

ปฏิบัติการที่ 8 คุณสมบัติการสืบทอด (Inheritance)

บทเรียนย่อย

- คุณสมบัติและโครงสร้างการสืบทอด (Inheritance)
- การ Overriding method
- สมาชิกคลาสประเภท protected

วัตถุประสงค์การเรียนรู้

- เรียนรู้และทำความเข้าใจคุณสมบัติการสืบทอด (Inheritance)
- เรียนรู้การ Overriding method
- เรียนรู้ และทำความเข้าใจสมาชิกคลาสประเภท protected

ให้นักศึกษาสร้าง Directory ชื่อว่า Lab08 สำหรับทดลองปฏิบัติการดังต่อไปนี้

ตอนที่ 1 คุณสมบัติและโครงสร้างการสืบทอด (Inheritance)

การสืบทอดคุณสมบัติ คือการสร้างคลาสใหม่ขึ้นมาจากคลาสเดิมที่มีอยู่ โดยที่คลาสที่สร้างขึ้นใหม่นั้นจะมีคุณสมบัติทั้งหมดเช่นเดียวกับคลาสที่มีอยู่เดิมทุกประการ ("is-a") ซึ่งคลาสใหม่ที่สร้างขึ้นนั้นเรียกว่า คลาสลูก (derived classes) และคลาสที่มีอยู่เดิมนั้นเรียกว่า คลาสแม่ (base classes) การประกาศสืบทอดคุณสมบัติ (Inheritance) สามารถทำได้ดังนี้

```
class className: memberAccessSpecifier baseClassName
{
    member list
};
```

โดยกำหนดให้ memberAccessSpecifier คือประเภทของการเข้าถึงสมาชิก public, protected และ private และในกรณีที่ไม่มีกำหนด memberAccessSpecifier จะมีค่าเป็น private โดยอัตโนมัติ

```
เช่น class circle: public shape
{
}
```

เป็นการสร้างคลาส circle ที่สืบทอดคุณสมบัติมาจากคลาส shape

```
และ class circle: private shape
{
}
```

เป็นการสร้างคลาส circle ที่สืบทอดคุณสมบัติจากคลาส shape แบบ private ซึ่งมีผลให้สมาชิกในคลาส shape ที่เป็นประเภท public ทุกตัวเปลี่ยนเป็นสมาชิกประเภท private ภายใต้คลาส circle ทำให้สมาชิกดังกล่าวนี้ไม่สามารถเข้าถึงจากภายนอกได้อีก

ให้นักศึกษาทำรายละเอียดย่อยตัวอย่างโปรแกรม ในโฟลเดอร์ ex01_inheritance จากนั้นทำการ make และรันโปรแกรม แล้วตอบคำถามดังต่อไปนี้

รหัสนิสิต ชื่อ-สกุล กลุ่ม.....

Base Class คือ

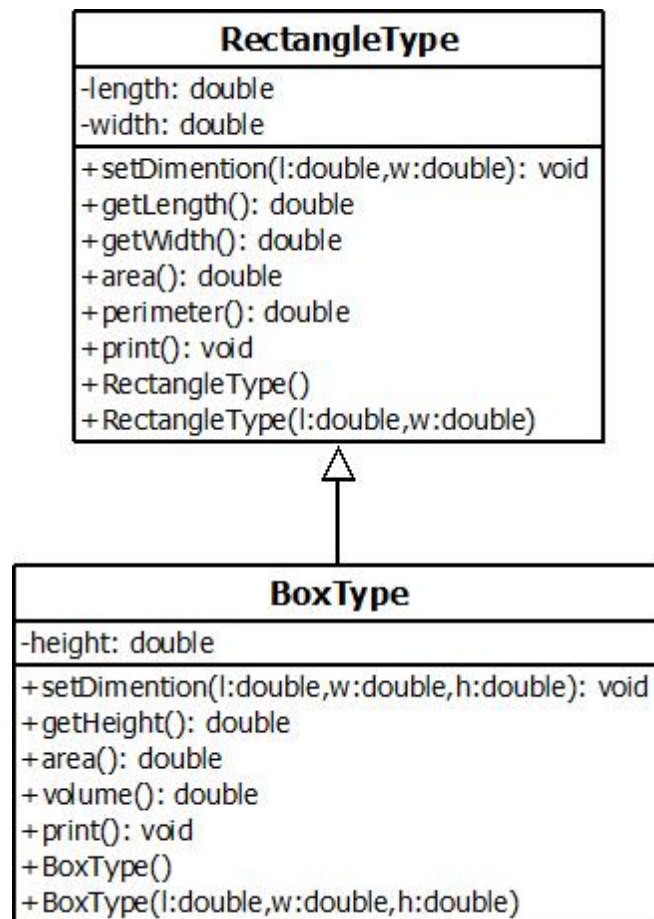
Derived Class คือ

แล้วทำการเขียนแผนภาพ Class Diagram



ตอนที่ 2 การ Overriding method

เมื่อคลาสลูกที่สืบทอดคุณสมบัติจากคลาสแม่ นอกจากคลาสลูกสามารถเพิ่มแอททริบิวต์ และเมธอดจากที่มีอยู่เดิมในคลาสแม่ได้แล้ว คลาสลูกนั้นสามารถสร้างเมธอดที่ชื่อเมธอดซ้ำเดิมกับคลาสแม่ (เหมือนกันทั้งจำนวน และชนิดของพารามิเตอร์) เรียกว่า overriding method ซึ่งเมื่อทำการ overriding method แล้วเมื่อมีเรียกใช้งานเมธอดดังกล่าวจะมาทำงานในเมธอดของคลาสลูกแทน ให้นิสิตศึกษาถึงรายละเอียดตัวอย่างโปรแกรมในไฟล์เดอร์ ex02_overriding จากนั้นทำการ make และรันโปรแกรมเพื่อสังเกตผล



สังเกต constructor ของคลาส BoxType ที่

```

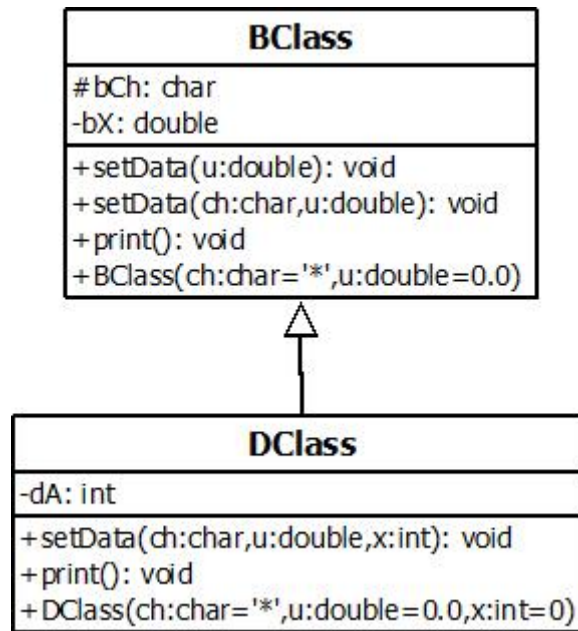
BoxType::BoxType(double l, double w, double h) : RectangleType(l, w)
{
    ...
}
    
```

หมายถึงภายใน constructor ของคลาส BoxType มีการเรียกใช้งาน constructor ของคลาสแม่ RectangleType โดยส่งพารามิเตอร์สองตัวคือ l และ w แทน default constructor

ตอนที่ 3 สมาชิกคลาสประเภท protected

สมาชิกคลาสประเภท protected เป็นสมาชิกที่สามารถเข้าถึงได้จากคลาสลูกที่สืบทอดคุณสมบัติ (Inheritance) จากคลาสแม่ นั่น แต่ไม่สามารถเข้าถึงได้จากภายนอกอื่นๆ

ให้นักศึกษาถึงรายละเอียดโปรแกรมในโฟลเดอร์ ex03_protected จากนั้นทำการ make และรัน โปรแกรมเพื่อสังเกตผล



1. ตัวแปร bCh เป็นประเภทใด และสามารถเข้าถึงได้จากคลาสใดบ้าง

.....

.....

.....

2. ตัวแปร bX เป็นประเภทใด และสามารถเข้าถึงได้จากคลาสใดบ้าง

.....

.....

.....

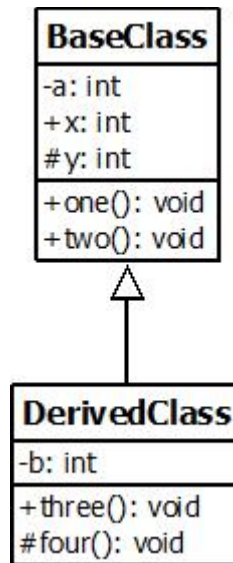
3. ตัวแปร dA เป็นประเภทใด และสามารถเข้าถึงได้จากคลาสใดบ้าง

.....

.....

.....

ให้นักศึกษาถึงรายละเอียดของโปรแกรมในไฟล์เตอร์ ex04_pvtInhr จากนั้นทำการ make และรันโปรแกรม



4. ให้นักนิสิตทดสอบเขียนโปรแกรมเพิ่มในไฟล์ main.cpp

cout << endl << endl; //Line 3

ab.one(); //Line 4

ทดสอบ make และรันโปรแกรม ผลเป็นอย่างไร เพราะเหตุใด

.....

.....

.....

แบบฝึกหัด

1. ให้นิสิตสร้างไฟล์เดอร์ ex08_circleNcylinder แล้วทำการออกแบบและเขียนคลาส CircleType ซึ่งเป็นคลาสเกี่ยวกับวงกลม โดยเขียนแบบแยกส่วน โครงสร้างคลาส รายละเอียดคลาส ออกจากกัน โดยมีรายละเอียดดังนี้

- มี attribute เก็บรัศมี (r) เป็นประเภท protected
- มี method สำหรับหาเส้นรอบวงกลม (circumference)
- มี method สำหรับหาพื้นที่วงกลม (area)

จากนั้นให้นิสิตออกแบบ และเขียนคลาส CylinderType เป็นคลาสเกี่ยวกับทรงกระบอก ที่สืบทอด คุณสมบัติมาจากคลาส CircleType โดยมีรายละเอียดดังนี้

- มี attribute เก็บค่าความสูง (h)
- มี method หาพื้นที่หน้าตัดทรงกระบอก (area) ซึ่ง overriding จาก method area เดิม
- มี method หาปริมาตรของทรงกระบอก

จากนั้นให้นิสิตเขียนไฟล์ main.cpp เพื่อทดสอบการใช้งานคลาส CircleType และ CylinderType ใน ทุก method และสร้าง Makefile สำหรับคอมไพล์โปรแกรม

หลังจากนั้น ให้นิสิตเขียนแผนภาพ Class Diagram ที่นิสิตได้ออกแบบในพื้นที่ด้านล่างนี้ พร้อมคัดลอกโปรแกรมลงบนกระดาษ

2. ให้นิสิตออกแบบและเขียนคลาส Person เป็นคลาสเกี่ยวกับบุคคล โดยเขียนแบบแยกส่วน โครงสร้างคลาส รายละเอียดคลาส ออกจากกัน โดยมีรายละเอียดดังนี้

- มี attribute สำหรับเก็บค่าชื่อ
- มี attribute สำหรับเก็บค่านามสกุล
- มี method สำหรับ get/set ข้อมูลต่าง ๆ
- มี method สำหรับพิมพ์ (print) ชื่อและนามสกุล

จากนั้นให้นิสิตออกแบบ และเขียนคลาส Student เป็นคลาสเกี่ยวกับนักศึกษา ที่สืบทอด คุณสมบัติมาจาก คลาส Person โดยมีรายละเอียดดังนี้

- มี attribute เก็บรหัสประจำตัวนักศึกษา (studentId)
- มี method สำหรับ get/set ข้อมูลรหัสประจำตัว
- มี method สำหรับพิมพ์ (print) โดยทำการพิมพ์ รหัสประจำตัว ชื่อและนามสกุล โดย method print นี้ให้ทำการ override method print จากคลาส Person
- มี method checkEqual(Student otherStudent) สำหรับตรวจสอบว่าตรวจสอบว่าเป็นนักศึกษามีรหัสประจำตัวเหมือนกันหรือไม่ หากรหัสนักศึกษาเหมือนกันให้คืนค่าจริง (true) หากไม่ใช่ให้คืนค่าเท็จ (false)

หลังจากนั้นให้นิสิตออกแบบ และเขียนคลาส UnderGraduate เป็นคลาสเกี่ยวกับนักศึกษาที่กำลังเรียน ที่สืบทอด คุณสมบัติมาจากคลาส Student โดยมีรายละเอียดดังนี้

- มี attribute เก็บค่าชั้นปี (level)
- มี method สำหรับ get/set ข้อมูลชั้นปี (ให้กำหนดค่าของชั้นปีอยู่ในช่วง 1 – 4 เท่านั้น)
- มี method สำหรับพิมพ์ (print) โดยทำการพิมพ์ รหัสประจำตัว ชื่อ นามสกุล และชั้นปี โดย method print นี้ให้ทำการ override method print จากคลาส Student

จากนั้นให้นิสิตเขียนไฟล์ main.cpp เพื่อทดสอบการใช้งานคลาส Person คลาส Student และ UnderGraduate ใน ทุก method และสร้าง Makefile สำหรับคอมไพล์โปรแกรม เช่น

- สร้าง object จาก Student และ UnderGraduate แล้วสั่ง print
- สร้าง object จาก Student จำนวน 2 object แล้วเปรียบเทียบกัน (checkEqual)
- สร้าง object จาก UnderGraduate จำนวน 2 object แล้วเปรียบเทียบกัน (checkEqual)

หลังจากนั้น ให้นิสิตเขียนแผนภาพ Class Diagram ที่นิสิตได้ออกแบบในพื้นที่ด้านล่างนี้ พร้อมคัดลอก โปรแกรมลงบนกระดาษ