

Class Diagram

เรื่องที่จะศึกษา

- Class
- Relationship
- สัญลักษณ์ที่ใช้ใน class diagram
- หลักการสร้าง class diagram

จุดประสงค์

- สามารถจำลองภาพของ Class ที่มีใน Problem Domain ในรูปของ Class Diagram ได้
- สามารถแสดงความสัมพันธ์ หรือ Abstraction ระหว่าง Class ใน Problem Domain ได้

Class Diagram คืออะไร

- Class diagram คือ แผนภาพที่ใช้แสดง class และความสัมพันธ์ (relationship) ระหว่าง class
- ความสัมพันธ์ที่แสดงเป็นความสัมพันธ์เชิงสถิตย (static) ไม่ใช่ความสัมพันธ์ที่เกิดขึ้นเนื่องจากกิจกรรม (dynamic)

Relationship คืออะไร

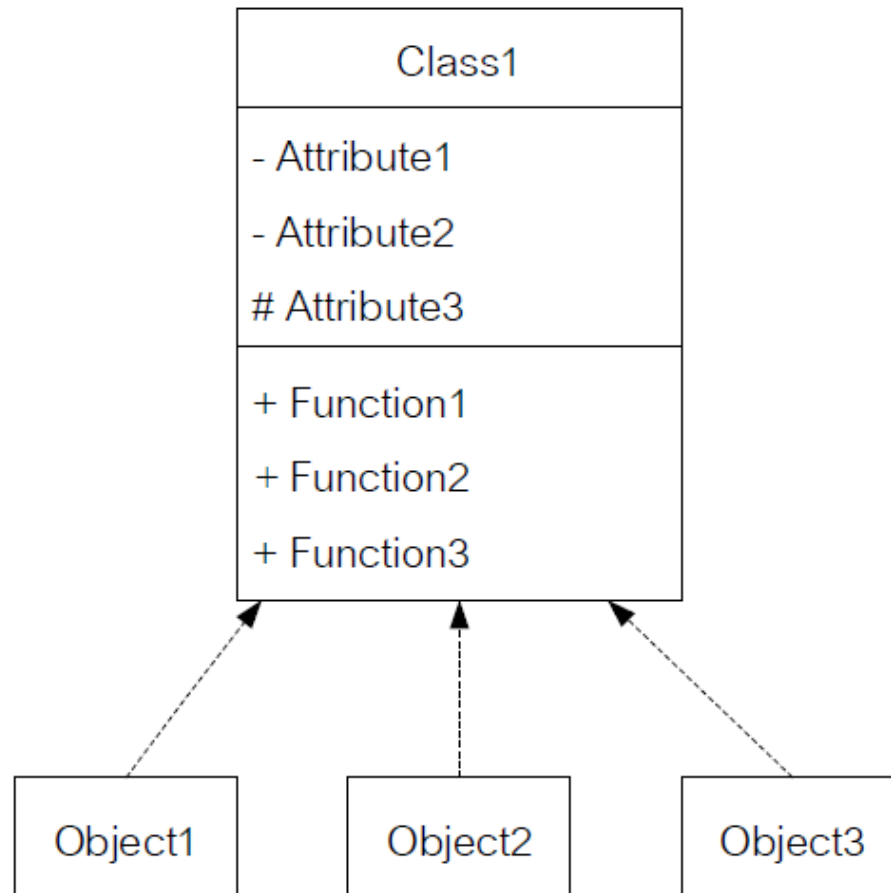
Relationship คือความสัมพันธ์ระหว่าง Objects ต่างๆ แบ่งได้ 2 รูปแบบ

- **Static relationship** ความสัมพันธ์เชิงสถิต เป็นความสัมพันธ์ที่คงทน มักจะไม่แปรสภาพไปตามเวลา
 - เจ้าของบัญชีเป็นเจ้าของบัญชีเงินฝาก
- **Dynamic relationship** ความสัมพันธ์เชิงกิจกรรม เป็นความสัมพันธ์ที่ไม่คงทน มักจะแปรสภาพไปตามเวลา
 - เจ้าของบัญชีฝากเงินเข้าบัญชีเงินฝาก
 - เจ้าของบัญชีถอนเงินจากบัญชีเงินฝาก
 - เจ้าของบัญชีปรับปรุงยอดบัญชีเงินฝาก

สัญลักษณ์ Class

Man
- Name # Surname - Age
+ Tell_Name + Tell_Age

สัญลักษณ์ที่ใช้แทน Class และ Relationship แบบต่างๆ



Visibility ของ Classes

- กำหนดระดับความเป็นส่วนตัวของ **members** ของ classes
 - member function หรือ methods
 - member variable หรือ attributes
- ใช้ควบคุมการมองเห็นหรือเรียกใช้ได้โดยตรงจากภายนอก

สัญลักษณ์ Visibility

- Private แทนด้วย -
- Protected แทนด้วย #
- Public แทนด้วย +

Visibility แบบ Public

- มองเห็นและเรียกใช้ได้โดยตรงจากภายนอก
- เข้าไปเปลี่ยนค่า อ่านค่า หรือเรียกใช้งานได้ทันทีโดยอิสระจากภายนอก
- มักใช้กับ Functions มากกว่า Attributes
- จะใช้เครื่องหมาย (+) กำกับไว้ข้างหน้า

Visibility แบบ Private

- ไม่สามารถเห็นได้จากภายนอก จะเห็นได้ภายในเฉพาะตัว class เองเท่านั้น
- หากภายนอกต้องการแก้ไข หรืออ่านค่า ทำได้วิธีเดียวคือ ผ่าน Function ที่เกี่ยวข้อง
- โดยทั่วไปมักใช้กับ Attributes มากกว่า Functions
- จะใช้เครื่องหมาย (-) กำกับไว้ข้างหน้า

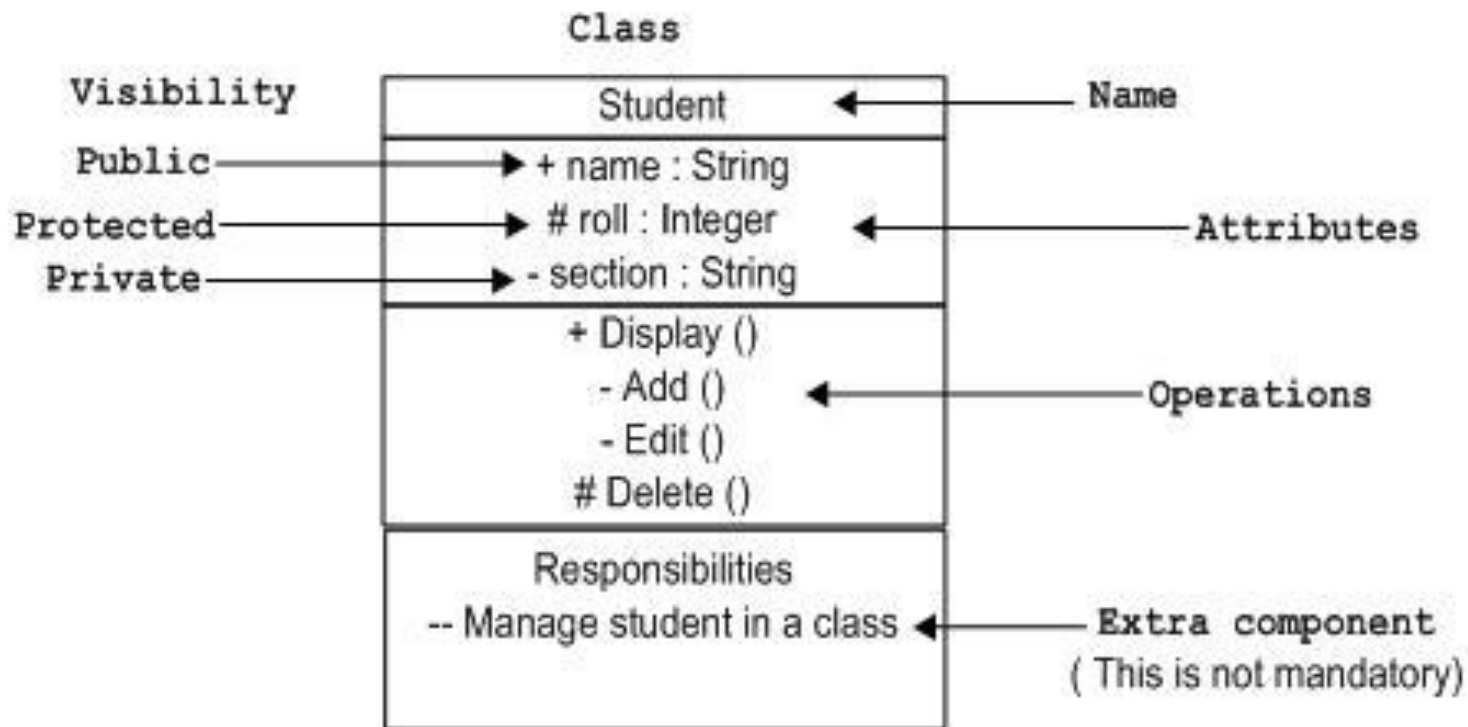
Visibility แบบ Protected

- สงวนไว้สำหรับการทำ Inheritance โดยเฉพาะ
- โดยปกติจะเป็นของ Superclass
- เมื่อทำ inheritance แล้ว Attributes และ Functions เหล่านี้จะใช้ได้ทั้ง Private หรือ Protect ซึ่งขึ้นอยู่กับภาษาที่ใช้
- จะใช้เครื่องหมาย (#) กำกับไว้หน้า

ตัวอย่าง Class คน


คน
<ul style="list-style-type: none">- เลขบัตรประจำตัวประชาชน- ชื่อ# นามสกุล- อายุ- หมู่เลือด+ สีผม
<ul style="list-style-type: none">+ บอกเลขบัตรประชาชน+ บอกชื่อ+ บอกนามสกุล+ บอกอายุ+ บอกหมู่เลือด

สัญลักษณ์ที่ใช้แทน Class และ Relationship



 Association

 Inheritance

 Realize/ Implimentation

 Dependancy

 Aggregation

 Composition

หลักการในการสร้าง Class Diagram

- การจำลอง Objects ในโลกแห่งความเป็นจริง มาเป็นคลาสและ relationship
 - ควรมีอยู่จริงในระบบ
 - ครบถ้วน ไม่ขาดหาย
 - ไม่มากเกินไปจนจำเป็น

หลักการในการสร้าง Class Diagram

- กำหนดกรอบของ Problem Domain ให้ชัดเจน
 - เขียน use case diagram ของ problem domain ที่กำหนดไว้
 - พิจารณาว่าในแต่ละ use case มี object ใดอยู่บ้าง
 - ทำให้ครบทุก use case

หลักการในการสร้าง Class Diagram...

- พิจารณาหา Tangible objects ให้ครบทุกตัว
 - ในกรณีที่มีหลายตัวใน problem domain เดียวกัน ให้หาตัวแทน object นั้น
- พิจารณาหา Intangible objects ให้ครบทุกตัว
 - ในกรณีที่มีหลายตัวใน problem domain เดียวกัน ให้หาตัวแทน object นั้น

หลักการในการสร้าง Class Diagram...

- ใช้ Classification Abstraction เพื่อแยกแยะและสร้าง class จาก object ที่มีอยู่
- หา attributes และ functions ที่มีอยู่ใน class นั้น ๆ
- วาด class ที่ได้ลงใน class diagram

หลักการในการสร้าง Class Diagram...

- หา Aggregation Abstraction โดยพิจารณา class ที่มีความสัมพันธ์แบบเป็นส่วนหนึ่งหรือประกอบด้วยกับ class อื่น ๆ
- ระบุชนิด aggregation ได้แก่
 - One to One หรือ
 - Many to One
- ใส่ Cardinality ให้ถูกต้อง

หลักการในการสร้าง Class Diagram...

- ใช้ Generalization มาพิจารณา class ต่าง ๆ ใน class diagram
- หากมีความสัมพันธ์แบบ generalization หรือ specialization ให้เพิ่มลงไปใน class diagram
- อาจมีการสร้าง class ใหม่เพื่อเป็น generalized class ได้

หลักการในการสร้าง Class Diagram...

- ใช้ Association มาพิจารณา class ต่าง ๆ ใน class diagram
- เพิ่มเติมสัญลักษณ์ของ Association ลงใน Class Diagram
- พิจารณาประเภทของความสัมพันธ์และ Cardinality ให้ถูกต้อง

หลักการในการสร้าง Class Diagram...

- พิจารณาว่าทุก class ควรมีความสัมพันธ์ แบบใดแบบหนึ่งกับ class อื่น
- หากพบ class ที่ยังไม่มีความสัมพันธ์กับ class อื่น อาจมีสาเหตุจาก
 - class นั้นเป็น class ที่เกินความจำเป็น ซึ่งไม่จำเป็นต้องมีในระบบได้
 - หรือต้องเพิ่มเติม class อื่นที่มีความสัมพันธ์กับ class ดังกล่าวเข้าไป

ตัวอย่างการสร้าง Class Diagram

Problem Domain ที่กำหนดคือ

“ในคณะวิทยาศาสตร์ของสถาบันการศึกษาแห่งหนึ่ง มีบุคลากรหลายประเภทด้วยกัน ได้แก่ อาจารย์ นักศึกษา และเจ้าหน้าที่ โดยที่อาจารย์แต่ละท่านมีหน้าที่ในการสอน วิชาใดวิชาหนึ่งหรือมากกว่า 1 วิชาก็ได้ และนักศึกษาก็มี หน้าที่ในการศึกษาวิชาวิชาหนึ่ง หรือมากกว่า 1 วิชาก็ได้ ใน เวลาเดียวกันเจ้าหน้าที่ของภาควิชา คือ เจ้าหน้าที่ประจำ ห้องทดลองต่าง ๆ โดยกำหนดว่าใน 1 ห้องทดลองจะต้องมี เจ้าหน้าที่ 1 คนเสมอ”

หา use case จาก problem domain

- use case ของระบบคือ
 - การเรียนการสอน
 - การใช้ห้องทดลอง
 - การดูแลห้องทดลอง

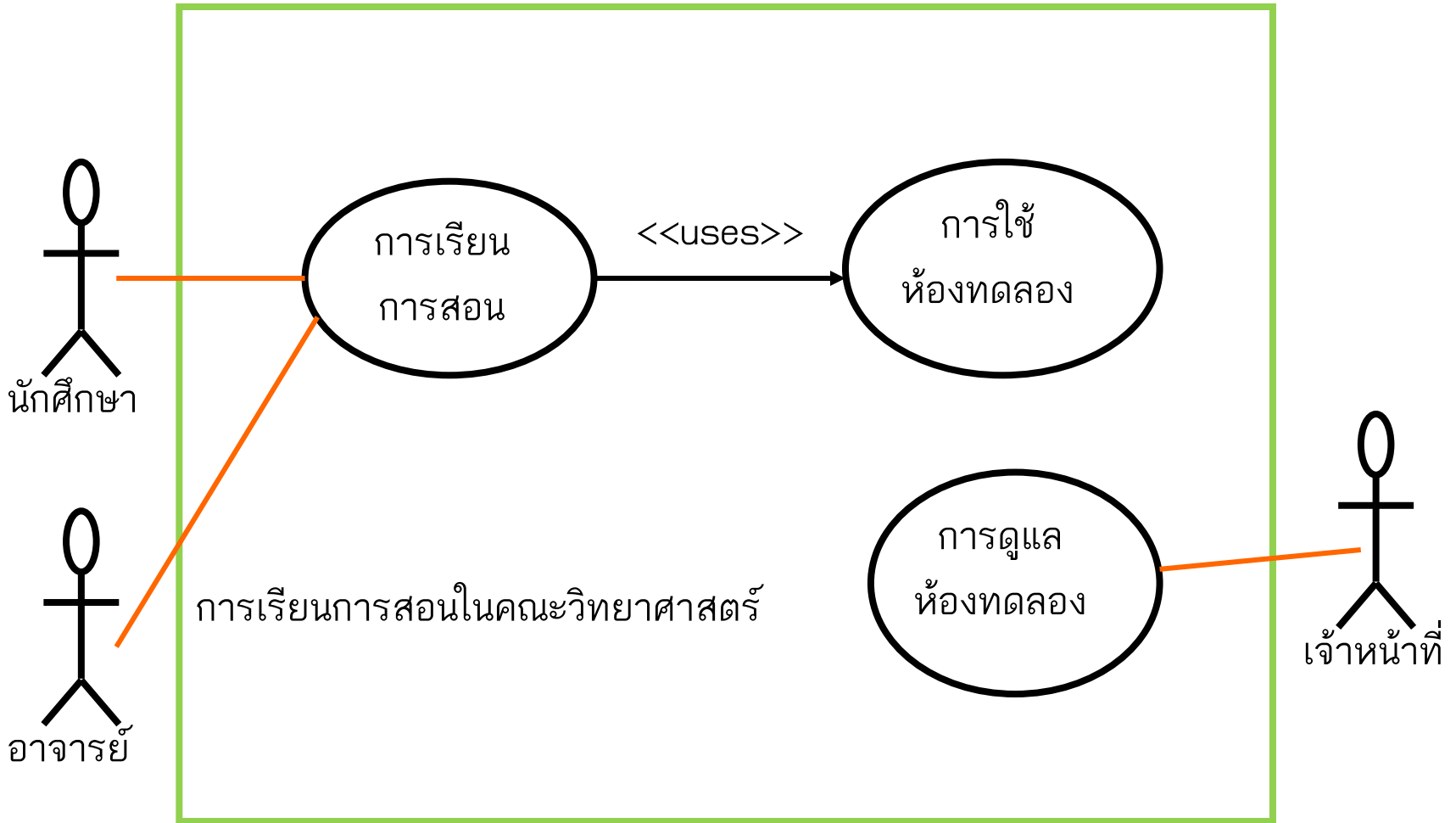
หา object/class จาก use case

- use case การเรียนการสอน
 - นักเรียน อาจารย์
 - ห้องเรียน วิชาเรียน ชั่วโมงเรียน
- use case การใช้ห้องทดลอง
 - นักเรียน อาจารย์
 - ห้องทดลอง
- use case การดูแลห้องทดลอง
 - เจ้าหน้าที่
 - ห้องทดลอง

หา actor จาก use case

- สรุป actors ที่มีจาก use case คือ
 - นักเรียน
 - อาจารย์
 - เจ้าหน้าที่

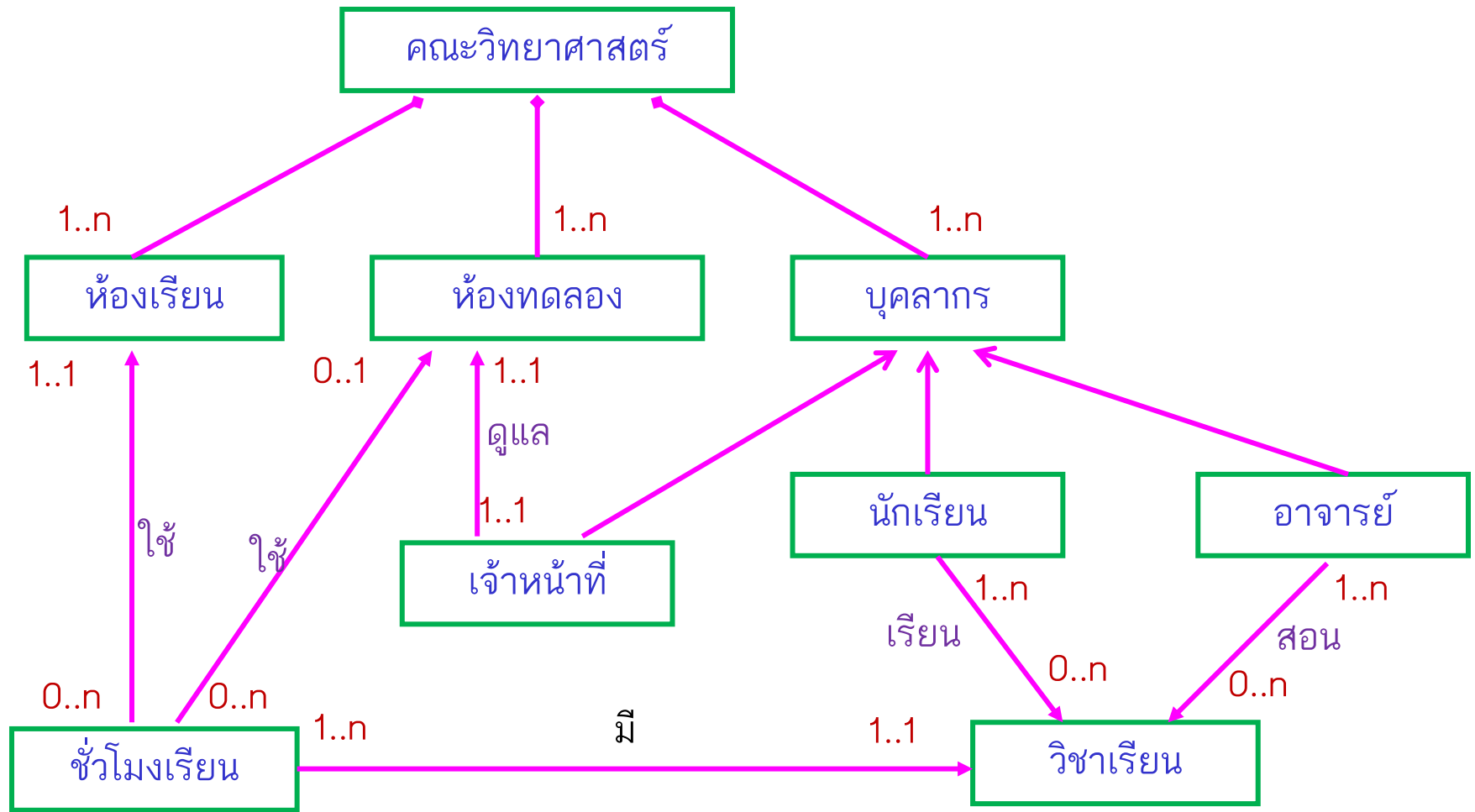
เขียน Use Case Diagram



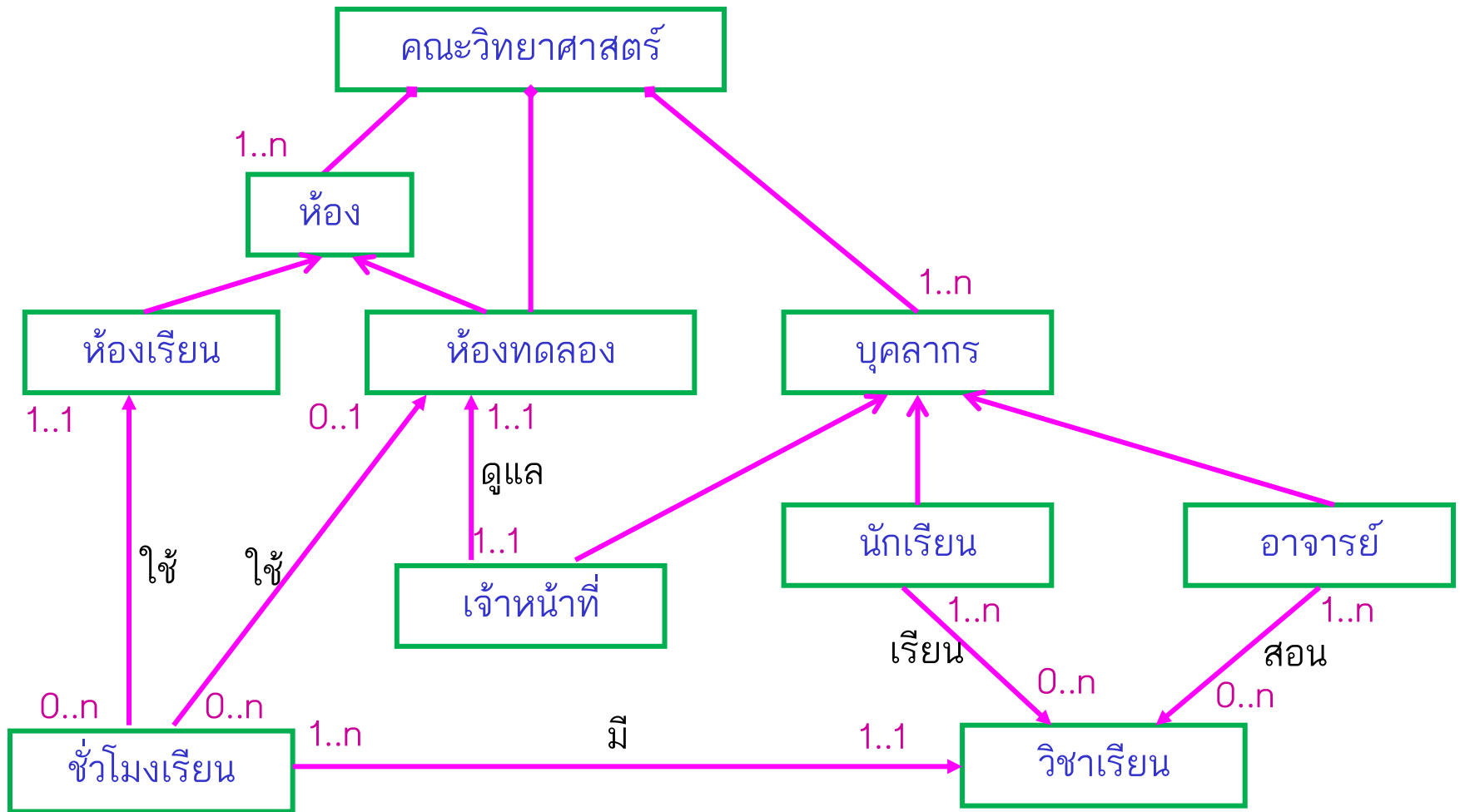
object/class ทั้งระบบ

- นักเรียน
- อาจารย์
- เจ้าหน้าที่
- ห้องเรียน
- วิชาเรียน
- ชั่วโมงเรียน
- ห้องทดลอง

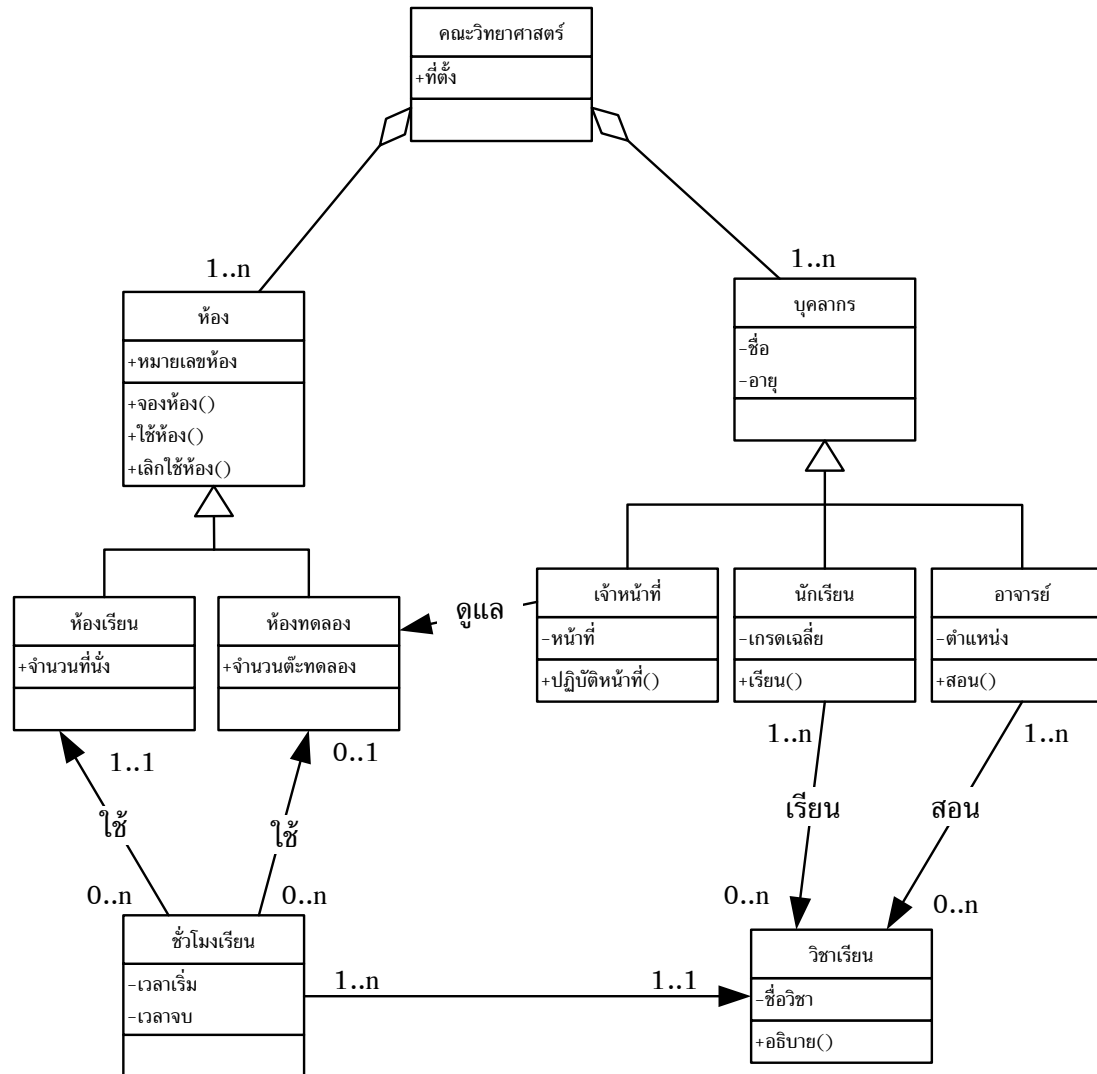
เขียน Class Diagram เบื้องต้น



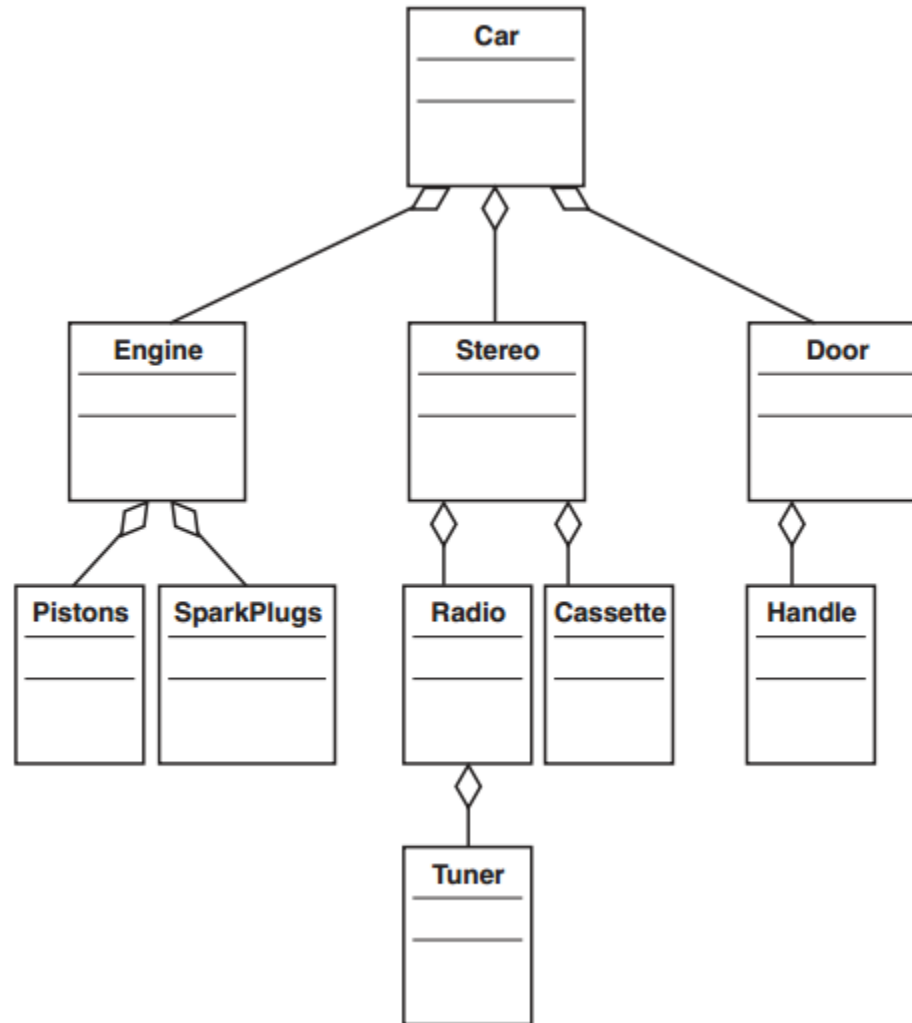
ปรับเปลี่ยน Class Diagram ให้สมบูรณ์ขึ้น



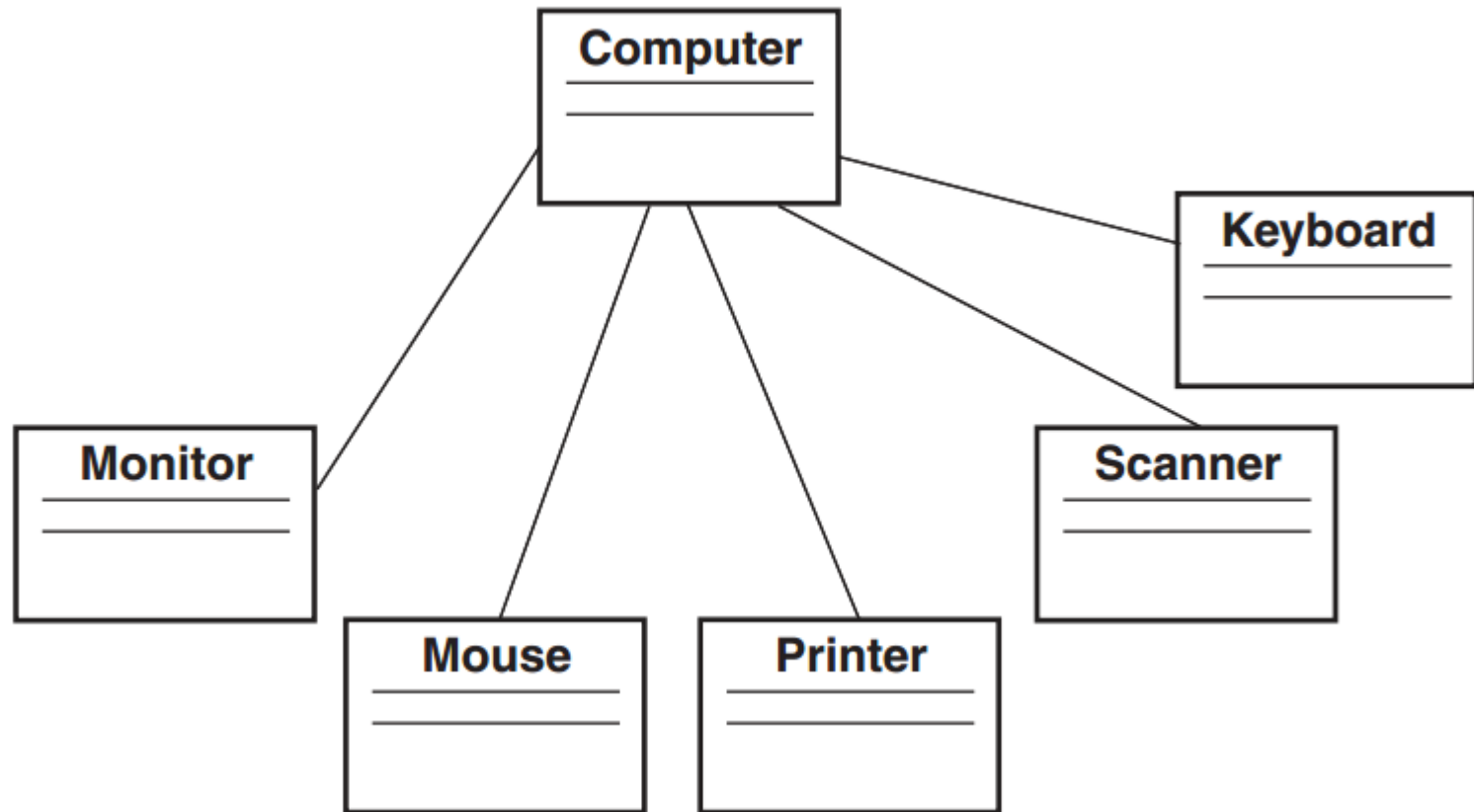
ปรับปรุง Class Diagram ให้สมบูรณ์



จงทำให้สมบูรณ์



จงทำให้สมบูรณ์



ตัวอย่าง class diagram

“คำถาม...”