รายการอ้างอิง

ภาษาอังกฤษ

A. Biliris, S. Dar, N.H. Gehani. Software-Practice and Experience, Vol. 23(12) : Making C++ Objects Persistent. John Wiley & Sons, 1993.

Al Stevens. C++ Database Development. Second Edition. MIS: Press, 1994.

Cattell, R.G.G. (Roderic Geoffrey Galton). Object Data Management : Object - Oriented and Extended Relational Database Systems. Revised Edition. Addison-Wesley Publishing Company, 1994.

Douglas K. Barry. The Object Database Handbook : How to Select, Implement, and Use Object-Oriented Databases. John Wiley & Sons, 1996.

Jiri Soukup. Taming C++ Pattern Classes and Persistence for Large Projects. Addison-Wesley Publishing Company, 1994.

Loomis, Mary E.S. Object Databases: The essentials. Massachusetts: Addison-Wesley Publishing Commpany, 1995.

Paul S. Wang. C++ with Object-oriented Programming. Boston: PWS Publishing Company, 1994.

Peter Coad, David North, Mark Mayfield. Object Models Strategies, Patterns and Applications. Yourdon Press Prentice Hall Building, 1995.

Peter Coad, Jill Nicola. Object-Oriented Programming. New Jersey: Prentice Hall International, 1993.

Tom Swan. Tom Swan's Mastering Borland C++ 5. First Edition. Indianapolis: SAMS Publishing, 1996.

ภาคผนวก ก.
คำอธิบายศัพท์ที่ใช้สำหรับโปรแกรมเชิงวัตถุ

ออบเจ็คต์	การรวบรวมกลุ่มข้อมูลและการทำงานต่างๆเข้าเป็นหน่วยเดียวกัน
คลาส	ชนิดหรือรูปแบบของออบเจ็คต์
การสืบทอด (inheritance)	การสืบทอดคลาสใหม่จากคลาสพ่อ (parent class) ที่มีอยู่เดิม โดยสามารถใช้ข้อมูลและวิธีการที่มีอยู่เดิมในคลาสพ่อ และสามารถเพิ่มเติมใหม่ได้อีก
การปิดบังข้อมูล (encapsulation)	ความสามารถในการปกปิดข้อมูลภายใน ขณะที่มีการเชื่อมโยงกับภายนอกตามการกำหนดของผู้ใช้งาน
โพลิมอร์ฟีซึม (polymorphism)	การส่งข่าวสารอันเดียวกันไปยังออบเจ็คต์ต่างๆ ออบเจ็คต์เหล่านั้นจะตอบสนองต่อข่าวสารที่ได้รับแตกต่างกัน
พลับบลิค (public)	ส่วนของข้อมูลหรือฟังก์ชันซึ่งยอมให้ใช้เป็นสาธารณะทั่วไปได้
โพรเทค (protected)	ส่วนที่ป้องกันสำหรับสมาชิกในคลาสนั้น คลาสที่สืบทอด หรือคลาสที่เป็นเฟรนด์กับมัน
ไพรเวท (private)	ส่วนของข้อมูลหรือฟังก์ชันที่ใช้เป็นส่วนตัวสำหรับสมาชิกในคลาสนั้นหรือคลาสที่เป็นเฟรนด์
คอนสตรัคเตอร์ (constructor)	ฟังก์ชันที่ถูกเรียกใช้ตอนสร้างออบเจ็คต์ขึ้นมา เป็นการกำหนดค่าเริ่มต้นสำหรับโครงสร้างและสมาชิกของออบเจ็คต์
ดีคอนสตรัคเตอร์ (deconstructor)	ฟังก์ชันที่ถูกเรียกตอนออบเจ็คต์ถูกทำลาย เป็นการยกเลิกการกำหนดค่าต่างๆสำหรับโครงสร้างและสมาชิกของวัตถุเมื่อเลิกใช้งาน
เฟรนด์ (friends)	คีย์เวิร์ดสำหรับคลาสหรือฟังก์ชัน ซึ่งอนุญาตให้คลาสหรือฟังก์ชันอื่นๆสามารถเข้าถึงข้อมูลหรือฟังก์ชันส่วนตัวหรือส่วนป้องกันได้
สแตติก (static)	จะมีตัวแปรเพียงตัวเดียวที่กำหนดขึ้นมาเป็นของคลาส ไม่ต้องกำหนดให้แต่ละออบเจ็คต์
เวอร์ชวล (virtual)	สำหรับคลาสกรณีการสืบทอดที่เป็นเวอร์ชวลจะทำให้ใช้คลาสมูลฐานร่วมกัน ส่วนการประกาศฟังก์ชันในคลาสมูลฐานให้เป็นเวอร์ชวลจะทำให้ใช้ฟังก์ชันในคลาสที่สืบทอดก่อนในคลาสมูลฐาน

เทมเพลท (template)	โครงร่างของโปรแกรมที่มีการรับค่าพารามิเตอร์ของชนิดข้อมูล ซี++ ใช้เทมเพลทสำหรับการประกาศคลาสหรือฟังก์ชัน ทำให้การใช้งานยืดหยุ่นขึ้นอยู่กับชนิดของพารามิเตอร์

ภาคผนวก ข.
การติดตั้งและการใช้งาน

1. การติดตั้งระบบ

การนำระบบจัดการออบเจ็คต์ถาวรไปใช้ต้องเตรียมการดังนี้

1. เครื่องคอมพิวเตอร์ที่มี CPU 80486 หรือสูงกว่า มีหน่วยความจำอย่างน้อย 8 M ขึ้นไป ใช้ระบบปฏิบัติการ MS ดอสเวอร์ชัน 6.0 ขึ้นไป หรือเป็นวินโดว์ส 95 ทำงานในโหมดดอส

2. ก่อนที่จะให้โปรแกรมทำงานต้องแน่ใจว่าในแฟ้ม CONFIG.SYS ของเครื่องติดตั้งดีไวซ์ไดรเวอร์ ANSI.SYS ไว้แล้ว มิเช่นนั้นจอภาพอาจแสดงผลได้ไม่ถูกต้อง

3. ถ้าต้องมีการคอมไพล์โปรแกรมบนเครื่องนั้นใหม่ ให้ใช้โปรแกรมบอร์แลนด์ซี++ เวอร์ชัน 4.X

2. ไลบรารีของระบบจัดการออบเจ็คต์ถาวร

ไลบรารีของระบบจัดการออบเจ็คต์ถาวรมีชื่อว่า OBJMAN.LIB แฟ้มข้อมูลที่รวมอยู่ในไลบรารีได้แก่

1. Objman.h, Objman.cpp
2. Type.h, Type.cpp
3. Linklist.h
4. Key.h, Key.cpp
5. Node.h, Node.cpp
6. Tnode.cpp
7. Btree.h, Btree.cpp
8. Screen.h, Screen.cpp

3. การเปลี่ยนแปลงแก้ไขค่าที่กำหนดการทำงาน

ค่าคงที่และตัวแปรต่างๆซึ่งสามารถแก้ไขให้การทำงานของระบบจัดการออบเจ็คต์ถาวรได้แก่

Constant	Description	Default	Header File
SCREENWIDTH	ความกว้างของส่วนแสดงผล	80	Screen.h
SCREENHEIGHT	ความสูงของส่วนแสดงผล	25	Screen.h
FCOLOR	สีของตัวอักษร	White	Screen.h
LCOLOR	สีของเส้น	Lightgray	Screen.h
SCRNAME	ชื่อจอแสดงผล		Screen.h
nodelenght	ขนาดของโหนดข้อมูล	64	Objman.h

ตารางแสดงค่าคงที่และตัวแปรที่ใช้

4. การคอมไพล์โปรแกรม

ถ้ามีการเปลี่ยนแปลงแก้ไขโปรแกรมหรือสร้างโปรแกรมประยุกต์ขึ้นใหม่ อาจต้องมีการคอมไพล์โปรแกรมในส่วนที่เกี่ยวข้องใหม่ แฟ้มข้อมูลที่ใช้ในการคอมไพล์ได้แก่

1. OBJMAN.BLD ใช้ในการสร้างไลบรารี OBJMAN.LIB สำหรับระบบจัดการออบเจ็คต์ถาวรแสดงได้ดังนี้

```
-+ objman.obj    &
-+ btree.obj     &
-+ key.obj       &
-+ tnode.obj     &
-+ node.obj      &
-+ screen.obj      &
-+ type.obj
```

2. MAKEFILE ใช้ในการคอมไพล์และลิงค์โปรแกรมของระบบจัดการออบเจ็คต์ รวมทั้งโปรแกรมใช้งานมีตัวอย่างดังนี้

```
# ============================================
# makefile - Borland C++
# ============================================
DEBUG = -v -vi
MODEL = -ml
CMPL = $(DEBUG) $(MODEL)
```

```
.cpp.obj :
    bcc $(CMPL) -c {$* }
all :  person.exe   \
       index.exe
       echo all done

# ========================================
# define the Personnel system class library
# ========================================
AOBJ = employee.obj dept.obj project.obj
PLINK = $&.obj $(AOBJ)

index.exe : index.obj $(AOBJ) objman.lib
    bcc $(CMPL) $(PLINK) objman.lib
person.exe : person.obj $(AOBJ) objman.lib
    bcc $(CMPL) $(PLINK) objman.lib

# Build  OBJMAN.LIB class library
objman.lib : objman.obj   \
          btree.obj    \
          key.obj      \
          tnode.obj    \
          node.obj     \
          screen.obj   \
          type.obj
   tlib objman @objman.bld
```

5. ตัวอย่างการนำระบบจัดการออบเจ็คต์ถาวรไปใช้

สร้างโปรแกรมใช้งานที่มีการสร้างฐานข้อมูล Personel เพื่อจัดเก็บออบเจ็คต์ของคลาส Employee, Department และ Project แฟ้มข้อมูลสำหรับโปรแกรมตัวอย่างได้แก่

1. Person.h, Person.cpp
2. Employee.h, Employee.cpp
3. Dept.h, Dept.cpp
4. Project.h, Project.cpp
5. Index.cpp ใช้สำหรับสร้างแฟ้มดัชนีใหม่

เมื่อคอมไพล์โปรแกรมแล้วจะได้โปรแกรมที่สามารถทำงานได้เป็น .EXE จากตัวอย่างคือ PERSON.EXE และ INDEX.EXE ซึ่งสามารถเรียกมาใช้งานด้วยการเรียกที่พร็อมพ์ของคำสั่งในโหมดดอส

ภาคผนวก ค.
คลาสในไลบรารี

Class Persistent

Attributes :

ObjectHeader objhdr	ส่วนหัวของออบเจ็คต์
ObjAddr objectaddr	ตำแหน่งที่อยู่โหนดของออบเจ็คต์
Objman& objman	ฐานข้อมูลที่เก็บออบเจ็คต์
short int indexcount	จำนวนคีย์ในออบเจ็คต์
short int instances	จำนวนอินสแตนท์
Node *node	โหนดปัจจุบันสำหรับการอ่านและเขียน
short int offset	ตำแหน่งตัวอักษรปัจจุบัน
bool restored	มีการเรียก Restore
bool stored	มีการเรียก Store
bool changed	ออบเจ็คต์มีการเปลี่ยนแปลงหรือไม่
bool deleted	ออบเจ็คต์ถูกลบหรือไม่
bool added	ออบเจ็คต์ถูกเพิ่มหรือไม่
static bool usenew	ออบเจ็คต์ถูกสร้างด้วย new
streampos filepos	ตำแหน่งในแฟ้ม
static Persistent *objconstructed	ตัวชี้ไปยังคีย์ของออบเจ็คต์ที่ถูกสร้าง
static Persistent *objdestroyed	ตัวชี้ไปยังคีย์ของออบเจ็คต์ที่ถูกทำลาย
LinkList<ObjKey> keys	ลิสต์ของคีย์ที่เกี่ยวข้องกับออบเจ็คต์

Methods :

Persistent()

virtual ~Persistent()

virtual void Write() = 0

virtual void Read() = 0

void Restore(ObjAddr nd = 0)

void Store()

bool AddObj()

bool ChangeObj()

bool DeleteObj()

bool ObjectExist() const

Persistent& FindObj(ObjKey *key)

Persistent& CurrentObj(ObjKey *key = 0)

Persistent& FirstObj(ObjKey *key = 0)

Persistent& LastObj(ObjKey *key = 0)

Persistent& NextObj(ObjKey *key = 0)

Persistent& PrevObj(ObjKey *key = 0)

ObjAddr ObjectAddress() const

static void Destroy(Persistent *pp)

template <class T> void ReadObj(T& t)

template <class T> void WriteObj(const T& t)

Class Objman

Attributes :

DataFile datafile	แฟ้มข้อมูลของออบเจ็คต์
IndexFile indexfile	แฟ้มดัชนีของออบเจ็คต์
LinkList<Persistent> objects	ลิสต์ของออบเจ็คต์ที่ถูกทำให้เกิดขึ้น
LinkList<Class> classes	ลิสต์ของคลาสที่ลงทะเบียนแล้ว
LinkList<ObjBtree> btrees	ไบนารีทรีของฐานข้อมูล
ObjAddr rebuildnode	อินเด็กซ์กำลังถูกสร้างใหม่
static Objman *opendatabase	ฐานข้อมูลที่เปิดใช้งาน

Methods :

Objman(const string& name)

~Objman()

static Objman *OpenDatabase()

void GetObjectHeader(ObjAddr nd, ObjectHeader& objhdr)

ClassID GetClassID(const char *classname)

void RebuildIndexes(ObjAddr nd)

Class_ObjKey

Attributes :

NodeNo fileaddr	ตำแหน่งที่อยู่ของออบเจ็คต์
IndexNo indexno	หมายเลขคีย์ 0 = คีย์หลัก, >0 = คีย์รอง
KeyLength keylength	ความยาวคีย์

Methods :

ObjKey(NodeNo fa = 0)

virtual ~ObjKey()

virtual void ReadKey(IndexFile& bfile)

virtual void WriteKey(IndexFile& bfile)

virtual bool isNull() const

virtual ObjKey *MakeKey() const

virtual int operator > (const ObjKey& key) const

virtual int operator == (const ObjKey& key) const

virtual ObjKey& operator = (const ObjKey& key)

void Relate(const Type_info *ti)

KeyLength GetKeyLength() const

void SetKeyLength(KeyLength kylen)

Class_template <class T> class Key

Attributes : T ky ค่าของคีย์

Methods :

Key(const T& key)

virtual ~Key()

const T& KeyValue( ) const

T& KeyValue()

void SetKey(const T& key)

ประวัติผู้วิจัย

น.ส. เจตนา อัครวงษ์ เกิดวันที่ 3 กุมภาพันธ์ 2510 ที่จังหวัดนครสวรรค์ สำเร็จการศึกษาระดับปริญญาตรี วิศวกรรมศาสตร์บัณฑิต สาขาวิชาคอมพิวเตอร์ จากสถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหารลาดกระบัง ในปีการศึกษา 2531 และเข้าศึกษาต่อในหลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2537 ปัจจุบันทำงานที่การไฟฟ้าฝ่ายผลิตแห่งประเทศไทย