

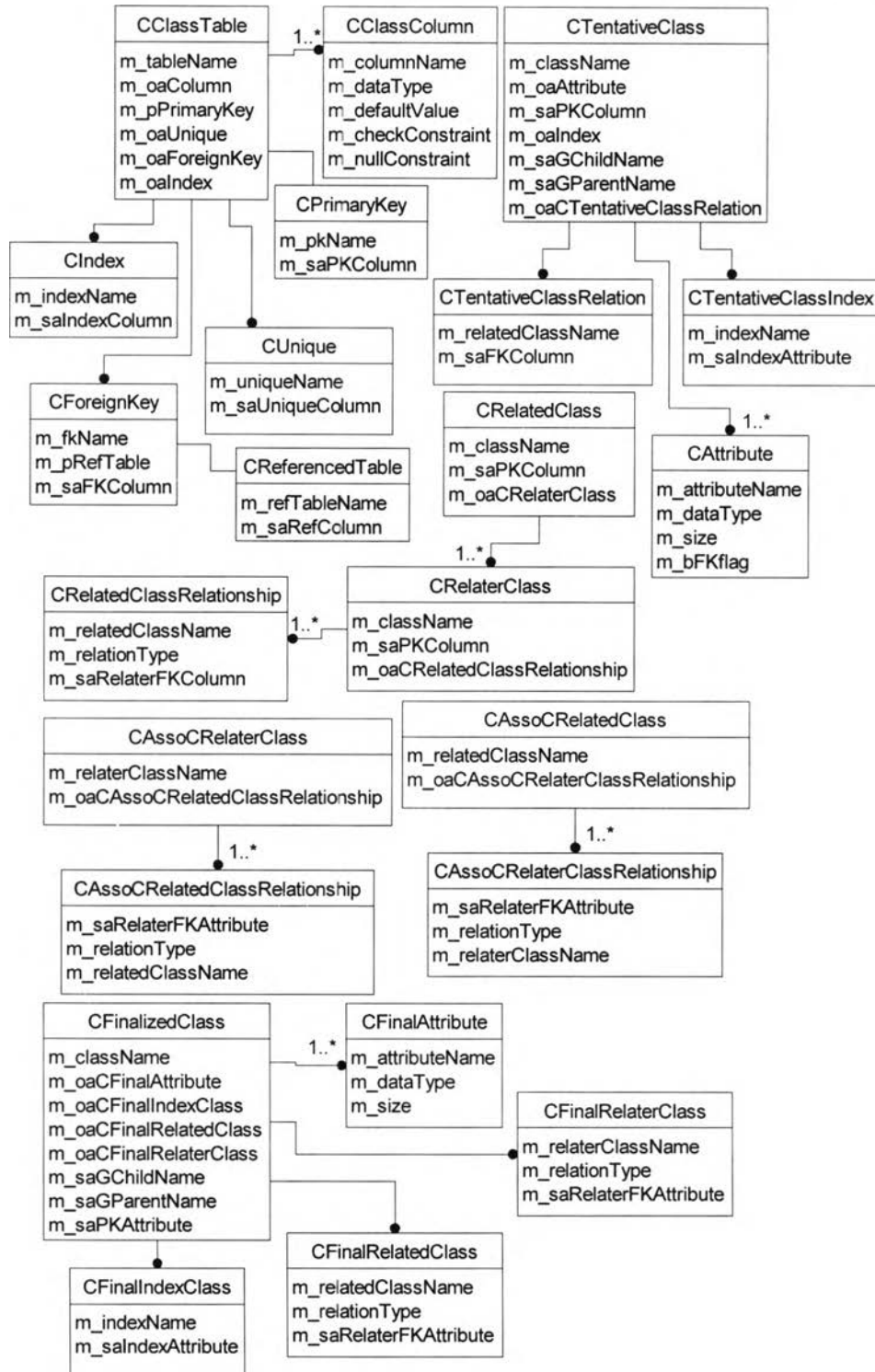


การพัฒนาระบบแปลงฐานข้อมูลเชิงสัมพันธ์เป็นฐานข้อมูลเชิงวัตถุ

จากการวิเคราะห์และออกแบบระบบแปลงฐานข้อมูลเชิงสัมพันธ์เป็นฐานข้อมูลเชิงวัตถุ ในบทที่ 3 โปรแกรมที่จะต้องพัฒนาแยกเป็นกลุ่มของคลาสดังนี้

1. กลุ่มคลาสกระจายสคริปต์
2. กลุ่มคลาสเตรียมแผนแบบเชิงวัตถุเบื้องต้น
3. กลุ่มคลาสจำแนกคลาส
4. กลุ่มคลาสดำเนินการสืบทอดคุณสมบัติ
5. กลุ่มคลาสดำเนินการหาความเกี่ยวพัน
6. กลุ่มคลาสปรับแผนแบบเชิงวัตถุขั้นสุดท้าย
7. กลุ่มคลาสสร้างแฟ้มข้อความ HCD และสร้างฐานข้อมูล

ทุกกลุ่มคลาสใช้ Microsoft Visual C++ version 6.0 เป็นเครื่องมือในการพัฒนา



รูปที่ 4.1 คลาสต่างๆในระบบแปลงฐานข้อมูลเชิงสัมพันธ์เป็นฐานข้อมูลเชิงวัตถุ

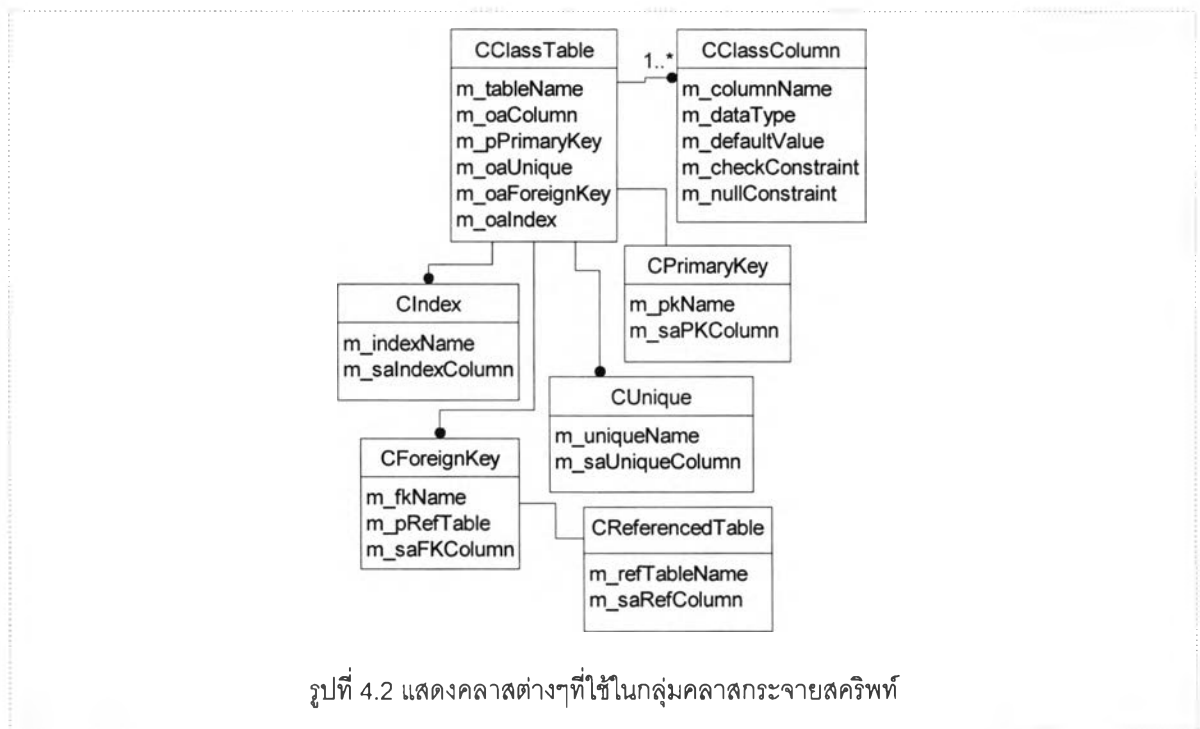
รูปที่ 4.1 แสดงคลาสต่างๆและความสัมพันธ์ระหว่างคลาส ซึ่งใช้ในการพัฒนาโปรแกรมแปลงฐานข้อมูลเชิงสัมพันธ์เป็นฐานข้อมูลเชิงวัตถุ

4.1 กลุ่มคลาสกระจายสคริปต์

วัตถุประสงค์เพื่อกระจายสคริปต์ (ซึ่งอยู่ในรูปของแฟ้มข้อความ) และคัดเลือกรายละเอียดที่จำเป็นต้องใช้ นำมาสร้างเป็นอ็อบเจกต์สำหรับรายละเอียดนั้นเพื่อความสะดวกในการนำไปใช้ในขั้นตอนต่อไป

โปรแกรมจะมีการสร้างอ็อบเจกต์ต่างๆจากนิยามคลาสดังต่อไปนี้

- CClassTable
- CClassColumn
- CPrimaryKey
- CUnique
- CIndex
- CForeignKey
- CReferencedTable



รูปที่ 4.2 แสดงคลาสต่างๆที่ใช้ในกลุ่มคลาสกระจายสคริปต์

รูปที่ 4.2 แสดงความสัมพันธ์ระหว่างคลาสต่างๆซึ่งจำเป็นต้องใช้ในการกระจายสคริปต์

อ็อบเจกต์ CClassTable

เป็นอ็อบเจกต์สำหรับตาราง สร้างจากนิยามคลาส CClassTable ประกอบด้วยแอททริบิวต์ต่างๆดังนี้

- m_tableName - ใช้เก็บชื่อตาราง
- m_oaColumn - เป็นแอเรย์ของพอยท์เตอร์ ใช้เก็บพอยท์เตอร์ชี้ไปยังอ็อบเจกต์ CClassColumn
- m_pPrimaryKey - ใช้เก็บพอยท์เตอร์ชี้ไปยังอ็อบเจกต์ CPrimaryKey
- m_oaUnique - เป็นแอเรย์ของพอยท์เตอร์ ใช้เก็บพอยท์เตอร์ชี้ไปยังอ็อบเจกต์ CUnique
- m_oaForeignKey - เป็นแอเรย์ของพอยท์เตอร์ ใช้เก็บพอยท์เตอร์ชี้ไปยังอ็อบเจกต์ CForeignKey
- m_oaIndex - เป็นแอเรย์ของพอยท์เตอร์ ใช้เก็บพอยท์เตอร์ชี้ไปยังอ็อบเจกต์ CIndex

อ็อบเจกต์สำหรับตารางที่ได้เก็บในตัวแปรแอเรย์ของอ็อบเจกต์ m_oaAllTables

อ็อบเจกต์ CClassColumn

เป็นอ็อบเจกต์สำหรับคอลัมน์ สร้างจากนิยามคลาส CClassColumn ประกอบด้วยแอททริบิวต์ต่างๆดังนี้

- m_columnName - ใช้เก็บชื่อคอลัมน์
- m_dataType - ใช้เก็บประเภทข้อมูลและขนาดของคอลัมน์นั้นๆ
- m_defaultValue - ใช้เก็บค่าปริยายของข้อมูลในคอลัมน์นั้นๆ
- m_checkConstraint - ใช้เก็บข้อกำหนดของค่าที่เก็บในคอลัมน์นั้นๆ
- m_nullConstraint - ใช้เก็บข้อกำหนดว่างที่เก็บในคอลัมน์นั้นๆ

อ็อบเจกต์ CPrimaryKey

เป็นอ็อบเจกต์สำหรับคีย์หลัก สร้างจากนิยามคลาส CPrimaryKey ประกอบด้วยแอททริบิวต์ต่างๆดังนี้

- m_pkName - ใช้เก็บชื่อคีย์หลัก
- m_saPKColumn - เป็นแอเรย์ของสตริง ใช้เก็บชื่อคอลัมน์ที่ประกอบเป็นคีย์หลัก

อ็อบเจกต์ CUnique

เป็นอ็อบเจกต์สำหรับ Unique Constraint สร้างจากนิยามคลาส CUnique ประกอบด้วยแอททริบิวต์ต่างๆดังนี้

- m_uniqueName - ใช้เก็บชื่อ Unique Constraint
- m_saUniqueColumn - เป็นแอเรย์ของสตริง ใช้เก็บชื่อคอลัมน์ที่ประกอบเป็น Unique Constraint

อ็อบเจกต์ CIndex

เป็นอ็อบเจกต์สำหรับดรรชนี สร้างจากนิยามคลาส CIndex ประกอบด้วยแอททริบิวต์ต่างๆดังนี้

- m_indexName - ใช้เก็บชื่อดรรชนี
- m_saIndexColumn - เป็นแอเรย์ของสตริง ใช้เก็บชื่อคอลัมน์ที่ประกอบเป็นดรรชนี

อ็อบเจกต์ CForeignKey

เป็นอ็อบเจกต์สำหรับคีย์นอก สร้างจากนิยามคลาส CForeignKey ประกอบด้วยแอททริบิวต์ต่างๆดังนี้

- m_fkName - ใช้เก็บชื่อคีย์นอก
- m_pRefTable - เป็นพอยน์เตอร์ชี้ไปยังตารางที่อ้างอิงถึง
- m_saFKColumn - เป็นแอเรย์ของสตริง ใช้เก็บชื่อคอลัมน์ที่ประกอบเป็นคีย์นอก

อ็อบเจกต์ CReferencedTable

อ็อบเจกต์สำหรับตาราง ซึ่งถูกอ้างอิงโดยคีย์นอกของอ็อบเจกต์ CClassTable สร้างจากนิยามคลาส CReferencedTable ประกอบด้วยแอททริบิวต์ต่างๆดังนี้

- m_refTableName - ใช้เก็บชื่อตาราง ซึ่งถูกอ้างอิงโดยคีย์นอกของอ็อบเจกต์ CClassTable
- m_saRefColumn - เป็นแอเรย์ของสตริง ใช้เก็บชื่อคอลัมน์ที่ประกอบเป็นคีย์หลักของตาราง ซึ่งถูกอ้างอิงโดยคีย์นอกของอ็อบเจกต์ CClassTable

ขั้นตอนการกระจายสคริปต์

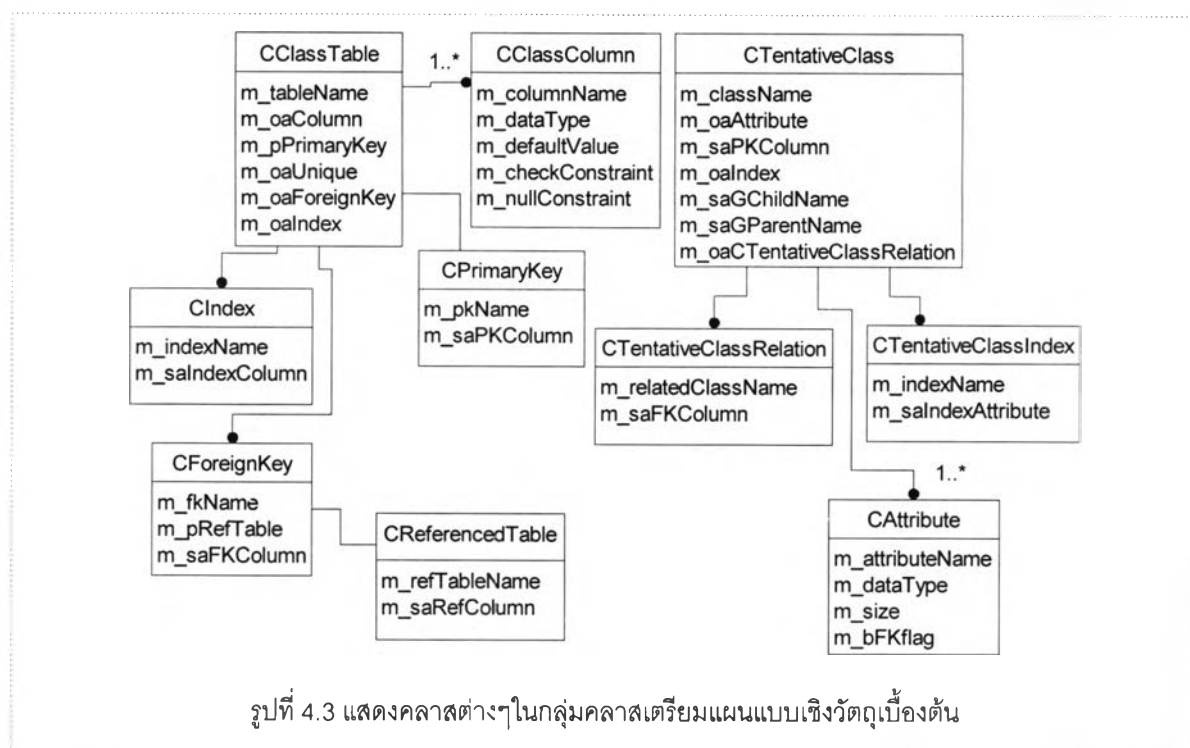
1. สร้างอ็อบเจกต์จากนิยามคลาส CClassTable
 - กำหนดค่าให้แอททริบิวต์ m_tableName เป็นชื่อตาราง
2. สร้างอ็อบเจกต์สำหรับคอลัมน์ในตารางจากนิยามคลาส CClassColumn
 - กำหนดค่าให้ m_columnName เป็นชื่อคอลัมน์ในตาราง
 - กำหนดค่าให้ m_dataType เป็นประเภทข้อมูลพร้อมขนาดของข้อมูลที่เก็บในคอลัมน์นั้น
 - กำหนดค่าให้ m_defaultValue เป็นค่าโดยปริยายสำหรับคอลัมน์
 - กำหนดค่าให้ m_checkConstraint เป็นข้อกำหนดสำหรับค่าที่เก็บในคอลัมน์
 - กำหนดค่าให้ m_nullConstraint เป็นข้อกำหนดว่างสำหรับคอลัมน์
3. เก็บพอยน์เตอร์ของอ็อบเจกต์ CClassColumn ลงในแอททริบิวต์ m_oaColumn ของอ็อบเจกต์ CClassTable
4. สร้างอ็อบเจกต์สำหรับคีย์หลักในตารางจากนิยามคลาส CPrimaryKey
 - กำหนดค่าให้ m_pkName เป็นชื่อคีย์หลักของตาราง
 - กำหนดค่าให้ m_saPKColumn เป็นชื่อคอลัมน์ที่ประกอบเป็นคีย์หลัก
5. เก็บพอยน์เตอร์ของอ็อบเจกต์ CPrimaryKey ลงในแอททริบิวต์ m_pPrimaryKey ของอ็อบเจกต์ CClassTable
6. สร้างอ็อบเจกต์สำหรับ Unique Constraint ในตารางจากนิยามคลาส CUnique
 - กำหนดค่าให้ m_uniqueName เป็นชื่อ Unique Constraint ของตารางนั้นๆ
 - กำหนดค่าให้ m_saUniqueColumn เป็นชื่อคอลัมน์ที่ประกอบเป็น Unique Constraint
7. เก็บพอยน์เตอร์ของอ็อบเจกต์ CUnique ลงในแอททริบิวต์ m_oaUnique ของอ็อบเจกต์ CClassTable
8. สร้างอ็อบเจกต์สำหรับคีย์นอกในตารางจากนิยามคลาส CForeignKey
 - กำหนดค่าให้ m_fkName เป็นชื่อคีย์นอกของตาราง
 - กำหนดค่าให้ m_saFKColumn เป็นชื่อคอลัมน์ที่ประกอบเป็นคีย์นอก
9. สร้างอ็อบเจกต์สำหรับตารางที่ถูกอ้างอิงโดยคีย์นอก จากนิยามคลาส CReferencedTable

- กำหนดค่าให้ `m_refTableName` เป็นชื่อตารางที่ถูกอ้างอิงโดยคีย์นอก
 - กำหนดค่าให้ `m_saRefColumn` เป็นชื่อคอลัมน์ที่ประกอบเป็นคีย์หลักของตารางที่ถูกอ้างอิง
10. เก็บพอยน์เตอร์ของอ็อบเจกต์ `CReferencedTable` ลงในแอททริบิวต์ `m_pRefTable` ของอ็อบเจกต์ `CForeignKey`
 11. เก็บพอยน์เตอร์ของอ็อบเจกต์ `CForeignKey` ลงในแอททริบิวต์ `m_oaForeignKey` ของอ็อบเจกต์ `CClassTable`
 12. เก็บพอยน์เตอร์ของอ็อบเจกต์สำหรับตาราง `CClassTable` ลงในตัวแปรแอดเรสของอ็อบเจกต์ `m_oaAllTables`

4.2 กลุ่มคลาสเตรียมแผนแบบเชิงวัตถุเบื้องต้น

วัตถุประสงค์ เพื่อสร้างอ็อบเจกต์สำหรับคลาสและอ็อบเจกต์ต่างๆที่เกี่ยวข้อง จากอ็อบเจกต์สำหรับตาราง (`CClassTable`) และอ็อบเจกต์อื่นๆที่เกี่ยวข้องกับตาราง โดยจะมีการสร้างอ็อบเจกต์สำหรับคลาสและอ็อบเจกต์ต่างๆที่เกี่ยวข้องจากนิยามคลาสดังต่อไปนี้

- `CTentativeClass`
- `CAttribute`
- `CTentativeClassRelation`
- `CTentativeClassIndex`



รูปที่ 4.3 แสดงคลาสต่างๆและความสัมพันธ์ระหว่างคลาส ซึ่งจำเป็นต้องใช้ในการเตรียมแผนแบบเชิงวัตถุเบื้องต้น โปรแกรมจะอ่านอ็อบเจกต์ CClassTable ที่ละอ็อบเจกต์ อ่านอ็อบเจกต์ CClassColumn อ็อบเจกต์ CPrimaryKey อ็อบเจกต์ CUnique อ็อบเจกต์ CIndex อ็อบเจกต์ CForeignKey และอ็อบเจกต์ CClassColumn ที่สัมพันธ์กับอ็อบเจกต์ CClassTable นำมาสร้างอ็อบเจกต์สำหรับคลาสและอ็อบเจกต์ต่างๆที่สัมพันธ์กับอ็อบเจกต์สำหรับคลาสนั้นๆ

อ็อบเจกต์ CTentativeClass

อ็อบเจกต์สำหรับคลาส สร้างจากนิยามคลาส CTentativeClass ประกอบด้วยแอททริบิวต์ต่างๆดังนี้

- m_className - ใช้เก็บชื่อคลาส ซึ่งได้จากแอททริบิวต์ m_tableName ของอ็อบเจกต์ CClassTable
- m_oaAttribute - เป็นแอเรย์ของพอยท์เตอร์ ใช้เก็บพอยท์เตอร์ที่ไปยังอ็อบเจกต์ CAttribute
- m_saPKColumn - เป็นแอเรย์ของสตริง ใช้เก็บชื่อคอลัมน์ที่ประกอบเป็นคีย์หลักได้จากแอททริบิวต์ m_saPKColumn ของอ็อบเจกต์ CPrimaryKey
- m_oaIndex - เป็นแอเรย์ของพอยท์เตอร์ ใช้เก็บพอยท์เตอร์ที่ไปยังอ็อบเจกต์ CTentativeClassIndex
- m_saGChildName - เป็นแอเรย์ของสตริง ใช้เก็บชื่อคลาสลูกสำหรับคลาสนี้

- m_saGParentName - เป็นแอเรียของสตริง ใช้เก็บชื่อคลาสพ่อสำหรับคลาสนี้
- m_oaCTentativeClassRelation - เป็นแอเรียของพอยท์เตอร์ ใช้เก็บพอยท์เตอร์
ที่ไปยังอ็อบเจกต์ CTentativeClassRelation

อ็อบเจกต์สำหรับคลาสที่ได้เก็บในตัวแปรแอเรียของอ็อบเจกต์

m_oaCTentativeClass

อ็อบเจกต์ CAttribute

อ็อบเจกต์สำหรับแอททริบิวต์ สร้างจากนิยามคลาส CAttribute ประกอบด้วย
แอททริบิวต์ต่างๆดังนี้

- m_attributeName - ใช้เก็บชื่อแอททริบิวต์ ได้จากแอททริบิวต์
m_columnName ของอ็อบเจกต์ CClassColumn
- m_dataTypeName - ใช้เก็บชื่อประเภทข้อมูล ได้จากแอททริบิวต์
m_dataType ของอ็อบเจกต์ CClassColumn
- m_size - ใช้เก็บขนาดของข้อมูล ได้จากแอททริบิวต์ m_dataType ของ
อ็อบเจกต์ CClassColumn
- m_bFKFlag - แฟล็กแสดงสถานะว่าแอททริบิวต์นี้ใช้เป็นคีย์นอกด้วยหรือไม่

อ็อบเจกต์ CTentativeClassIndex

อ็อบเจกต์สำหรับดรรชนี สร้างจากนิยามคลาส CTentativeClassIndex ประกอบด้วย
แอททริบิวต์ต่างๆดังนี้

- m_indexName - ใช้เก็บชื่อดรรชนี ได้จากแอททริบิวต์ m_indexName ของ
อ็อบเจกต์ CIndex
- m_saIndexAttribute - ใช้เก็บชื่อคอลัมน์ที่ประกอบเป็นดรรชนี ได้จาก
แอททริบิวต์ m_saIndexColumn ของอ็อบเจกต์ CIndex

อ็อบเจกต์ CTentativeClassRelation

อ็อบเจกต์สำหรับดรรชนี สร้างจากนิยามคลาส CTentativeClassRelation
ประกอบด้วยแอททริบิวต์ต่างๆดังนี้

- m_relatedClassName - ใช้เก็บชื่อคลาสที่ถูกอ้างอิงโดยคลาสนี้ ได้จาก
แอททริบิวต์ m_refTableName ของอ็อบเจกต์ CReferencedTable
- m_saFKColumn - ใช้เก็บชื่อคอลัมน์ที่ประกอบเป็นคีย์นอก ได้จากแอททริบิวต์
m_saFKColumn ของอ็อบเจกต์ CForeignKey

ขั้นตอนการเตรียมแผนแบบเชิงวัตถุเบื้องต้น

1. อ่านอ็อบเจกต์สำหรับตาราง CClassTable จากตัวแปร m_oaAllTable ที่ละอ็อบเจกต์
พร้อมอ็อบเจกต์ CPrimaryKey CForeignKey CClassColumn CIndex และ
CReferencedTable
2. สร้างอ็อบเจกต์สำหรับคลาส จากนิยามคลาส CTentativeClass
 - กำหนดค่าให้แอททริบิวต์ m_className เป็นค่าจากแอททริบิวต์ m_tableName
ของอ็อบเจกต์ CClassTable
 - กำหนดค่าให้แอททริบิวต์ m_saPKColumn เป็นค่าจากแอททริบิวต์
m_saPKColumn ของอ็อบเจกต์ CPrimaryKey
3. เก็บพอยน์เตอร์ของอ็อบเจกต์สำหรับคลาส CTentativeClass ลงในตัวแปรแอเรย์ของ
อ็อบเจกต์ m_oaCTentativeClass
4. สร้างอ็อบเจกต์สำหรับแอททริบิวต์ในคลาส จากนิยามคลาส CAttribute
 - กำหนดค่าให้ m_attributeName เป็นค่าจากแอททริบิวต์ m_columnName ของ
อ็อบเจกต์ CClassColumn
 - กำหนดค่าให้ m_dataType เป็นค่าบางส่วนจากแอททริบิวต์ m_dataType ของ
อ็อบเจกต์ CClassColumn
 - กำหนดค่าให้ m_size เป็นค่าบางส่วนจากแอททริบิวต์ m_dataType ของอ็อบเจกต์
CClassColumn
 - กำหนดค่าให้ m_bFKFlag เป็นค่า TRUE เมื่อแอททริบิวต์นี้เป็นคีย์นอกด้วย นอกนั้น
เป็น FALSE
5. เก็บพอยน์เตอร์ของอ็อบเจกต์ CAttribute ลงในแอททริบิวต์ m_oaAttribute ของ
อ็อบเจกต์ CTentativeClass
6. สร้างอ็อบเจกต์สำหรับดรรชนีในคลาส จากนิยามคลาส CTentativeClassIndex

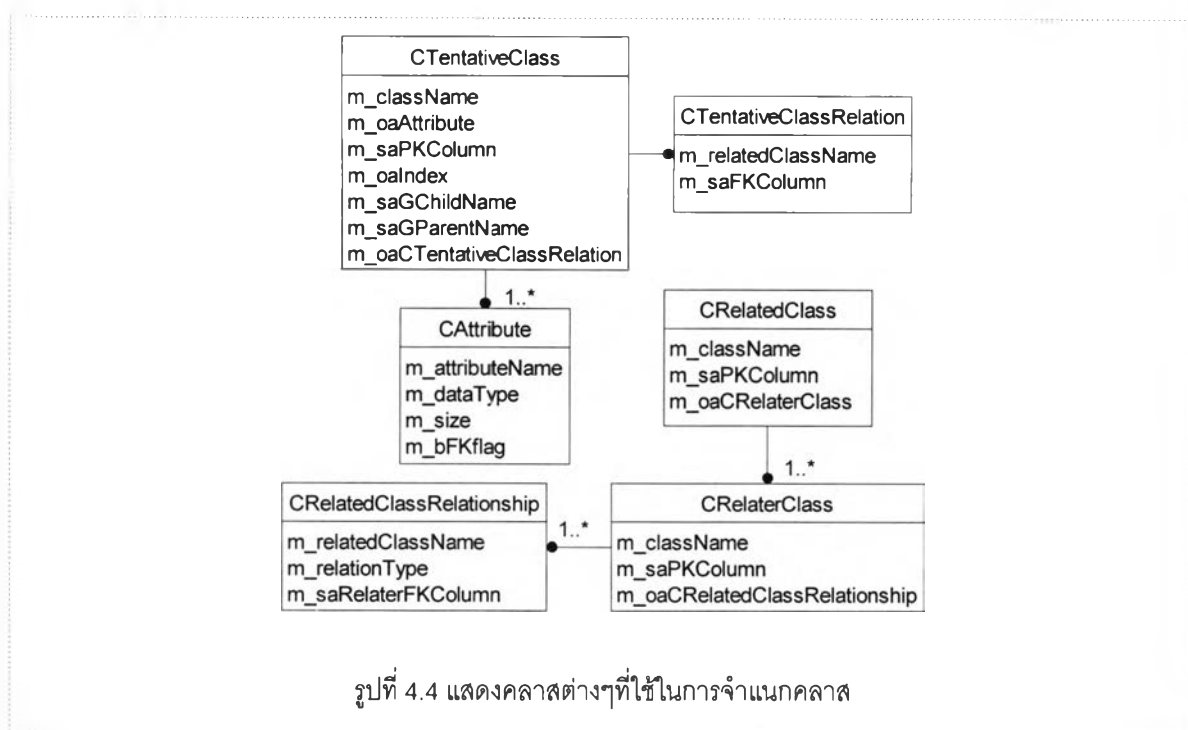
- กำหนดค่าให้ `m_indexName` เป็นค่าจากแอททริบิวต์ `m_indexName` ของอ็อบเจกต์ `CIndex`
 - กำหนดค่าให้ `m_saIndexAttribute` เป็นค่าจากแอททริบิวต์ `m_saIndexColumn` ของอ็อบเจกต์ `CIndex`
7. เก็บพอยน์เตอร์ของอ็อบเจกต์ `CTentativeClassIndex` ลงในแอททริบิวต์ `m_oaIndex` ของอ็อบเจกต์ `CTentativeClass`
 8. สร้างอ็อบเจกต์สำหรับสำหรับแสดงความสัมพันธ์ระหว่างคลาส จากนิยามคลาส `CTentativeClassRelation`
 - กำหนดค่าให้ `m_relatedClassName` เป็นค่าจากแอททริบิวต์ `m_refTableName` ของอ็อบเจกต์ `CReferencedTable`
 - กำหนดค่าให้ `m_saFKColumn` เป็นค่าจากแอททริบิวต์ `m_m_saFKColumn` ของอ็อบเจกต์ `CForeignKey`
 9. เก็บพอยน์เตอร์ของอ็อบเจกต์ `CTentativeClassRelation` ลงในแอททริบิวต์ `m_oaCTentativeClassRelation` ของอ็อบเจกต์ `CTentativeClass`
 10. เริ่มขั้นตอนที่ 1 ใหม่จนครบทุกอ็อบเจกต์

4.3 กลุ่มคลาสจำแนกคลาส

วัตถุประสงค์เพื่อแยกคลาสที่ได้ในแผนแบบเชิงวัตถุเบื้องต้นเป็น 2 กลุ่มคลาสคือ คลาสผู้อ้างอิงไปยังคลาสอื่นเรียกว่า Relater Class (อ้างอิงผ่านคีย์นอก) และคลาสผู้ถูกอ้างอิงโดยคลาสอื่นเรียกว่า Related Class

โปรแกรมจะอ่านอ็อบเจกต์สำหรับคลาส `CTentativeClass` จากแอเรย์ `m_oaCTentativeClass` ที่ละอ็อบเจกต์เพื่อจำแนกคลาสผู้อ้างอิงและคลาสผู้ถูกอ้างอิงโดยมีการสร้างอ็อบเจกต์สำหรับคลาสผู้อ้างอิง อ็อบเจกต์สำหรับคลาสผู้ถูกอ้างอิง และอ็อบเจกต์สำหรับแสดงความสัมพันธ์ระหว่างคลาสผู้อ้างอิงกับคลาสผู้ถูกอ้างอิง ตามนิยามของคลาสดังต่อไปนี้

- `CRelaterClass`
- `CRelatedClass`
- `CRelatedClassRelationship`



รูปที่ 4.4 แสดงคลาสต่างๆและความสัมพันธ์ระหว่างคลาส ซึ่งจำเป็นต้องใช้ในการจำแนก
 คลาส

อ็อบเจกต์ CRelaterClass

เป็นอ็อบเจกต์สำหรับคลาสผู้อ้างอิงคลาสอื่น สร้างจากนิยามคลาส CRelaterClass ประกอบด้วยแอททริบิวต์ต่างๆดังนี้

- m_className - ใช้เก็บชื่อคลาส ซึ่งได้จากแอททริบิวต์ m_className ของอ็อบเจกต์ CTentativeClass
- m_saPKColumn - ใช้เก็บชื่อแอททริบิวต์ที่ประกอบเป็นคีย์หลัก ซึ่งได้จากแอททริบิวต์ m_saPKColumn ของอ็อบเจกต์ CTentativeClass
- m_oaCTentativeClassRelation - เป็นแอเรย์ของพอยน์เตอร์ ใช้เก็บพอยน์เตอร์ของอ็อบเจกต์ CTentativeClassRelation

อ็อบเจกต์ CRelatedClassRelationship

เป็นอ็อบเจกต์แสดงความสัมพันธ์ระหว่างคลาสผู้อ้างอิงกับคลาสผู้ถูกอ้างอิง สร้างจากนิยามคลาส CRelatedClassRelationship ประกอบด้วยแอททริบิวต์ต่างๆดังนี้

- m_relatedClassName - เก็บชื่อคลาสที่ถูกอ้างอิง ได้จากแอททริบิวต์ m_relatedClassName ของอ็อบเจกต์ CTentativeClassRelation
- m_relationType - เก็บความสัมพันธ์ระหว่างคลาสผู้ถูกอ้างอิงกับคลาสผู้อ้างอิง ("ONE-TO-ONE" หรือ "ONE-TO-MANY")
- m_saRelaterFKColumn - เก็บชื่อแอททริบิวต์ซึ่งใช้เป็นคีย์นอก ได้จากแอททริบิวต์ m_saFKColumn ของอ็อบเจกต์ CTentativeClassRelation

อ็อบเจกต์ CRelatedClass

เป็นอ็อบเจกต์สำหรับคลาสผู้ถูกอ้างอิง สร้างจากนิยามคลาส CRelatedClass ประกอบด้วยแอททริบิวต์ต่างๆดังนี้

- m_className - เก็บชื่อคลาสผู้ถูกอ้างอิง ได้จากแอททริบิวต์ m_relatedClassName ของอ็อบเจกต์ CTentativeClassRelation
- m_saPKColumn - เก็บชื่อแอททริบิวต์ที่ประกอบเป็นคีย์หลัก ได้จากแอททริบิวต์ m_saPKColumn ของอ็อบเจกต์ CTentativeClass
- m_oaCRelaterClass - เป็นแอเรียของพอยน์เตอร์ ใช้เก็บพอยน์เตอร์ของอ็อบเจกต์ CRelaterClass

ขั้นตอนการจำแนกคลาส

1. อ่านอ็อบเจกต์สำหรับคลาส CTentativeClass จากตัวแปร m_oaCTentativeClass ที่ละอ็อบเจกต์พร้อมอ็อบเจกต์ CTentativeClassRelation
2. ตรวจสอบการมีความสัมพันธ์กับคลาสอื่นจากแอททริบิวต์ m_oaCTentativeClassRelation ของอ็อบเจกต์ CTentativeClass ถ้ามีความสัมพันธ์กับคลาสอื่น
 - สร้างอ็อบเจกต์สำหรับคลาสผู้ถูกอ้างอิงจากนิยามคลาส CRelaterClass สำหรับอ็อบเจกต์ CTentativeClass นี้
 - กำหนดให้ m_className เป็นค่าใน m_className ของ CTentativeClass
 - กำหนดให้ m_saPKColumn เป็นค่าใน m_saPKColumn ของอ็อบเจกต์ CTentativeClass
 - สร้างอ็อบเจกต์จากนิยามคลาส CRelatedClassRelationship สำหรับคลาสผู้ถูกอ้างอิงโดย CRelaterClass

- กำหนดให้ `m_relatedClassName` เป็นค่าใน `m_relatedClassName` ของ `CTentativeClassRelation`
 - กำหนดให้ `m_saRelaterFKColumn` เป็นค่าใน `m_saFKColumn` ของอ็อบเจ็กต์ `CTentativeClassRelation`
 - หาความสัมพันธ์ระหว่างคลาสผู้ถูกอ้างอิงกับคลาสผู้อ้างอิง
 - อ่านข้อมูลในตารางที่สัมพันธ์กับคลาสผู้ถูกอ้างอิงและคลาสผู้อ้างอิง
 - ถ้า 1 รายการข้อมูลในตารางของถูกอ้างอิงมีมากกว่า 1 รายการในตารางผู้อ้างอิงแล้วความสัมพันธ์ระหว่างคลาสผู้ถูกอ้างอิงกับคลาสผู้อ้างอิงเป็น "ONE-TO-MANY"
 - ถ้า 1 รายการข้อมูลในตารางของถูกอ้างอิงมีเพียง 1 รายการในตารางผู้อ้างอิงแล้วความสัมพันธ์ระหว่างคลาสผู้ถูกอ้างอิงกับคลาสผู้อ้างอิงเป็น "ONE-TO-ONE"
 - กำหนดค่าให้ `m_relationType` เป็น "ONE-TO-ONE" หรือ "ONE-TO-MANY" แล้วแต่ความสัมพันธ์ที่หาได้
 - เก็บพอยน์เตอร์ของอ็อบเจ็กต์ `CRelatedClassRelationship` ในแอททริบิวต์ `m_oaCRelatedClassRelationship` ของอ็อบเจ็กต์ `CRelaterClass`
 - สร้างอ็อบเจ็กต์จากนิยามคลาส `CRelatedClass` สำหรับคลาสผู้ถูกอ้างอิงโดย `CRelaterClass`
 - กำหนดให้ `m_className` เป็นค่าใน `m_relatedClassName` ของอ็อบเจ็กต์ `CTentativeClassRelation`
 - กำหนดให้ `m_saPKColumn` เป็นค่าใน `m_saPKColumn` ของอ็อบเจ็กต์ `CTentativeClass`
 - เก็บพอยน์เตอร์ของอ็อบเจ็กต์ `CRelatedClass` ในแอททริบิวต์ `m_oaCRelaterClass` ของอ็อบเจ็กต์ `CRelatedClass`
 - เก็บพอยน์เตอร์ของอ็อบเจ็กต์ `CRelatedClass` ในตัวแปรแอเรย์ของอ็อบเจ็กต์ `m_oaCRelaterClass`
 - เก็บพอยน์เตอร์ของอ็อบเจ็กต์ `CRelaterClass` ในตัวแปรแอเรย์ของอ็อบเจ็กต์ `m_oaCRelaterClass`
3. เริ่มขั้นตอนที่ 1 ใหม่จนครบทุกอ็อบเจ็กต์

4.4 กลุ่มคลาสค้นหาการสืบทอดคุณสมบัติ

วัตถุประสงค์เพื่อค้นหาการสืบทอดคุณสมบัติระหว่างคลาสซึ่งแทนด้วยอ็อบเจกต์ของ CTentativeClass แยกเป็น 2 กลุ่มคลาสน้อยคือ

- กลุ่มคลาสค้นหาคลาสซึ่งมีแอสทริบิวท์ซ้ำกัน
- กลุ่มคลาสค้นหาคลาสซึ่งมีคีย์หลักเป็นซับเซตคีย์หลักของคลาสอื่น

4.4.1 กลุ่มคลาสค้นหาคลาสซึ่งมีแอสทริบิวท์ซ้ำกัน

วัตถุประสงค์เพื่อค้นหาคลาสซึ่งมีแอสทริบิวท์ซ้ำกัน ถ้าพบสร้างคลาสใหม่โดยมีแอสทริบิวท์เป็นแอสทริบิวท์ที่ซ้ำกันของคลาสเหล่านั้น

ขั้นตอนการค้นหาคลาสซึ่งมีแอสทริบิวท์ซ้ำกัน

1. อ่านอ็อบเจกต์สำหรับคลาส CTentativeClass จากตัวแปร m_oaCTentativeClass ลำดับที่เพิ่มทีละ 1
 - ก. อ่านอ็อบเจกต์สำหรับคลาส CTentativeClass จากตัวแปร m_oaCTentativeClass ลำดับถัดจากข้อ 1 เพิ่มทีละ 1
 - ข. เปรียบเทียบแอสทริบิวท์ของอ็อบเจกต์ในข้อ ก. กับแอสทริบิวท์ของอ็อบเจกต์ในข้อ 1. ทีละแอสทริบิวท์
 - ถ้าเหมือนกันทุกประการ (ทั้งชื่อแอสทริบิวท์ ประเภทข้อมูลและขนาด) เก็บชื่อแอสทริบิวท์พร้อมประเภทข้อมูลและขนาดในแอเรย์ของสตริง saAttrib saDataType และ saSize ตามลำดับ
 - วนกลับข้อ ข. จนครบทุกแอสทริบิวท์
 - ค. จากข้อ ข. ถ้ามีแอสทริบิวท์ซ้ำกัน สร้างอ็อบเจกต์สำหรับคลาสจากนิยามคลาส CTentativeClass
 - กำหนดค่าให้แอสทริบิวท์ m_className เป็นค่าในแอสทริบิวท์ m_className ของอ็อบเจกต์ในข้อ 1. และค่าในแอสทริบิวท์ m_className ของอ็อบเจกต์ในข้อ ก. นำมาต่อกันด้วยเครื่องหมาย "_"
 - กำหนดค่าให้แอสทริบิวท์ m_saGChildName เป็นค่าในแอสทริบิวท์ m_className ของอ็อบเจกต์ในข้อ 1. และค่าในแอสทริบิวท์ m_className ของอ็อบเจกต์ในข้อ ก.

- สร้างอ็อบเจ็กต์สำหรับแอททริบิวต์ในคลาส จากนิยามคลาส CAttribute โดยใช้ค่าในแอตทริบิวต์ saAttrb saDataType และ saSize
- เก็บพอยน์เตอร์ของอ็อบเจ็กต์ CAttribute ลงในแอททริบิวต์ m_oaAttribute ของอ็อบเจ็กต์ CTentativeClass ในข้อ ค.
- เก็บอ็อบเจ็กต์สำหรับคลาส CTentativeClass ในข้อ ค. ในตัวแปร m_oaCTentativeClass
- กำหนดค่าให้แอททริบิวต์ m_saGParentName ของอ็อบเจ็กต์ ในข้อ 1. และข้อ ก. เป็นค่าเดียวกันกับค่าใน m_className ของอ็อบเจ็กต์ในข้อ ค.
- ลบอ็อบเจ็กต์ CAttribute ที่ซ้ำใน m_oaAttribute ของอ็อบเจ็กต์ในข้อ 1. ข้อ ก. ง. วนกลับไปทำข้อ ก.

2. วนกลับไปทำข้อ 1.

4.4.2 กลุ่มคลาสค้นหาคลาสซึ่งมีคีย์หลักเป็นซับเซตคีย์หลักของคลาสอื่น

วัตถุประสงค์เพื่อค้นหาคลาสซึ่งมีคีย์หลักเป็นซับเซตคีย์หลักของคลาสอื่น ถ้าพบสร้างคลาสใหม่ โดยคลาสใหม่นี้ประกอบด้วยแอททริบิวต์ซึ่งประกอบด้วยคีย์หลักที่เป็นซับเซตของคีย์หลักของคลาสอื่น

ขั้นตอนการค้นหาคลาสซึ่งมีแอททริบิวต์ซ้ำกัน

1. อ่านอ็อบเจ็กต์สำหรับคลาส CTentativeClass จากตัวแปร m_oaCTentativeClass ลำดับที่ j (เริ่มที่ 0)
 - ก. อ่านอ็อบเจ็กต์สำหรับคลาส CTentativeClass จากตัวแปร m_oaCTentativeClass ลำดับที่ k จนหมดทุกอ็อบเจ็กต์ (เริ่มที่ 0 ถ้า k เท่ากับ j แล้วเพิ่มค่า k อีก 1)
 - ข. ตรวจสอบดูว่าค่าในแอททริบิวต์ m_saPKColumn ของอ็อบเจ็กต์ CTentativeClass ในข้อ ก. เป็นซับเซตกับค่าในแอททริบิวต์ m_saPKColumn ของอ็อบเจ็กต์ CTentativeClass ในข้อ 1. หรือไม่
 - ค. ถ้าใช่ สร้างอ็อบเจ็กต์สำหรับคลาสจากนิยามคลาส CTentativeClass
 - กำหนดค่าให้แอททริบิวต์ m_className เป็นค่าในแอททริบิวต์ m_className ของอ็อบเจ็กต์ในข้อ 1. และค่าในแอททริบิวต์ m_className ของอ็อบเจ็กต์ในข้อ ก. นำมาต่อกันค้นด้วยเครื่องหมาย “_”

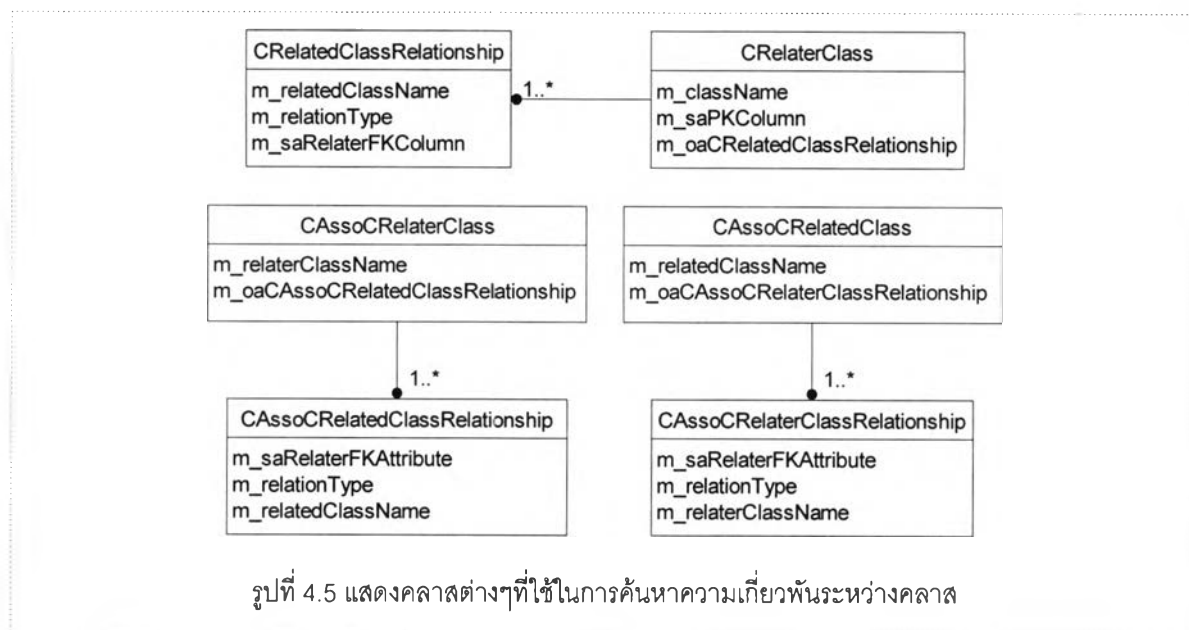
- กำหนดค่าให้แอททริบิวต์ `m_saGChildName` เป็นค่าในแอททริบิวต์ `m_className` ของอ็อบเจกต์ในข้อ 1. และค่าในแอททริบิวต์ `m_className` ของอ็อบเจกต์ในข้อ ก.
 - สร้างอ็อบเจกต์สำหรับแอททริบิวต์ในคลาส จากนิยามคลาส `CAttribute` โดยใช้ค่า `m_saPKColumn` ของอ็อบเจกต์ `CTentativeClass` ในข้อ ก.
 - เก็บพอยน์เตอร์ของอ็อบเจกต์ `CAttribute` ลงในแอททริบิวต์ `m_oaAttribute` ของอ็อบเจกต์ `CTentativeClass` ในข้อ ค.
 - เก็บอ็อบเจกต์สำหรับคลาส `CTentativeClass` ในข้อ ค. ในตัวแปร `m_oaCTentativeClass`
 - กำหนดค่าให้แอททริบิวต์ `m_saGParentName` ของอ็อบเจกต์ ในข้อ 1. และข้อ ก. เป็นค่าเดียวกันกับค่าใน `m_className` ของอ็อบเจกต์ในข้อ ค.
 - ลบอ็อบเจกต์ `CAttribute` ที่ซ้ำใน `m_oaAttribute` ของอ็อบเจกต์ในข้อ 1. ข้อ ก.
- ง. วนกลับไปข้อ ก.

2. วนกลับไปข้อ 1.

4.5 กลุ่มคลาสค้นหาความเกี่ยวพัน

วัตถุประสงค์เพื่อค้นหาความเกี่ยวพัน (Association) ระหว่างคลาส ผลของการทำงานของขั้นตอนนี้จะมีการสร้างอ็อบเจกต์ต่างๆจากนิยามคลาสดังต่อไปนี้

- `CAssoCRelatedClass`
- `CAssoCRelaterClass`
- `CAssoCRelatedClassRelationship`
- `CAssoCRelaterClassRelationship`



รูปที่ 4.5 แสดงคลาสต่างๆและความสัมพันธ์ระหว่างคลาส ซึ่งจำเป็นต้องใช้ในการค้นหาความสัมพันธ์ระหว่างคลาส

อ็อบเจกต์ CAssoCRelaterClass

เป็นอ็อบเจกต์สำหรับคลาสผู้อ้างอิงคลาสอื่น สร้างจากนิยามคลาส

CAssoCRelaterClass ประกอบด้วยแอตทริบิวต์ต่างๆดังนี้

- m_relaterClassName - ใช้เก็บชื่อคลาสที่ผ่านการกำหนดความสัมพันธ์แล้ว
- m_oaCAssoCRelatedClassRelationship - เป็นแอเรียของพอยน์เตอร์ ใช้เก็บพอยน์เตอร์ที่ไปยังอ็อบเจกต์ CAssoCRelatedClassRelationship

อ็อบเจกต์ CAssoCRelatedClassRelationship

เป็นอ็อบเจกต์แสดงความสัมพันธ์ระหว่างคลาสผู้อ้างอิงคลาสอื่นกับคลาสผู้ถูกอ้างอิง สร้างจากนิยามคลาส CAssoCRelatedClassRelationship ประกอบด้วยแอตทริบิวต์ต่างๆดังนี้

- m_saRelaterFKAttribute - ใช้เก็บแอตทริบิวต์ที่ประกอบเป็นคีย์นอกของคลาสผู้อ้างอิง
- m_relationType - ใช้แสดงความสัมพันธ์ระหว่างคลาสผู้ถูกอ้างอิงกับคลาสผู้อ้างอิง
- m_relatedClassName - ใช้เก็บชื่อคลาสผู้ถูกอ้างอิง

อ็อบเจกต์ CAssoCRelatedClass

เป็นอ็อบเจกต์สำหรับคลาสผู้ถูกอ้างอิงโดยคลาสอื่น สร้างจากนิยามคลาส CAssoCRelatedClass ประกอบด้วยแอททริบิวต์ต่างๆดังนี้

- m_relatedClassName - ใช้เก็บชื่อคลาสผู้ถูกอ้างอิง
- m_oaCAssoCRelaterClassRelationship - เป็นแอเรียของพอยน์เตอร์ ใช้เก็บพอยน์เตอร์ชี้ไปยังอ็อบเจกต์ CAssoCRelaterClassRelationship

อ็อบเจกต์ CAssoCRelaterClassRelationship

- m_saRelaterFKAttribute - ใช้เก็บแอททริบิวต์ที่ประกอบเป็นคีย์นอกของคลาสผู้อ้างอิง
- m_relationType - ใช้แสดงความสัมพันธ์ระหว่างคลาสผู้อ้างอิงกับคลาสผู้ถูกอ้างอิง
- m_relaterClassName - ใช้เก็บชื่อคลาสผู้อ้างอิง

ขั้นตอนการค้นหาความเกี่ยวพันระหว่างคลาส

1. อ่านอ็อบเจกต์สำหรับคลาส CRelaterClass จากตัวแปร m_oaCRelaterClass
 - ก. หาคีย์หลักของคลาสผู้อ้างอิง
 - ข. หาจำนวนคีย์นอกของคลาสผู้อ้างอิงซึ่งชี้ไปยังคลาสอื่น โดยหาจากจำนวนอ็อบเจกต์ของ CRelatedClassRelationship ซึ่งผูกกับอ็อบเจกต์ CRelaterClass (เก็บใน m_oaCRelatedClassRelationship)
 - ค. กรณีจำนวนคีย์นอกมีเพียง 1 ตัว
 - ถ้าคีย์นอกนั้นเป็นคีย์หลักด้วย
 - พิจารณาคลาสผู้ถูกอ้างอิงว่ามีคลาสอื่นที่อ้างอิงคลาสผู้ถูกอ้างอิงนี้ในลักษณะเช่นเดียวกันนี้หรือไม่
 - ถ้ามี แสดงว่าคลาสผู้ถูกอ้างอิงกับคลาสผู้อ้างอิงเหล่านี้มีการสืบทอดคุณสมบัติกัน จะได้คลาสผู้ถูกอ้างอิงเป็นคลาสพ่อและคลาสผู้อ้างอิงเป็นคลาสลูก
 - กำหนดค่าแอททริบิวต์ m_saGParentName ของอ็อบเจกต์ CTentativeClass สำหรับคลาสผู้อ้างอิงเป็นชื่อคลาสผู้ถูกอ้างอิง

- กำหนดค่าแอททริบิวต์ m_saGChildName ของอ็อบเจ็กต์ CTentativeClass สำหรับคลาสผู้ถูกอ้างอิงเป็นชื่อคลาสผู้อ้างอิง
- ถ้าไม่มี แสดงว่าคลาสผู้ถูกอ้างอิงกับคลาสผู้อ้างอิงมีความเกี่ยวพันกันและเป็นความเกี่ยวพันกันแบบ "ONE-TO-ONE"
 - สร้างอ็อบเจ็กต์สำหรับคลาสผู้อ้างอิง จากนิยามคลาส CAssoCRelaterClass
 - กำหนดค่าให้ m_relaterClassName จากแอททริบิวต์ m_relaterName ของอ็อบเจ็กต์ CRelaterClass
 - สร้างอ็อบเจ็กต์ CAssoCRelatedClassRelationship
 - กำหนดค่าให้ m_relationType เป็น "ONE-TO-ONE"
 - เก็บอ็อบเจ็กต์ CAssoCRelatedClassRelationship ใน m_oaCAssoCRelatedClassRelationship ของอ็อบเจ็กต์ CAssoCRelaterClass
 - เก็บอ็อบเจ็กต์ CAssoCRelaterClass ในตัวแปรเอเรย์อ็อบเจ็กต์ m_oaCAssoCRelaterClass
 - สร้างอ็อบเจ็กต์สำหรับคลาสผู้ถูกอ้างอิง จากนิยามคลาส CAssoCRelatedClass
 - กำหนดค่าให้ m_relatedClassName โดยคัดลอกค่าจาก m_relatedClassName ของอ็อบเจ็กต์ CRelatedClassRelationship
 - สร้างอ็อบเจ็กต์ CAssoCRelaterClassRelationship
 - กำหนดค่าให้ m_relationType เป็น "ONE-TO-ONE"
 - เก็บอ็อบเจ็กต์ CAssoCRelaterClassRelationship ใน m_oaCAssoCRelaterClassRelationship ของอ็อบเจ็กต์ CAssoCRelatedClass
 - เก็บอ็อบเจ็กต์ CAssoCRelatedClass ในตัวแปรเอเรย์ของอ็อบเจ็กต์ m_oaCAssoCRelatedClass
- ถ้าคีย์นอกนั้นไม่เป็นคีย์หลัก
 - สร้างอ็อบเจ็กต์สำหรับคลาสผู้อ้างอิงจากนิยามคลาส CAssoCRelaterClass
 - กำหนดค่าให้ m_relaterClassName โดยคัดลอกค่าจาก m_relaterName ของอ็อบเจ็กต์ CRelaterClass

- สร้างอ็อบเจกต์ CAssoCRelatedClassRelationship
 - กำหนดค่าให้ m_relationType เป็นค่าความสัมพันธ์จากคลาสผู้ถูกอ้างอิงไปยังคลาสผู้อ้างอิง
 - เก็บอ็อบเจกต์ CAssoCRelatedClassRelationship ใน m_oaCAssoCRelatedClassRelationship ของอ็อบเจกต์ CAssoCRelaterClass
 - เก็บอ็อบเจกต์ CAssoCRelaterClass ในตัวแปรแอเรย์ของอ็อบเจกต์ m_oaCAssoCRelaterClass
 - สร้างอ็อบเจกต์สำหรับคลาสผู้ถูกอ้างอิง จากนิยามคลาส CAssoCRelatedClass
 - กำหนดค่าให้ m_relatedClassName จากแอททริบิวต์ m_relatedClassName ของอ็อบเจกต์ CRelatedClassRelationship
 - สร้างอ็อบเจกต์ CAssoCRelaterClassRelationship
 - กำหนดค่าให้ m_relationType เป็นค่าความสัมพันธ์จากคลาสผู้อ้างอิงไปยังคลาสผู้ถูกอ้างอิง
 - เก็บอ็อบเจกต์ CAssoCRelaterClassRelationship ใน m_oaCAssoCRelaterClassRelationship ของอ็อบเจกต์ CAssoCRelatedClass
 - เก็บอ็อบเจกต์ CAssoCRelatedClass ในตัวแปรแอเรย์ของอ็อบเจกต์ m_oaCAssoCRelatedClass
- ง. กรณีจำนวนคีย์นอกมี 2 ตัว
- หาคีย์หลักของคลาสผู้อ้างอิงจาก m_saPKColumn ของอ็อบเจกต์ CRelaterClass
 - หาคีย์นอกตัวที่หนึ่งและคีย์นอกตัวที่สอง โดยได้จาก m_relatedClassName ของอ็อบเจกต์ CRelatedClassRelationship
 - ถ้าทุกแอททริบิวต์ของคลาสผู้อ้างอิงประกอบด้วยคีย์นอกเท่านั้น
 - กำหนดค่าในแอททริบิวต์ m_bFlag ของอ็อบเจกต์ CTentativeClass ที่สัมพันธ์กับคลาสผู้อ้างอิงเป็น FALSE
 - ถ้าคีย์หลักประกอบด้วยคีย์นอก 2 ตัว
 - สร้างอ็อบเจกต์จากนิยามคลาส CAssoRelaterClass

- กำหนดค่า m_relaterClassName เป็นชื่อคลาสผู้ถูกอ้างอิงตัวที่หนึ่ง
- สร้างอ็อบเจกต์จากนิยามคลาส CAssoRelatedClassRelationship
 - กำหนดค่า m_relationType เป็น "MANY-TO-MANY"
 - กำหนดค่า m_relatedClassName เป็นชื่อคลาสผู้ถูกอ้างอิงตัวที่สอง
- เก็บอ็อบเจกต์ CAssoRelatedClassRelationship ใน m_oaCAssoRelatedClassRelationship ของอ็อบเจกต์ CAssoRelaterClass
- เก็บอ็อบเจกต์ CAssoRelaterClass ในตัวแปรเอเรย์อ็อบเจกต์ m_oaCAssoRelaterClass
- สร้างอ็อบเจกต์จากนิยามคลาส CAssoRelatedClass
 - กำหนดค่า m_relatedClassName เป็นชื่อคลาสผู้ถูกอ้างอิงตัวที่สอง
 - สร้างอ็อบเจกต์จากนิยามคลาส CAssoRelaterClassRelationship
 - กำหนดค่า m_relationType เป็น "MANY-TO-MANY"
 - กำหนดค่า m_relaterClassName เป็นชื่อคลาสผู้ถูกอ้างอิงตัวที่หนึ่ง
 - เก็บอ็อบเจกต์ CAssoRelaterClassRelationship ใน m_oaCAssoRelaterClassRelationship ของอ็อบเจกต์ CAssoRelatedClass
- เก็บอ็อบเจกต์ CAssoRelatedClass ในตัวแปรเอเรย์ของอ็อบเจกต์ m_oaCAssoRelatedClass
- ถ้าคีย์หลักประกอบด้วยคีย์นอกตัวใดตัวหนึ่ง
 - สร้างอ็อบเจกต์จากนิยามคลาส CAssoRelaterClass
 - กำหนดค่า m_relaterClassName เป็นชื่อคลาสผู้ถูกอ้างอิงตัวด้วยคีย์นอกของคลาสผู้อ้างอิงซึ่งไม่เป็นคีย์หลัก
 - สร้างอ็อบเจกต์จากนิยามคลาส CAssoRelatedClassRelationship
 - กำหนดค่า m_relationType เป็นค่าความสัมพันธ์จากคลาสผู้อ้างอิงด้วยคีย์นอกซึ่งไม่เป็นคีย์หลักไปยังคลาสผู้ถูกอ้างอิง
 - กำหนดค่า m_relatedClassName เป็นชื่อคลาสผู้ถูกอ้างอิงตัวซึ่งถูกอ้างอิงด้วยคีย์หลักของคลาสผู้อ้างอิง

- เก็บอ็อบเจ็กต์ CAssoRelatedClassRelationship ใน m_oaCAssoRelatedClassRelationship ของอ็อบเจ็กต์ CAssoRelaterClass
 - เก็บอ็อบเจ็กต์ CAssoRelaterClass ในตัวแปรแอสเรย์ของอ็อบเจ็กต์ m_oaCAssoRelaterClass
 - สร้างอ็อบเจ็กต์จากนิยามคลาส CAssoRelatedClass
 - กำหนดค่า m_relatedClassName เป็นชื่อคลาสผู้ถูกอ้างอิงตัวด้วยคีย์นอกของคลาสผู้อ้างอิงซึ่งเป็นคีย์หลักด้วย
 - สร้างอ็อบเจ็กต์จากนิยามคลาส CAssoRelaterClassRelationship
 - กำหนดค่า m_relationType เป็นค่าความสัมพันธ์จากคลาสผู้ถูกอ้างอิงไปยังคลาสผู้อ้างอิงด้วยคีย์นอกซึ่งไม่เป็นคีย์หลัก
 - กำหนดค่า m_relaterClassName เป็นชื่อคลาสผู้ถูกอ้างอิงตัวซึ่งถูกอ้างอิงด้วยคีย์นอกซึ่งไม่เป็นคีย์หลักของคลาสผู้อ้างอิง
 - เก็บอ็อบเจ็กต์ CAssoRelaterClassRelationship ใน m_oaCAssoRelaterClassRelationship ของอ็อบเจ็กต์ CAssoRelatedClass
 - เก็บอ็อบเจ็กต์ CAssoRelatedClass ในตัวแปรแอสเรย์อ็อบเจ็กต์ m_oaCAssoRelatedClass
- จ. กรณีคลาสผู้อ้างอิงมีจำนวนคีย์นอก 2 ตัวและคลาสผู้อ้างอิงนี้มีแอททริบิวต์อื่นๆนอกเหนือจากแอททริบิวต์ที่เป็นคีย์นอก หรือกรณีคลาสผู้อ้างอิงมีจำนวนคีย์นอกมากกว่า 2 ตัว
- สร้างอ็อบเจ็กต์จากนิยามคลาส CAssoRelaterClass
 - กำหนดค่า m_relaterClassName เป็นชื่อคลาสผู้อ้างอิง
 - สร้างอ็อบเจ็กต์จากนิยามคลาส CAssoRelatedClassRelationship
 - กำหนดค่า m_relationType เป็นค่าความสัมพันธ์จากคลาสผู้ถูกอ้างอิงไปยังคลาสผู้อ้างอิง
 - กำหนดค่า m_relatedClassName เป็นชื่อคลาสผู้ถูกอ้างอิง
 - เก็บอ็อบเจ็กต์ CAssoRelatedClassRelationship ใน m_oaCAssoRelatedClassRelationship ของอ็อบเจ็กต์ CAssoRelaterClass

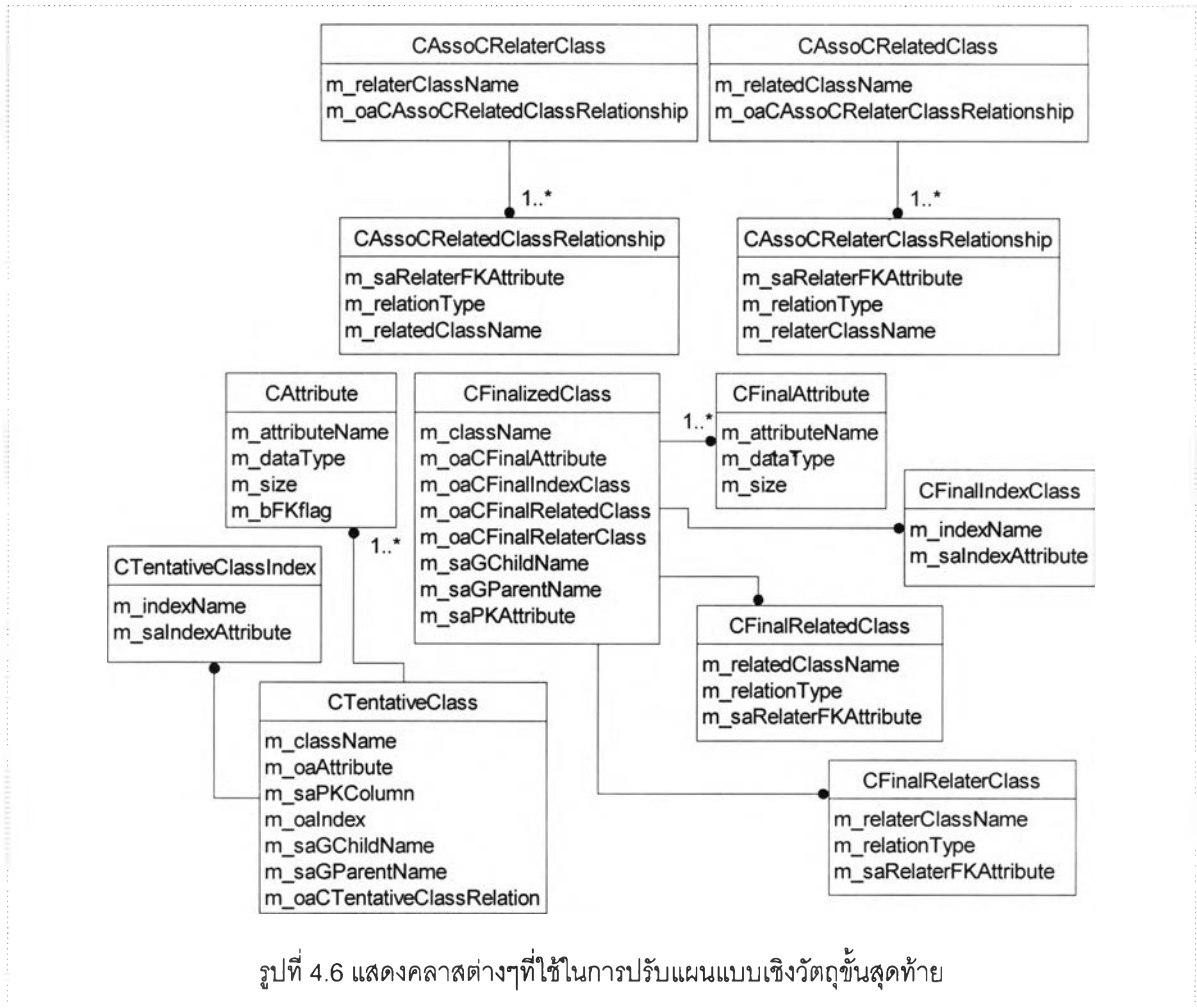
- เก็บอ็อบเจ็กต์ CAssoRelaterClass ในตัวแปรแอเรย์อ็อบเจ็กต์
m_oaCAssoRelaterClass
- สร้างอ็อบเจ็กต์จากนิยามคลาส CAssoRelatedClass
 - กำหนดค่า m_relatedClassName เป็นชื่อคลาสผู้ถูกอ้างอิง
 - สร้างอ็อบเจ็กต์จากนิยามคลาส CAssoRelaterClassRelationship
 - กำหนดค่า m_relationType เป็นค่าความสัมพันธ์จากคลาสผู้อ้างอิงไปยังคลาสผู้ถูกอ้างอิง
 - กำหนดค่า m_relaterClassName เป็นชื่อคลาสผู้อ้างอิง
 - เก็บอ็อบเจ็กต์ CAssoRelaterClassRelationship ใน
m_oaCAssoRelaterClassRelationship ของอ็อบเจ็กต์
CAssoRelatedClass
- เก็บอ็อบเจ็กต์ CAssoRelatedClass ในตัวแปรแอเรย์อ็อบเจ็กต์
m_oaCAssoRelatedClass

2. วนกลับไปข้อ 1. จนครบทุกอ็อบเจ็กต์ CRelaterClass

4.6 กลุ่มคลาสปรับแผนแบบเชิงวัตถุขั้นสุดท้าย

หลังจากได้แผนแบบเชิงวัตถุที่สมบูรณ์แล้ว ขั้นตอนต่อไปเป็นการปรับและรวบรวมอ็อบเจ็กต์ที่ได้จากการค้นหาความเกี่ยวพัน นำมาสร้างอ็อบเจ็กต์ใหม่เพื่อความสะดวกในการนำไปใช้งานในการสร้างแฟ้ม HCD และสร้างฐานข้อมูลต่อไป ขั้นตอนนี้จะมีการสร้างอ็อบเจ็กต์จากนิยามคลาสดังต่อไปนี้

- CFinalizedClass
- CFinalAttribute
- CFinalIndexClass
- CFinalRelaterClass
- CFinalRelatedClass



รูปที่ 4.6 แสดงคลาสต่างๆและความสัมพันธ์ระหว่างคลาส ซึ่งจำเป็นต้องใช้ในการปรับแผนแบบเชิงวัตถุขั้นสุดท้าย

อ็อบเจกต์ CFinalizedClass

เป็นอ็อบเจกต์สำหรับคลาสที่ผ่านการปรับแก้ไขเรียบร้อยแล้ว สร้างจากนิยามคลาส CFinalizedClass ประกอบด้วยแอททริบิวต์ต่างๆดังนี้

- m_className - ใช้เก็บชื่อคลาส ได้จากแอททริบิวต์ m_className ของอ็อบเจกต์ CTentativeClass
- m_oaCFinalAttribute - เป็นแอเรย์ของพอยท์เตอร์ ใช้เก็บพอยท์เตอร์ชี้ไปยังอ็อบเจกต์ CFinalAttribute
- m_oaCFinalIndexClass - เป็นแอเรย์ของพอยท์เตอร์ ใช้เก็บพอยท์เตอร์ชี้ไปยังอ็อบเจกต์ CFinalIndexClass

- m_oaCFinalRelatedClass - เป็นแอเรย์ของพอยท์เตอร์ ใช้เก็บพอยท์เตอร์ชี้ไปยังอ็อบเจกต์ CFinalRelatedClass
- m_oaCFinalRelaterClass - เป็นแอเรย์ของพอยท์เตอร์ ใช้เก็บพอยท์เตอร์ชี้ไปยังอ็อบเจกต์ CFinalRelaterClass
- m_saGChildName - เป็นแอเรย์ของสตริง ใช้เก็บชื่อคลาสลูก
- m_saGParentName - เป็นแอเรย์ของสตริง ใช้เก็บชื่อคลาสพ่อ
- m_saPKAttribute - เป็นแอเรย์ของสตริง ใช้เก็บชื่อแอททริบิวท์ที่ประกอบเป็นคีย์หลัก

อ็อบเจกต์ CFinalAttribute

เป็นอ็อบเจกต์สำหรับแอททริบิวท์ของคลาสซึ่งผ่านการปรับแก้ไขเรียบร้อยแล้ว สร้างจากนิยามคลาส CFinalAttribute ประกอบด้วยแอททริบิวท์ต่างๆดังนี้

- m_attributeName - ใช้เก็บชื่อแอททริบิวท์ ได้จากแอททริบิวท์ m_attributeName ของอ็อบเจกต์ CAttribute
- m_dataType - ใช้เก็บชื่อประเภทข้อมูล ได้จากแอททริบิวท์ m_dataType ของอ็อบเจกต์ CAttribute
- m_size - ใช้เก็บขนาดของข้อมูล ได้จากแอททริบิวท์ m_dataType ของอ็อบเจกต์ CAttribute

อ็อบเจกต์ CFinalIndexClass

เป็นอ็อบเจกต์สำหรับดรรชนีของคลาสซึ่งผ่านการปรับแก้ไขเรียบร้อยแล้ว สร้างจากนิยามคลาส CFinalIndexClass ประกอบด้วยแอททริบิวท์ต่างๆดังนี้

- m_indexName - ใช้เก็บชื่อดรรชนี ได้จากแอททริบิวท์ m_indexName ของอ็อบเจกต์ CTentativeClassIndex
- m_saIndexAttribute - ใช้เก็บชื่อคอลัมน์ที่ประกอบเป็นดรรชนี ได้จากแอททริบิวท์ m_saIndexAttribute ของอ็อบเจกต์ CTentativeClassIndex

อ็อบเจกต์ CFinalRelaterClass

เป็นอ็อบเจกต์สำหรับคลาสผู้อ้างอิงโดยคลาสอื่นซึ่งผ่านการปรับแก้ไขเรียบร้อยแล้ว สร้างจากนิยามคลาส CFinalRelaterClass ประกอบด้วยแอททริบิวท์ต่างๆดังนี้

- m_relaterClassName – ใช้เก็บชื่อคลาสผู้อ้างอิง ได้จากแอททริบิวต์ m_relaterClassName ของอ็อบเจกต์ CAssoCRelaterClassRelationship
- m_relationType – ใช้เก็บความสัมพันธ์ระหว่างคลาสผู้อ้างอิงกับคลาสผู้ถูกอ้างอิง ได้จากแอททริบิวต์ m_relationType ของอ็อบเจกต์ CAssoCRelaterClassRelationship
- m_saRelaterFKAttribute - ใช้เก็บชื่อแอททริบิวต์ที่ประกอบเป็นคีย์นอก ได้จากแอททริบิวต์ m_saRelaterFKAttribute ของอ็อบเจกต์ CAssoCRelaterClassRelationship

อ็อบเจกต์ CFinalRelatedClass

เป็นอ็อบเจกต์สำหรับคลาสผู้ถูกอ้างอิงโดยคลาสอื่นซึ่งผ่านการปรับแก้ไขเรียบร้อยแล้ว สร้างจากนิยามคลาส CFinalRelatedClass ประกอบด้วยแอททริบิวต์ต่างๆดังนี้

- m_relatedClassName – ใช้เก็บชื่อคลาสผู้ถูกอ้างอิง ซึ่งได้จากแอททริบิวต์ m_relatedClassName ของอ็อบเจกต์ CAssoCRelatedClassRelationship
- m_relationType – ใช้เก็บความสัมพันธ์ระหว่างคลาสผู้ถูกอ้างอิงกับคลาสผู้อ้างอิง ได้จากแอททริบิวต์ m_relationType ของอ็อบเจกต์ CAssoCRelatedClassRelationship
- m_saRelaterFKAttribute - ใช้เก็บชื่อแอททริบิวต์ที่ประกอบเป็นคีย์นอก ได้จากแอททริบิวต์ m_saRelaterFKAttribute ของอ็อบเจกต์ CAssoCRelatedClassRelationship

ขั้นตอนการปรับแผนแบบเชิงวัตถุขั้นสุดท้าย

1. อ่านอ็อบเจกต์สำหรับคลาส CTentativeClass จากตัวแปร m_oaCTentativeClass ที่ละอ็อบเจกต์พร้อมอ็อบเจกต์ CAttribute CTentativeClassIndex CAssoRelaterClass CAssoRelatedClass CAssoRelatedClassRelationship และ CAssoRelaterClassRelationship
2. สร้างอ็อบเจกต์สำหรับคลาส จากนิยามคลาส CFinalizedClass
 - ก. กำหนดค่าให้ m_className เป็นค่าจากแอททริบิวต์ m_className ของอ็อบเจกต์ CTentativeClass

- ข. กำหนดค่าให้ `m_saGChildName` เป็นค่าจากแอททริบิวต์ `m_saGChildName` ของอ็อบเจ็กต์ `CTentativeClass`
- ค. กำหนดค่าให้ `m_saGParentName` เป็นค่าจากแอททริบิวต์ `m_saGParentName` ของอ็อบเจ็กต์ `CTentativeClass`
- ง. กำหนดค่าให้ `m_saPKAttribute` เป็นค่าจากแอททริบิวต์ `m_saPKAttribute` ของอ็อบเจ็กต์ `CTentativeClass`
- จ. สร้างอ็อบเจ็กต์ `CFinalAttribute` จากนิยามคลาส `CFinalAttribute`
- กำหนดค่าให้ `m_attributeName` เป็นค่าจากแอททริบิวต์ `m_attributeName` ของอ็อบเจ็กต์ `CAttribute`
 - กำหนดค่าให้ `m_dataType` เป็นค่าจากแอททริบิวต์ `m_dataType` ของอ็อบเจ็กต์ `CAttribute`
 - กำหนดค่าให้ `m_size` เป็นค่าจากแอททริบิวต์ `m_size` ของอ็อบเจ็กต์ `CAttribute`
- ฉ. เก็บอ็อบเจ็กต์ `CFinalAttribute` ในแอททริบิวต์ `m_oaCFinalAttribute` ของอ็อบเจ็กต์ `CFinalizedClass`
- ช. สร้างอ็อบเจ็กต์ `CFinalIndex` จากนิยามคลาส `CFinalIndex`
- กำหนดค่าให้ `m_indexName` เป็นค่าจากแอททริบิวต์ `m_indexName` ของอ็อบเจ็กต์ `CTentativeClassIndex`
 - กำหนดค่าให้ `m_saIndexAttribute` เป็นค่าจากแอททริบิวต์ `m_saIndexAttribute` ของอ็อบเจ็กต์ `CTentativeClassIndex`
- ซ. เก็บอ็อบเจ็กต์ `CFinalAttribute` ในแอททริบิวต์ `m_oaCFinalAttribute` ของอ็อบเจ็กต์ `CFinalizedClass`
- ฅ. สร้างอ็อบเจ็กต์ `CFinalRelaterClass` จากนิยามคลาส `CFinalRelaterClass`
- กำหนดค่าให้ `m_relaterClassName` เป็นค่าจากแอททริบิวต์ `m_relaterClassName` ของอ็อบเจ็กต์ `CAssoCRelaterClassRelationship`
 - กำหนดค่าให้ `m_relationType` เป็นค่าจากแอททริบิวต์ `m_relationType` ของอ็อบเจ็กต์ `CAssoCRelaterClassRelationship`
 - กำหนดค่าให้ `m_saFKAttribute` เป็นค่าจากแอททริบิวต์ `m_saFKAttribute` ของอ็อบเจ็กต์ `CAssoCRelaterClassRelationship`
- ญ. เก็บอ็อบเจ็กต์ `CFinalRelaterClass` ในแอททริบิวต์ `m_oaCFinalRelaterClass` ของอ็อบเจ็กต์ `CFinalizedClass`
- ฎ. สร้างอ็อบเจ็กต์ `CFinalRelatedClass` จากนิยามคลาส `CFinalRelatedClass`

- กำหนดค่าให้ `m_relatedClassName` เป็นค่าจากแอททริบิวต์ `m_relatedClassName` ของอ็อบเจ็กต์ `CAssoCRelatedClassRelationship`
 - กำหนดค่าให้ `m_relationType` เป็นค่าจากแอททริบิวต์ `m_relationType` ของอ็อบเจ็กต์ `CAssoCRelatedClassRelationship`
 - กำหนดค่าให้ `m_saFKAttribute` เป็นค่าจากแอททริบิวต์ `m_saFKAttribute` ของอ็อบเจ็กต์ `CAssoCRelatedClassRelationship`
 - เก็บอ็อบเจ็กต์ `CFinalRelatedClass` ในแอททริบิวต์ `m_oaCFinalRelatedClass` ของอ็อบเจ็กต์ `CFinalizedClass`
3. เปลี่ยนประเภทข้อมูลในแอททริบิวต์ `m_dataType` ของอ็อบเจ็กต์ `CFinalAttribute` เป็นประเภทข้อมูลของ POET C++ Data Type

4.7 กลุ่มคลาสสร้างเพิ่มความ HCD และสร้างฐานข้อมูล

วัตถุประสงค์เพื่อนำแผนแบบเชิงวัตถุที่ได้ ซึ่งเก็บในรูปของอ็อบเจ็กต์ `CFinalizedClass` `CFinalAttribute` `CFinalIndexClass` `CFinalRelaterClass` และ `CFinalRelaterClass` มาสร้างเพิ่มความ POET C++ Class Definition หรือเพิ่ม HCD สำหรับคลาสต่างๆ โดยกำหนดให้ 1 คลาสเป็น 1 เพิ่ม ชื่อคลาสเป็นชื่อเพิ่มนามสกุล HCD

เพิ่มความ HCD ที่ได้จะถูกนำไปสร้างฐานข้อมูลบนระบบจัดการฐานข้อมูล POET ต่อไป

```

#include <ptcomp.hxx>
#include <ptblob.hxx>
#include <ptdatetime.hxx> } 1

persistent class Entry;
persistent class Customer; } 2

persistent class Invoice
{
public:
    Customer *pCustomer;
    lset<Entry *> psetEntry; } 4

    long invoiceID;
    PtString invoiceName;
    long customerID; } 5

    Invoice& operator = ( const Invoice& );
    Invoice( const Invoice& ); } 6

public:
    Invoice();
    virtual ~Invoice(); } 7

    useindex NameIndex; } 8
};

indexdef NameIndex : Invoice
{
    invoiceName;
} } 9

```

รูปที่ 4.7 โครงสร้างเพิ่มข้อความ POET C++ Class Definition หรือเพิ่ม HCD

รูปที่ 4.7 แสดงโครงสร้างเพิ่มข้อความ POET C++ Class Definition หรือเพิ่ม HCD มีส่วนประกอบ 9 ส่วนดังนี้

1. Include Files ซึ่งจำเป็นสำหรับเพิ่มข้อมูล HCD โปรแกรมจะใส่ 3 บรรทัดให้กับทุกๆ เพิ่ม HCD
2. Forward Declaration สำหรับการอ้างอิงคลาสอื่น ในที่นี้มีการอ้างอิงคลาส Customer และ Entry ในส่วนที่ 4 ส่วนนี้โปรแกรมจะอาศัยอ็อบเจกต์ CFinalRelaterClass และ CFinalRelatedClass สำหรับการทำ Forward Declaration

3. ส่วนหัวของการนิยามคลาส โปรแกรมจะเติมคำว่า persistent เพื่อเป็นการบอกว่า อ็อบเจกต์ของคลาสนี้จะถูกเก็บในฐานข้อมูล

ถ้าคลาสที่กำลังนิยาม มีการสืบทอดคุณสมบัติจากคลาสอื่น โปรแกรมจะต่อท้ายชื่อคลาสด้วยเครื่องหมาย Colon (:) และตามด้วยชื่อคลาสที่ถูกสืบทอดคุณสมบัติ
4. พอยท์เตอร์หรือกลุ่มของพอยท์เตอร์อ้างอิงไปยังคลาสอื่น โปรแกรมจะกำหนดพอยท์เตอร์หรือกลุ่มของพอยท์เตอร์ให้ โดยโปรแกรมทราบจากอ็อบเจกต์ CFinalRelatedClass และ CFinalRelaterClass
5. กลุ่มของ Data Member ซึ่งโปรแกรมอ่านจากอ็อบเจกต์ CFinalAttribute
6. Copy Constructor เพื่อเป็นไปตามมาตรฐานกำหนด Class Definition ซึ่งระบบจัดการฐานข้อมูล POET การกำหนดหรือไม่กำหนด Copy Constructor ไม่มีความแตกต่างกัน
7. กำหนด Constructor และ Virtual Constructor สำหรับคลาส
8. กำหนดว่ามีการใช้ Index ชื่ออะไรสำหรับอ็อบเจกต์ของคลาสนั้นๆ
9. กำหนดนิยามของ Index โดยอาศัยอ็อบเจกต์ CFinalIndex

ขั้นตอนการสร้างแฟ้มข้อความ HCD และสร้างฐานข้อมูล

1. กำหนดตำแหน่งปริยายสำหรับการเก็บแฟ้ม HCD และตำแหน่งสำหรับการสร้างฐานข้อมูลเป็น "C:\\DBThesis"
2. กำหนดชื่อฐานข้อมูลข้อมูลเป็น "base"
3. กำหนดชื่อพจนานุกรมข้อมูลสำหรับฐานข้อมูลเป็น "dict"
4. อ่านอ็อบเจกต์ CFinalizedClass ที่ละอ็อบเจกต์และอ็อบเจกต์ที่เกี่ยวข้อง
 - ก. ใช้ค่าใน m_className ของอ็อบเจกต์ CFinalizedClass เป็นชื่อแฟ้มข้อความนามสกุล HCD เช่นถ้าค่าใน m_className เป็น ClassA แฟ้มที่ได้ชื่อ ClassA.hcd
 - ข. สร้างแฟ้มข้อความชื่อและนามสกุลตามข้อ ก.
 - 3 บรรทัดแรกบรรจุข้อความดังนี้


```
#include <ptcomp.hxx>
\n#include <ptblob.hxx>
\n#include <ptdatetime.hxx>
```

ค. ตรวจสอบดูว่ามีคลาสอื่นอ้างอิงถึงหรือไม่ ถ้ามี

- บรรทัดถัดมากำหนด Forward Declaration สำหรับคลาสนั้น เช่นสมมุติคลาสที่อ้างอิงชื่อ ClassB กำหนดดังนี้

```
persistent class ClassB;
```

- ตรวจสอบความสัมพันธ์ระหว่างคลาสผู้ถูกอ้างอิงกับคลาสที่อ้างอิง
 - ถ้าความสัมพันธ์เป็น "MANY-TO-ONE" หรือ "MANY-TO-MANY" กำหนดในส่วน public ดังนี้

```
lset< ClassB * > psetClassB;
```

- ถ้าความสัมพันธ์เป็น "ONE-TO-ONE" กำหนดในส่วน public ดังนี้

```
ClassB *pClassB;
```

ง. ตรวจสอบดูว่ามีการอ้างอิงไปยังคลาสอื่นหรือไม่ ถ้ามี

- บรรทัดถัดมากำหนด Forward Declaration สำหรับคลาสนั้น เช่นสมมุติคลาสที่อ้างอิงชื่อ ClassC กำหนดดังนี้

```
persistent class ClassC;
```

- ตรวจสอบความสัมพันธ์ระหว่างคลาสผู้อ้างอิงกับคลาสผู้ถูกอ้างอิง
 - ถ้าความสัมพันธ์เป็น "ONE-TO-ONE" หรือ "ONE-TO-MANY" กำหนดในส่วน public ดังนี้

```
ClassC *pClassC;
```

- ถ้าความสัมพันธ์เป็น "MANY-TO-MANY" กำหนดในส่วน public ดังนี้

```
lset< ClassC * > psetClassC;
```

จ. ตรวจสอบดูว่าเป็นคลาสลูกคลาสอื่นหรือไม่โดยการอ่านจากค่าใน

m_saGParentName ของอ็อบเจกต์ CFinalizedClass ถ้าเป็น

- จากรูปที่ 4.7 หมายเลข 3 กำหนดการสืบทอดคุณสมบัติจากคลาสพ่อ เช่นถ้าทราบว่าคลาสพ่อชื่อ ParentClass กำหนดดังนี้

```
persistent class ClassA : ParentClass
```

ฉ. กำหนด Constructor และ Destructor ในส่วน public

สมมุติคลาสที่ได้ชื่อ ClassA กำหนด Constructor และ Destructor ดังนี้

```
ClassA();
```

```
~ClassA();
```


- ข. อ่านอ็อบเจกต์ CFinalAttribute ที่ละอ็อบเจกต์
 สมมุติค่าในแอททริบิวท์ m_attributeName เป็น "invoiceName"
 ค่าในแอททริบิวท์ m_dataType เป็น "PtString"
 ค่าในแอททริบิวท์ m_size ว่าง
 นำค่าจากทั้ง 3 แอททริบิวท์มาต่อกันโดยมีช่องว่างคั่นปิดท้ายด้วยเครื่องหมาย ";"
 กำหนดในส่วน public ผลจะได้ดังนี้

```
PtString invoiceName;
```

ทำเช่นนี้จนครบทุกแอททริบิวท์

5. วนไปทำข้อ 4. สำหรับคลาสต่อไปจนครบทุกคลาส
6. เมื่อครบทุกคลาสแล้ว สร้างแฟ้ม base.hcd และ CreateDB.bat ภายในแฟ้ม base.hcd จะบรรจุบรรทัดของการอินคลูดแฟ้มของคลาสทั้งหมดที่เกี่ยวข้องสำหรับการสร้างฐานข้อมูล และแฟ้ม CreateDB.bat จะบรรจุคำสั่งสำหรับการสร้างฐานข้อมูลบนระบบจัดการฐานข้อมูล POET
7. สั่งให้มีการทำงานตามคำสั่งในแฟ้ม CreateDB.bat เพื่อสร้างฐานข้อมูลบนระบบจัดการฐานข้อมูล POET

```
#include "Customer.hcd"
#ifndef _Product
#define _Product
#include "Product.hcd"
#endif

#include "Car.hcd"
#include "PrescriptionDrug.hcd"
#include "Invoice.hcd"
#include "Entry.hcd"
#ifndef _ReplicateAttribute1_ReplicateAttribute2
#define _ReplicateAttribute1_ReplicateAttribute2
#include "ReplicateAttribute1_ReplicateAttribute2.hcd"
#endif

#include "ReplicateAttribute1.hcd"
#include "ReplicateAttribute2.hcd"
#ifndef _InclusionDependency1_InclusionD2
#define _InclusionDependency1_InclusionD2
#include "InclusionDependency1_InclusionD2.hcd"
#endif

#include "InclusionDependency1.hcd"
#include "InclusionD2.hcd"
```

ตารางที่ 4.1 ตัวอย่างแฟ้ม base.HCD

ตารางที่ 4.1 เป็นตัวอย่างเพิ่ม base.hcd ซึ่งจะถูกสร้างขึ้นหลังจบการสร้างเพิ่มความ HCD และสร้างฐานข้อมูล

```
ptxx -xd dict base.hcd  
ptadmin -c -B base -D dict
```

ตารางที่ 4.2 ตัวอย่างเพิ่ม CreateDB.bat

ตารางที่ 4.2 เป็นตัวอย่างเพิ่ม CreateDB.bat ซึ่งจะถูกสร้างขึ้นสำหรับเรียกใช้สร้างฐานข้อมูล ในตอนท้ายของการสร้างเพิ่มความ HCD และสร้างฐานข้อมูล