



รายงาน
การออกแบบโปรแกรมสำหรับร้านชาวม

โดย
นายพงศธร ชมภูษ รหัศ 634245022

เสนอ
อาจารย์ นันทิยา หลิมศิริรัตน์

รายงานนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาบัณฑิต
สาขาวิทยาการคอมพิวเตอร์
ภาคเรียนที่ 1 ปีการศึกษา 2564
มหาวิทยาลัยราชภัฏนครปฐม

ที่มา

คลาสที่สร้างเป็นการนำแนวคิดของการขายขนมร้านขนมมาสร้างโปรแกรมคอมพิวเตอร์สำหรับการสั่งสินค้า การคิดราคา การแสดงรายละเอียดของคำสั่งซื้อ และการเช็คสินค้าที่เหลือในคลังสินค้า โดยสามารถสร้างร้านค้าจาก คลาสได้ หลายแบบตาม overloading class โดยร้านค้าจะมีชื่อร้าน และ ยอดเงินคงเหลือและไม่สามารถเปลี่ยนแปลง ยอดเงินได้จากภายนอก การเรียกใช้งานเป็นการจำลองสถานการณ์ โดยจะมีลูกค้ามาสั่งสินค้า และไม่ต้องทำอะไรกับการตั้งนั้นเพียงป้อนข้อมูลที่ต้องการซื้อและ ร้านค้าจะทำการสร้างและคิดราคา รวมทั้งจำนวนของขนมแล้วแก้วของร้านค้าอัตโนมัติ

Class MilkTeaStore

```
/*
เป็นคลาสสำหรับ "ร้านชานมไข่มุก" สามารถคิดราคาและสร้าง order โดยร้านจะต้องตั้งชื่อร้านก่อนเปิดร้าน และมี
- เมนู "Coco", "Milk", "Matcha", "Chocolate", "Chocolate", "Caramel"
- ชื่อสาขา (branchName)
- ยอดเงินตั้งต้น (balance) เป็น 0
- จำนวนแก้ว (glass)
- ผงชานม (powder)
*/
public class MilkTeaStore {
    private String branchName; // ชื่อสาขา
    private int balance; // ยอดเงิน
    private int glass; // จำนวนแก้ว
    private double powder; // จำนวนผงชานม
}
```

Class MilkTeaStore คือการทำงานภายในร้าน โดยจะมี properties ทั้งหมด 4 ตัว คือ

- เมนู "Coco", "Milk", "Matcha", "Chocolate", "Chocolate", "Caramel"
- ชื่อสาขา (branchName)
- ยอดเงินตั้งต้น (balance) เป็น 0
- จำนวนแก้ว (glass)
- ผงชานม (powder)

Overloading class

สำหรับการสร้าง ร้านค้าในหลากหลายรูปแบบ

```
// overloading class 1
// ตั้งชื่อร้าน โดยจำนวนแก้วคือ 20 และ ผงชาวมคือ 100.0
// โดยจะเรียก overloading 4
public MilkTeaStore(String branchName) {
    this(branchName, glass: 20, powder: 100.0);
}

// overloading class 2
// ตั้งชื่อร้านและจำนวนแก้ว โดยจำนวนแก้วตาม user และ ผงชาวมคือ 100.0
// โดยจะเรียก overloading 4
public MilkTeaStore(String branchName, int glass) {
    this(branchName, glass, powder: 100.0);
}

// overloading class 3
// ตั้งชื่อร้านและจำนวนผงชาวม โดยจำนวนผงชาวมตาม user และ จำนวนแก้วคือ 20
// โดยจะเรียก overloading ต่อไป
public MilkTeaStore(String branchName, double powder) {
    this(branchName, glass: 20, powder);
}

// overloading class 4
// ตั้งชื่อร้านและจำนวนผงชาวม โดยจำนวนผงชาวมและจำนวนแก้วตาม user
public MilkTeaStore(String branchName, int glass, double powder) {
    this.branchName = branchName;
    this.glass = glass;
    this.powder = powder;
    this.balance = 0;

    //แสดงข้อความเมื่อสร้างร้านค้าเสร็จ
    this.showStock();
}
```

Getter & Setter method

โดยสามารถเรียกใช้จากข้างนอก class ได้

```
// getter and setter
// สามารถเรียกใช้จากภายนอกได้
//get ชื่อร้าน
public String getBranchName() {
    return branchName;
}
//set ชื่อร้าน
public void setBranchName(String branchName) {
    this.branchName = branchName;
}
//get จำนวนแก้ว
public int getGlass() {
    return glass;
}
//set จำนวนแก้ว
public void setGlass(int glass) {
    this.glass = glass;
}
//get จำนวนผงชาวม
public double getPowder() {
    return powder;
}
//set จำนวนผงชาวม
public void setPowder(double powder) {
    this.powder = powder;
}
//get ยอดเงินของร้าน โดยสามารถ get ได้อย่างเดียว
public int getBalance() {
    return balance;
}
```

Function สำหรับแสดงรายละเอียดของร้านค้าและสินค้าคงเหลือในคลัง

```
//แสดงข้อความจำนวนคงเหลือของสินค้าในร้านค้าและยอดเงิน
public void showStock(){
    System.out.println("ชื่อร้าน: " +branchName);
    System.out.println("จำนวนแก้ว: " +glass);
    System.out.println("จำนวนผง: " +powder);
    System.out.println("ยอดเงิน: " +balance);
    System.out.println("-----");
}
```

Overloading method สำหรับการเรียกใช้งานการสั่งซื้อ โดยจะมี ทั้งหมด 4 method ที่สามารถสั่งได้

```
//overloading function order 1 เรียก overloading order 4
//เป็นการสั่ง order โดยไม่ระบุอะไรมา
public void order(String menu){
    this.order(menu, num: 1, topping: false);
}

//overloading function order 2 เรียก overloading order 4
//เป็นการสั่ง order และระบุจำนวนแก้ว
public void order(String menu, int num){
    this.order(menu, num, topping: false);
}

//overloading function order 3 เรียก overloading order 4
//เป็นการสั่ง order และระบุ topping
public void order(String menu, boolean topping){
    this.order(menu, num: 1, topping);
}

//overloading function order 4
//เป็นการสั่ง order โดยระบุ จำนวนแก้วและ topping
public void order(String menu, int num, boolean topping){
    //เช็ค ว่า สินค้าเหลือพอหรือไม่
    if(powder*num <=8 || glass < num){
        System.out.println("Sorry product is not enough");
        return;
    }
    //แสดงว่าสั่ง menu อะไรมาที่แก้ว และ ใส่ topping หรือไม่
    System.out.println("รับ order : "+ menu + " "+num+ " แก้ว ใส่ topping: "+ topping);
    int price = 0;
    //check ถ้าใส่ topping จะคิดเงิน เพิ่ม 5 บาท
    if(topping){
        price = price + 5;
    }
}
```

Function สำหรับการรับคำสั่งซื้อจากลูกค้าและเช็คว่เมนูนั้นมีหรือไม่
พร้อมทั้งทำการเช็ค สินค้าคงเหลือ ในคลังหากไม่มีจะแสดงข้อความให้ ลูกค้า หากมีจะทำการ
คิดราคาและแสดงข้อมูลของ คำสั่งซื้อนั้น และมีการเช็คว่ลูกค้าใส่ topping เพิ่มหรือไม่ หาก
เพิ่มจะทำการคิดเงินเพิ่ม 5 บาท

```
public void order(String menu, int num, boolean topping){  
    //เช็คว่ สินค้าเหลือพอหรือไม่  
    if(powder*num <=8 || glass < num){  
        System.out.println("Sorry product is not enough");  
        return;  
    }  
    //แสดงว่สั่ง menu อะไรมากี่แก้ว และ ใส่ topping หรือไม่  
    System.out.println("รับ order : "+ menu + " "+num+ " แก้ว ใส่ topping: "+ topping);  
    int price = 0;  
    //check ถ้าใส่ topping จะคิดเงิน เพิ่ม 5 บาท  
    if(topping){  
        price = price + 5;  
    }  
    //เช็คว่มีเมนูในร้านหรือไม่หากมีแล้วจะทำการบวกราคาเพิ่มและ ทำการสร้าง  
    if(menu.equals("Coco")){  
        price = price + 35;  
        this.make( powder: 12.5, num);  
    }else if(menu.equals("Milk")){  
        price = price + 25;  
        this.make( powder: 8, num);  
    }else if(menu.equals("Matcha")){  
        price = price + 45;  
        this.make( powder: 15, num);  
    }else if(menu.equals("Chocolate")){  
        price = price + 30;  
        this.make( powder: 11, num);  
    }else if(menu.equals("Caramel")){  
        price = price + 40;  
        this.make( powder: 13.5, num);  
    }else{ // หากไม่มี menu นั้น จะทำการแสดงข้อความ และ หยุดการทำงาน
```

ทำการเช็คว่ามีเมนูที่ลูกค้าสั่งมีหรือไม่ หากไม่มี จะทำการแจ้งลูกค้า
และทำการเรียกการคิดเงินและเก็บเงินจากลูกค้า รวมทั้งการใช้สินค้าในคลัง
พร้อมทั้งแสดงรายละเอียดของ order จากการเรียก method เดียว

```
// เช็คว่ามีเมนูในร้านหรือไม่หากมีแล้วจะทำการบวกราคาเพิ่มและ ทำการสร้าง
if(menu.equals("Coco")){
    price = price + 35;
    this.make( powder: 12.5, num);
}else if(menu.equals("Milk")){
    price = price + 25;
    this.make( powder: 8, num);
}else if(menu.equals("Matcha")){
    price = price + 45;
    this.make( powder: 15, num);
}else if(menu.equals("Chocolate")){
    price = price + 30;
    this.make( powder: 11, num);
}else if(menu.equals("Caramel")){
    price = price + 40;
    this.make( powder: 13.5, num);
}else{ // หากไม่มี เมนู นั้น จะทำการแสดงข้อความ และ หยุดการทำงาน
    System.out.println("Sorry, we don't have that menu "+menu);
    return;
}

//ทำการคิดเงินและเพิ่มยอด balance โดยส่งราคาเป็น parameter
this.calculate(price);

//สร้าง order จาก constructor โดยส่ง parameter 3 ตัว
Order order = new Order(menu, price: price * num, topping, num);
//แสดงข้อความเมื่อ สร้าง order เสร็จแล้ว
order.showOrder();
}
```


Function make() สำหรับการสร้าง เมนูนั้น ๆ ตามคำสั่งซื้อและทำการลดค่าสินค้าในคลัง คือลดจำนวนแก้วตาม ตัวแปร num และ จำนวนผงชานม ตามตัวแปร powder

Function calculate() สำหรับการเพิ่มยอดเงินจากราคาของคำสั่งซื้อนั้น

โดยจะเป็น private method ที่สามารถเรียกใช้ได้ภายใน class เท่านั้น เปรียบเสมือนการทำงานในร้านค้า โดยที่ลูกค้าไม่ต้องคำนึงถึง

```
//function สร้าง ชานมตามจำนวนและ ทำการ ลดค่า powder ตามจำนวนแก้ว
private void make(double powder, int num){
    this.powder = this.powder - (powder * num);
    this.glass = this.glass - num;
}
//function ในการคิดเงินและเพิ่มยอด balance
private void calculate(int price){
    this.balance = balance + price;
}
```

Class Order

ใช้สำหรับในการแสดงรายละเอียดของคำสั่งซื้อนั้นๆ

โดยมี properties ทั้งหมด 4 ตัว ดังนี้

Name ชื่อเมนู

Price ราคารวม

Topping ใส่ topping หรือไม่

Num จำนวนแก้วที่สั่ง

```
// คลาสสำหรับ แสดงข้อมูล order ที่สร้าง
public class Order {
    private String name; //ชื่อเมนู
    private int price; //ราคารวม
    private boolean topping; //ส่วนเสริม
    private int num; //จำนวนแก้วที่สั่ง

    // overloading class ที่ไม่ต้องใส่ parameter
    public Order() {
    }

    //overloading class โดยมี parameter 3 ตัว
    public Order(String name, int price, boolean topping, int num) {
        this.name = name;
        this.price = price;
        this.topping = topping;
        this.num = num;
    }
}
```

Getter & Setter

```
//getter and setter ของ ข้อมูล
public String getName() {
    return name;
}

public void setName(String name) {
    this.name = name;
}

public int getPrice() {
    return price;
}

public void setPrice(int price) {
    this.price = price;
}

public boolean isTopping() {
    return topping;
}

public void setTopping(boolean topping) {
    this.topping = topping;
}

public int getNum() {
    return num;
}

public void setNum(int num) {
    this.num = num;
}
```

Method สำหรับไว้แสดงรายละเอียดของคำสั่งซื้อนั้น

```
//แสดงรายละเอียดของ order
public void showOrder(){
    System.out.println("เมนู: "+name);
    System.out.println("ราคา: "+price);
    System.out.println("จำนวน: "+num);
    System.out.println("ใส่ topping: "+topping);
    System.out.println("-----");
}
```

Class MainStore

ทำการสร้างร้านค้าจาก overloading class โดยมีการกำหนดชื่อและ properties ต่างๆ ตามต้องการ

```
public class MainStore {  
    //method main สำหรับเรียกใช้งาน  
    public static void main(String[] args) {  
        //สร้างร้านชาวมโดยใส่แค่ ชื่อร้าน  
        MilkTeaStore milkTeaStore1 = new MilkTeaStore( branchName: "ร้านนมสดใส");  
        //สร้างร้านชาวมโดยใส่แค่ ชื่อร้าน และจำนวนแก้ว  
        MilkTeaStore milkTeaStore2 = new MilkTeaStore( branchName: "ร้านนมลิล่า", glass: 50);  
        //สร้างร้านชาวมโดยใส่แค่ ชื่อร้าน และจำนวนผงชาวม  
        MilkTeaStore milkTeaStore3 = new MilkTeaStore( branchName: "ร้านนมวิลล่า", powder: 200.0);  
        //สร้างร้านชาวมโดยใส่แค่ ชื่อร้าน จำนวนแก้วและจำนวนผงชาวม  
        MilkTeaStore milkTeaStore4 = new MilkTeaStore( branchName: "ร้านนมวอนวอน", glass: 30, powder: 200.0);  
    }  
}
```

Output: โดย หลังสร้าง Class จะแสดงรายละเอียดตามรูปภาพ

```
ชื่อร้าน: ร้านนมสดใส  
จำนวนแก้ว: 20  
จำนวนผง: 100.0  
ยอดเงิน: 0  
-----
```

```
ชื่อร้าน: ร้านนมลิล่า  
จำนวนแก้ว: 50  
จำนวนผง: 100.0  
ยอดเงิน: 0  
-----
```

```
ชื่อร้าน: ร้านนมวิลล่า  
จำนวนแก้ว: 20  
จำนวนผง: 200.0  
ยอดเงิน: 0  
-----
```

```
ชื่อร้าน: ร้านนมวอนวอน  
จำนวนแก้ว: 30  
จำนวนผง: 200.0  
ยอดเงิน: 0  
-----
```

เรียกใช้งาน Order และสั่ง Menu “Coco” และแสดงยอดคงเหลือสินค้าในคลัง

```
//สั่ง order Coco
milkTeaStore2.order( menu: "Coco", num: 1);
//ทำการแสดงคลังสินค้าโดยจะเห็นจำนวนแก้วและ ผงชานมลดลง
milkTeaStore2.showStock();
```

Output:

```
รับ order : Coco 1 แก้ว ใส่ topping: false
เมนู: Coco
ราคา: 35
จำนวน: 1
ใส่ topping: false
-----
ชื่อร้าน: ร้านนมลีลา
จำนวนแก้ว: 49
จำนวนผง: 87.5
ยอดเงิน: 35
-----
```

ทำการเรียกใช้งาน overloading method

```
//สั่ง order Milk
milkTeaStore1.order( menu: "Milk"); // จำนวนแก้วจะเท่ากับ 1 และ ไม่ใส่ topping
milkTeaStore1.order( menu: "Milk", num: 1); //จำนวนแก้วจะเท่ากับ 1 และ ไม่ใส่ topping
milkTeaStore1.order( menu: "Milk", num: 1, topping: true); //จะเห็นราคาเพิ่มขึ้นอีก 5 บาท
milkTeaStore1.order( menu: "Milk", topping: true); //จะเห็นราคาเพิ่มขึ้นอีก 5 บาท
milkTeaStore1.order( menu: "Milk", num: 2); //จะเห็นราคาเพิ่มขึ้นสองเท่าเพราะ สั่งสองแก้ว

//ทำการแสดงคลังสินค้า โดยจะเห็นจำนวนแก้วและ ผงชานมลดลง
milkTeaStore1.showStock();
```

Output:

```
รับ order : Milk 1 แก้ว ใส่ topping: false
เมนู: Milk
ราคา: 25
จำนวน: 1
ใส่ topping: false
-----
รับ order : Milk 1 แก้ว ใส่ topping: false
เมนู: Milk
ราคา: 25
จำนวน: 1
ใส่ topping: false
-----
รับ order : Milk 1 แก้ว ใส่ topping: true
เมนู: Milk
ราคา: 30
จำนวน: 1
ใส่ topping: true
-----
รับ order : Milk 1 แก้ว ใส่ topping: true
เมนู: Milk
ราคา: 30
จำนวน: 1
ใส่ topping: true
-----
รับ order : Milk 2 แก้ว ใส่ topping: false
เมนู: Milk
ราคา: 50
จำนวน: 2
ใส่ topping: false
```

แสดงยอดคงเหลือ

```
-----
ชื่อร้าน: ร้านนมสดใส
จำนวนแก้ว: 14
จำนวนผง: 52.0
ยอดเงิน: 135
-----
```

ทดสอบสั่งซื้อเมนูที่ไม่มี

```
//สั่ง order ที่ไม่มีในร้านค้า
milkTeaStore4.order( menu: "ThaiMilk", num: 1);
//ทำการแสดงคลังสินค้า โดยจะเห็นจำนวนแก้วและ ผงชานมเท่าเดิม
milkTeaStore4.showStock();
```

Output:

```
-----
รับ order : ThaiMilk 1 แก้ว ใส่ topping: false
Sorry, we don't have that menu ThaiMilk
ชื่อร้าน: ร้านนมวอนวอน
จำนวนแก้ว: 30
จำนวนผง: 200.0
ยอดเงิน: 0
-----
```

ทดสอบเรียก Getter & Setter ของ class

```
//เรียกใช้ get and set ใน class
milkTeaStore1.setGlass(0);
System.out.println("จำนวนแก้วคือ: "+milkTeaStore1.getGlass());
```

Output:

```
จำนวนแก้วคือ: 0
```


ทดสอบเรียกสินค้าในคลังไม่พอ โดนการ set จำนวนแก้วให้เป็น 0

```
//สินค้าไม่พอใน stock  
MilkTeaStore milkTeaStore5 = new MilkTeaStore( branchName: "ร้านนมแมว", glass: 0, powder: 200.0);  
//ทดลอง สั่ง order แสดงข้อความ "Sorry product is not enough"  
milkTeaStore5.order( menu: "Milk");// จำนวนแก้วจะเท่ากับ 1 และ ไม่ใส่ topping
```

Output:

```
Sorry product is not enough
```

```
Process finished with exit code 0
```