หอสมุดกลาง สถาบันวิทยบริการ จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 2

หลักการของการโปรแกรมเชิงวัตถุ

ในบทนี้ผู้วิจัยจะกล่าวถึงหลักการของการโปรแกรมเชิงวัตถุ ซึ่งใช้เป็นพื้นฐานในการออกแบบโปรแกรมวิเคราะห์ภาพถ่ายรังสีเอกซ์ของกระโหลกศีรษะ สาเหตุที่ผู้วิจัยเลือกใช้วิธีการของการโปรแกรมเชิงวัตถุเนื่องจาก ผู้วิจัยต้องจัดทำโปรแกรมสำหรับสภาวะแวดล้อมแบบวินโดว์ซึ่งมีการออกแบบในลักษณะการโปรแกรมเชิงวัตถุ ดังมีรายละเอียดของหลักการโปรแกรมเชิงวัตถุดังนี้

โลกประกอบด้วยสิ่งมีชีวิตและวัตถุต่าง ๆ มากมาย ซึ่งมีความสัมพันธ์กันอย่างซับซ้อน แต่มนุษย์สามารถจำแนกสิ่งมีชีวิตและวัตถุต่าง ๆ ได้อย่างเป็นระบบ และสามารถประยุกต์ใช้ให้เกิดประโยชน์ได้อย่างมีประสิทธิภาพ โดยมนุษย์อาศัยความสามารถในการจัดกลุ่ม การจำแนกความแตกต่าง และการกำหนดสัญลักษณ์เพื่อแทนความหมาย

การโปรแกรมเชิงวัตถุ เป็นวิธีการในการเขียนโปรแกรมโดยเลียนแบบความสามารถตามธรรมชาติของมนุษย์ ในการมองและแยกแยะสิ่งต่าง ๆ เป็นวัตถุ การมองสิ่งมีชีวิตหรือวัตถุต่าง ๆ มนุษย์ไม่ได้มองลึกถึงรายละเอียดภายในของวัตถุ แต่มองถึงลักษณะภายนอกและคุณสมบัติของวัตถุที่แสดงออกมา

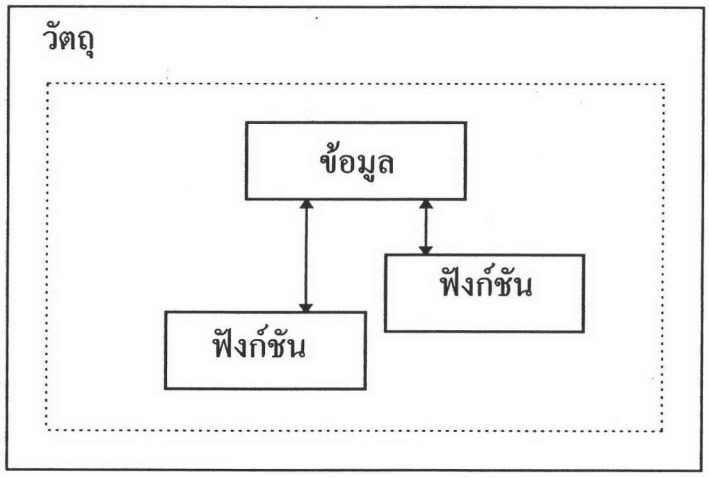
Holzner (1993) ได้อธิบายหลักการการโปรแกรมเชิงวัตถุ โดยยกตัวอย่างการมองอุปกรณ์ในห้องครัว ห้องครัวอาจมีอุปกรณ์ต่าง ๆ ดังนี้ ตู้เย็น เตาอบ และเครื่องล้างจาน เมื่อผู้ใช้ใส่ของเข้าไปในตู้เย็น ผู้ใช้ไม่ต้องทราบถึงขั้นตอนรายละเอียดในการสร้างความเย็น แต่ทราบเพียงคุณสมบัติการทำความเย็นของตู้เย็น เช่นเดียวกันผู้ใช้ใส่ของเข้าไปในเตาอบเพื่อให้ของร้อน ซึ่งสามารถเปรียบกับหลักการของการโปรแกรมเชิงวัตถุ ที่จะมีการกำหนดวัตถุและคุณสมบัติต่าง ๆ เพื่อให้ง่ายในการจดจำและจัดการกับวัตถุนั้น ๆ

ลักษณะที่สำคัญของการโปรแกรมเชิงวัตถุ

การโปรแกรมเชิงวัตถุนั้น จะมองวัตถุแยกเป็น 2 ส่วน คือ ลักษณะหรือข้อมูลของวัตถุนั้น และคุณสมบัติของวัตถุที่แสดงออก ในที่นี้จะเรียกว่าฟังก์ชัน โดยจะจัดวัตถุที่คุณสมบัติพื้นฐานร่วมกันเป็นกลุ่มเรียกว่าคลาส และมีลักษณะที่สำคัญ 3 ประการดังนี้

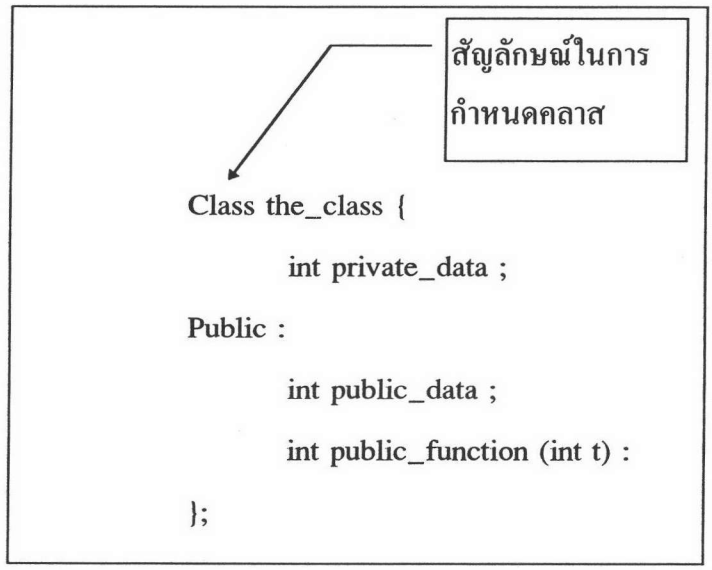
1. เอนแคปซูเลชั่น (Encapsulation)

เป็นการผนึกข้อมูลและฟังก์ชันเข้าด้วยกัน เป็นวัตถุชิ้นเดียวกัน ดังแสดงในรูปที่ 2.1 โดยข้อมูลที่อยู่ในวัตถุจะถูกเรียกใช้จากฟังก์ชันที่อยู่ในวัตถุเดียวกันเท่านั้น จากหลักการนี้จะช่วยแก้ปัญหาการเขียนโปรแกรมแบบโครงสร้างซึ่งหลายฟังก์ชันจะมีการใช้ข้อมูลร่วมกันทำให้การทดสอบและแก้ไขโปรแกรมทำได้โดยลำบาก เนื่องจากข้อมูลอาจถูกเปลี่ยนแปลงโดยฟังก์ชันอื่นได้ง่ายและถ้ามีการแก้ไขโครงสร้างข้อมูลก็อาจต้องแก้ไขโปรแกรมเกือบทั้งหมด ซึ่งต่างจากหลักการของการโปรแกรมเชิงวัตถุที่มีคุณสมบัติเอนแคปซูเลชั่นทำให้ข้อมูลและฟังก์ชันถูกผนึกอยู่ด้วยกัน ข้อมูลจะถูกเปลี่ยนแปลงจากฟังก์ชันที่อยู่ในวัตถุเท่านั้น ทำให้สามารถตรวจสอบความผิดพลาดได้ง่ายและถูกจำกัดอยู่เฉพาะในวัตถุเท่านั้น และเมื่อมีการแก้ไขโครงสร้างข้อมูลก็ไม่มีผลกระทบต่อส่วนอื่นของโปรแกรม ผู้พัฒนาโปรแกรมเพียงแต่แก้ไขฟังก์ชันที่เกี่ยวข้องในวัตถุเท่านั้น



รูปที่ 2.1 แสดงการผนึกข้อมูลและฟังก์ชันเข้าด้วยกัน

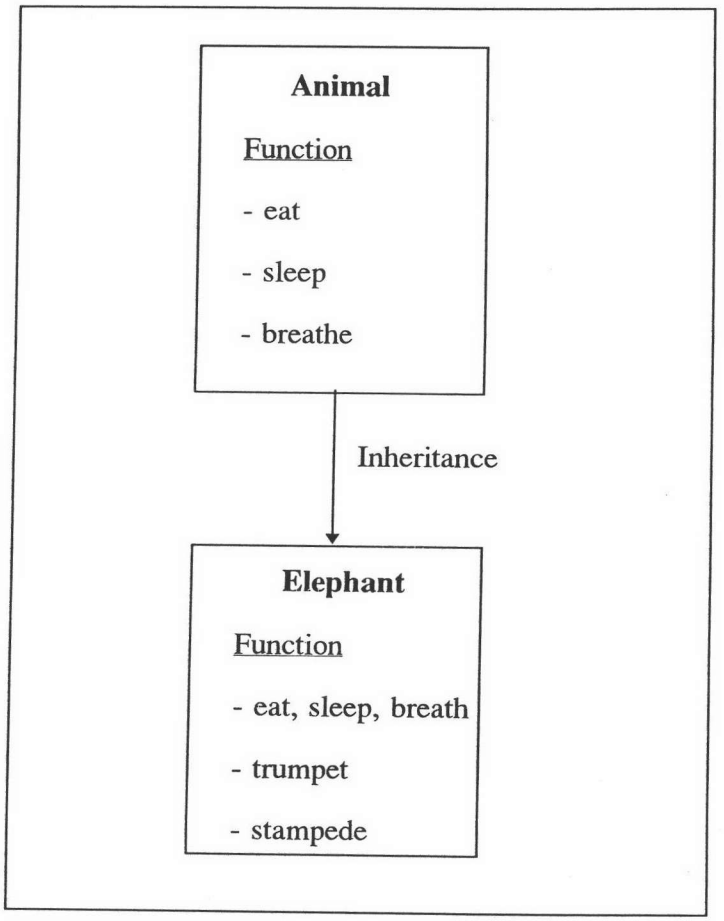
ในภาษา C++ นั้น จะเรียกการผนึกข้อมูลและฟังก์ชันเข้าด้วยกันนี้ว่า คลาสและมีรูปแบบการกำหนดคลาส ดังแสดงในรูปที่ 2.2



รูปที่ 2.2 แสดงการกำหนดคลาสในภาษา C++

2. อินเฮอร์ริแทนซ์ (Inheritance)

คือการสร้างคลาสใหม่ ซึ่งได้รับการถ่ายทอดข้อมูลและฟังก์ชันจากคลาสเดิม โดยสามารถเปลี่ยนแปลงเพิ่มข้อมูลหรือฟังก์ชันของคลาสเดิมได้ ดังตัวอย่างในรูปที่ 2.3 คลาส Elephant เป็นคลาสใหม่ที่ได้รับการถ่ายทอดข้อมูลและฟังก์ชัน จากคลาส Animal โดยที่ คลาส Elephant จะได้รับการถ่ายทอดฟังก์ชัน eat ฟังก์ชัน sleep และฟังก์ชัน breathe จากคลาส Animal และคลาส Elephant ยังมีการกำหนดฟังก์ชันเพิ่มเติมอีก 2 ฟังก์ชัน คือ ฟังก์ชัน trumpet และฟังก์ชัน stampede



รูปที่ 2.3 แสดงการสร้างคลาส elephant โดยได้รับการถ่ายทอดจากคลาส Animal

ในภาษา C++ นั้น จะแทนการถ่ายทอดคลาสโดยใช้สัญลักษณ์ ":" ดังแสดงในรูปที่ 2.4

```
Class class_name : acess base_class {
        Private data and function list
Public :
        Public data and function list
} ;
```

รูปที่ 2.4 แสดงสัญลักษณ์ในการถ่ายทอดคลาส

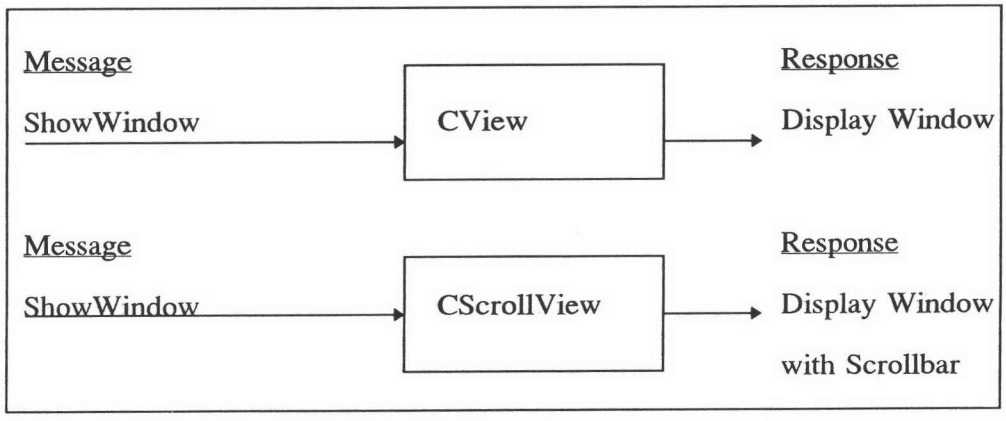
จากตัวอย่างการถ่ายทอดจากคลาส Animal เป็นคลาส Elephant สามารถแสดงเป็นภาษา C++ ได้ดังรูปที่ 2.5

```
Class Animal {
Public :
        Void eat (Void)      ;
        Void sleep (Void)    ;
        Void breathe (Void) ;
} ;
Class Elephant : Public Animal {
Public :
        int trumpet (Void)   ;
        int stampede (Void) ;
} ;
```

รูปที่ 2.5 แสดงการสร้างคลาส elephant โดยได้รับการถ่ายทอดจากคลาส Animal ในภาษา C++

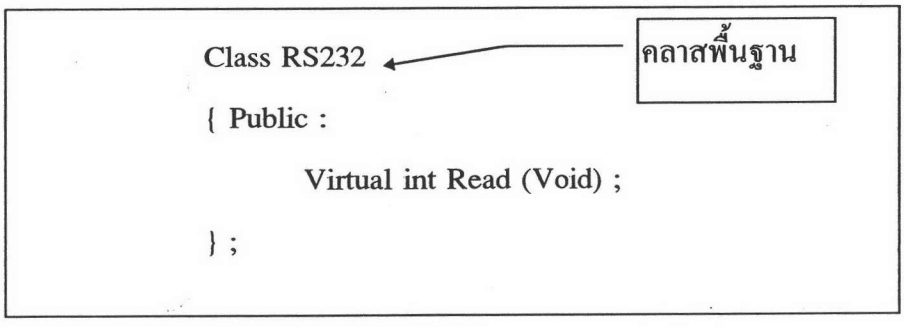
3. โพลิมอร์ฟิซึ่ม (Polymorphism)

คือการที่วัตถุที่มีความสัมพันธ์ อาจมีการถ่ายทอดระดับเดียว หรือมากกว่า 1 ระดับ ตอบสนองต่อข้อความในลักษณะเดียวกัน โดยมีวิธีการเฉพาะของแต่ละวัตถุที่ไม่เหมือนกัน ดังตัวอย่างในรูปที่ 2.6 คลาส CScrollView เป็นคลาสที่ถ่ายทอดมาจากคลาส CView และมีฟังก์ชัน ShowWindow เช่นเดียวกันแต่มีคุณสมบัติของฟังก์ชันต่างกัน เมื่อวัตถุทั้งสองได้รับข้อความให้ทำงานฟังก์ชัน ShowWindow เหมือนกัน วัตถุทั้งสองจะตอบสนองต่อข้อความในลักษณะเดียวกันแต่มีวิธีการเฉพาะของแต่ละวัตถุที่ไม่เหมือนกัน โดยวัตถุ Cview แสดงรูปวินโดว์อย่างเดียว แต่คลาส CScrollView จะแสดงรูปวินโดว์พร้อมแถบเลื่อนด้านข้าง

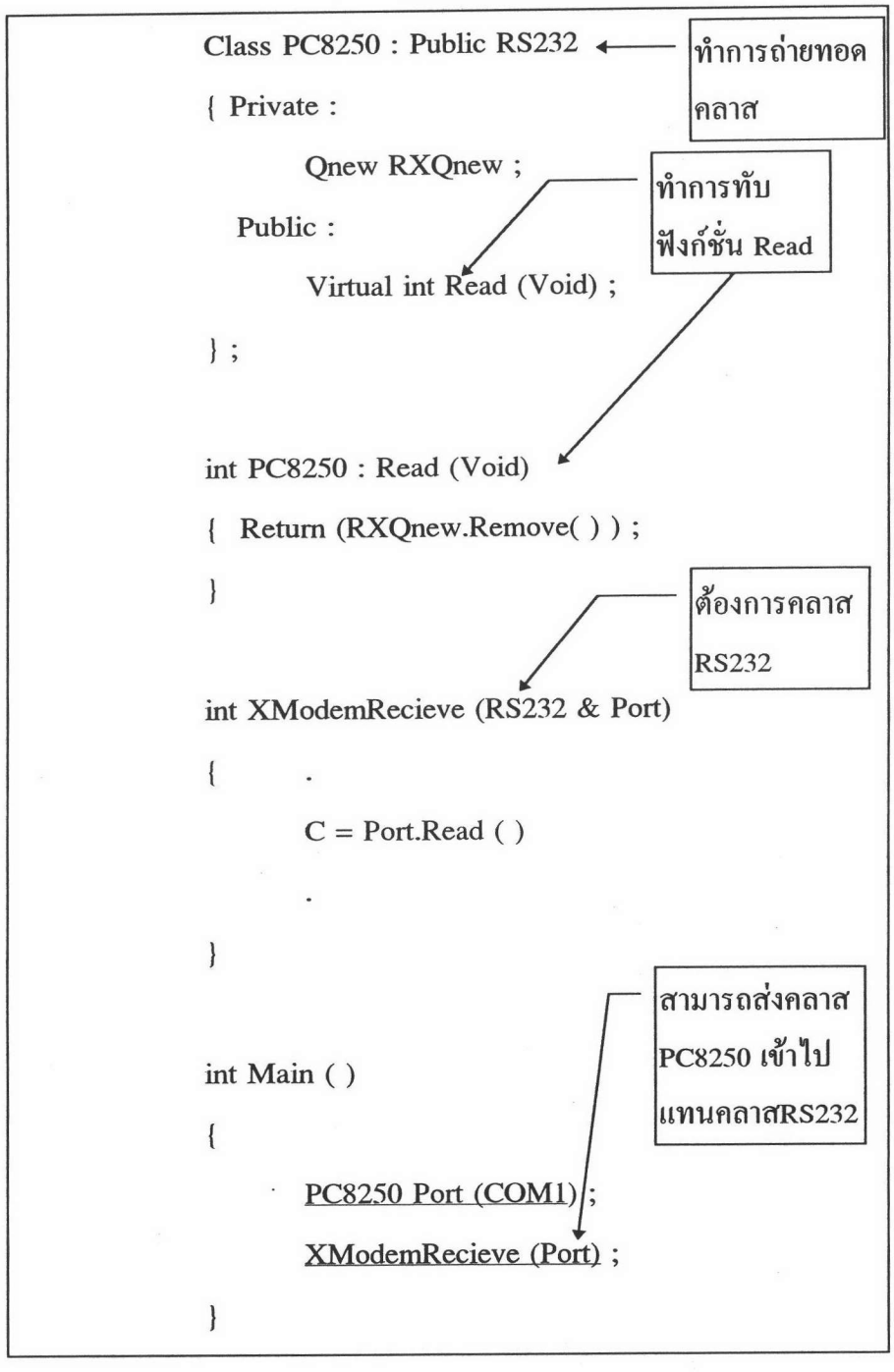


รูปที่ 2.6 แสดงการตอบสนองข้อความเดียวกันของวัตถุ Cview และ CScrollView

Nelson (1992) ได้ยกตัวอย่างการใช้คุณสมบัติ Polymorphism ในการตอบสนองต่อข้อความเดียวกันในภาษา C++ ดังนี้ คลาส RS232 เป็นคลาสพื้นฐาน และมีฟังก์ชัน Read ส่วนคลาส PC8250 เป็นคลาสที่ถ่ายทอดมาจากคลาส RS232 และมีการทับฟังก์ชัน Read ของคลาส RS232 ด้วยฟังก์ชัน Read ของตัวเอง จากรูปที่ 2.7 ฟังก์ชัน XModemRecieve รับตัวแปรชนิดคลาส RS232 และจะทำการส่งข้อความให้คลาส RS232 ทำฟังก์ชัน Read ของตนเอง แต่จากตัวอย่างพบว่า เราสามารถส่งคลาส PC8250 เข้าไปแทนคลาส RS232 ทั้งที่ฟังก์ชัน XModemRecieve ต้องการคลาส RS232



รูปที่ 2.7 แสดงตัวอย่างการใช้คุณสมบัติโพลิมอร์ฟิซึ่ม ในภาษา C++



รูปที่ 2.7 แสดงตัวอย่างการใช้คุณสมบัติโพลิมอร์ฟิซึ่ม ในภาษา C++ (ต่อ)