

Generalization Abstraction

เรื่องที่จะศึกษา

- Generalization Abstraction
- Specialization Abstraction
- Inheritance

Generalization และ Specialization

- ในโลกแห่งความเป็นจริง วัตถุทุกชนิด จะ แบ่งปัน ลักษณะหรือความเป็นอยู่ ซึ่งกันและกัน
- เราสามารถจัดกลุ่มของ วัตถุ ที่มีคุณสมบัติหรือลักษณะ ใกล้เคียงกัน ให้อยู่ในคลาสเดียวกัน
- ตัวอย่าง การจำแนก classes ของสัตว์...

ตัวอย่าง คลาสของสัตว์

○ Mammals

- Dogs, cats, horses, duckbill platypuses, kangaroos, dolphins and whales are all mammals
- ตัวอ่อนดื่มนมเป็นอาหาร และมีขนตามร่างกาย

○ Birds

- นกทุกชนิดมีขน และเกิดจากไข่ที่มีเปลือกแข็ง
- ขนบนปีกและหาง จะทับซ้อนกันอยู่ ซึ่งทำให้ได้ลม และทำให้นกบินและร่อนลงได้

ตัวอย่าง คลาสของสัตว์

- Fish
 - Fish เป็นสัตว์มีกระดูกสันหลัง
 - อาศัยในน้ำ มี เหงือก (gills), เกล็ด (scales) และ ครีบ (fins)
- Reptiles (สัตว์เลื้อยคลาน)
 - Reptiles เป็น class ของสัตว์ที่มีเกล็ดบนผิวหนัง
 - เป็นสัตว์เลือดเย็นและเกิดบนบก

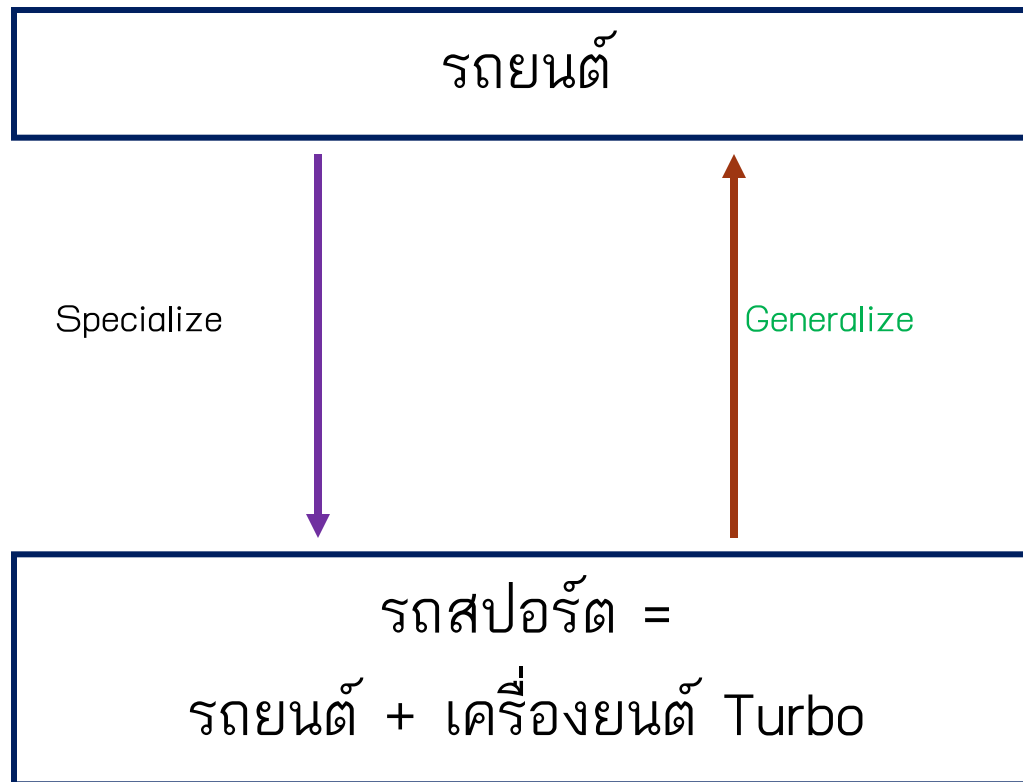
ตัวอย่าง คลาสของสัตว์

- Amphibians (สัตว์ครึ่งบกครึ่งน้ำ)
 - Amphibians เกิดในน้ำ เมื่อแรกเกิดจะหายใจด้วยเหงือกคล้ายปลา
 - เมื่อโตขึ้นจะพัฒนาปอดขึ้นมาและอาศัยบนบกเป็นหลัก
- Arthropods
 - Arthropods เป็น class ของสัตว์ที่มีมากกว่า 4 ขา
 - แมลงต่างๆ รวมถึงแมงมุมอยู่ในประเภทนี้

Generalization และ Specialization

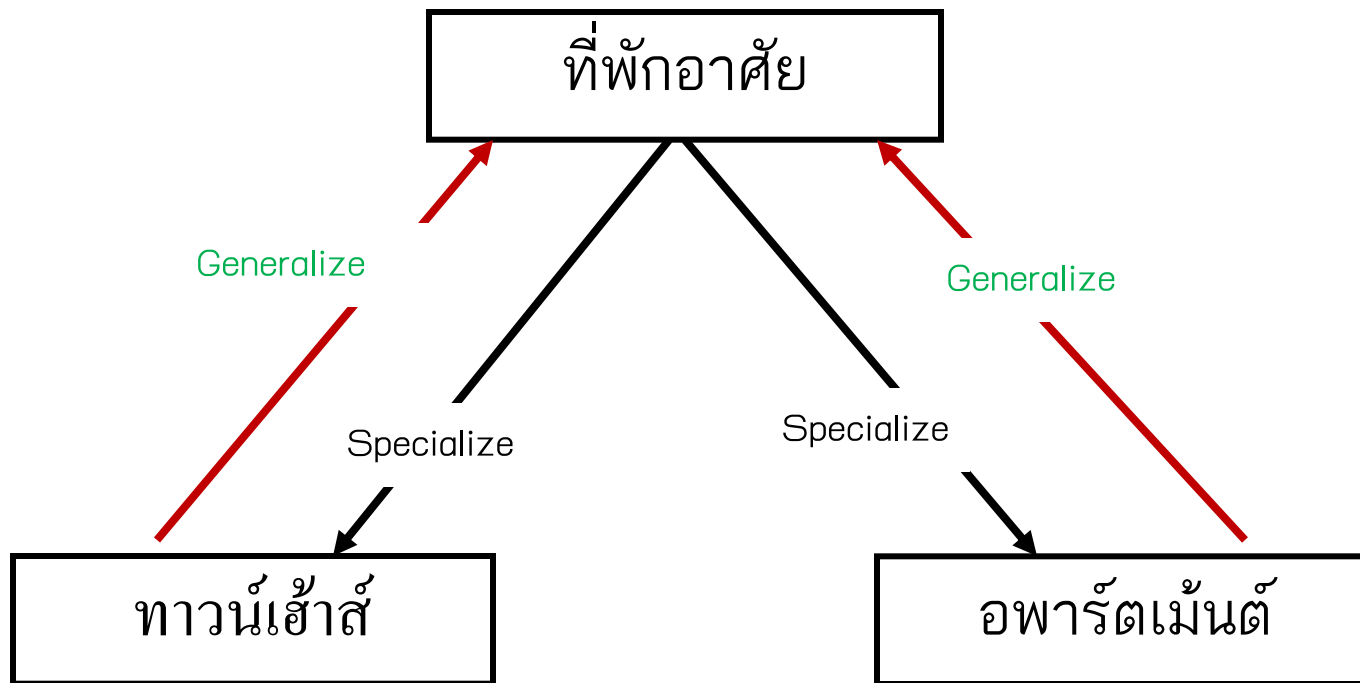
- การให้ Concept ใหม่ กับ Class ใด Class หนึ่ง (จากหลายๆ Objects) โดยละเอียดหรือตัดคุณสมบัติพิเศษบางอย่างออกไป เพื่อให้ Class ดังกล่าวมีลักษณะเป็นสามัญ (General) เรียกว่า กระบวนการ Generalize
- การให้ Concept ใหม่ กับ Class ใด Class หนึ่งที่มีอยู่แล้ว โดยพิจารณาหรือเพิ่มเติมคุณสมบัติใหม่ๆ ให้ Class (สามารถนำไปสร้างเป็น Object) ที่มีลักษณะพิเศษ (Special) เพิ่มขึ้นกว่าเดิม เรียกว่า กระบวนการ Specialize

ตัวอย่าง Generalization และ Specialization ของรถยนต์และรถสปอร์ต



Generalization และ Specialization

ของที่พักอาศัย ทาวน์เฮ้าส์ และ อพาร์ทเมนต์



กิจกรรม

- ให้นักศึกษาบอกถึงการทำให้ generalization และ specialization ที่พบในชีวิตประจำวันมาคนละกรณี

Inheritance

- กลไกที่เกิดจากการทำ Generalization

Abstraction:

- Subclass, Superclass และ Inheritance
- กฎเกณฑ์ของการทำ Inheritance
- Multiple Inheritance
- Polymorphism

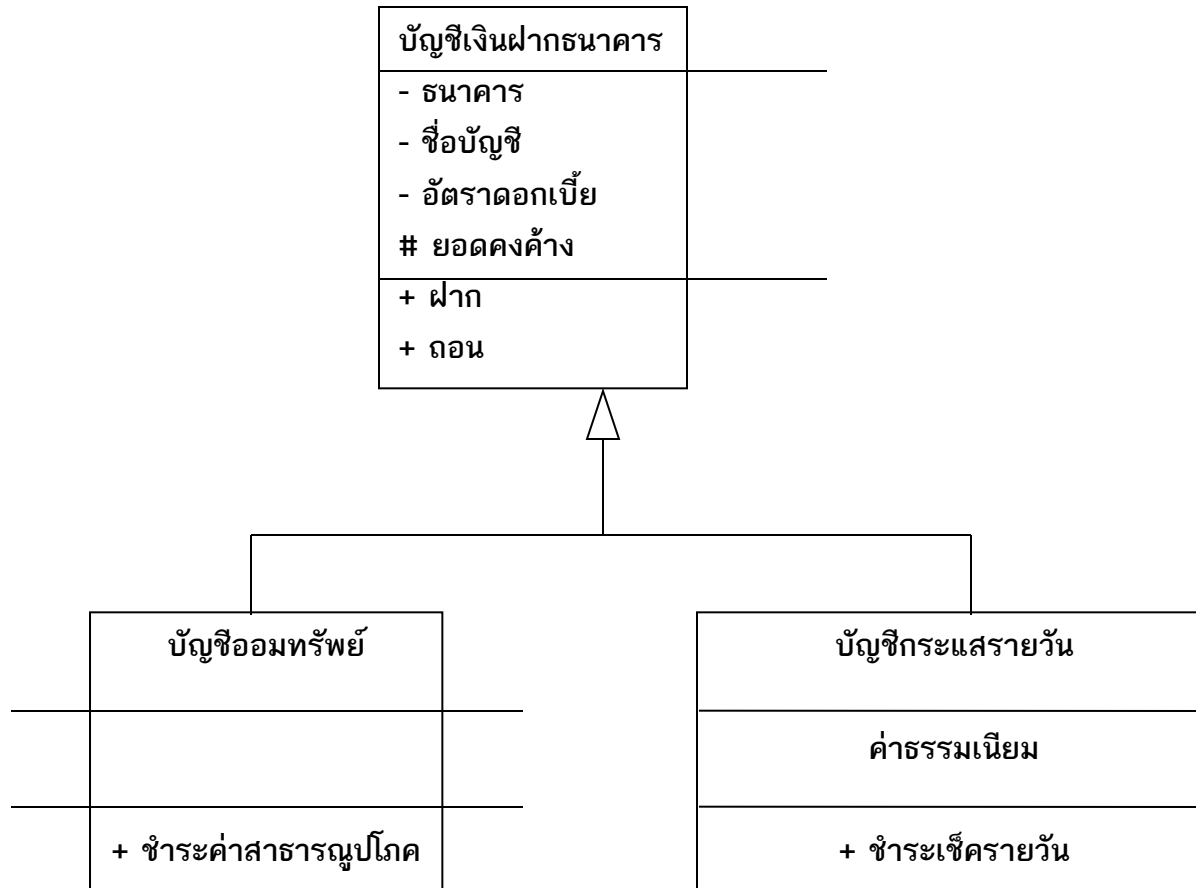
Subclass, Superclass และ Inheritance

- Superclass = Class เริ่มต้น หรือ class แม่ (base class)
- Subclass = Class ที่เกิดจากการทำ Specialize
- Inheritance = กระบวนการ Specialization

ข้อควรจำในการทำ Inheritance

- Subclass ที่ inherit มาจาก superclass นั้นจะต้องมีคุณสมบัติทุกอย่างของ superclass (attribute & function) ผสมกับคุณสมบัติพิเศษที่เพิ่มเข้าไปในแต่ละ subclass เสมอ
- ใช้สัญลักษณ์ลูกศรซึ่งมีหัวเป็นรูปสามเหลี่ยมใส่ชี้จาก subclass ไปยัง superclass

สัญลักษณ์แสดงการทำ Inheritance



กฎเกณฑ์ของการทำ Inheritance

- Private Attributes/Functions จะถ่ายทอดมาเป็น Private Attributes/Functions ของ Subclass
 - ไม่สามารถเข้าถึง private attribute ที่ inherit มาได้
 - ไม่สามารถเข้าถึงได้จาก Function ที่ไม่ได้มาจากการ Inheritance (ฟังก์ชันของ subclass เอง)

กฎเกณฑ์ของการทำ Inheritance (2)

- Protected Attributes/Functions จะถ่ายทอดมาเป็น Protected Attributes/Functions ของ Subclass
 - สามารถเข้าถึงได้จากทุก Function ใน Subclass

กฎเกณฑ์ของการทำ Inheritance (3)

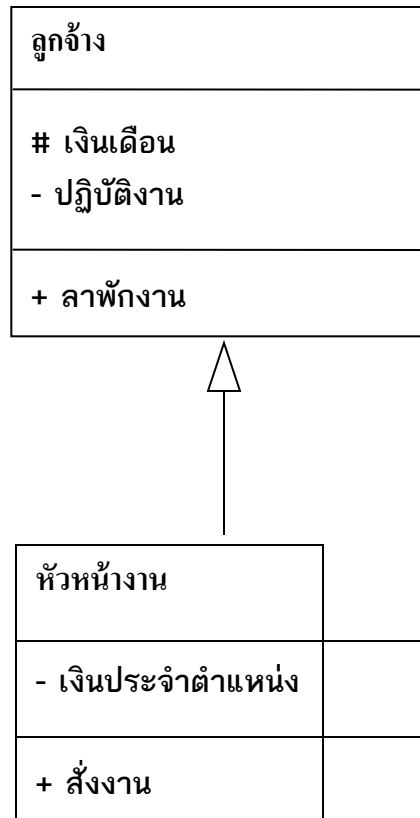
- Public Attributes/Functions จะถ่ายทอดมาเป็น Public Attributes/Functions ของ Subclass เสมอ

Inside View ของ Class บัญชีออมทรัพย์ และ Class บัญชีกระแสรายวัน

บัญชีออมทรัพย์	
- ธนาคาร - ชื่อบัญชี - อัตราดอกเบี้ย # ยอดคงค้าง	
+ ฝาก + ถอน + ชำระค่าสาธารณูปโภค	

บัญชีกระแสรายวัน	
- ธนาคาร - ชื่อบัญชี - อัตราดอกเบี้ย # ยอดคงค้าง - ค่าธรรมเนียม	
+ ฝาก + ถอน + ชำระเช็ครายวัน	

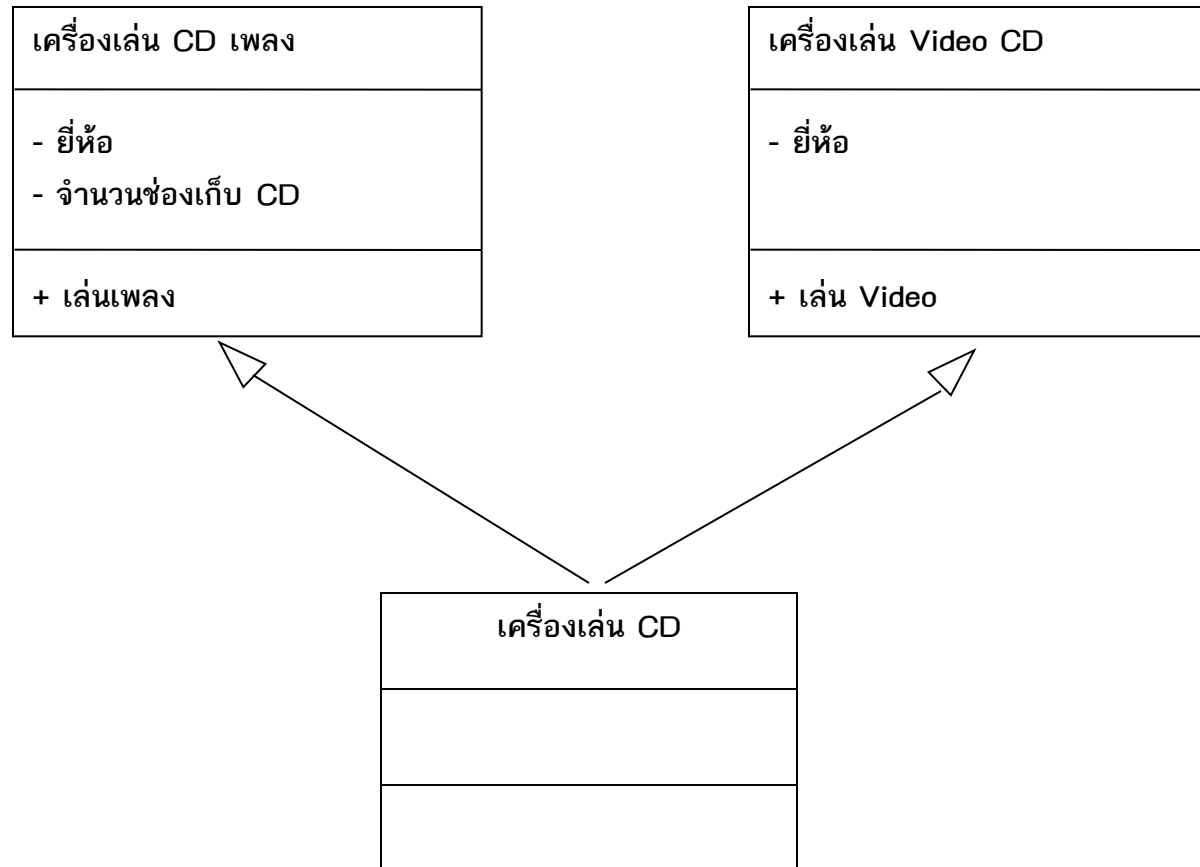
Inheritance ของพนักงาน - หัวหน้างาน



Multiple Inheritance

- เป็น Subclass ที่ inherit มาจาก superclass มากกว่า 1 ตัว เพื่อให้ได้ subclass ที่มีคุณสมบัติพิเศษเพียงตัวเดียว
- Subclass ที่เกิดจาก Multiple Inheritance นั้น จะเลือกเอา attribute และ function ที่ชื่อซ้ำกันจาก superclass ที่ได้ทำ inherit ก่อน เสมอ
- การทำ inherit จะทำ superclas ที่อยู่ ทางซ้าย ก่อนเสมอ

Multiple Inheritance ของเครื่องเล่น CD



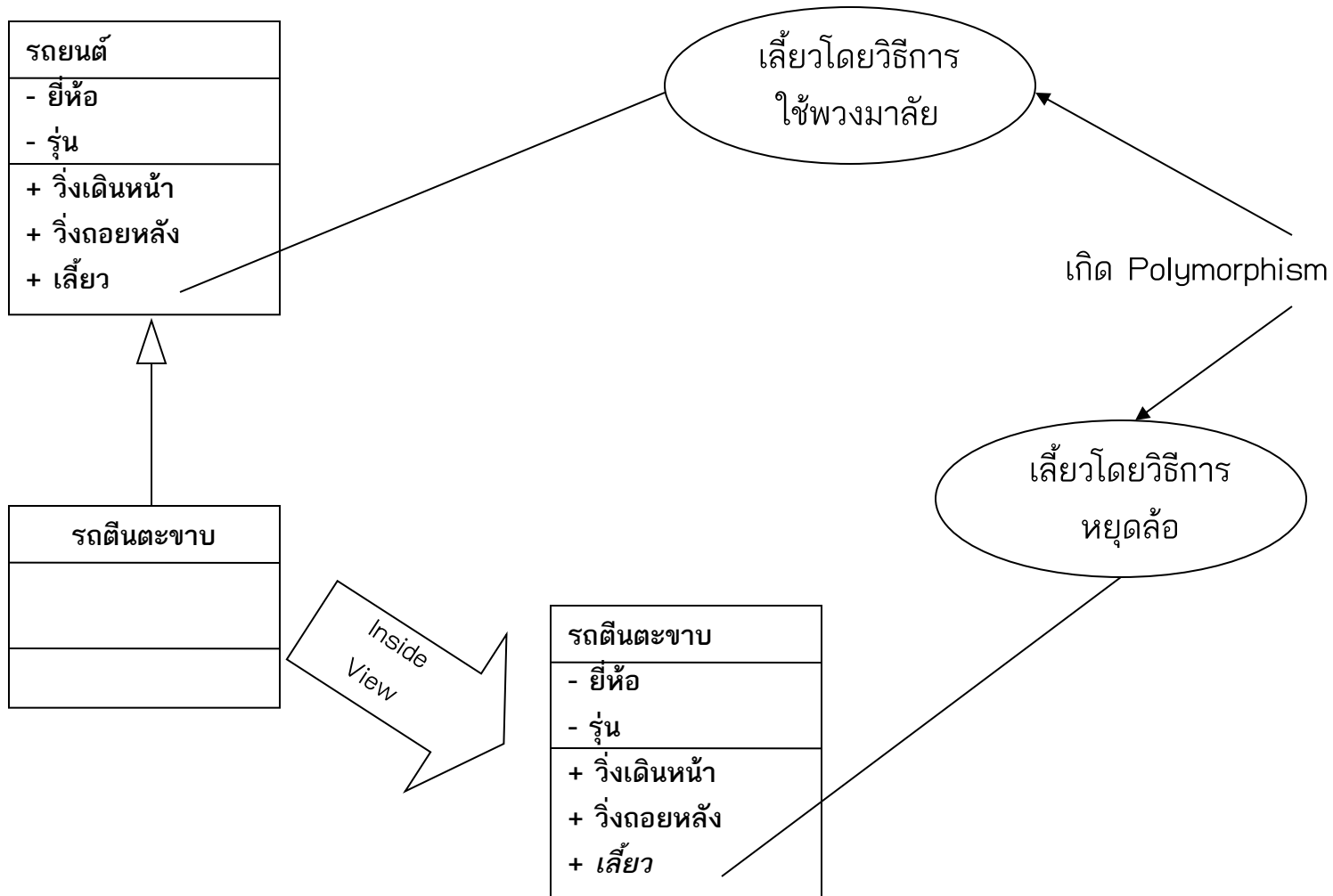
Inside View ของเครื่องเล่น CD

เครื่องเล่น CD
<ul style="list-style-type: none">- ยี่ห้อ- จำนวนช่องเก็บ CD
<ul style="list-style-type: none">+ เล่นเพลง+ เล่น Video

Polymorphism

- เป็นคุณสมบัติของ Subclass ที่มีการดัดแปลง function บางอย่างโดยไม่ได้ยึดตาม superclass ทั้งหมด

ภาพจำลองแสดง Polymorphism



References

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