

บทที่ 1

บทนำ



ปัจจุบันออบเจ็คต์เทคโนโลยีได้เข้ามามีบทบาทกับการพัฒนาโปรแกรมมากขึ้นเรื่อยๆ จะเห็นได้จากการมีภาษาการเขียนโปรแกรมเชิงวัตถุ(Object-Oriented Programming) เกิดขึ้นมากมาย ภาษาซี++ เป็นภาษาหนึ่งที่ได้รับคามนิยมแพร่หลาย การเขียนโปรแกรมเชิงวัตถุจะเกี่ยวข้องกับการออกแบบคลาสและการใช้ออบเจ็คต์ของคลาสต่างๆ

ความสำคัญของปัญหา

ออบเจ็คต์ที่ถูกสร้างขึ้นโดยโปรแกรมก็คือข้อมูลนั่นเอง ซึ่งจะมีประโยชน์มากขึ้นถ้าสามารถนำมาใช้ในการประมวลผลครั้งต่อไป ตามปกติออบเจ็คต์ในโปรแกรมจะถูกทำลายเมื่อการทำงานของโปรแกรมสิ้นสุดลง บางโปรแกรมจำเป็นต้องมีการจัดเก็บและเรียกใช้ออบเจ็คต์ซึ่งยังคงอยู่ได้หลังจากโปรแกรมสิ้นสุดลงเป็นออบเจ็คต์ถาวร (persistent object) มีการนำออบเจ็คต์ในคลาสต่างๆที่มีการจัดเก็บไว้มาใช้ภายหลัง

บางภาษาได้มีแนวทางการแก้ปัญหาซึ่งเหมาะกับงานบางประเภท ตัวอย่างเช่น Smalltalk สามารถเก็บภาพการทำงาน (image) ของสถานะปัจจุบัน รวมทั้งค่าของตัวแปรต่างๆในออบเจ็คต์ของโปรแกรม เมื่อระบบเริ่มต้นทำงานใหม่สถานะภายในของออบเจ็คต์จะถูกเรียกกลับคืนมา แต่บางภาษาไม่ได้รวมความสามารถในการจัดเก็บและเรียกใช้ออบเจ็คต์จากหน่วยเก็บข้อมูล ในซี++ ก็ต้องพิจารณาถึงการจัดเก็บออบเจ็คต์ไว้ใช้ภายหลัง

เนื่องจากออบเจ็คต์เป็นข้อมูลที่มีความยุ่งยากซับซ้อน การจัดเก็บไม่สามารถทำลงบนแฟ้มข้อมูลหรือตารางทั่วไปเช่นข้อมูลแบบพื้นฐานได้ จึงต้องหาวิธีในการจัดเก็บให้เหมาะสม

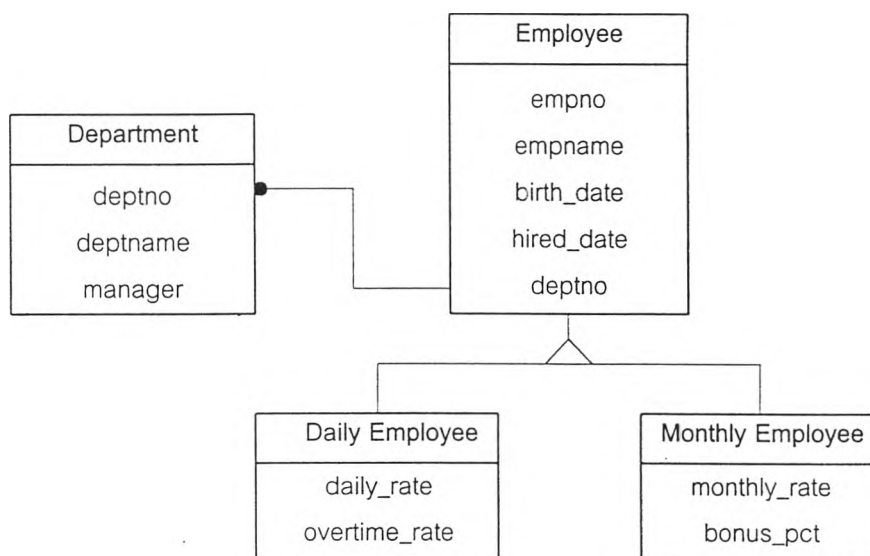
แนวคิดประกอบการวิจัย

การเขียนโปรแกรมเชิงวัตถุใช้แนวคิดของชนิดข้อมูลนามธรรม (abstract data type) ซึ่งเรียกว่าคลาส (classes) ประกอบด้วยสมาชิกที่เป็นข้อมูล (data elements) หรือแอททริบิวต์ (attributes) และสมาชิกที่เป็นฟังก์ชัน (member functions) หรือวิธีการ (methods) ออบเจกต์เป็นตัวแทน (instance) ของคลาส ซึ่งจะมีลักษณะอย่างไรสามารถกำหนดได้จากคลาส สมาชิกของคลาสชนิดข้อมูลจะเก็บสถานะต่างๆของออบเจกต์ และสมาชิกชนิดฟังก์ชันจะใช้ตอบสนองการทำงานต่อออบเจกต์อื่นในระบบ ฟังก์ชันเป็นส่วนที่ออบเจกต์ในคลาสใช้ร่วมกัน ออบเจกต์จึงประกอบด้วยสถานะของออบเจกต์เองและบทบาทที่ตอบสนองต่อออบเจกต์อื่นๆ

การประกาศคลาสมีการกำหนดส่วนที่เป็นสาธารณะ (public) ส่วนป้องกัน(protected) และส่วนตัว (private) ออบเจกต์จะมีการปิดบัง (encapsulate) ข้อมูลที่เป็นส่วนตัว การกระทำต่อออบเจกต์หรือการเข้าถึงข้อมูลของออบเจกต์สามารถทำได้ด้วยฟังก์ชันหรือวิธีการ โดยทั่วไปภาษาการเขียนโปรแกรมเชิงวัตถุมีการใช้ลำดับชั้น(hierarchy) ของชนิดออบเจกต์ด้วยการถ่ายทอด (inheritance) แอททริบิวต์และวิธีการ

แบบเชิงวัตถุ(object model) เป็นพื้นฐานของการออกแบบเชิงวัตถุ ประยุกต์ใช้หลักการนามธรรม การปิดบังข้อมูล การถ่ายทอด และโพลิมอร์ฟิซึม (polymorphism) มีแบบจำลองข้อมูล (data model) ของออบเจกต์อยู่หลายรูปแบบ มาตรฐานที่ใช้ก็ยังไม่เป็นแบบเดียวกัน เช่น ODMG-93 ของ OMG, SQL 3 ของ ANSI

หลักการเชิงวัตถุอาศัยการถ่ายทอดคุณสมบัติ คลาสหนึ่งอาจกำเนิด (derived) มาจากคลาสอื่น เช่น ลูกจ้างแบ่งเป็น 2 ชนิดคือ ลูกจ้างรายวัน (Daily Employee) และลูกจ้างรายเดือน (Monthly Employee) คลาสของลูกจ้างรายวันและลูกจ้างรายเดือนสืบทอดคุณสมบัติ (inherit) ของคลาสลูกจ้าง ได้แก่ เลขประจำตัว ชื่อของลูกจ้าง วันเกิด วันเข้าทำงาน รหัสแผนก และมีการเพิ่มคุณสมบัติเฉพาะของแต่ละคลาส ทั้ง 2 คลาสไม่ได้สืบทอดมาจากคลาสแผนก (Department) แต่คลาสลูกจ้างมีความสัมพันธ์กับคลาสแผนก เมื่อออกแบบคลาสแล้วต้องออกแบบสมาชิกชนิดข้อมูลในคลาส ซึ่งจะรวมทุกสมาชิกข้อมูลในคลาสเอง และสมาชิกข้อมูลที่สืบทอดมาจากคลาสผู้ให้กำเนิด ตัวอย่างการออกแบบคลาสและลำดับชั้นของคลาสแสดงในรูปที่ 1.1



รูปที่ 1.1 แสดงลำดับชั้นของคลาส

ความคงอยู่ถาวร (Persistence)

ความหมายของความคงอยู่ถาวร คือ ข้อมูลยังคงอยู่หลังจากการทำงานของโปรแกรมสิ้นสุดลง ข้อมูล transient จะมีอยู่ในช่วงการทำงานของโปรแกรมเท่านั้น ส่วนข้อมูล persistent คงอยู่จนกระทั่งไม่มีการใช้งานอีกต่อไป

มี 2 วิธีการหลักๆ ในการทำให้ออบเจกต์มีความคงอยู่ถาวร วิธีแรกโดยผ่านการสืบทอดจากคลาสพื้นฐาน (base class) วิธีนี้ต้องเตรียมคลาสซึ่งยอมให้มีข้อมูลที่คงอยู่ถาวรได้ ผู้เขียนโปรแกรมต้องประกาศคลาสซึ่งต้องการคงอยู่ถาวรให้สืบทอดมาจากคลาสพื้นฐานนี้ อีกวิธีหนึ่งโดยการกำหนดเป็นคลาสของหน่วยเก็บข้อมูล วิธีนี้จะง่ายในแง่การใช้งานของผู้เขียนโปรแกรม

มีความแตกต่างของข้อมูลชนิดคงอยู่ถาวรอยู่ 2 แบบคือ แบบแรกข้อมูลจะถูกเก็บบนหน่วยเก็บข้อมูล เมื่อมีการใช้งานจึงนำมาไว้ในหน่วยความจำ แบบที่สองข้อมูลจะอยู่ในหน่วยความจำ และมีการย้ายไปยังหน่วยเก็บข้อมูลเมื่อมีการจัดเก็บหรือสิ้นสุดการทำงาน

ออบเจกต์อาจจะถูกทำให้คงอยู่ถาวรตั้งแต่ตอนสร้างออบเจกต์ขึ้นมา ขึ้นอยู่กับชนิดของออบเจกต์ว่าเป็นแบบ persistent หรือ transient หรือผู้ใช้อาจกำหนดให้ออบเจกต์คงอยู่ถาวรเมื่อต้องการ

การใช้งานออบเจกต์ถาวรต้องการให้ออบเจกต์ที่เก็บอยู่ในหน่วยความจำถูกบันทึกลงหน่วยเก็บข้อมูล เช่น ฮาร์ดดิสก์ หรือดีสเกตต์ และสามารถเรียกออบเจกต์กลับมาใช้ตามชนิดและค่าแอททริบิวต์

วัตถุประสงค์

เพื่อออกแบบและพัฒนาต้นแบบระบบจัดการออบเจกต์ถาวรสำหรับภาษาซี++ สำหรับการจัดการหน่วยเก็บข้อมูลของออบเจกต์ถาวร

ขอบเขตการวิจัย

1. การวิจัยนี้ใช้เทคนิคการเขียนโปรแกรมเชิงวัตถุของภาษาซี++ ทำการออกแบบและพัฒนาต้นแบบระบบจัดการออบเจกต์ถาวร

- เพื่อจัดเก็บออบเจกต์ถาวรลงหน่วยเก็บข้อมูล
- สามารถเรียกออบเจกต์ที่จัดเก็บไว้ขึ้นมาใช้งาน
- สามารถเปลี่ยนแปลงแก้ไขข้อมูลของออบเจกต์และลบออบเจกต์เมื่อไม่ต้องการใช้อีกต่อไปได้

2. การพัฒนาทำบนเครื่องไมโครคอมพิวเตอร์ที่ใช้ระบบปฏิบัติการดอส โดยจัดทำเป็นคลาสไลบรารีของซี++

ขั้นตอนการดำเนินงาน

1. ศึกษาทฤษฎีและเทคนิคต่างๆเกี่ยวกับการเขียนโปรแกรมเชิงวัตถุในภาษาซี++ การจัดการฐานข้อมูลของออบเจกต์ และการจัดเก็บออบเจกต์ถาวร

2. วิเคราะห์และออกแบบระบบจัดการออบเจกต์ถาวรสำหรับภาษาซี++
3. พัฒนาและติดตั้งต้นแบบระบบจัดการออบเจกต์ถาวร
4. ทดสอบและปรับปรุงต้นแบบ
5. สรุปผลการวิจัยและข้อเสนอแนะ

ประโยชน์ที่คาดว่าจะได้รับ

1. ได้ต้นแบบระบบจัดการออบเจกต์ถาวรสำหรับภาษาซี++
2. เป็นแนวทางในการพัฒนาระบบจัดการออบเจกต์ถาวรที่มีความสามารถมากขึ้น และอาจพัฒนาไปสู่ระบบจัดการฐานข้อมูลของออบเจกต์ในระดับสูงขึ้น
3. สามารถนำไปประยุกต์ใช้งานกับโปรแกรมประยุกต์ต่างๆที่มีการออกแบบเชิงวัตถุโดยใช้ภาษาซี++