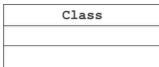
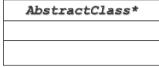
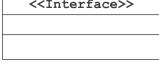
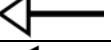
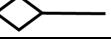


แบบฝึกปฏิบัติ ครั้งที่ 6

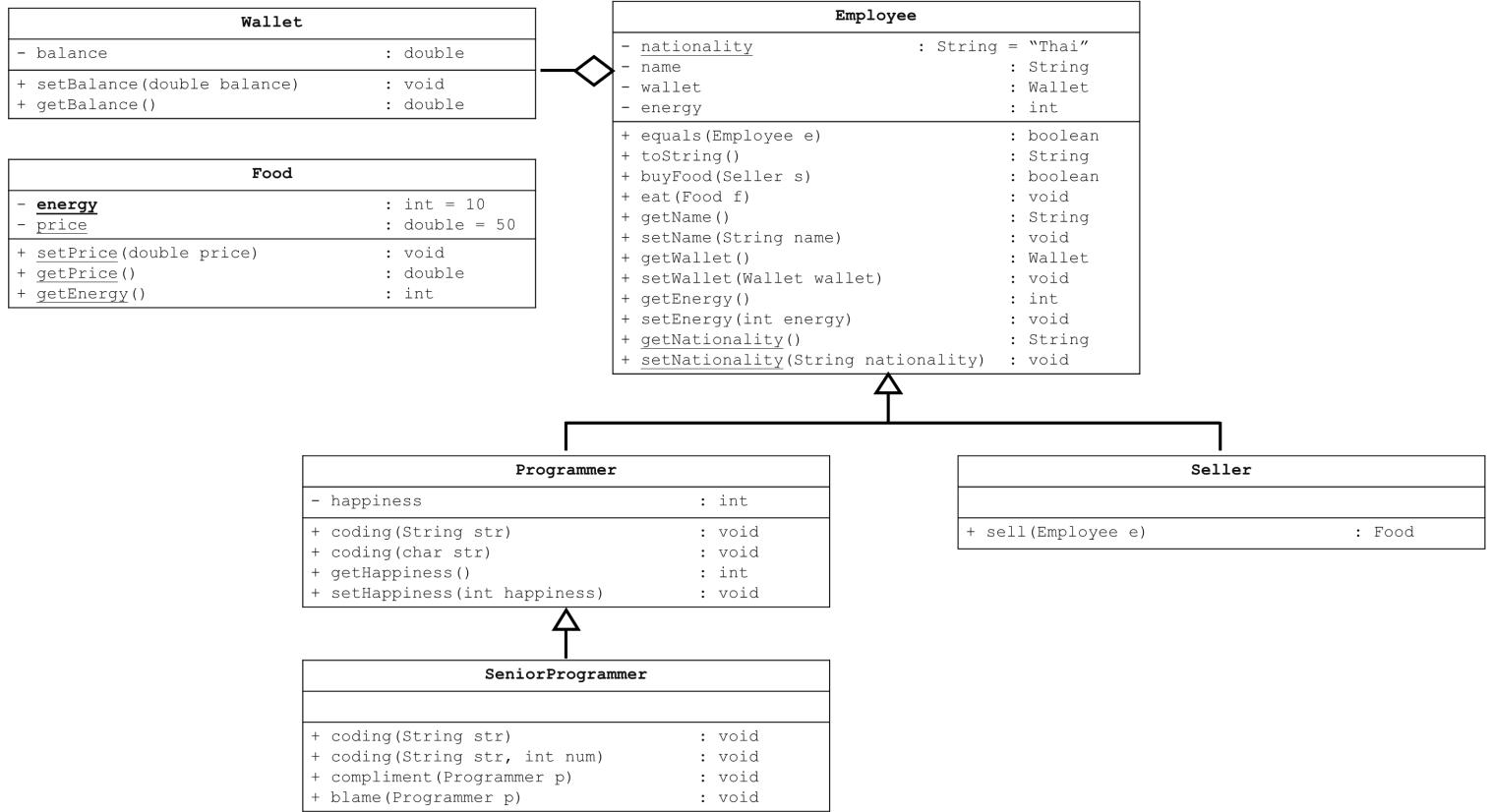
เรื่อง การเขียนโปรแกรมเชิงวัตถุ

วัตถุประสงค์ 1. เพื่อฝึกฝนการใช้คุณสมบัติการมีได้หลากหลายรูปแบบ (Polymorphism)

คำอธิบายเครื่องหมายดังแสดงในตารางต่อไปนี้

ลำดับ	สัญลักษณ์	ความหมาย
1	+	เป็นการกำหนด access mode เป็น public
2	-	เป็นการกำหนด access mode เป็น private
3	#	เป็นการกำหนด access mode เป็น protected
4		บ่งบอกว่าเป็นคลาสปกติ
5		ชื่อคลาสเป็นตัวเอียงหนา บ่งบอกว่าเป็นคลาสไม่สมบูรณ์
6		บ่งบอกว่าเป็นอินเตอร์เฟส
7		บ่งบอกการสืบทอด extends
8		บ่งบอกการ implements
9		บ่งบอกการเป็นส่วนหนึ่ง composite
10	<code>methodname () *</code>	ตัวเอียง บ่งบอกว่าเป็นเมธอดไม่สมบูรณ์
11	attribute หรือ methodname ()	ตัวหนา บ่งบอกความเป็น final
12	<u>attribute</u> หรือ <u>methodname ()</u>	<u>ตัวขีดเส้นใต้</u> บ่งบอกความเป็น static

ข้อที่ 1 ให้นักศึกษาสร้างคลาส **Employee**, **Programmer**, **SeniorProgrammer**, **Seller**, **Wallet**, และ **Food** จากคลาสไดอะแกรมดังต่อไปนี้



หมายเหตุ (1) 例外ทริบิวท์ที่กำหนดให้เป็น final จะไม่มีเมธอด setter

(2) 例外ทริบิวท์ที่กำหนดให้เป็นเมธอดของคลาส (static) เมธอดประเภท setter และ getter ต้องเป็นของคลาสด้วย

1.1. ให้นักศึกษาสร้างคลาส Food โดยกำหนดให้แต่ละเมธอดมีรายละเอียดดังต่อไปนี้

- สร้าง例外ทริบิวท์ตามที่กำหนดในคลาสไดอะแกรม
- สร้างเมธอดประเภท setter และ getter ของทุก例外ทริบิวท์ตามที่กำหนดในคลาสไดอะแกรม นอกจากนี้ เมธอด setPrice(double price) จะนำค่าพารามิเตอร์ price ไปกำหนดให้例外ทริบิวท์ price ก็ ต่อเมื่อพารามิเตอร์ price ต้องมีค่ามากกว่า例外ทริบิวท์ price ของเดิม ถ้าไม่ใช่ให้แสดงข้อความดังต่อไปนี้

Cannot update the food price.

1.2. ให้นักศึกษาสร้างคลาส Wallet โดยกำหนดให้แต่ละเมธอดมีรายละเอียดดังต่อไปนี้

- สร้าง例外ทริบิวท์ตามที่กำหนดในคลาสไดอะแกรม
- สร้างเมธอดประเภท setter และ getter ของทุก例外ทริบิวท์ตามที่กำหนดในคลาสไดอะแกรม

1.3. ให้นักศึกษาสร้างคลาส Employee โดยกำหนดให้แต่ละเมธอดมีรายละเอียดดังต่อไปนี้

- สร้าง例外ทริบิวท์ตามที่กำหนดในคลาสไดอะแกรม

- สร้างเมธอดประเภท setter และ getter ของทุกแอ็พทริบิวท์ตามที่กำหนดในคลาสไดอะแกรม
- สร้างเมธอด eat (Food f) ที่จะนำค่าแอ็พทริบิวท์ energy ของพารามิเตอร์ f ไปเพิ่มให้แอ็พทริบิวท์ energy ของตัวเอง
- สร้างเมธอด buyFood (Seller s) จะทำการซื้ออาหารผ่านพารามิเตอร์ s ด้วยเมธอด sell (...)

ถ้าการซื้อ สำเร็จ เมธอด buyFood (Seller s) จะดำเนินการดังต่อไปนี้

- พารามิเตอร์ s จะคืนค่าเป็นอปเจ็กที่สร้างมาจากคลาส Food ให้
 - ตนเองจะทานอปเจ็ก Food ผ่านเมธอด eat (...)
 - คืนค่าเป็น true

ถ้าการซื้อ ไม่สำเร็จ เมธอด buyFood (Seller s) จะดำเนินการดังต่อไปนี้

- พารามิเตอร์ s จะคืนค่า null
 - คืนค่าเป็น false

- ให้ overloaded เมธอด equals (Employee e) ซึ่งจะคืนค่าเป็น true ก็ต่อเมื่อแอ็พทริบิวท์ name ของพารามิเตอร์ e กับตัวเองมีค่าเหมือนกัน แต่ถ้าไม่ใช่จะคืนค่าเป็น false
- ให้ overridden เมธอด toString () ซึ่งจะคืนเป็นข้อความดังต่อไปนี้

My name is [แอ็พทริบิวท์ name]. \nI have [แอ็พทริบิวท์ energy] energy left.\nI have a balance of [แอ็พทริบิวท์ balance] ใน แอ็พทริบิวท์ wallet] baht.

1.4. ให้นักศึกษาสร้างคลาส Seller โดยกำหนดให้แต่ละเมธอดมีรายละเอียดดังต่อไปนี้

- สร้างเมธอด sell (Employee e) จะคืนค่าเป็นอปเจ็กจากคลาส Food ก็ต่อเมื่อผู้ซื้อ e มีเงินเพียงพอที่จะซื้ออาหาร โดยที่ ราคาอาหารจะอยู่ในแอ็พทริบิวท์ price ของคลาส Food

ถ้าผู้ซื้อมีเงินเพียงพอ กับราคาอาหารจะดำเนินการดังต่อไปนี้

- เพิ่มเงินของ Seller ขึ้นเท่ากับราคา Food
 - ลดเงินของ Employee ลงเท่ากับราคา Food
 - คืนค่าเป็นอปเจ็กใหม่ที่สร้างมาจากคลาส Food

ถ้าผู้ซื้อมีเงินไม่เพียงพอ กับราคาอาหารจะดำเนินการดังต่อไปนี้

- แสดงข้อความ "Your money is not enough."
 - คืนค่าเป็น null

1.5. ให้นักศึกษาสร้างคลาส Programmer โดยกำหนดให้แต่ละเมธอดมีรายละเอียดดังต่อไปนี้

- สร้างแอ็พทริบิวท์ตามที่กำหนดในคลาสไดอะแกรม
- สร้างเมธอดประเภท setter และ getter ของทุกแอ็พทริบิวท์ตามที่กำหนดในคลาสไดอะแกรม
- สร้างเมธอด coding (String str) และ coding (char str) โดยมีรายละเอียดดังต่อไปนี้

ถ้ามีค่าแอ็พทริบิวท์ energy มากกว่าหรือเท่ากับ 30 จะดำเนินการดังต่อไปนี้

- แสดงข้อความ "Your code is [ค่าพารามิเตอร์ str]"
 - ลดค่าแอ็พทริบิวท์ energy ลง 30

- ลดค่าแออททริบิวท์ happiness ลง 30

ถ้ามีค่าแออททริบิวท์ energy น้อยกว่า 30 จะดำเนินการดังต่อไปนี้

- แสดงข้อความ "Error Error Error"
- ลดค่าแออททริบิวท์ energy ลง 30
- ลดค่าแออททริบิวท์ happiness ลง 30

1.6. ในนักศึกษาสร้างคลาส SeniorProgrammer โดยกำหนดให้แต่ละเมธอดมีรายละเอียดดังต่อไปนี้

- ให้ overridden เมธอด coding (String str) โดยมีรายละเอียดดังต่อไปนี้

ถ้ามีค่าแออททริบิวท์ energy มากกว่าหรือเท่ากับ 10 จะดำเนินการดังต่อไปนี้

- แสดงข้อความ "I'm coding about [ค่าพารามิเตอร์ str]"
- ลดค่าแออททริบิวท์ energy ลง 5
- ลดค่าแออททริบิวท์ happiness ลง 5

ถ้ามีค่าแออททริบิวท์ energy น้อยกว่า 10 จะดำเนินการดังต่อไปนี้

- แสดงข้อความ "ZzZzZz"
- ลดค่าแออททริบิวท์ energy ลง 5
- ลดค่าแออททริบิวท์ happiness ลง 5

- ให้ overloaded เมธอด coding (String str, int num) ที่ทำงานด้านล่างนี้ช้าจำนวน num รอบ โดยมีรายละเอียดดังต่อไปนี้

ถ้ามีค่าแออททริบิวท์ energy มากกว่าหรือเท่ากับ 10 จะดำเนินการดังต่อไปนี้

- แสดงข้อความ "I'm coding about [ค่าพารามิเตอร์ str]"
- ลดค่าแออททริบิวท์ energy ลง 5
- ลดค่าแออททริบิวท์ happiness ลง 5

ถ้ามีค่าแออททริบิวท์ energy น้อยกว่า 10 จะดำเนินการดังต่อไปนี้

- แสดงข้อความ "ZzZzZz"
- ลดค่าแออททริบิวท์ energy ลง 5
- ลดค่าแออททริบิวท์ happiness ลง 5

- การเรียกใช้เมธอด coding (char str) จากอปเจกท์สร้างมาจากคลาส SeniorProgrammer จะลดค่าแออททริบิวท์ energy และค่าแออททริบิวท์ happiness ลง 5 ดังแสดงในกรณีที่ 3 (ด้านล่าง) โดยมีข้อกำหนดว่า ห้ามนักศึกษา overridden เมธอด coding (char str) จากคลาส Programmer เด็ดขาด

- สร้างเมธอด compliment (Programmer p) ซึ่งจะเพิ่มค่าแออททริบิวท์ happiness ขึ้น 20 แต้มให้กับพารามิเตอร์ p และแสดงข้อความดังต่อไปนี้

[แออททริบิวท์ name ของพารามิเตอร์ p] in a good mood

- สร้างเมธอด blame (Programmer p) ซึ่งจะลดค่าแออททริบิวท์ happiness ลง 20 แต้มให้กับพารามิเตอร์ p และแสดงข้อความดังต่อไปนี้

[แออททริบิวท์ name ของพารามิเตอร์ p] in a bad mood

กำหนดโค๊ดสำหรับทดสอบความถูกต้องของคลาสข้างต้นที่นักศึกษาได้พัฒนาขึ้น

กรณีที่ 1

```
public class Main {  
    public static void main(String[] args) {  
        Wallet w1 = new Wallet();  
        Wallet w2 = new Wallet();  
        w1.setBalance(200);  
        w2.setBalance(100);  
  
        Seller s1 = new Seller();  
        s1.setName("Peter");  
        s1.setEnergy(100);  
        s1.setWallet(w1);  
  
        Employee e1 = new Employee();  
        e1.setName("Adam");  
        e1.setEnergy(100);  
        e1.setWallet(w2);  
  
        System.out.println(s1);  
        System.out.println(e1);  
        e1.buyFood(s1);  
        System.out.println(s1);  
        System.out.println(e1);  
    }  
}
```

ตัวอย่างผลลัพธ์

```
My name is Peter.  
I have 100 energy left.  
I have a balance of 200.0 baht.  
My name is Adam.  
I have 100 energy left.  
I have a balance of 100.0 baht.  
My name is Peter.  
I have 100 energy left.  
I have a balance of 250.0 baht.  
My name is Adam.  
I have 110 energy left.  
I have a balance of 50.0 baht.
```

กรณีที่ 2

```
public class Main {
    public static void main(String[] args) {
        Wallet w1 = new Wallet();
        Wallet w2 = new Wallet();
        w1.setBalance(200);
        w2.setBalance(100);

        Programmer p1 = new Programmer();
        p1.setName("Sara");
        p1.setEnergy(100);
        p1.setWallet(w1);
        p1.setHappiness(100);

        SeniorProgrammer sp1 = new SeniorProgrammer();
        sp1.setName("Kim");
        sp1.setEnergy(100);
        sp1.setWallet(w2);
        sp1.setHappiness(100);

        System.out.println(p1 + "\nHappiness : " +p1.getHappiness());
        System.out.println(sp1 + "\nHappiness : " +sp1.getHappiness());
        sp1.compliment(p1);
        sp1.compliment(p1);
        System.out.println(p1 + "\nHappiness : " +p1.getHappiness());
        System.out.println(sp1 + "\nHappiness : " +sp1.getHappiness());
        sp1.blame(p1);
        System.out.println(p1 + "\nHappiness : " +p1.getHappiness());
        System.out.println(sp1 + "\nHappiness : " +sp1.getHappiness());
    }
}
```

ตัวอย่างผลลัพธ์

```
My name is Sara.
I have 100 energy left.
I have a balance of 200.0 baht.
Happiness : 100
My name is Kim.
I have 100 energy left.
I have a balance of 100.0 baht.
Happiness : 100
Sara in a good mood
Sara in a good mood
My name is Sara.
I have 100 energy left.
I have a balance of 200.0 baht.
Happiness : 140
My name is Kim.
I have 100 energy left.
I have a balance of 100.0 baht.
Happiness : 100
Sara in a bad mood
My name is Sara.
I have 100 energy left.
I have a balance of 200.0 baht.
Happiness : 120
My name is Kim.
I have 100 energy left.
I have a balance of 100.0 baht.
Happiness : 100
```

กรณีที่ 3

```
public class Main {
    public static void main(String[] args) {
        Wallet w1 = new Wallet();
        Wallet w2 = new Wallet();
        w1.setBalance(200);
        w2.setBalance(100);

        Programmer p1 = new Programmer();
        p1.setName("Boy");
        p1.setEnergy(100);
        p1.setWallet(w1);
        p1.setHappiness(100);

        SeniorProgrammer sp1 = new SeniorProgrammer();
        sp1.setName("Ploy");
        sp1.setEnergy(100);
        sp1.setWallet(w2);
        sp1.setHappiness(100);

        System.out.println(sp1 + "\nHappiness : " +sp1.getHappiness());
        sp1.coding('A');
        sp1.coding("Bugggggg");
        sp1.coding("Bugggggg",2);
        System.out.println(sp1 + "\nHappiness : " +sp1.getHappiness());
        System.out.println("-----");
        System.out.println(p1 + "\nHappiness : " +p1.getHappiness());
        p1.coding('B');
        p1.coding("Deathhhhhhhh");
        System.out.println(p1 + "\nHappiness : " +p1.getHappiness());
    }
}
```

ตัวอย่างผลลัพธ์

```
My name is Ploy.
I have 100 energy left.
I have a balance of 100.0 baht.
Happiness : 100
I'm coding about A
I'm coding about Bugggggg
I'm coding about Bugggggg
I'm coding about Bugggggg
My name is Ploy.
I have 80 energy left.
I have a balance of 100.0 baht.
Happiness : 80
-----
My name is Boy.
I have 100 energy left.
I have a balance of 200.0 baht.
Happiness : 100
Your code is B
Your code is Deathhhhhhhh
My name is Boy.
I have 40 energy left.
I have a balance of 200.0 baht.
Happiness : 40
```