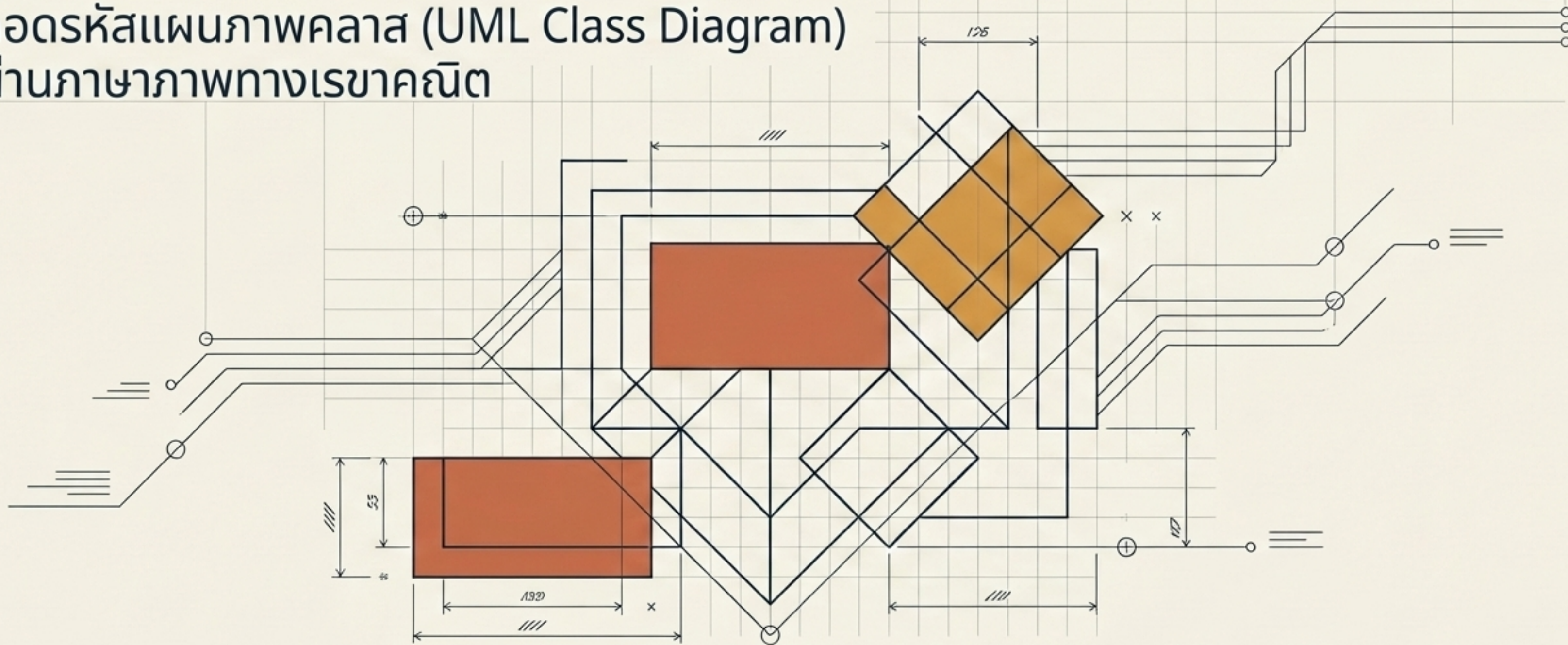


แผนภาพคลาส (Class Diagram): สถาปัตยกรรมของการออกแบบเชิงวัตถุ

ถอดรหัสแผนภาพคลาส (UML Class Diagram)
ผ่านภาษาภาพทางเรขาคณิต

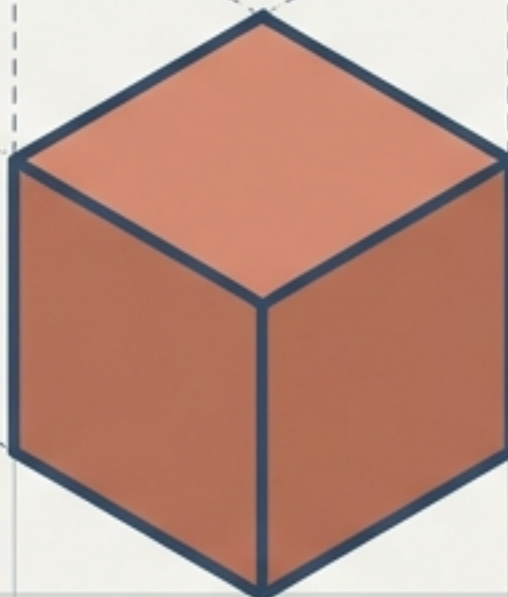


นามธรรมสู่รูปธรรม (Abstract to Object)

แม่พิมพ์สี่เหลี่ยม / Class



คลาส (Class) ไม่สามารถดำเนินกิจกรรมใดๆ
ได้ด้วยตัวเอง แต่มันคือ 'นามธรรม' (Abstract)
ที่ทำหน้าที่เป็นเสมือนแม่พิมพ์ (Template)



การหล่อหลอมแม่พิมพ์เรขาคณิตให้กลายเป็นสิ่งที่
มีอยู่จริงในระบบเรียกว่า Object หรือ Instance
(เช่น คลาส 'สินค้า' ถูกสร้างเป็น 'สินค้าชิ้นที่ 1')

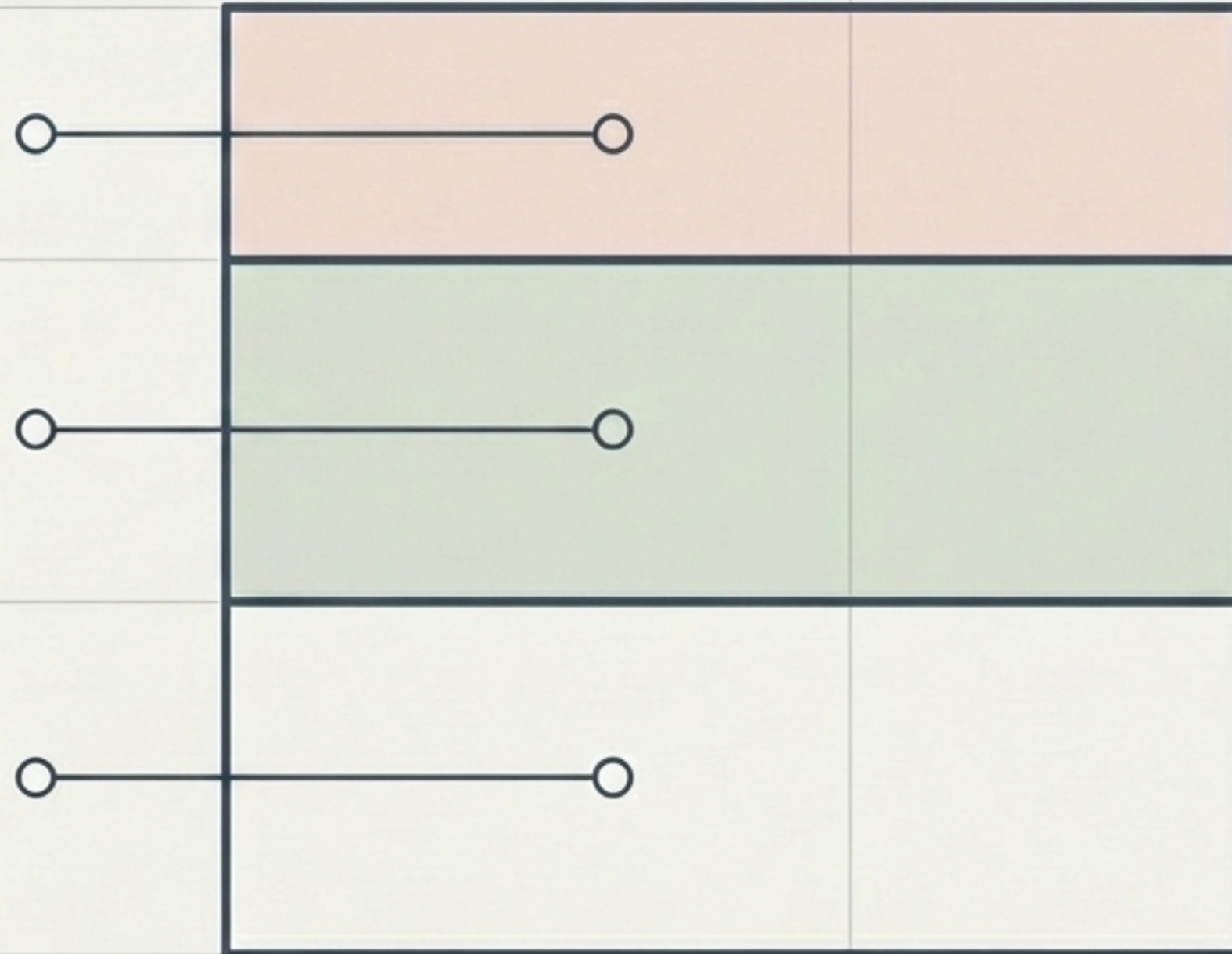
วัตถุ / Object

องค์ประกอบของแผนภาพคลาส (Anatomy of a Class)

ส่วนที่ 1: ชื่อคลาส (Class Name)
หลักการ: ใช้ 'คำนาม' และขึ้นต้นด้วย 'ตัวพิมพ์ใหญ่' (เช่น Human, Goldmember)

ส่วนที่ 2: แอตทริบิวต์ (Attributes)
หลักการ: คุณลักษณะหรือข้อมูลที่จัดเก็บ ใช้ 'คำนาม' และเริ่มด้วย 'ตัวพิมพ์เล็ก' (เช่น age, studentName)

ส่วนที่ 3: โอเปอเรชัน (Operations)
หลักการ: พฤติกรรมหรือหน้าที่การทำงาน ใช้ 'คำกริยา', เริ่มด้วย 'ตัวพิมพ์เล็ก' และตามด้วย () (เช่น setName(), checkpasswd())



การสร้างแผนภาพคลาส (Class Diagram Construction)

โครงสร้างของแอตทริบิวต์ (Attributes)

[Visibility] **[attributeName]** : **[Attribute Type]**

ตัวอย่าง: - **studentName** : string

ความหมาย: แอตทริบิวต์ส่วนตัว ชื่อการศึกษา ชนิดข้อมูลเป็นสายอักขระ

โครงสร้างของโอเปอเรชัน (Operations)

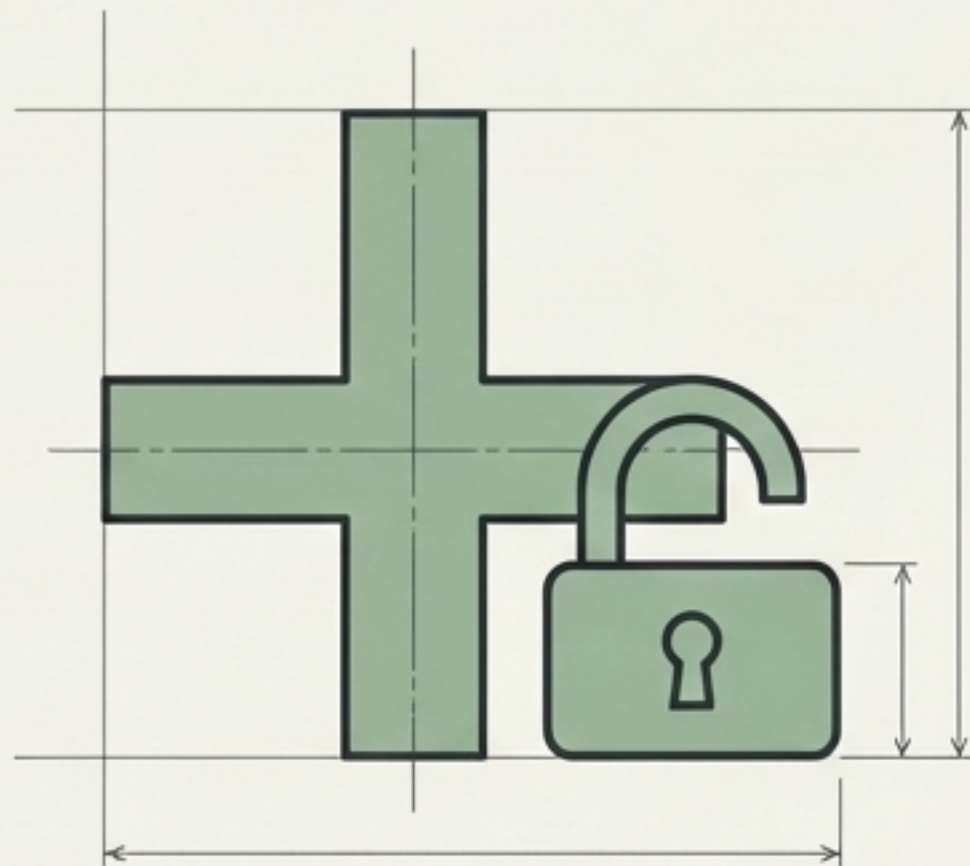
[Visibility] **[OperationName]** (**[Parameter]** : **[Type]**) : **[Return Type]**

ตัวอย่าง: + **setStudentId** (**size** : integer)

ความหมาย: โอเปอเรชันสาธารณะ รับค่าตัวแปร size ชนิดจำนวนเต็ม

ข้อสังเกต: Abstract Data Type (ADT) สามารถเป็นคลาสอื่นได้ เช่น จอภาพ, เม้าส์

สัญลักษณ์ที่ใช้แทนระดับการมองเห็นและการเข้าถึง (Visibility And Accessibility)



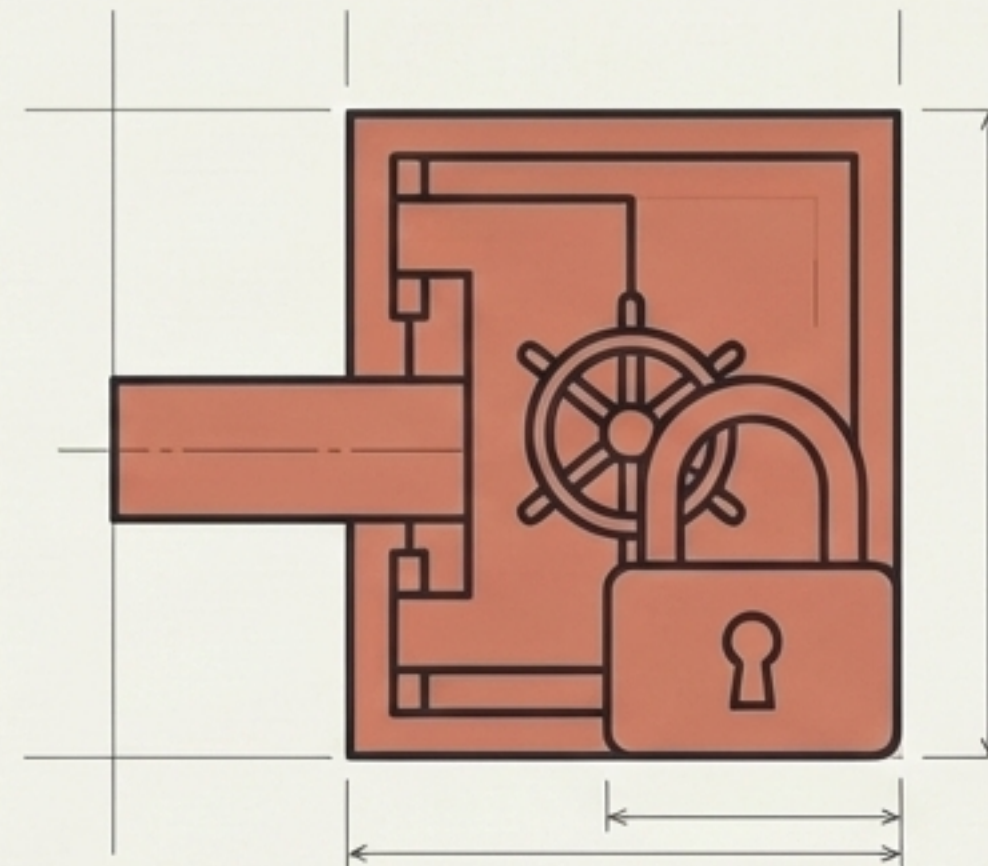
+ สาธารณะ (Public)

อนุญาตให้คลาสอื่นมองเห็นและใช้งานได้ทันที (มักใช้กับ Operations)



ป้องกัน (Protected)

อนุญาตให้คลาสอื่นมองเห็น แต่ไม่อนุญาตให้ใช้งานโดยตรง

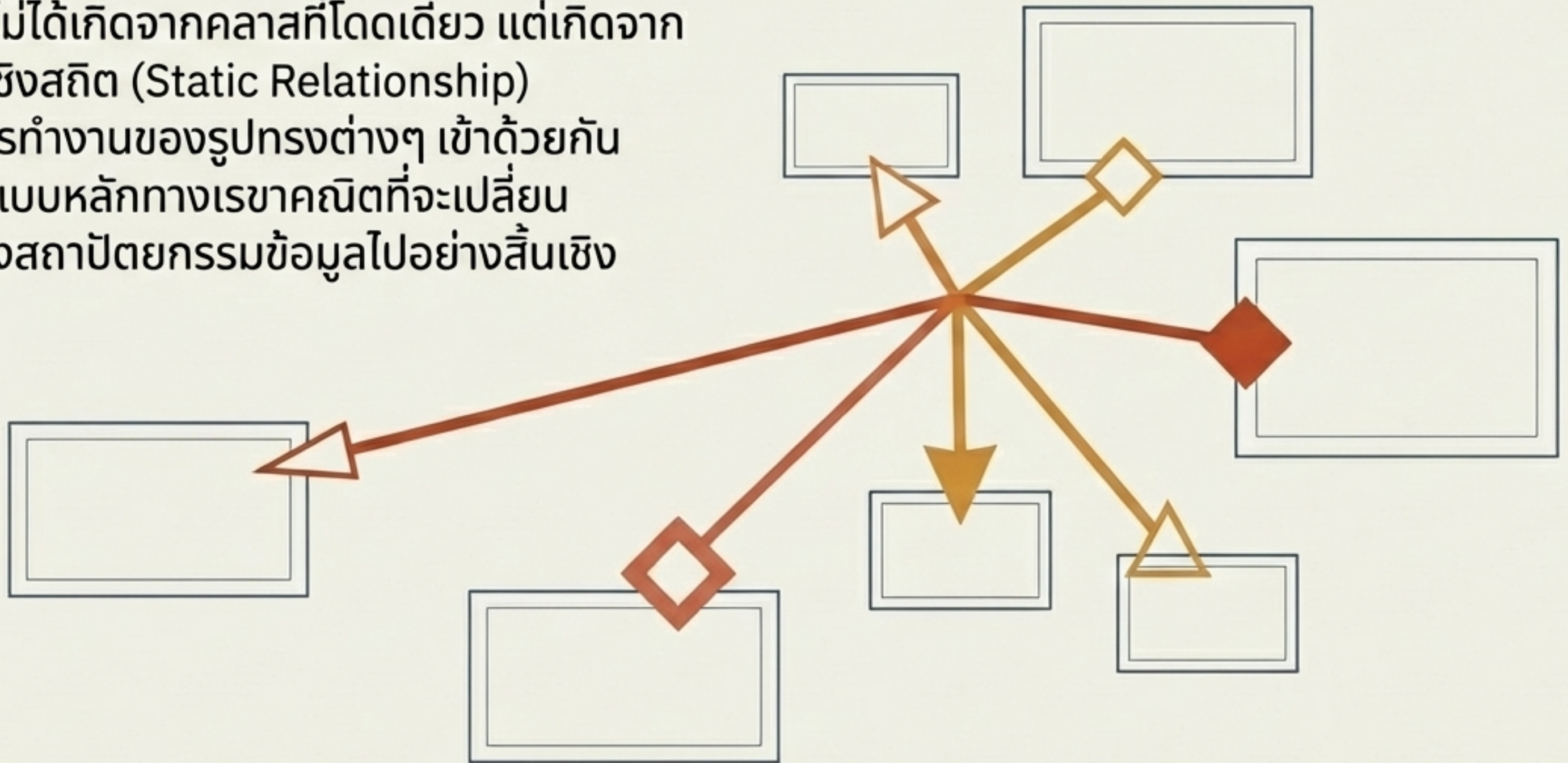


- ส่วนตัว (Private)

ซ่อนตัวอย่างสมบูรณ์ คลาสอื่นมองไม่เห็นและใช้งานไม่ได้ (มักใช้กับ Attributes เพื่อความปลอดภัย)

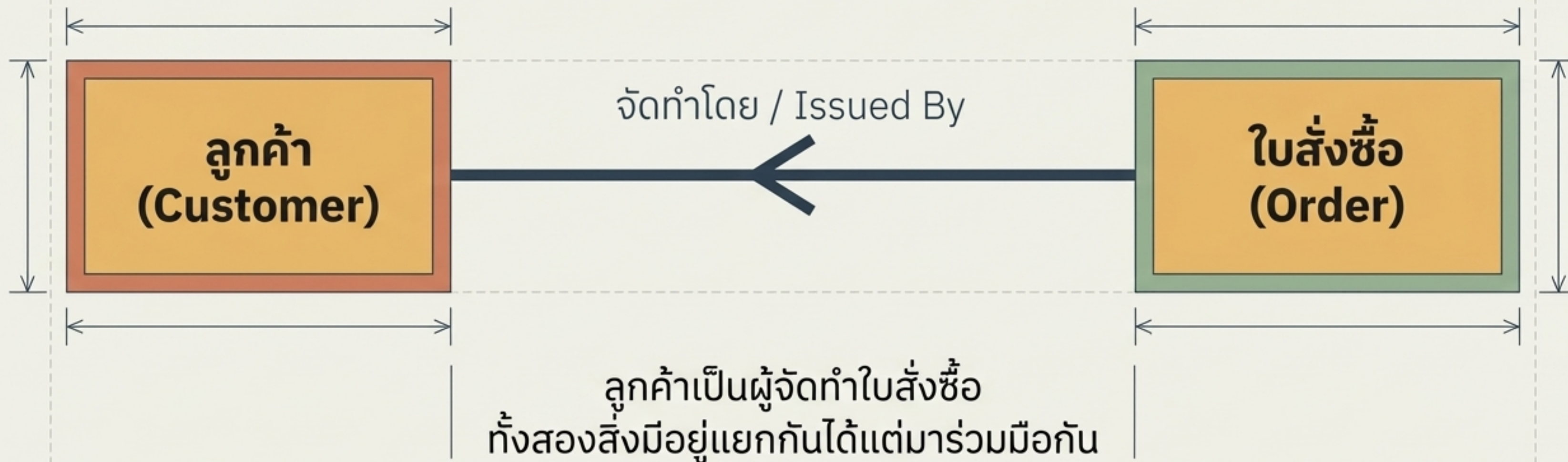
ความสัมพันธ์ระหว่างคลาส (The Geometry of Connections)

ระบบที่แท้จริงไม่ได้เกิดจากคลาสที่โดดเดี่ยว แต่เกิดจาก
ความสัมพันธ์เชิงสถิต (Static Relationship)
การประสานการทำงานของรูปทรงต่างๆ เข้าด้วยกัน
แบ่งเป็น 4 รูปแบบหลักทางเรขาคณิตที่จะเปลี่ยน
ความหมายของสถาปัตยกรรมข้อมูลไปอย่างสิ้นเชิง



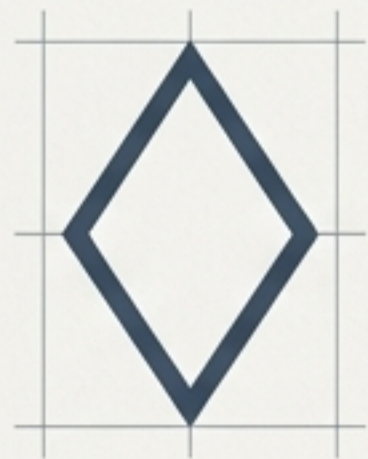
เส้นตรงแห่งความเกี่ยวข้องกัน (Association)

ความสัมพันธ์ระดับสหาย คลาสทั้งสองมีความสำคัญเท่าเทียมกัน
ไม่มีใครเป็นเจ้าของใคร เป็นโครงสร้างพื้นฐานที่สุดในระบบ

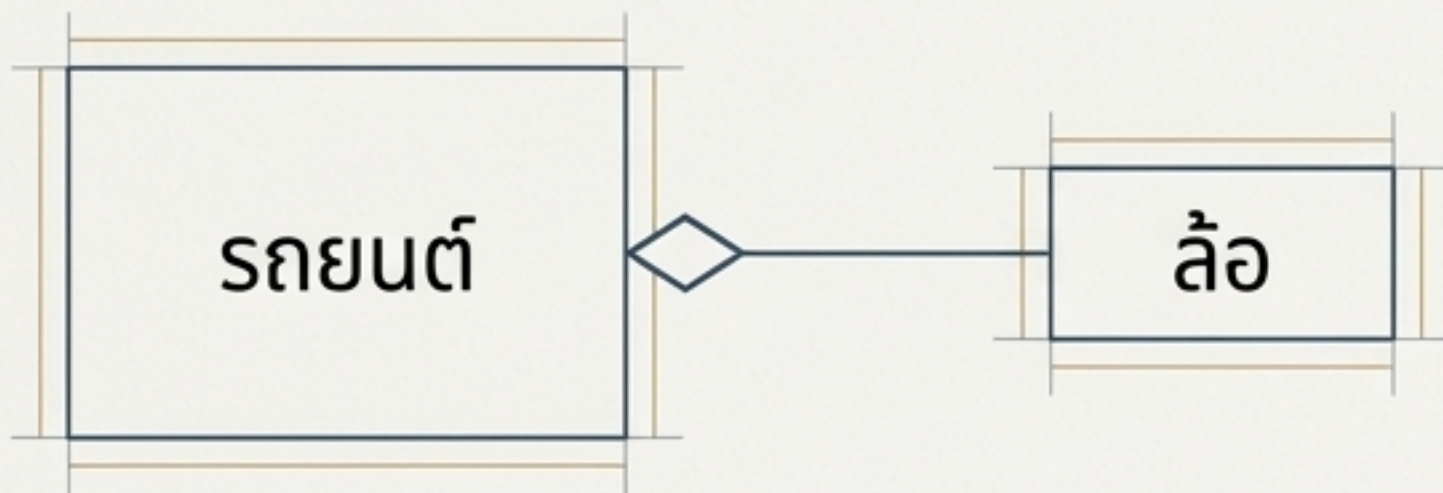


เมทริกซ์สี่เหลี่ยมข้าวหลามตัด (Part-Whole Relationships)

การเป็นส่วนหนึ่ง (Aggregation)

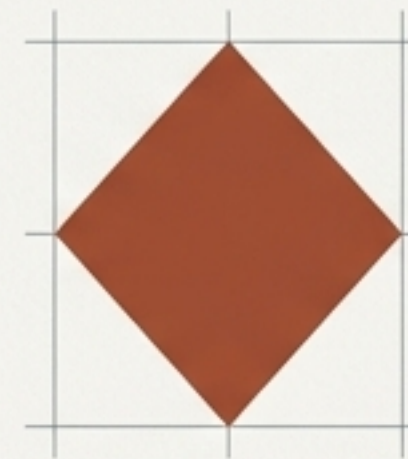


◇ สี่เหลี่ยมข้าวหลามตัดโปร่ง
แยกส่วนได้ - เมื่อลบ Whole Class
ออกไป Part Class ยังคงอยู่ได้

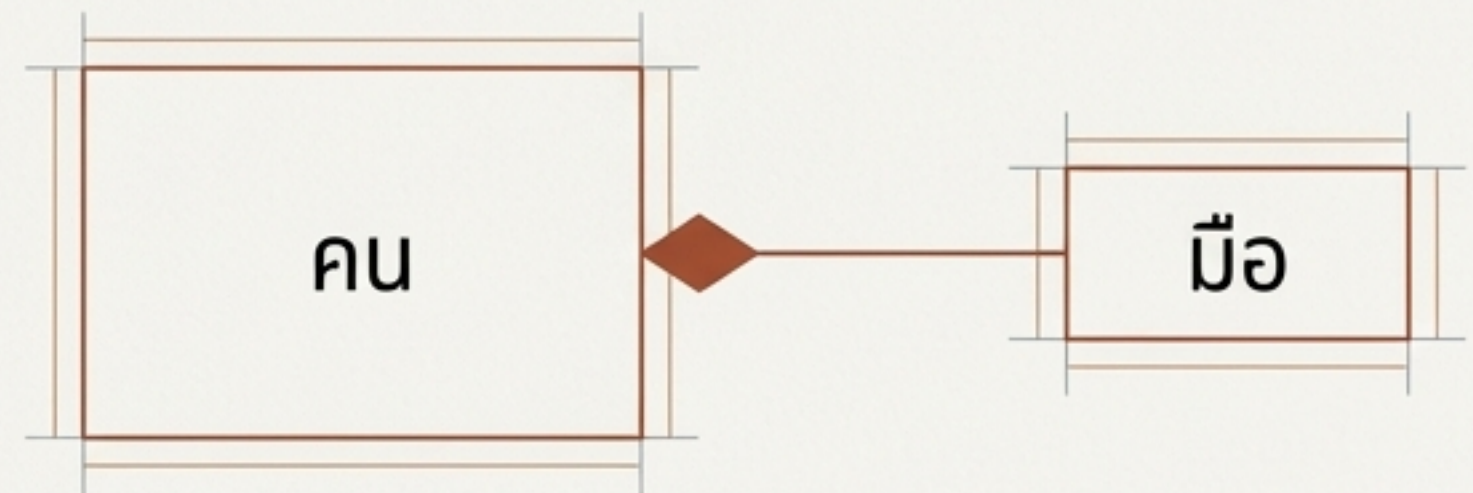


ตัวอย่าง: รถยนต์ ◇--- ล้อ. ถ้ารถพัง ล้อยังเอาไปใช้ได้ต่อได้

การเป็นองค์ประกอบ (Composition)



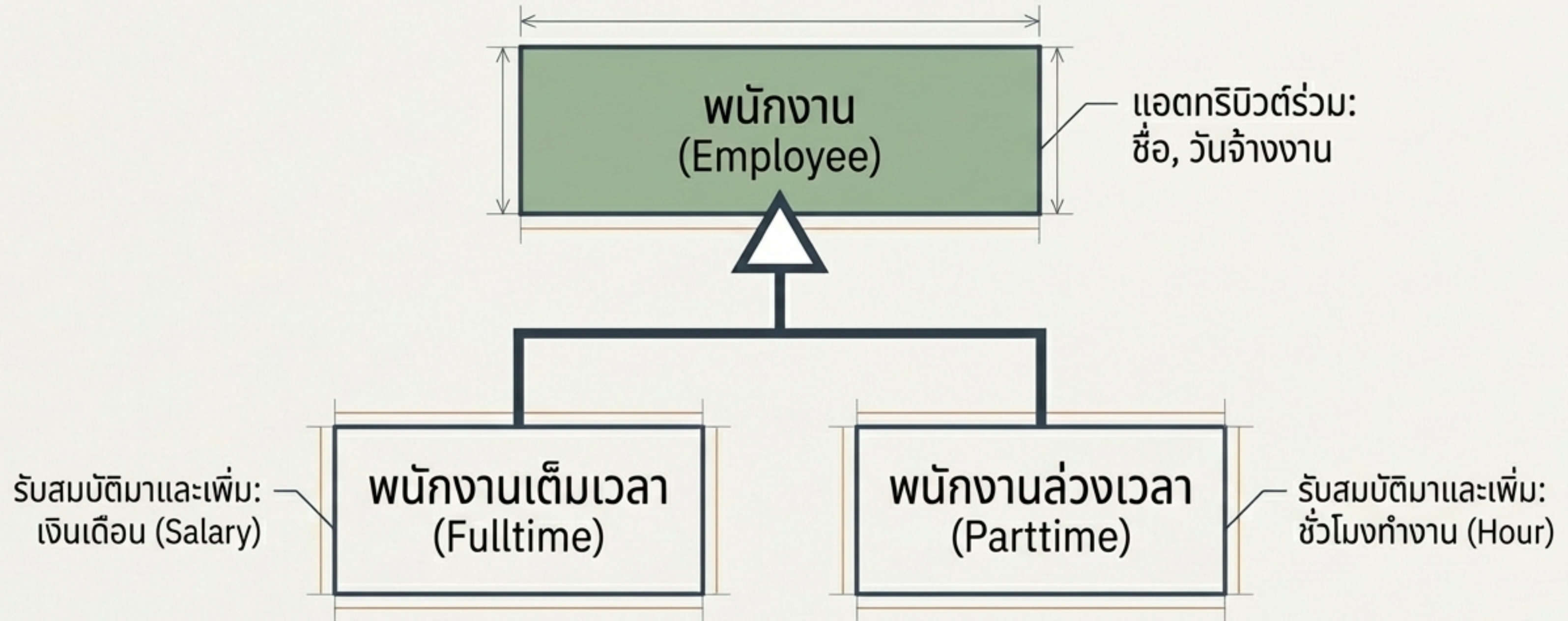
◆ สี่เหลี่ยมข้าวหลามตัดทึบ
ขาดกันไม่ได้ - ความสัมพันธ์เข้มงวด
เมื่อลบ Whole Class ออกไป
Part Class จะหายไปด้วย



ตัวอย่าง: คน ◆--- มือ. ถ้าไม่มีคน มือก็ไม่สามารถอยู่ได้ด้วยตัวเอง

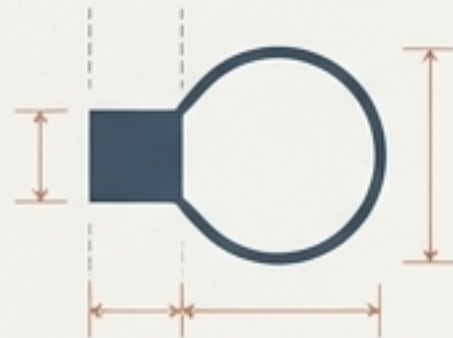
สามเหลี่ยมแห่งสายเลือด (Generalization)

การถ่ายทอดคุณสมบัติ (Inheritance): โครงสร้างลำดับชั้นเพื่อลดความซ้ำซ้อนของสถาปัตยกรรม

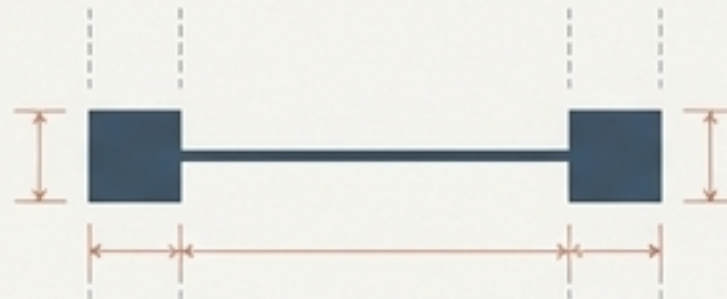


แกนตัวเลขแห่งความสัมพันธ์ (Degree & Multiplicity)

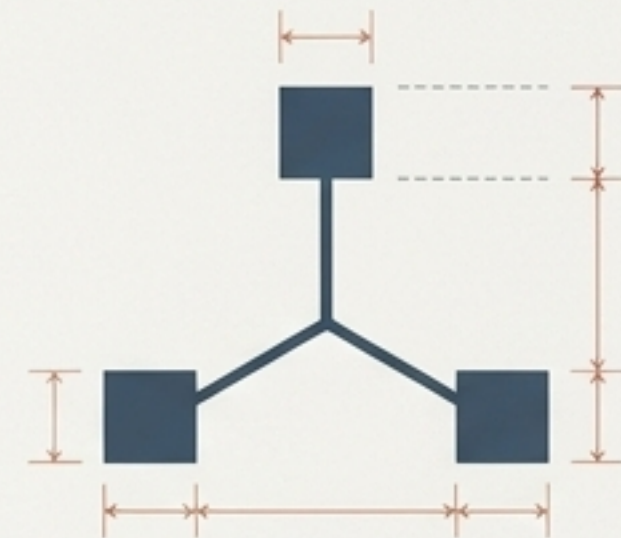
ดีกรีของความสัมพันธ์ (Degree)



Unary: เส้นม้วนกลับหาตัวเอง (1 คลาส)



Binary: เส้นเชื่อมระหว่าง 2 จุด (2 คลาส)



Ternary: เส้น 3 แฉกเชื่อมตรงกลาง (3 คลาส)

จำนวนวัตถุที่สัมพันธ์กัน (Multiplicity)

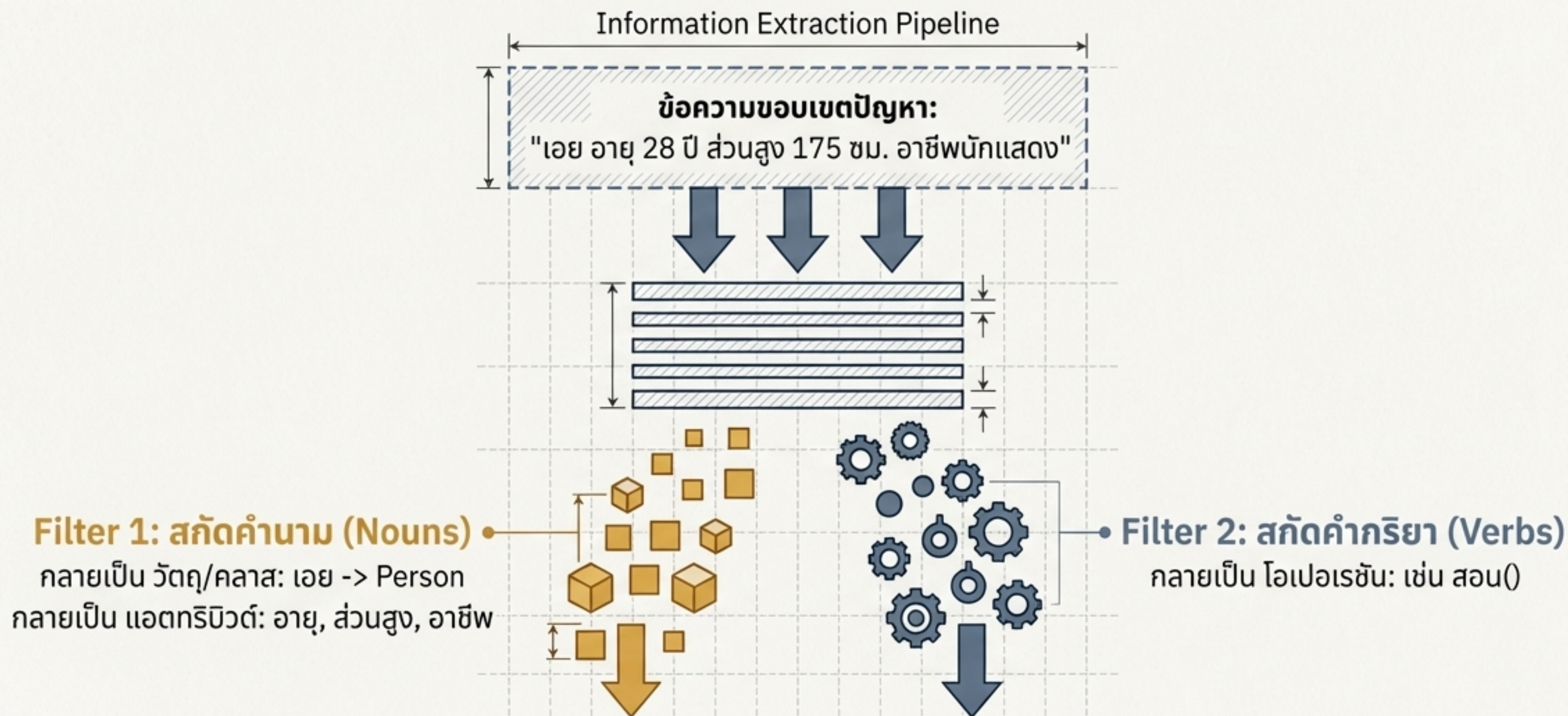
การระบุปริมาณที่ปลายเส้นฝั่งตรงข้าม



อ่านค่า: ภาพยนตร์ 1 เรื่อง บันทึกลงดีวีดีได้ 1 แผ่นหรือมากกว่า / ดีวีดี 1 แผ่น อาจไม่บันทึกภาพยนตร์เลยหรือบันทึกได้ 1 เรื่อง

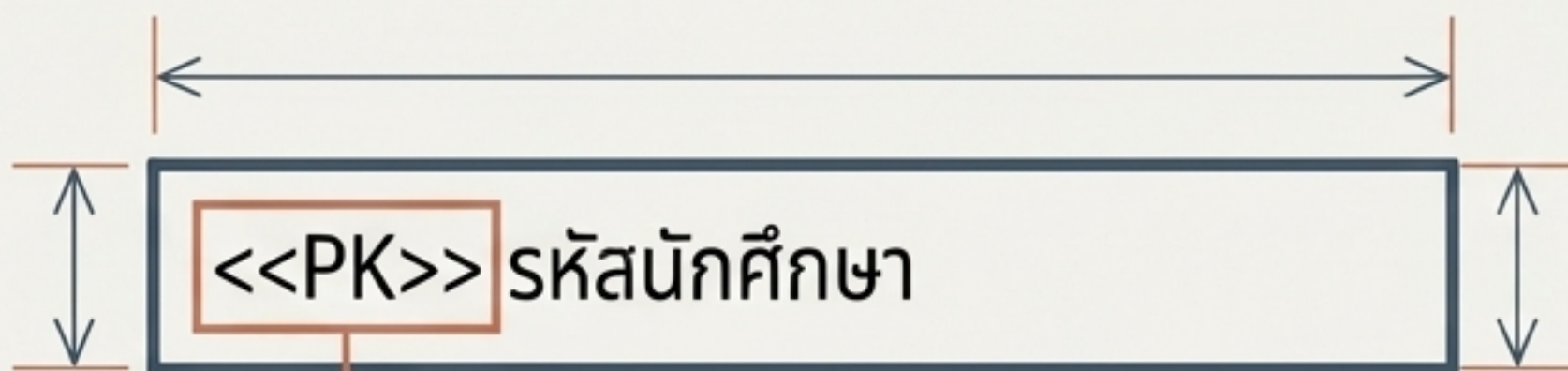
สกัดข้อมูลเป็นสถาปัตยกรรม (Geometric Extraction)

การแปลงข้อมูลดิบสู่โครงสร้างเชิงโครงสร้าง (Structuring Data)



กำหนดคุณสมบัติระดับไมโคร (Building the Blocks) (Building





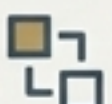
การฝังรหัสประจำตัว (Identifier)



<<PK>> คีย์หลัก (Primary Key)
เป็นตัวบ่งชี้ที่ไม่ซ้ำกัน
เช่น <<PK>> รหัสนักศึกษา

Blueprint Spec Sheet

Diagnostic table

	ประเภทข้อมูล (Data Type Mapping)
	String (สายอักขระ) - สำหรับชื่อ, รหัส, ที่อยู่
	Integer (เลขจำนวนเต็ม) - สำหรับชั้นปี, หน่วยกิต
	Date (วันที่) - สำหรับวันที่ลงทะเบียน
	Currency (ค่าเงิน) - สำหรับเงินค่าลงทะเบียน
	Boolean (ตรรกะ) - สำหรับสถานะ (True/False)

โครงสร้างระบบลงทะเบียน (The Architecture Layout)

จาก Use Case สู่โครงสร้างสถาปัตยกรรมพื้นฐาน นำคลาสพื้นฐานมาวางบนกระดานเพื่อเตรียมประกอบร่าง

1. นักศึกษา
(Student)

2. พนักงาน
(Employee)

3. วิชา
(Course)

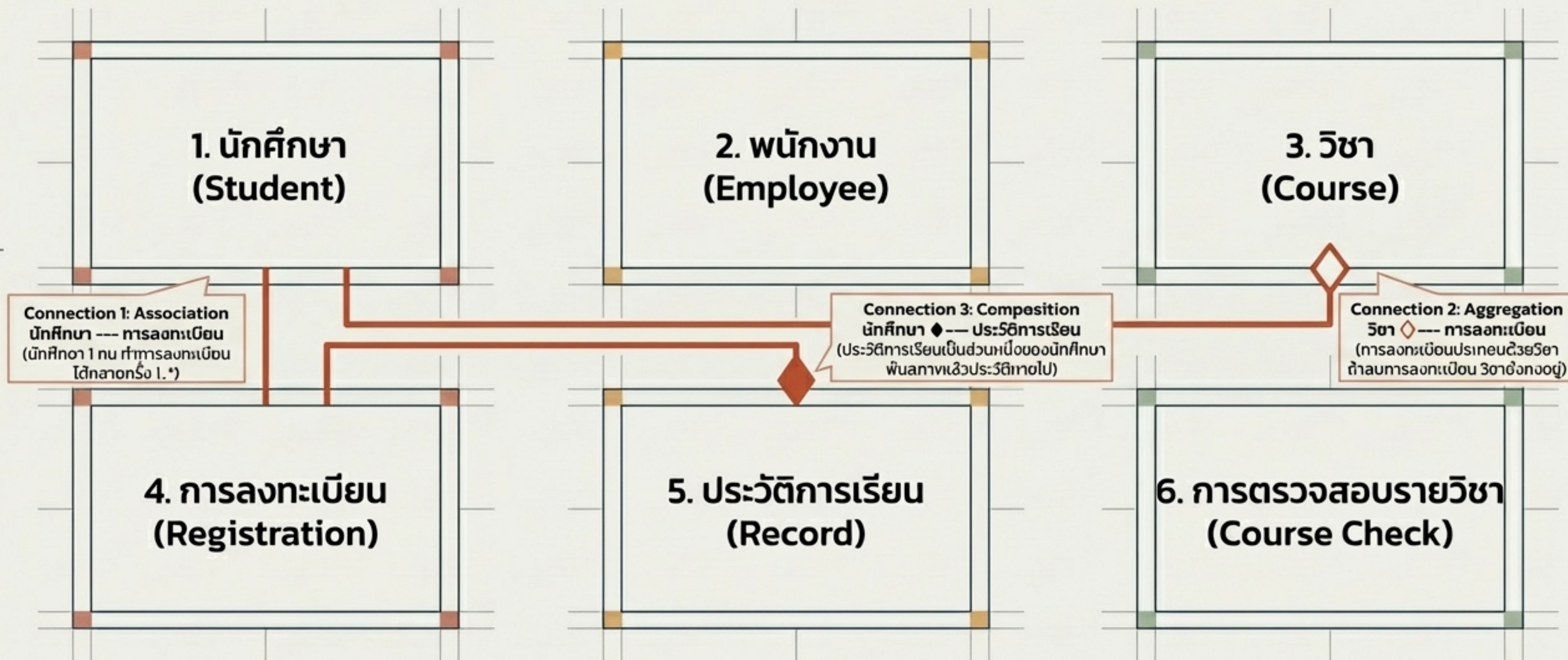
4. การลงทะเบียน
(Registration)

5. ประวัติการเรียน
(Record)

6. การตรวจสอบรายวิชา
(Course Check)

เชื่อมต่อวงจร (Wiring the System)

สาธิตการเลือกใช้เส้นความสัมพันธ์เรขาคณิตแต่ละรูปแบบลงในระบบจริง



เรขาคณิตเปลี่ยนโลก (The Complete UML Canvas)

แผนภาพคลาสไม่ใช่เพียงแค่การวาดรูปสี่เหลี่ยมและเส้นตรง
แต่มันคือ 'การใช้รูปทรงเรขาคณิตเพื่อสะท้อนความจริงของระบบ' (The geometric reflection of reality)

จากนามธรรมที่จับต้องไม่ได้ ถูกสกัด จัดระเบียบ และเชื่อมโยงด้วยหลักการทางเรขาคณิต
จนกลายเป็นพิมพ์เขียวที่วิศวกรซอฟต์แวร์สามารถนำไปสร้างระบบที่เปลี่ยนแปโลกได้จริง

