

บทที่ 4

การสืบทอด Inheritance





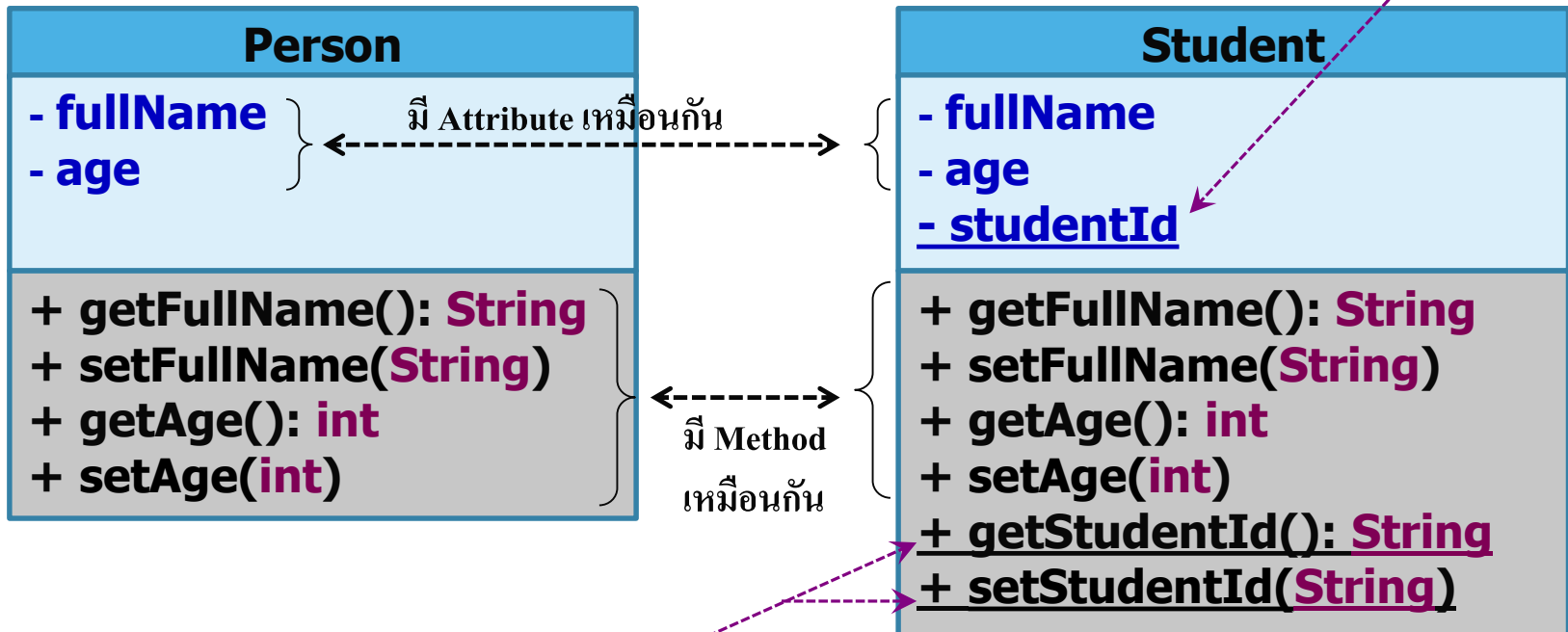
การสืบทอดคลาส

- ❖ การสืบทอดคลาส หรือ Inheritance คือ การส่งต่อ attribute และ method จากคลาสหนึ่งไปยังคลาสใหม่
- ❖ คลาสลูก จะได้รับ attribute และ method จากคลาสที่สืบทอด หรือ คลาสแม่ มาทุกอย่าง
- ❖ การสืบทอดทำเพื่อเพิ่มส่วน attribute และ method เพื่อให้ทำงานเฉพาะเจาะจงกว่าคลาสเดิมที่มีอยู่

เมื่อไหร่ต้องสืบทอด



Attribute ที่เพิ่มเติมขึ้นมา



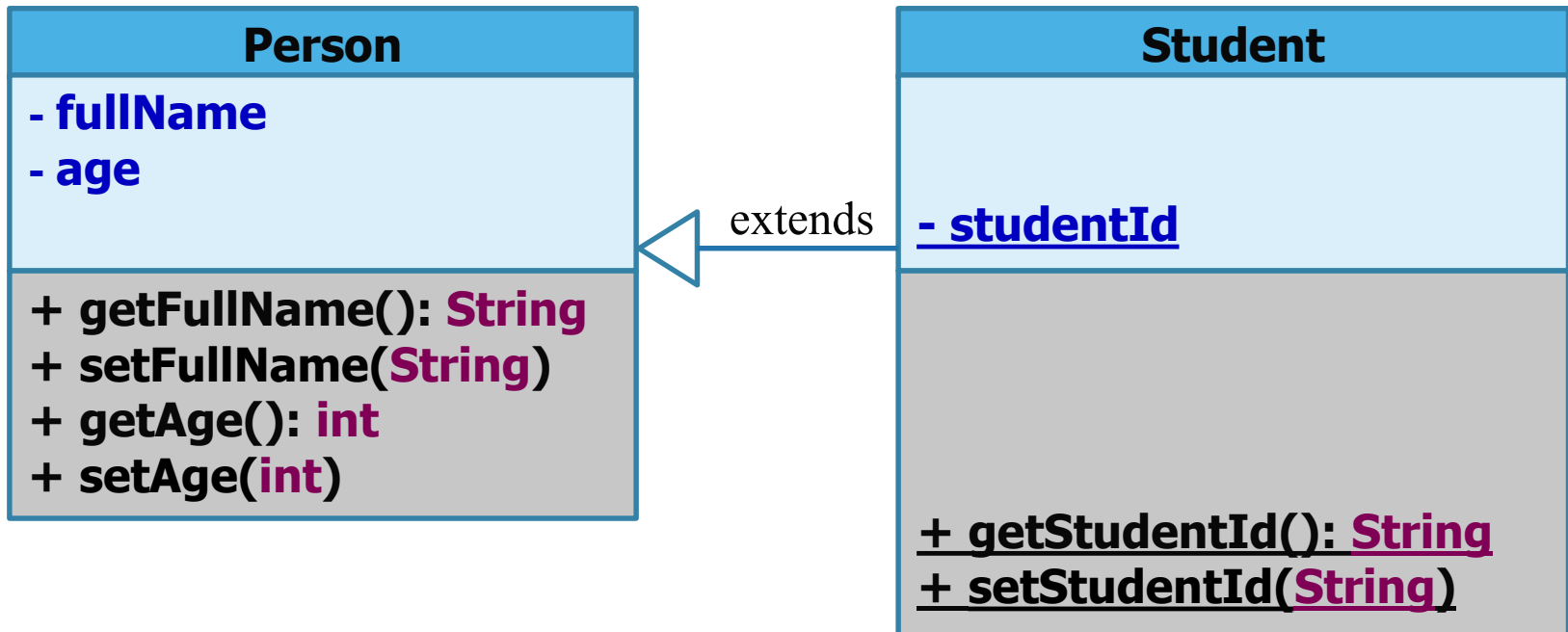
Method ที่เพิ่มเติมขึ้นมา



แยกส่วนที่แตกต่างออกเป็นคลาสลูก

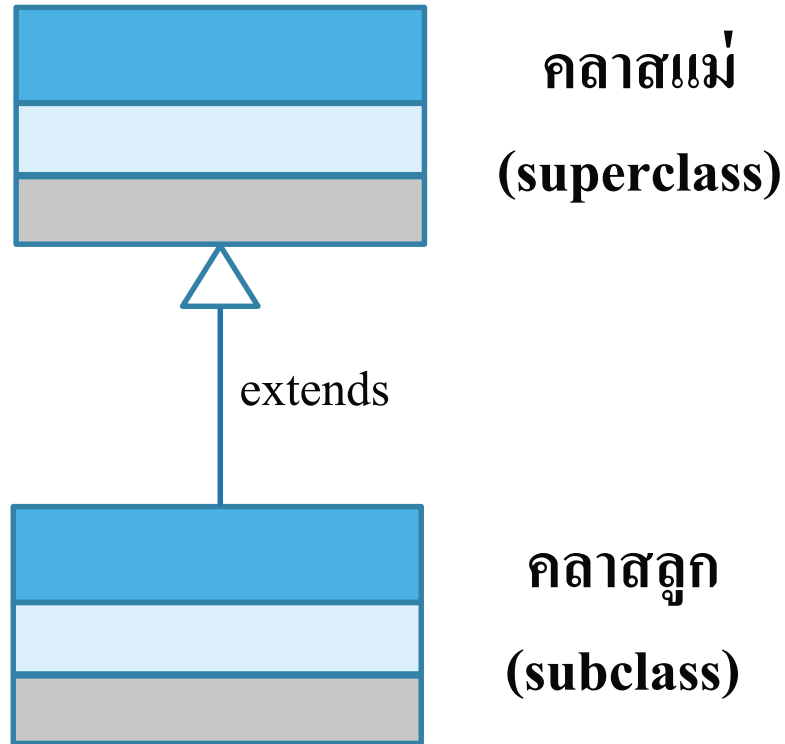
คลาสแม่

คลาสลูก





สัญลักษณ์การสืบทอด





รูปแบบคลาสแม่ และคลาสลูก

```
class ชื่อคลาสแม่ {
```

```
// Attribute
```

เปลี่ยนจาก **private** เป็น **protected**

เพื่อให้คลาสลูกเข้าถึงได้โดยไม่ต้อง

เรียกผ่าน getter/setter method

```
protected ชนิดตัวแปร ชื่อตัวแปรที่1;  
protected ชนิดตัวแปร ชื่อตัวแปรที่2;  
protected ชนิดตัวแปร ชื่อตัวแปรที่N;
```

```
// Getter&Setter Method
```

```
...
```

```
}
```

```
class ชื่อคลาสลูก extends ชื่อคลาสแม่ {
```

```
// Attribute
```

```
private ชนิดตัวแปร ชื่อตัวแปรที่1;  
private ชนิดตัวแปร ชื่อตัวแปรที่2;
```

```
// Getter&Setter Method
```

```
...
```

```
}
```



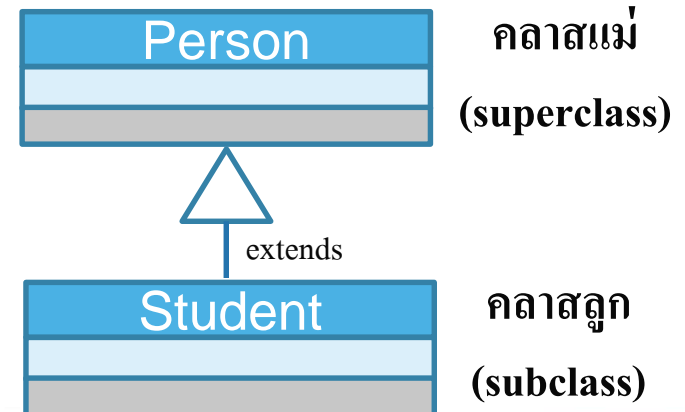
ตัวอย่าง

คลาสแม่ (superclass)

```
class Person {  
    // Attribute  
    protected String fullName;  
    protected int age;  
  
    // Getter&Setter Method  
    public String getFullName( ) {  
        return fullName;  
    }  
    public void setFullName(String newFullName) {  
        fullName = newFullName;  
    }  
  
    public int getAge( ) {  
        return age;  
    }  
    public void setAge(int newAge) {  
        age = newAge;  
    }  
}
```

คลาสลูก (subclass)

```
class Student extends Person {  
    private String studentId;  
  
    public String getStudentId() {  
        return studentId;  
    }  
  
    public void setStudentId(String id) {  
        studentId = id;  
    }  
}
```



```

public class TestApp { คลาสทดสอบ
    public static void main(String args[]) {

        Student pop = new Student();
        pop.setFullName("Poppy");
        pop.setAge(21);
        pop.setStudentId("61100000-0");

        System.out.println(pop.getStudentId());
        System.out.println(pop.getFullName());
        System.out.println(pop.getAge() + " ปี");
    }
}

```

object

pop: Student
fullName = Poppy age = 21 studentId = 61100000-0

ตัวอย่างหน้าจอ

```

61100000-0
Poppy
21 ปี

```

```

class Person { คลาสแม่
    // Attribute
    protected String fullName;
    protected int age;

    // Getter&Setter Method
    public String getFullName( ) {
        return fullName;
    }
    public void setFullName(String newFullName) {
        fullName = newFullName;
    }

    public int getAge( ) {
        return age;
    }
    public void setAge(int newAge) {
        age = newAge;
    }
}

```

```

class Student extends Person { คลาสลูก
    private String studentId;

    public String getStudentId() {
        return studentId;
    }

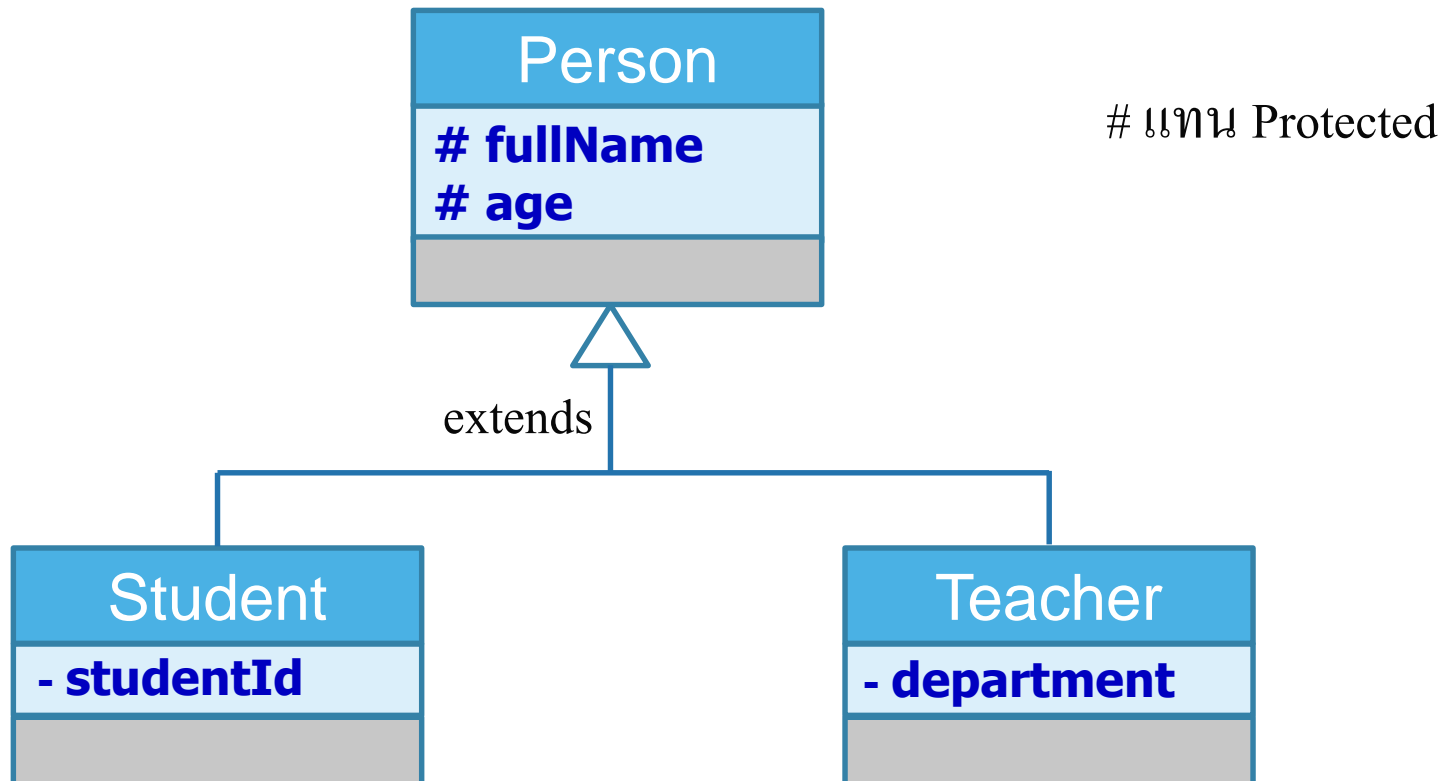
    public void setStudentId(String id) {
        studentId = id;
    }
}

```




กิจกรรม

❖ จงสร้างคลาส Teacher ที่สืบทอดจากคลาส Person โดยเพิ่ม attribute ชื่อ department





เมธอด toString()

❖ เมธอด toString จะถูกเรียกอัตโนมัติ เมื่ออ้างอิงชื่อ object นั้น โดยไม่ระบุชื่อ attribute หรือ method

คลาสทดสอบ

```
public class TestApp {
    public static void main(String args[]) {

        Student pop = new Student();
        pop.setFullName("Poppy");
        pop.setAge(21);
        pop.setStudentId("61100000-0");

        // เมธอด toString ถูกเรียกอัตโนมัติ
        System.out.println(pop);
    }
}
```

ตัวอย่างหน้าจอ

```
Student [fullName=Poppy,
age=21, id=61100000-0]
```

คลาสแม่

```
class Person {
    // Attribute
    protected String fullName;
    protected int age;
    // Setter Method
    public void setFullName(String newFullName) {
        fullName = newFullName;
    }
    public void setAge(int newAge) {
        age = newAge;
    }
}
```

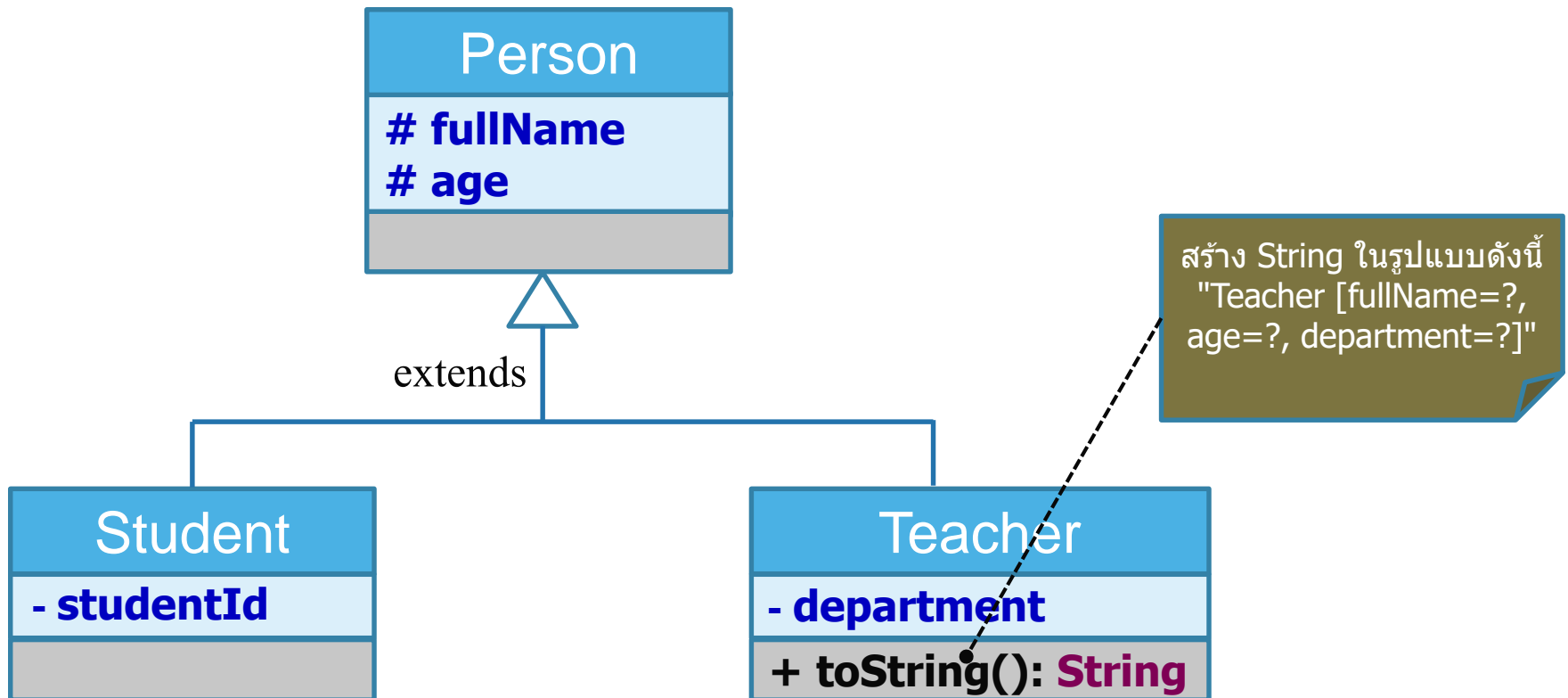
คลาสลูก

```
class Student extends Person {
    private String studentId;
    public void setStudentId(String id) {
        studentId = id;
    }
    public String toString() {
        return "Student [fullName=" + fullName +
            ", age=" + age + ", id=" + studentId + "];"
    }
}
```



กิจกรรม

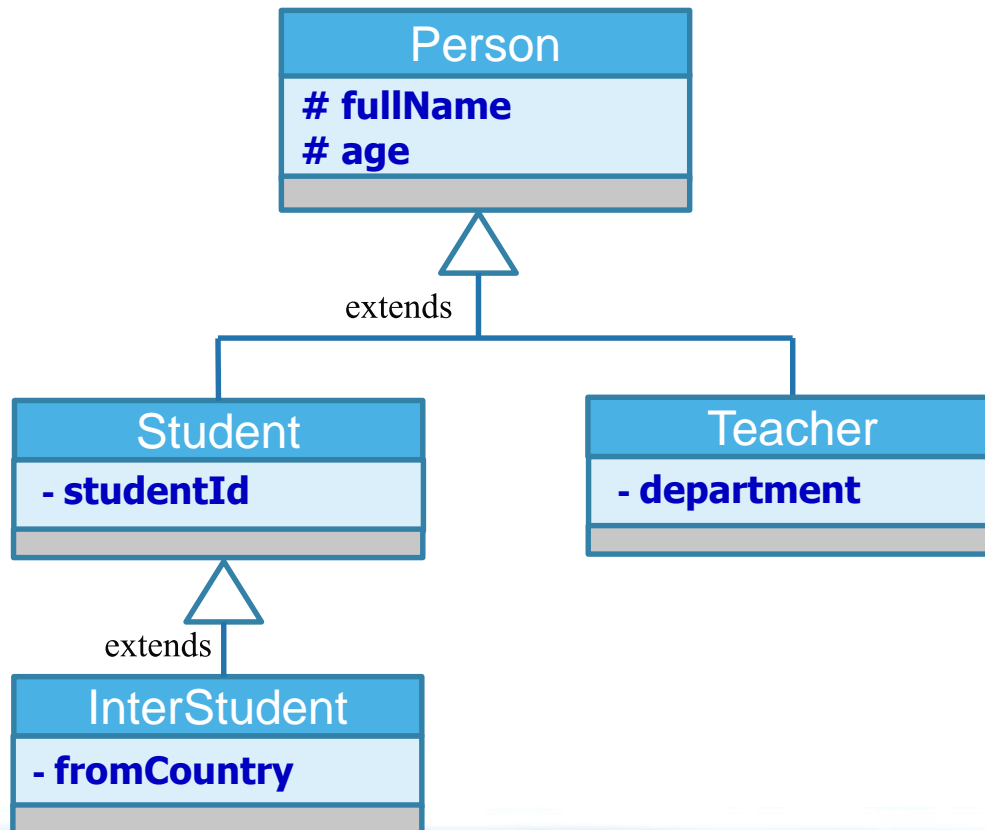
❖ จงเพิ่มเมธอด toString ในคลาส Teacher ในกิจกรรมก่อนหน้านี้ ตามรูปแบบที่กำหนด และทดสอบเรียกใช้





กิจกรรม

❖ สร้างคลาส InterStudent โดยให้สืบทอดจากคลาส Student และเพิ่ม attribute ชื่อ fromCountry เข้าไป





การใช้ super

❖ การใช้ super เป็นคำสั่งที่ใช้ในการระบุถึง object ของคลาสแม่ (superclass)

❖ รูปแบบการเรียก

super.ชื่อแอทริบิวต์ของคลาสแม่

super.ชื่อเมธอดของคลาสแม่()

❖ Constructor ของคลาสแม่สามารถเรียกจากคลาสลูกเช่นกัน รูปแบบดังนี้

super(); // เรียก Default Constructor ของคลาสแม่

super(arg_list, ...); // เรียก Constructor อื่นของคลาสแม่



การเรียก Constructor จากคลาสแม่

Constructor ของคลาสแม่จะไม่ได้รับการสืบทอดมาให้คลาสลูก ดังนั้นหากต้องการให้ Constructor ของคลาสแม่ทำงานต้องเขียนคำสั่งเรียกเองโดยใช้ **super(...)**;

คลาสทดสอบ

```
public class TestApp {  
    public static void main(String args[]) {  
        Student pop = new Student("Poppy", 21, "61100000-0");  
        System.out.println(pop);  
    }  
}
```

ตัวอย่างหน้าจอ

```
Person [Student  
[fullname=Poppy, age=21],  
id=61100000-0]
```

```
class Person { คลาสแม่  
    // Attribute  
    protected String fullName;  
    protected int age;  
  
    public Person(String fullName, int age) {  
        this.fullName = fullName;  
        this.age = age;  
    }  
}
```

คลาสลูก

```
class Student extends Person {  
    private String studentId;  
  
    public Student(String fullName, int age, String studentId) {  
        super(fullName, age);  
        this.studentId = studentId;  
    }  
  
    public String toString() {  
        return "Person [Student [fullname=" + fullName  
            + ", age=" + age + "], id=" + studentId + "];"  
    }  
}
```



กิจกรรม

❖ จงสร้างคลาสแม่ชื่อ Point2D และคลาสลูกชื่อ Point3D และทดสอบการทำงาน

