



การนำแบบจำลองไปใช้งาน

ผู้ช่วยศาสตราจารย์สมเกียรติ ช่อเหมือน (tko@webmail.npru.ac.th)

สาขาวิชาวิศวกรรมซอฟต์แวร์ คณะวิทยาศาสตร์และเทคโนโลยี

NAKHON PATHOM
RAJABHAT UNIVERSITY
NAKHON
PATHOM
RAJABHAT
UNIVERSITY

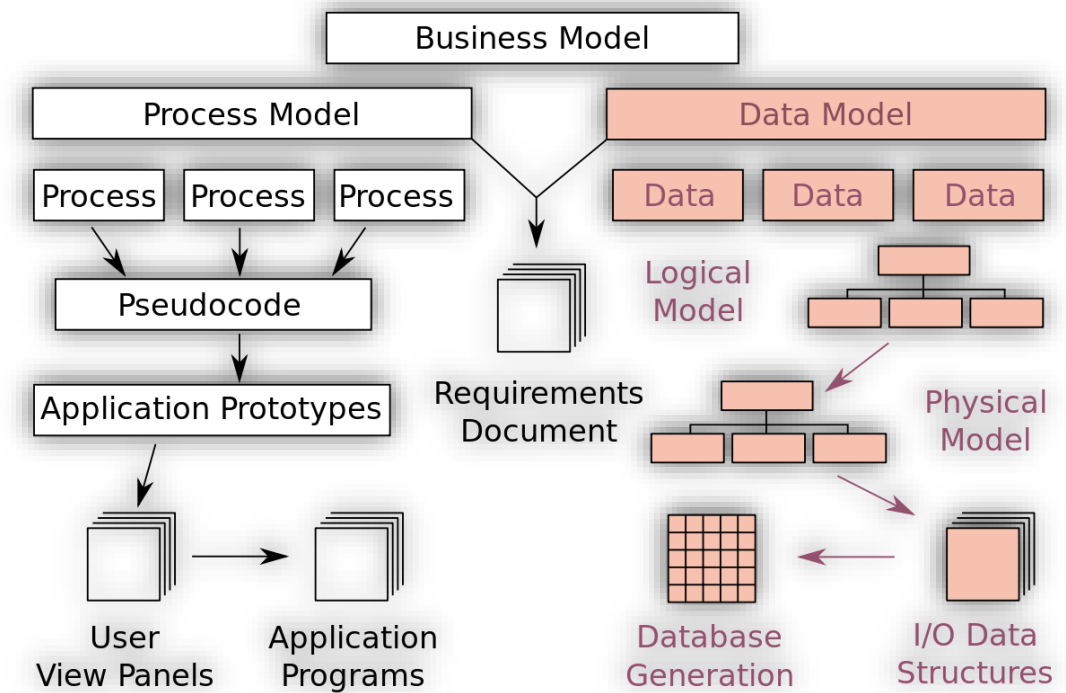
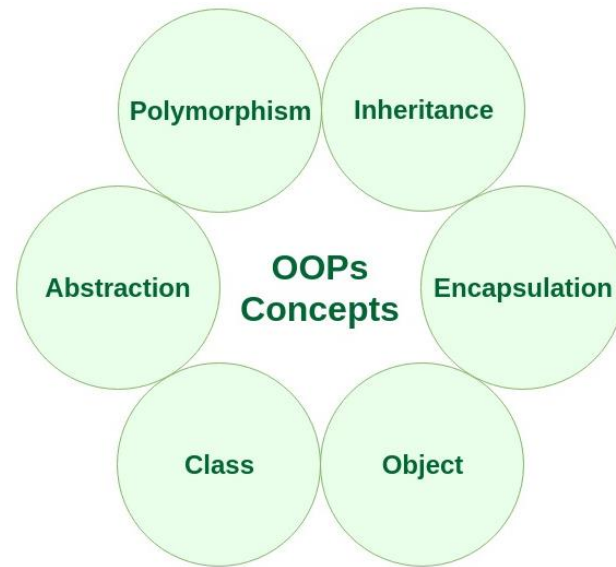
เนื้อหาการเรียนรู้

- การใช้งานแบบจำลองเชิงวัตถุ
- การเขียนโปรแกรมเชิงวัตถุ
- การกำหนดเงื่อนไขการทำงาน
- การกำหนดสถานะของวัตถุ
- การใช้งานข้อมูล
 - ฐานข้อมูล
 - ข้อมูลเชิงวัตถุ



Implementing an object-oriented design

- การดำเนินการตามที่ออกแบบเชิงวัตถุ
- object designs to databases
- object oriented programming language (**OOPL**)



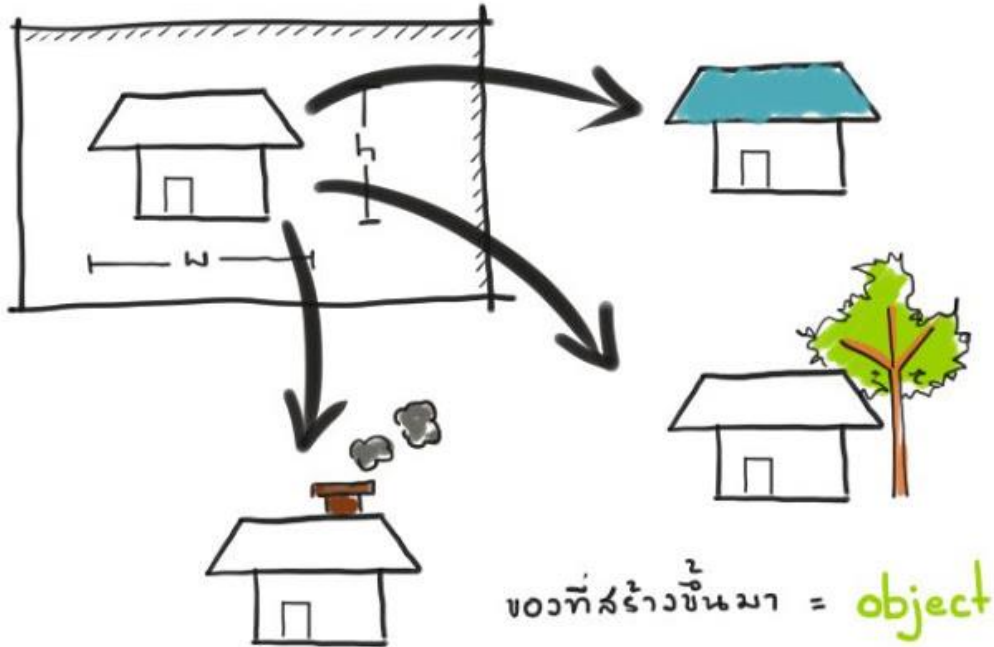
Programming Languages

- การแปลงจากแบบจำลองเชิงวัตถุไปเป็นโค้ดหรือโปรแกรม
- ภาษาโปรแกรมเชิงวัตถุ เช่น C++, Java, Smalltalk, C# และ Python
- การใช้คลาสเป็นตัวแทนของวัตถุ



class Circle using C++

บ้านแบบ = class



| Circle |
|--|
| - x-coord - y-coord # radius |
| + findArea() + findCircumference() + scale() |

```
class Circle
{
    private:
        double x_coord;
        double y_coord;
    protected:
        double radius;
    public:
        double findArea();
        double findCircumference();
        void scale();
};
```

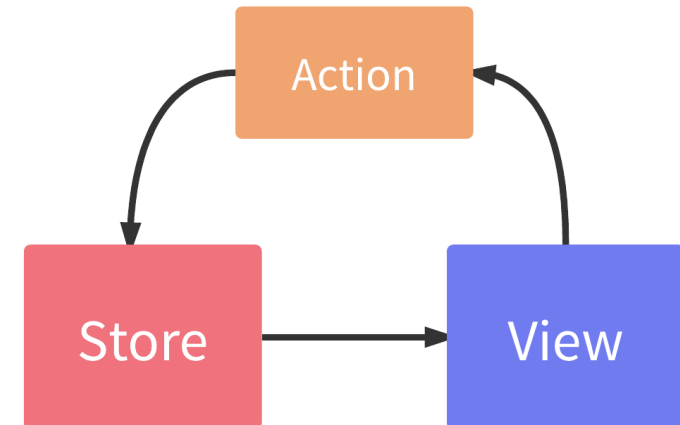
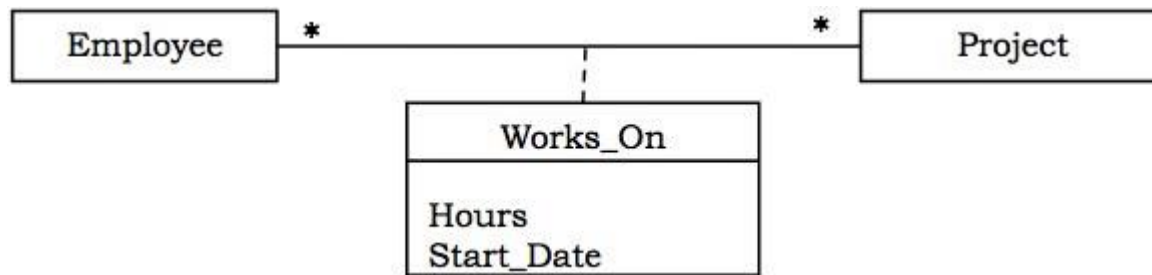
Implementing Associations

- ไม่ได้จัดทำโครงสร้างเพื่อใช้การเชื่อมโยงโดยตรง
- การเชื่อมโยงอาจเป็นแบบทิศทางเดียวหรือแบบสองทิศทาง
- การเชื่อมโยงแบบกลุ่ม
 - one-to-one (1:1)
 - one-to-many (1:M)
 - many-to-many (M:N)



Associations as Classes

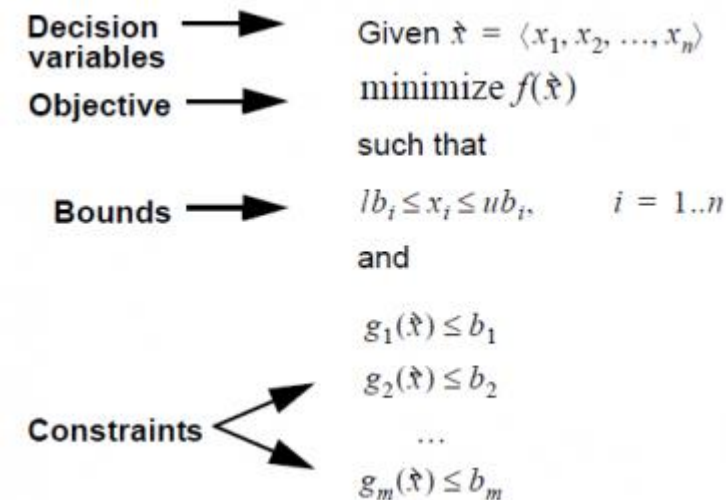
- การเชื่อมโยงมีคุณสมบัติบางอย่างที่เกี่ยวข้อง จำเนิการแยกคลาส



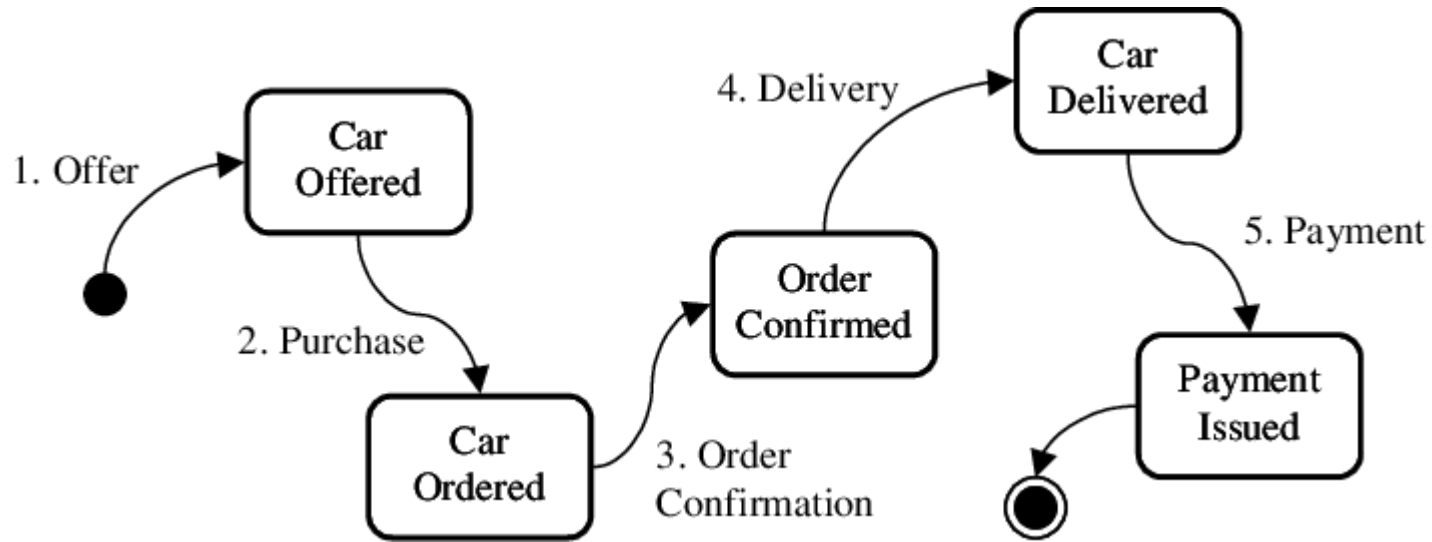
Implementing Constraints

- ข้อจำกัด
 - ช่วงและประเภทของค่าที่อาจใช้คุณลักษณะ
 - ค่าเริ่มต้นที่ถูกต้องจะถูกกำหนดให้กับแอดทริบิวต์

| Design variables | | Constraints |
|---------------------|---|--|
| ϵ_d | Dielectric constant of the insulator | $2 \leq \epsilon_d \leq 15$ |
| f_v | Frequency of the applied voltage (kHz) | $5 \leq f_v \leq 20$ |
| r_f | Positive-to-negative half cycle ratio | $0.5 \leq r_f \leq 1.5$ |
| Objective functions | | Definitions |
| $- F_{x,ST} $ | Time and domain averaged x-directional force (mN/m) | $F_{x,ST} \equiv \frac{1}{ST} \int_S \int_T F_x(\vec{r}, t) dt d\vec{r}$ |
| P_T | Power input for one cycle by the charge current through the upper electrode (W) | $P_T \equiv \frac{1}{T} \int_T I(t)V(t) dt$ where $I(t) = \int_{eu} (q_i n_i(t) u_{y,i}(t) - q_e n_e(t) u_{y,e}(t)) dx$ |

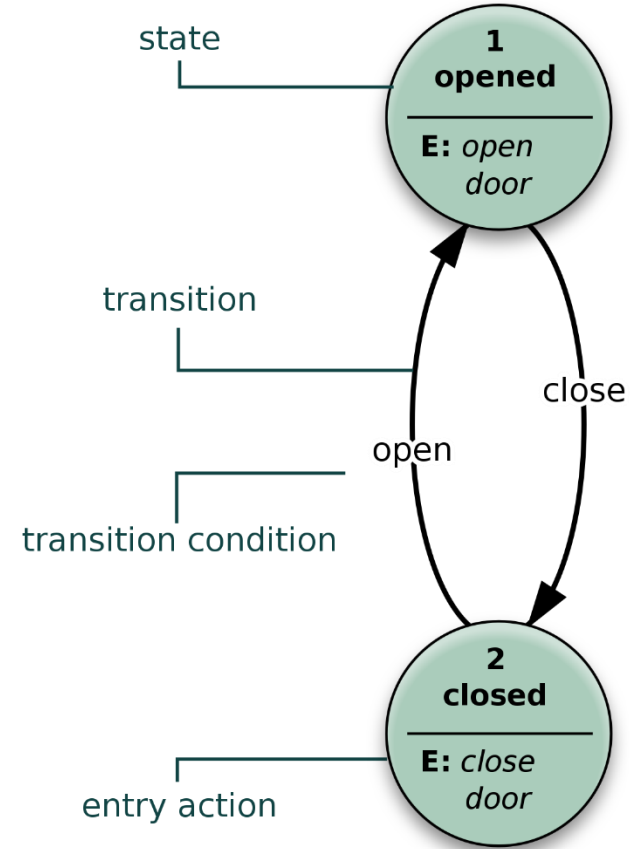


State Charts



สถานะของวัตถุ

Condition /Action

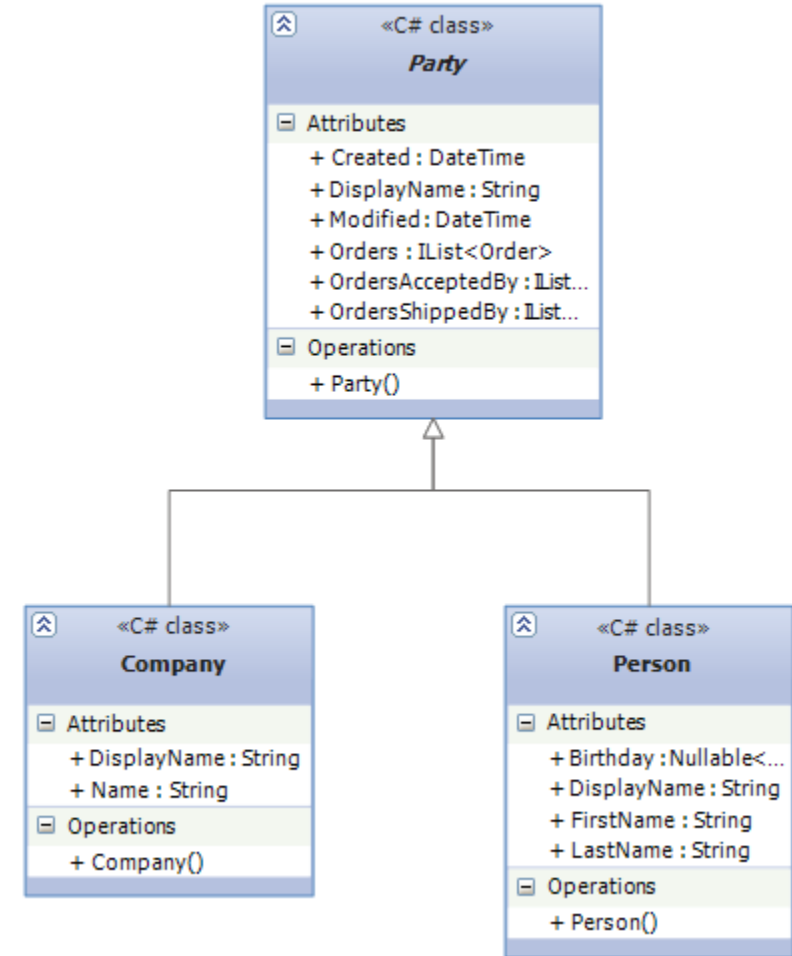
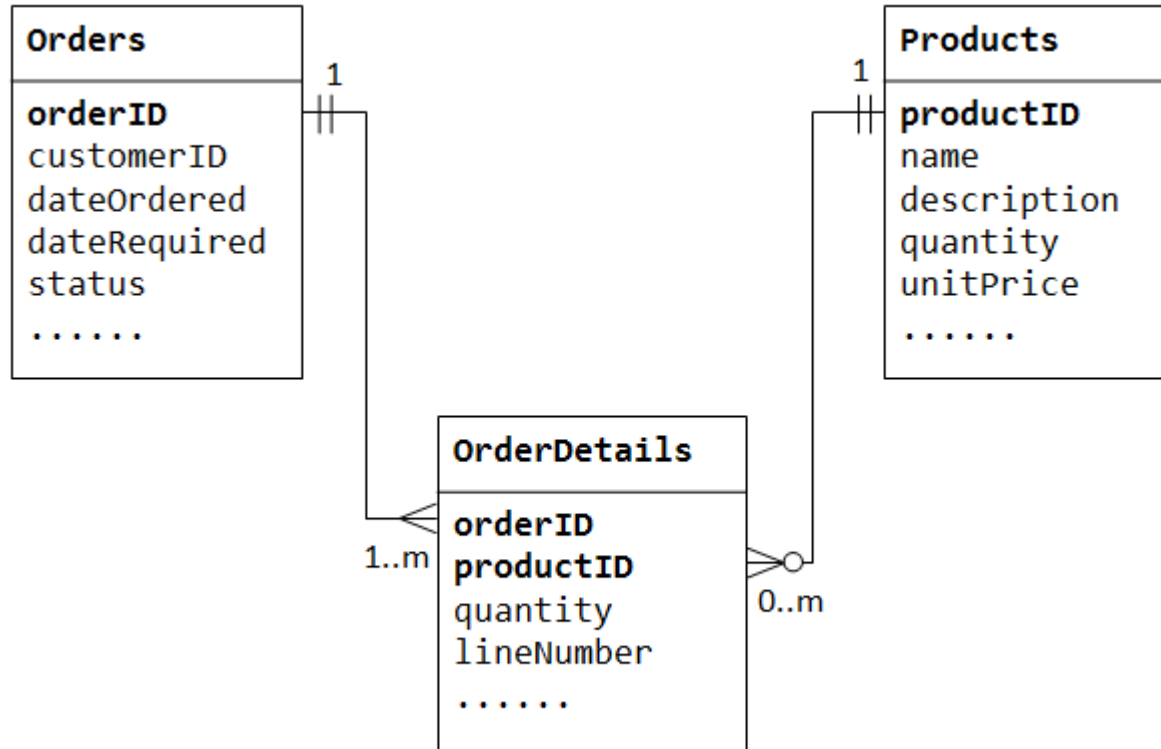


Object Mapping to Database System

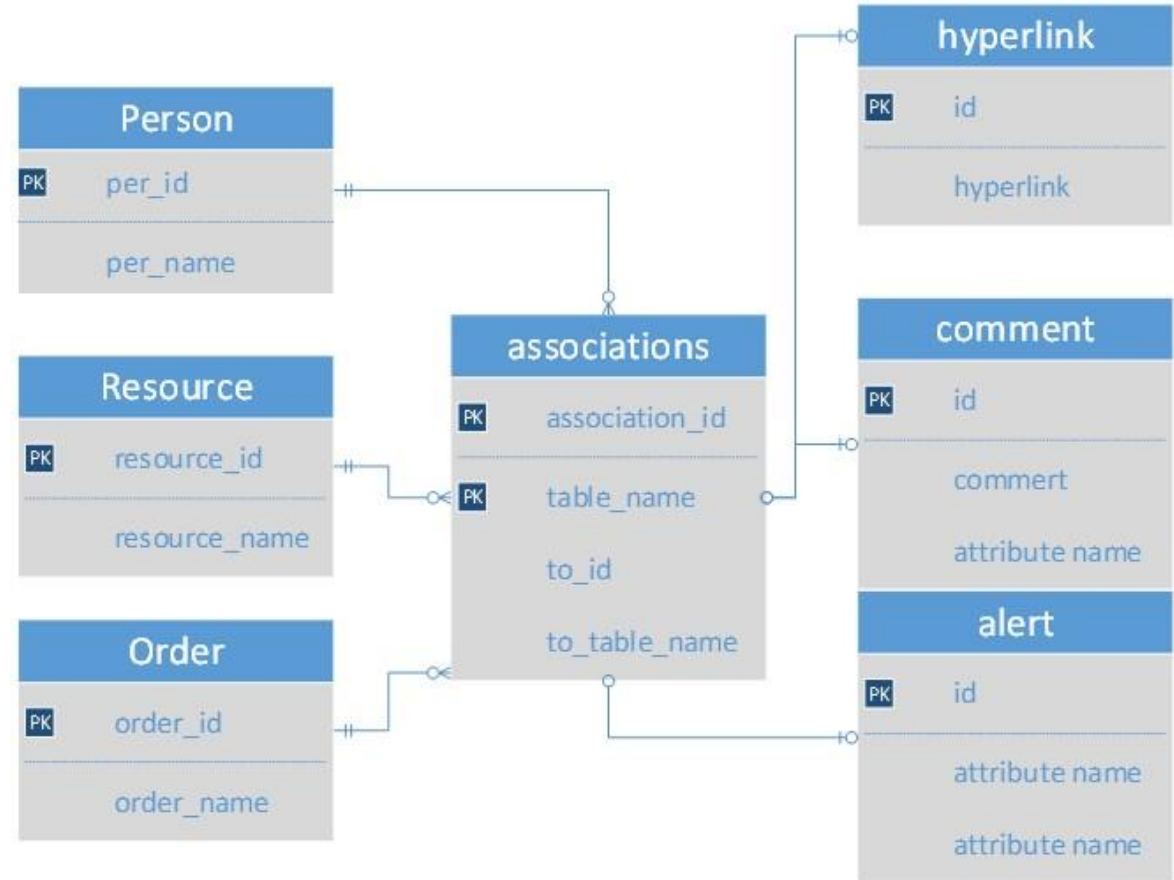
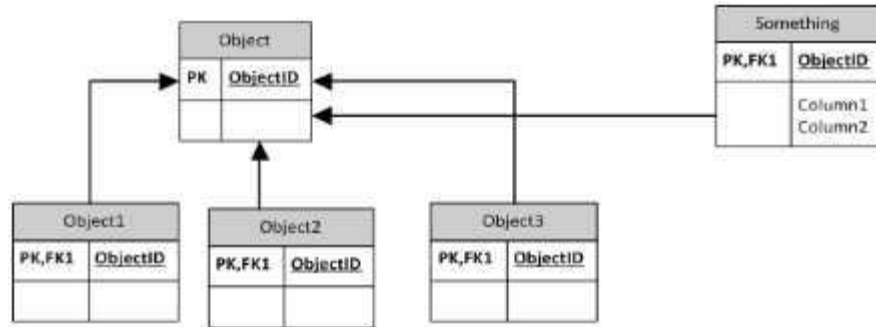
- Persistency of Objects
- RDBMS

| From (Object Concept) ... | To (Relational Concept) ... |
|---------------------------|-----------------------------|
| Package | Schema |
| Class | Table |
| OID | Identity field |
| Data type property | Field |
| Reference property | Reference field |
| Embedded object | Set of fields |
| List property | List field |
| Array property | Child table |
| Stream property | BLOB |
| Index | Index |
| Class method | Stored procedure |

Classes as Tables

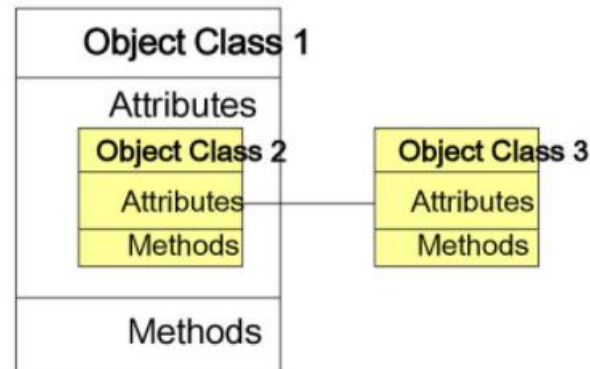


Associations to Database Tables



สรุปท้ายบท

- การนำแบบจำลองเชิงวัตถุที่ออกแบบมาใช้งาน
- การดำเนินการเขียนโปรแกรมเชิงวัตถุ
 - กำหนดเงื่อนไขการทำงาน
 - กำหนดสถานะของวัตถุ
- การเขียนโปรแกรมเพื่อใช้งาน
 - ฐานข้อมูล
 - ข้อมูลเชิงวัตถุ



```
class Test {
    public static void main(String[] args) {
        Account acc = new Account();
        acc.deposit(500);
        Customer c = new Customer("Test");
        System.out.println("InCustomer \\" +
            c.getName() + "\\" + " has accounts " +
            c.getAccount(acc).getBalance());
    }
}
```