

บทที่ 4

การทดสอบระบบ

การนำระบบจัดการออบเจกต์ถาวรไปใช้ต้องสร้างโปรแกรมภาษาซี++ให้ทำงานในโหมดคอส โดยต้องมีการรวม (include) เพิ่ม OBJMAN.H เข้าไปด้วย แล้วคอมไพล์และลิงก์เข้ากับไลบรารีของระบบคือ OBJMAN.LIB ในโปรแกรมต้องมีการประกาศฐานข้อมูลของออบเจกต์โดยประกาศออบเจกต์ของคลาส ObjMan คลาสของออบเจกต์ที่ต้องการคงอยู่ถาวรต้องสืบทอดมาจากคลาสพื้นฐาน Persistent โดยการจัดการออบเจกต์ทำผ่านฟังก์ชันของคลาส

ได้ทดสอบระบบจัดการออบเจกต์กับโปรแกรมที่มีการสร้างฐานข้อมูล Personel เพื่อจัดเก็บออบเจกต์ของคลาส Employee, Department และ Project สามารถสอบถามข้อมูลของออบเจกต์ เพิ่ม แก้ไข ลบออบเจกต์ และแสดงออบเจกต์ที่มีอยู่ได้ ตัวอย่างเพิ่มเฮดเดอร์ของโปรแกรม PERSON (Person.h)

```
#ifndef PERSON_H
#define PERSON_H

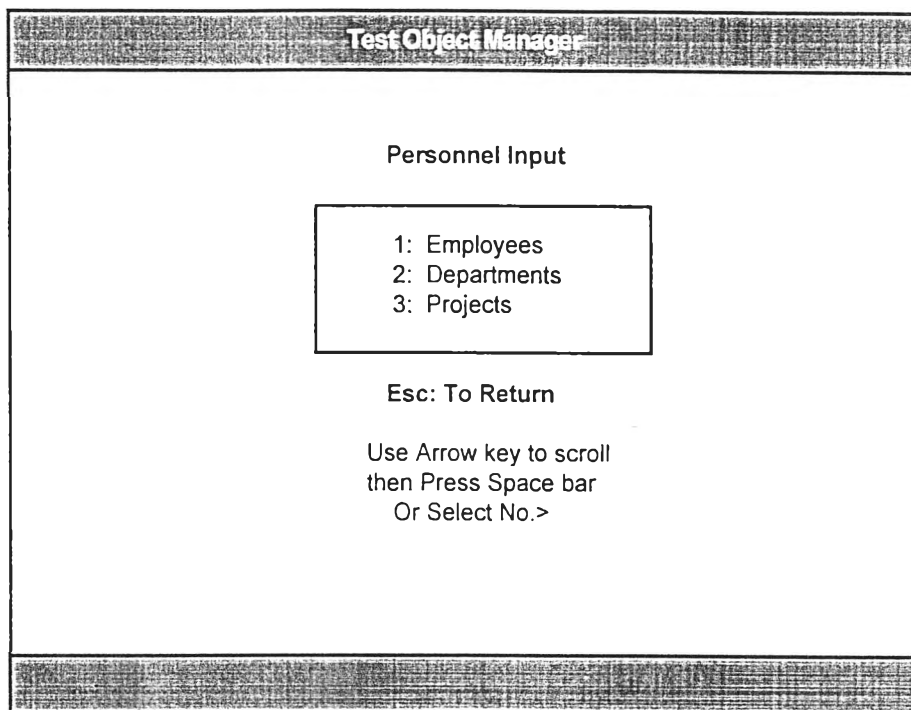
typedef short int EmplNumber;
typedef short int DeptNumber;
typedef short int ProjNumber;

#include "objman.h"
#include "screen.h"
#include "employee.h"
#include "dept.h"
#include "project.h"

extern SCR *scr;

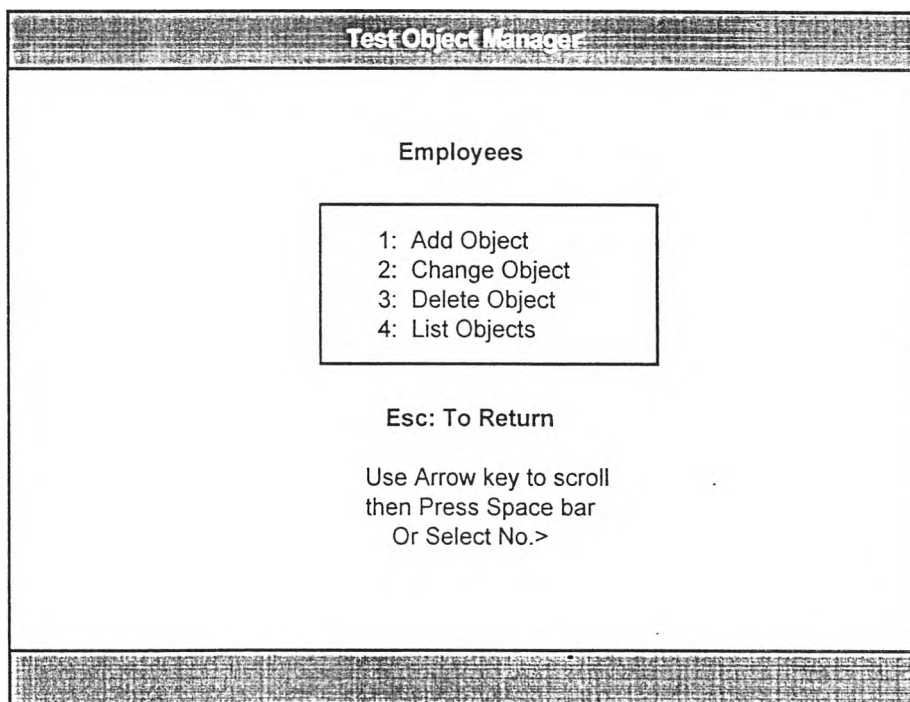
#endif
```

หลังจากคอมไพล์และลิงก์โปรแกรมที่เกี่ยวข้องแล้วจะได้ execute โปรแกรมซึ่งในการทดสอบนี้คือ PERSON.EXE และ INDEX.EXE สามารถเรียกมาทำงานได้ในเครื่องที่ใช้ระบบปฏิบัติการคอสหรือวินโดวส์ 95 ในโหมดคอส เมื่อเรียกโปรแกรม PERSON มาทำงานจะเป็นการสร้างฐานข้อมูล PERSONEL ขึ้นใหม่หรือเปิดฐานข้อมูลที่มีอยู่แล้วมาใช้ และแสดงเมนูหลักของโปรแกรกดังรูปที่ 4.1



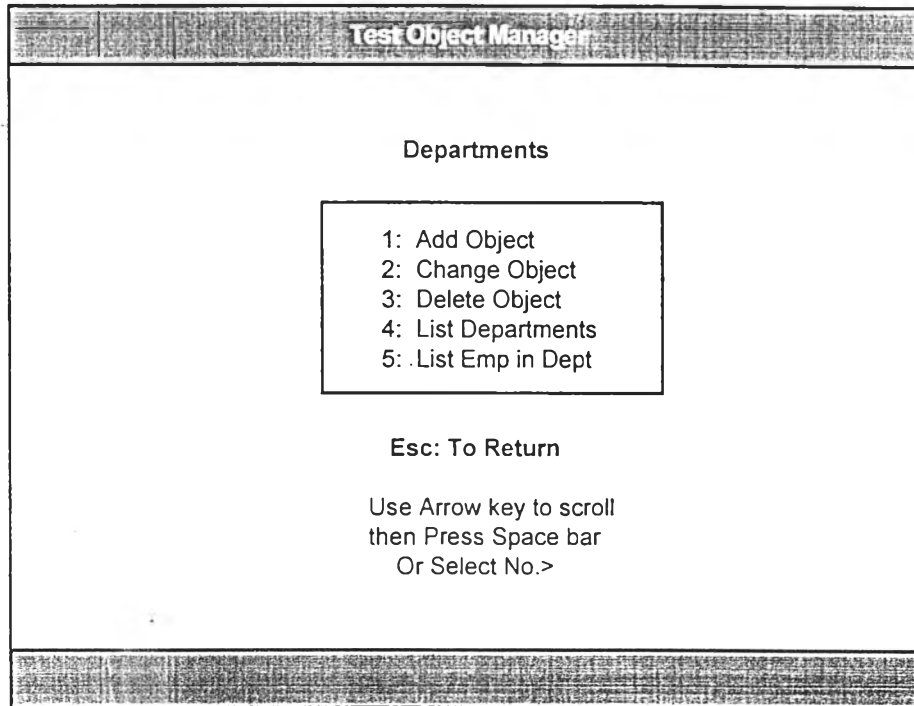
รูปที่ 4.1 แสดงเมนูหลักของโปรแกรม PERSON

เมื่อเลือกรายการจากเมนูหลักก็จะไปยังเมนูถัดไป จากตัวอย่างในรูป 4.2 เป็นเมนูที่เกิดจากการเลือกรายการ EMPLOYEES



รูปที่ 4.2 แสดงเมนู EMPLOYEES

ถ้าเลือกรายการ DEPARTMENTS จากเมนูหลักจะแสดงเมนูดังรูปที่ 4.3



รูปที่ 4.3 แสดงเมนู DEPARTMENTS

การจัดการออบเจกต์ตารางใน โปรแกรมจะเป็นการกระทำกับออบเจกต์ต่างๆของคลาสคู่ได้ จากคำสั่งในโปรแกรม การจัดเก็บออบเจกต์ทำบนแฟ้มข้อมูล PERSONEL.DAT และแฟ้มดัชนี PERSONEL.IDX การป้อนข้อมูลให้ออบเจกต์และการแสดงผลทำจากจอภาพและคีย์บอร์ดผ่าน ออบเจกต์ของคลาสการเชื่อมต่อ SCR ดังตัวอย่างการเพิ่มออบเจกต์ของคลาส Employee และการ แสดงออบเจกต์คลาส Employee ที่มีการจัดเก็บไว้

Enter employee number (0 to quit) :

Name :

Department no. (0 to cancel) :

Emp#	Employee	Dept#
1	Adul	10
2	Adisak	10
3	Chaiwat	20

4	Danai	30
5	Jintana	30

การค้นหาออบเจกต์ที่มีการจัดเก็บไว้สามารถทำได้รวดเร็วเนื่องจากมีการใช้ดัชนีเพื่อหาตำแหน่งที่อยู่ของออบเจกต์ในแฟ้มข้อมูล ถ้าออบเจกต์มีขนาดไม่ใหญ่เกินไปและจำนวนของออบเจกต์ไม่เยอะมากประสิทธิภาพของระบบจะดี ระบบนี้เหมาะกับออบเจกต์ที่มีขนาดของข้อมูลไม่ใหญ่มากนักเพราะการจัดเก็บเป็นลิงค์ลิสต์ของโหนดข้อมูล จำนวนออบเจกต์ที่สามารถเก็บได้ขึ้นอยู่กับจำนวนโหนดที่มีได้และขนาดของออบเจกต์ ซึ่งชนิดของตัวแปรที่เก็บหมายเลขโหนด NodeNo เป็นตัวกำหนดจำนวนโหนดที่มีได้

สิ่งที่ต้องคำนึงถึงในการใช้งาน

จากการทดสอบพบว่าการใช้งานระบบจัดการออบเจกต์ถาวรมีสิ่งที่ต้องคำนึงถึงหลายอย่าง ได้แก่

1. คลาสที่เกี่ยวข้องกัน

ออบเจกต์ในคลาสหนึ่งอาจเกี่ยวข้องกับออบเจกต์อื่นในอีกคลาสด้วยการอ้างอิงถึง ถ้าคลาสมีคีย์รอนชนิดเดียวกันกับคีย์หลักของอีกคลาสความสัมพันธ์ก็จะเกิดขึ้น ซึ่งอาจจะบังคับความสัมพันธ์หรือไม่ก็ได้ เช่น ไม่สามารถลบแผนกได้ถ้ายังมีพนักงานสังกัดอยู่ การบังคับความสัมพันธ์ขึ้นอยู่กับการทำงานของโปรแกรมประยุกต์

2. ออบเจกต์หลายสำเนา

การมีออบเจกต์หลายสำเนาอยู่ในหน่วยความจำ ระบบจัดการออบเจกต์จะไม่รู้ว่าต้องจัดเก็บอันไหนในฐานข้อมูล ดังนั้นจึงควรแน่ใจว่ามีออบเจกต์ถาวรอยู่เพียงสำเนาเดียวในหน่วยความจำ

เมื่อออบเจกต์ถูกสร้างขึ้นจะเพิ่มการอ้างอิงเข้าไปในลิสต์ของออบเจกต์ถาวรที่ถูกทำให้เกิดขึ้น ถ้าทำให้เกิดออบเจกต์ที่มีอยู่แล้วจะเกิดเอ็กเซพชันหรือหยุดการทำงาน เอ็กเซพชันจะให้ตัวชี้ที่มีค่าเป็นตำแหน่งของออบเจกต์แรกที่เกิดขึ้น ฟังก์ชัน `destroy` ในคลาส `Persistent` ใช้ลบออบเจกต์ให้เหลือสำเนาเดียวด้วยการใช้ตำแหน่งที่อยู่ของออบเจกต์เป็นพารามิเตอร์ อาจใช้คำสั่ง `delete` ก็ได้แต่จะมีผลเมื่อดีสตรัคเตอร์ของออบเจกต์ทำงาน ซึ่งทำให้เกิดปัญหาได้ ดังนั้นต้องระวังการทำให้เกิดออบเจกต์ที่มีอยู่แล้ว

3. การเปลี่ยนโครงสร้างคลาสของออบเจกต์ถาวร

ในการใช้งานโปรแกรมอาจมีการเปลี่ยนแปลงข้อกำหนดคลาสของออบเจกต์ถาวร สามารถเพิ่มคลาสใหม่และลบคลาสที่มีอยู่ได้ โดยฐานข้อมูลของออบเจกต์ยังคงใช้ได้ไม่ต้องมีการแก้ไข แต่ให้ระวังในการเพิ่มหรือลบแอททริบิวต์ของคลาส ต้องมีการแก้ไขฟังก์ชันของคลาสด้วย บางครั้งการเพิ่มหรือลบแอททริบิวต์อาจทำให้โครงสร้างของแฟ้มข้อมูลไม่ถูกต้อง ต้องมีการเขียนโปรแกรมเพื่อแปลงแฟ้มข้อมูลให้ถูกต้องใช้งานได้

4. การสร้างแฟ้มดัชนีใหม่

ความสัมพันธ์ระหว่างดัชนีและแฟ้มข้อมูลเป็นสิ่งที่อ่อนแอ ถ้าระบบทำงานผิดพลาด ทำให้ดัชนีไม่ชี้ไปยังออบเจกต์ที่ควรชี้ ฐานข้อมูลอาจจะใช้ไม่ได้ ความผิดพลาดอาจเกิดจากระบบล้มเหลวขณะที่ออบเจกต์และดัชนีกำลังเปลี่ยนแปลง ทำให้โปรแกรมไม่สามารถดึงออบเจกต์ได้ถูกต้อง ซึ่งสิ่งที่ควรทำคือการสร้างแฟ้มดัชนีใหม่ ควรเตรียมโปรแกรมสำหรับการสร้างดัชนีใหม่ให้ออบเจกต์ในฐานข้อมูล เช่น โปรแกรม INDEX ในการทดสอบนี้ ซึ่งเป็นการอ่านแฟ้มข้อมูลของออบเจกต์แล้วสร้างแฟ้มดัชนีใหม่และให้ผลการทำงานดังนี้

Rebuild Index for Personel Database...

Empl:	1	Adul	10	
Dept:	10	Account	1	
Dept:	20	Engineering	0	
Dept:	30	Personnel	5	
Proj:	10	Development	3	180
Empl:	2	Adisak	10	
Empl:	3	Chaiwat	20	
Empl:	4	Danai	30	
Empl:	5	Jintana	30	
Proj:	20	Training	4	90