำ Classification : Biology

- กลางทศวรรษ 1700, Carolus Lirmaeus นักพฤกษศาสตร์ชาว Swedish แบ่งสิ่งมีชีวิตออกตาม genus และ species
- ในยุค 1800 Darwin เสนอทฤษฎีวิวัฒนาการ มีการเลือกโดยธรรมชาติ สิ่งมีชีวิตที่พบเห็นในทุกวันนี้ วิวัฒนาการ มาจากสิ่งมีชีวิตในอดีต
- ในยุค 1900 เราแบ่งสิ่งมีชีวิตออกเป็น kingdom, phylum, subphylum, class, order, family, genus, และ species

ตัวอย่างการทำ Classification : Biology

- ปัจจุบัน เราแยกด้วย DNA



ตัวอย่างการทำ Classification : chemistry

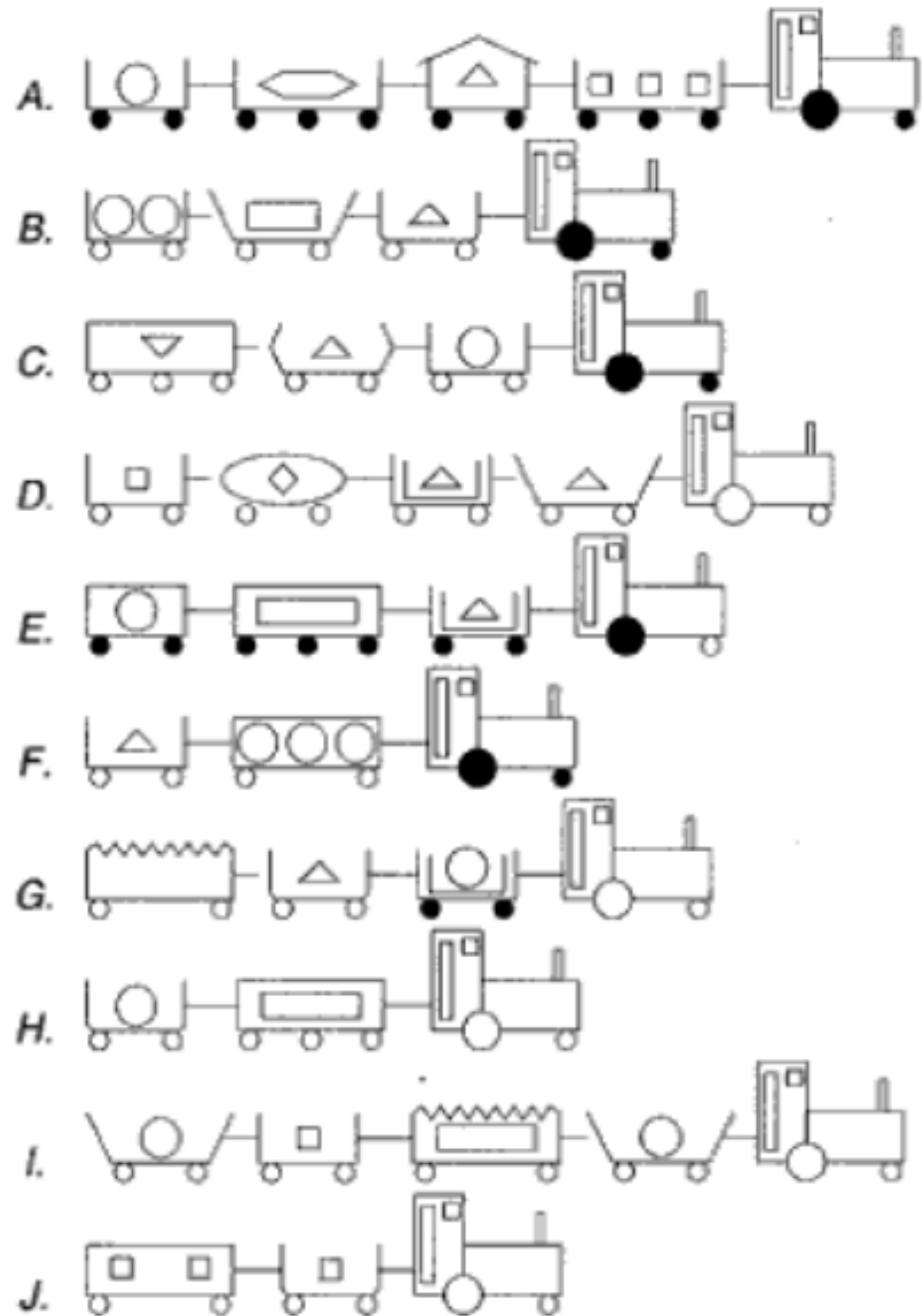
- ยุคโบราณ เชื่อกันว่า โลกประกอบด้วย ดิน น้ำ ลม ไฟ
- mid-1600s, Robert Boyle นำเสนอว่า elements (ธาตุ) คือ abstractions ของสารเคมี
- 1789, Lavoisier ตีพิมพ์ตารางธาตุ มี 23 ชนิด (และพบในภายหลังว่า บางชนิดไม่ใช่ธาตุ)
- 1869, Mendeleev นำเสนอ periodic law ที่สามารถระบุธาตุได้และทำนายคุณสมบัติของธาตุที่ยังไม่ค้นพบ

ตัวอย่างการทำ Classification : chemistry

- early 1900s, ค้นพบว่าธาตุ มีองค์ประกอบเหมือนกัน แตกกันที่ atomic weights นำไปสู่แนวคิดเรื่อง isotopes

❖ ไม่มีวิธีการใดที่จะทำ Abstraction ได้ 100%

○ จงทำ classification
ของรถไฟในภาพ



ประเภทของ Object

- สิ่งที่มีตัวตนสามารถจับต้องได้ (Tangible Objects)
 - คน สุนัข รถยนต์
 - อื่น ๆ (นักศึกษา ยกตัวอย่าง)
- สิ่งที่ไม่มีตัวตนและไม่สามารถจับต้องได้ (Intangible Objects)
 - บทบาท เหตุการณ์ ปฏิสัมพันธ์
 - อื่น ๆ (นักศึกษา ยกตัวอย่าง)

ตัวอย่างที่ 1

- “หนังสือเล่มหนึ่ง ปกสีเหลือง ภายในประกอบด้วย เนื้อหาเกี่ยวกับ Object Orientation หนังสือเล่มนี้มี จำนวน 50 หน้า”

ตัวอย่างที่ 1 : การวิเคราะห์ (1)

- ขั้นตอนที่ 1 : หาคำนาม
 - หนังสือเล่มหนึ่ง
 - ปกสีเหลือง
 - เนื้อหาเกี่ยวกับ Object Orientation
 - หน้า

ตัวอย่างที่ 1 : การวิเคราะห์ (2)

- ขั้นตอนที่ 2 : แยกประเภทของคำนาม
 - หนังสือเล่มหนึ่ง : Object
 - ปกสีเหลือง : Attribute
 - เนื้อหาเกี่ยวกับ Object Orientation : Attribute
 - หน้า : Attribute

การระบุ Class และ Object

- ในบาง Problem Domain อาจจะได้ทั้ง Class และ Object ในเวลาเดียวกัน ดังนั้น จำเป็นต้องระบุให้แน่ชัดว่าสิ่งใดคือ Class และสิ่งใดคือ Object

ตัวอย่างที่ 2

- “พยาบาลชื่อ ปราณี ฉีดยาป้องกันโรคบาดทะยักให้แก่
คนไข้ชื่อ กิตติ”

ตัวอย่างที่ 2 : การวิเคราะห์ (1)

- ขั้นตอนที่ 1 : หาคำนาม
 - พยาบาลชื่อ ปรานี
 - ยาป้องกันโรคบาดทะยัก
 - คนไข้ชื่อกิตติ

ตัวอย่างที่ 2 : การวิเคราะห์ (2)

- ขั้นตอนที่ 2 : แยกประเภทของคำนาม
 - พยาบาล : Class
 - ปราณี : Object หนึ่งใน Class พยาบาล
 - คนไข้ : Class
 - กิตติ : Object หนึ่งใน Class คนไข้
 - ยาป้องกันโรคบาดทะยัก : Class / Object

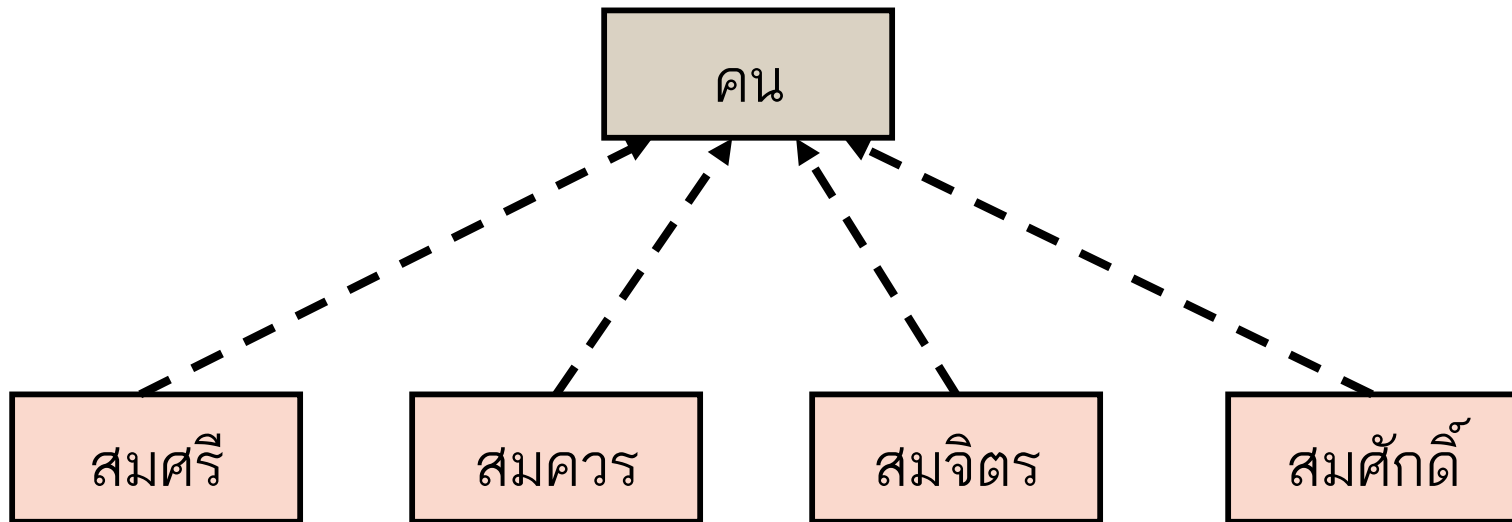
ตัวอย่างที่ 2 : การวิเคราะห์ (3)

- ยาป้องกันโรคบาดทะยัก : Class
 - เพราะเป็นการบอกอย่างกว้าง ๆ ว่าเป็นยาป้องกันบาดทะยัก แต่ไม่ได้ระบุยี่ห้อยา
- ยาป้องกันโรคบาดทะยัก : Object
 - ถือเป็น Object หนึ่งใน Class ยา

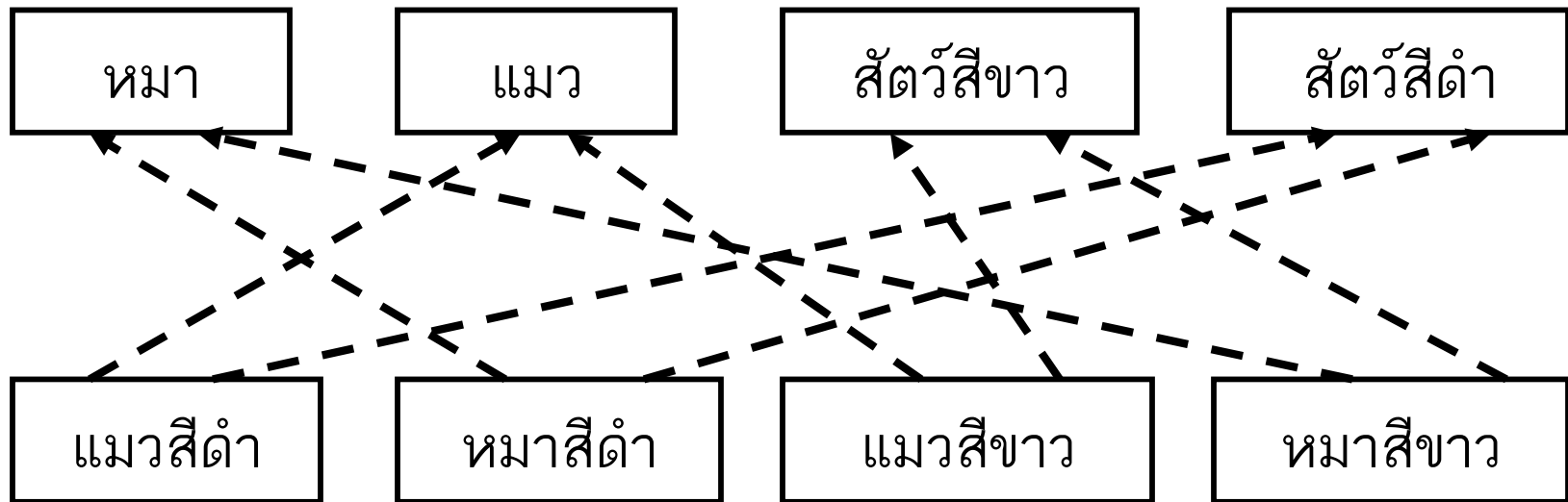
Classification Abstraction

- เป็นกระบวนการในการค้นหาว่ามี Object ใดบ้างใน Problem Domain
- เป็นการจำแนกแยกแยะว่า Object แต่ละตัวจัดอยู่ใน Class ใดบ้าง
- การทำ Classification Abstraction แสดงด้วยสัญลักษณ์ ลูกศรประทีที่ลากจาก Objects ไปยัง Class

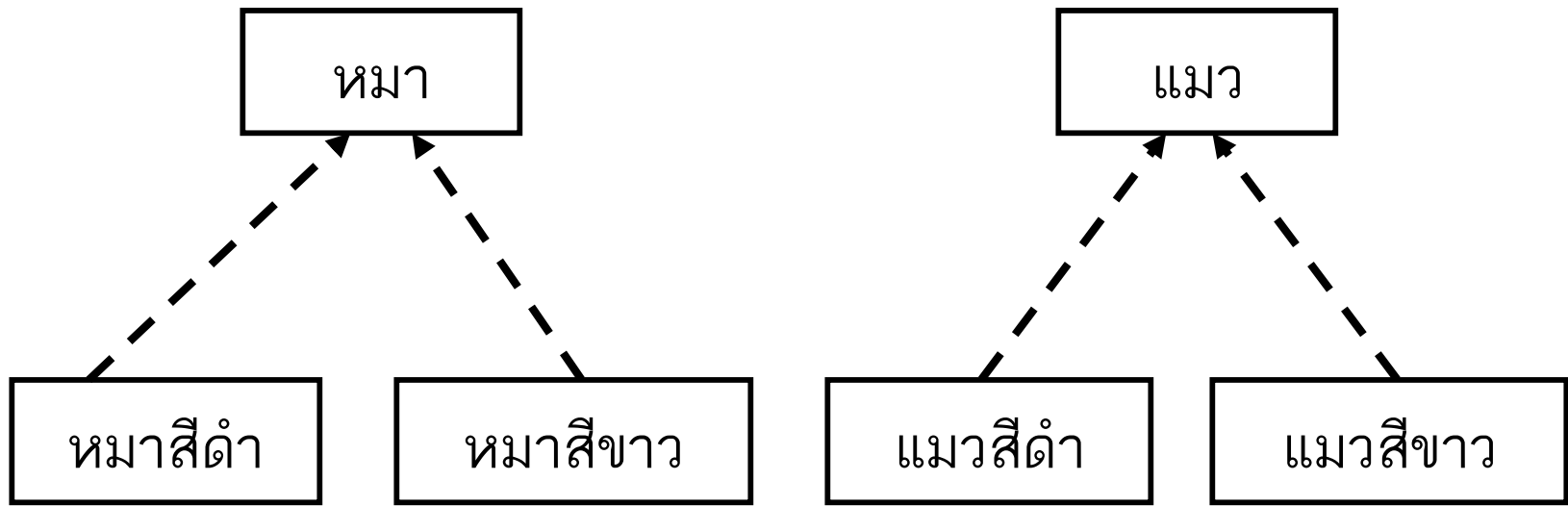
Classification ของ Class คน



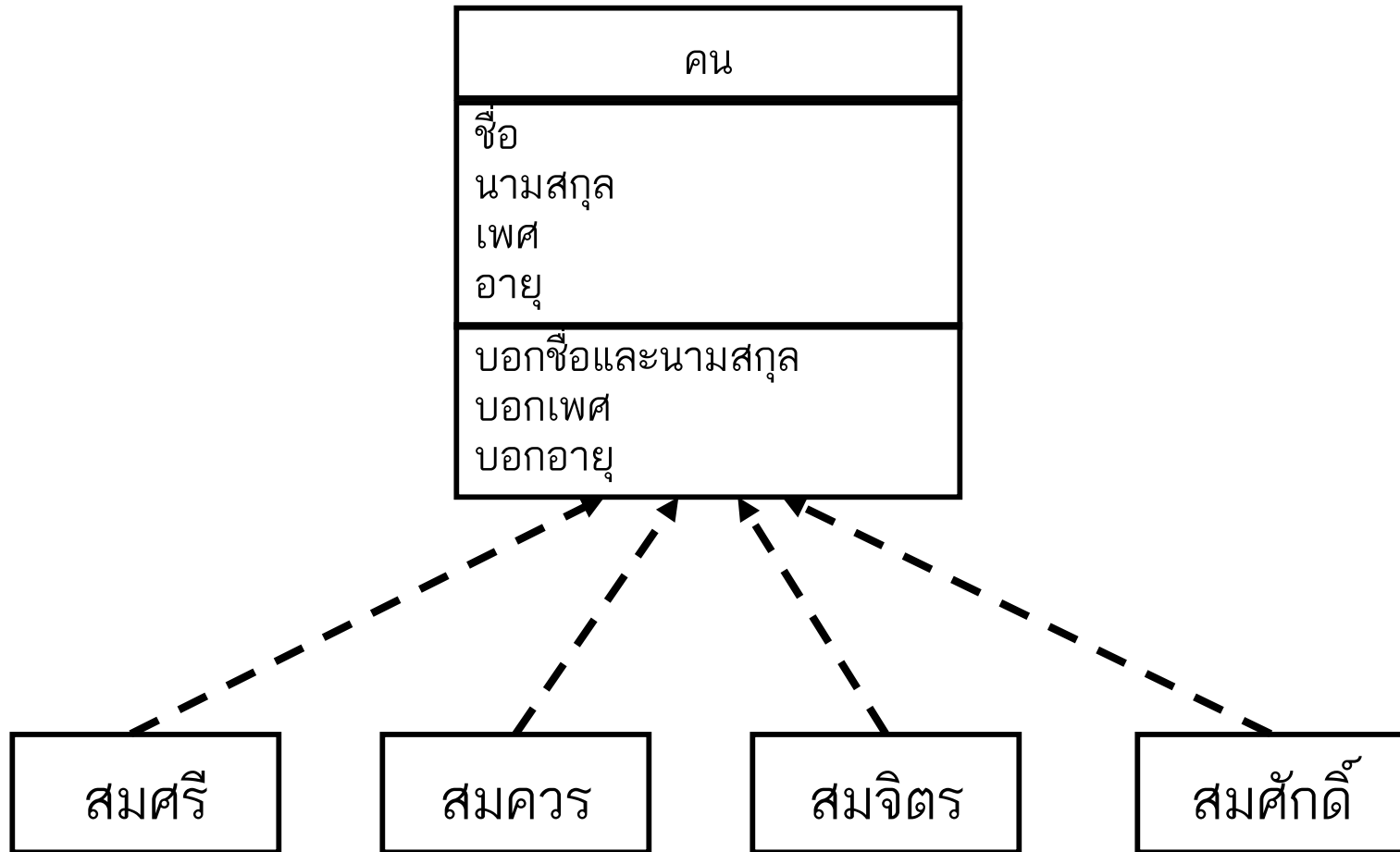
Classification ของ หมา แมว สัตว์สีขาว และสัตว์สีดำ



ปรับปรุงการทำ Classification ของหมา และแมว



Classification ของ Class คนที่มีการใส่ รายละเอียด



Encapsulation

- Encapsulation เปรียบเสมือนกับการนำเปลือกมาครอบ Attributes และ Function ของ Class เอาไว้
- ลักษณะของเปลือก
 - เปลือกใส จะสามารถมองได้จากภายนอก
 - เปลือกทึบ จะไม่สามารถเห็นได้จากภายนอก
- ภาพของ Class ที่มองเห็นได้จากภายนอกนั้นเรียกว่า Outside View

Outside View ของ Class คน

คน
อายุ
บอกอายุ ()

Class คน

คน
บอกอายุ ()

Outside View ของ Class
คน

การขอดู Attribute โดยการใช้ Method

ในภาษา C++ จะเรียก Method ว่า Member Function



การเขียน Function ที่ถูกใช้งาน

สมศักดิ์ : คน
อายุ
บอกอายุ ()

สมศรี : คน
อายุ
บอกอายุ ()

ส่วนของ Function
จะถูกเขียนด้วยตัวเอียง

Information Hiding

- คือ คุณสมบัติของ Object ที่เมื่อต้องการเข้าถึง Attribute บางตัวของ Object นั้น จะต้องอาศัยการทำผ่าน Function ที่สามารถมองเห็นและเรียกใช้ได้เท่านั้น
 - Attribute ถูกหุ้มด้วยเปลือกหีบ
 - Function ถูกหุ้มด้วยเปลือกใส

ประเภทของ Attribute และ Methods

- จำแนกตามความสามารถในการเห็นและเข้าถึง Attributes และ Methods เหล่านั้น (Visibility) ได้ 3 ประเภท
 - Private Attributes and Functions
 - Protected Attributes and Functions
 - Public Attributes and Functions

Private Attributes and Methods

- เป็น Attributes และ Method ที่ไม่สามารถเห็นได้เลยจากภายนอก
- การเข้าถึง Attribute เหล่านี้ได้ต้องผ่านทาง Method ที่มีไว้เท่านั้น
- จะใช้เครื่องหมาย (-) กำกับไว้หน้า Private Attribute และ Private Method
 - เช่น อายุของคน

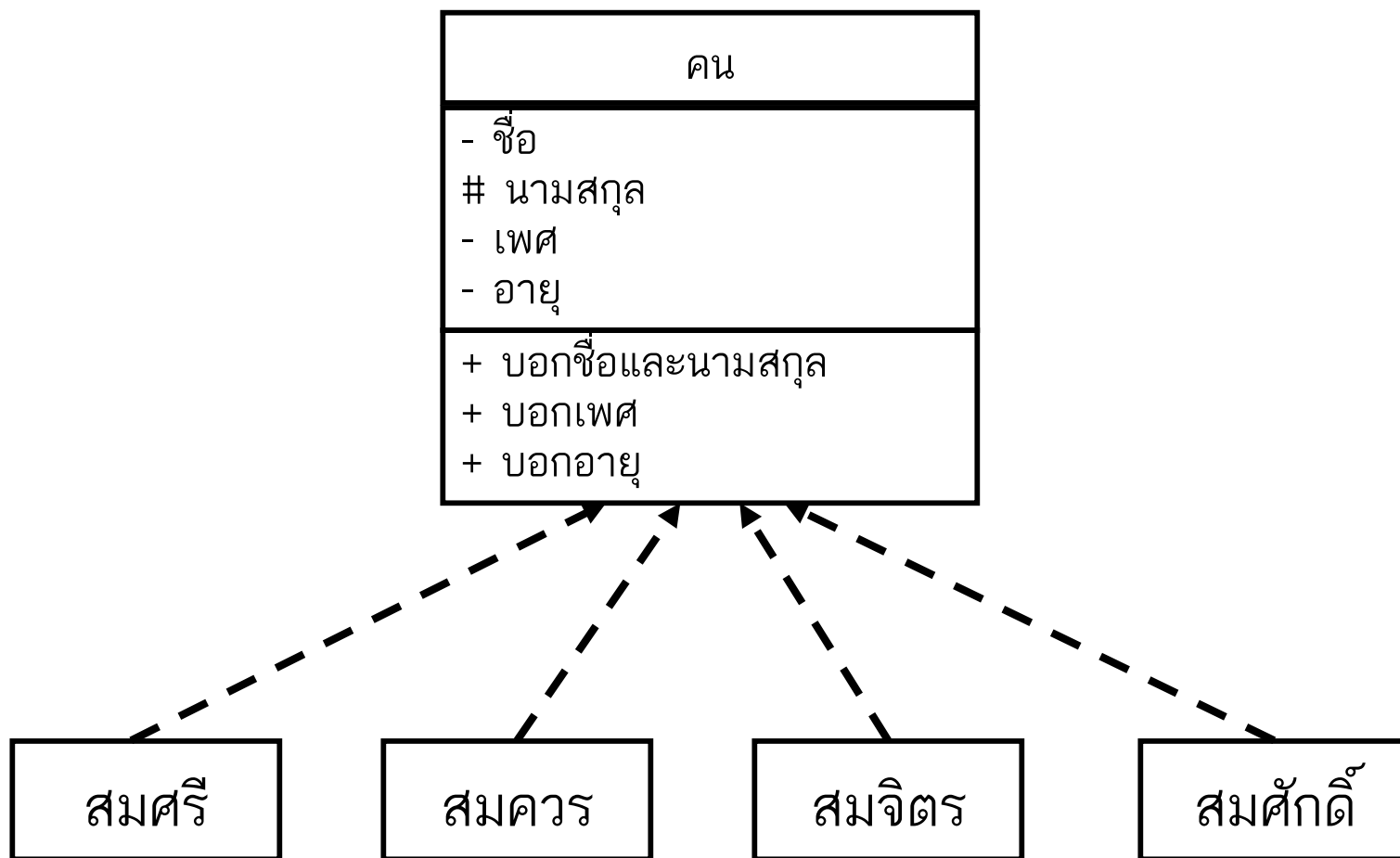
Protected Attributes and Functions

- เป็น Attributes และ Methods ที่ไม่สามารถเห็นได้จากภายนอกแต่เป็นส่วนที่สามารถส่งต่อให้ Inherited Class ได้เท่านั้น
- จะใช้เครื่องหมาย (#) กำกับไว้หน้า Protected Attribute และ Protected Methods
- เช่น ลักษณะทางกรรมพันธุ์ที่ลูกสืบทอดมาจากพ่อแม่

Public Attributes and Functions

- เป็น Attributes และ Methods ที่สามารถมองเห็นได้ และสามารถเรียกใช้ได้โดยตรงจากภายนอก
- จะใช้เครื่องหมาย (+) กำกับไว้หน้า Public Attribute และ Public Method
 - เช่น สีมม สีผิว

Classification ของ Class คน



References

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