

Object-Oriented Programming Concept

Maharakham University

อาจารย์พระ พฤกษ์ศรี
ภาควิชาวิทยาการคอมพิวเตอร์ คณะวิทยาการสารสนเทศ
Email: potchara.p@msu.ac.th

1

Object-Oriented Programming (OOP)

- การพัฒนาโปรแกรมประยุกต์เชิงวัตถุ หรือ Object-Oriented Programming (OOP)
- คำนิยามสำคัญ
 - Class
 - Object
 - Attributes
 - Method

Maharakham University

2

คลาส (Class)

- คลาส เปรียบเสมือนแบบพิมพ์ของวัตถุต่างๆ (วัตถุต่างๆ เรียกว่า วัตถุ) ซึ่งวัตถุแต่ละตัวนั้น จะต้องถูกสร้างมาจากคลาสเสมอ
- ตัวอย่าง

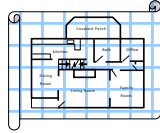
เอาแบบไปสร้างบ้าน

Maharakham University

3

คลาส (Class)

- แบบพิมพ์ของบ้าน = Class



ในการพัฒนาโปรแกรมเชิงวัตถุ

- Class เป็นต้นแบบที่จะนำไปสร้างวัตถุ (Object) ต่างๆ ในโปรแกรม
- Class ไม่สามารถถูกใช้งานได้โดยตรง จะต้องนำไปสร้างเป็น Object ตามแบบก่อนจึงจะใช้งานได้

Maharakham University



4

ออปเจ็ค (Object)

- ออปเจ็ค (Object) หรือเรียกว่า Instance
- Object คือสิ่งที่ถูกสร้างขึ้นมาจากต้นแบบ



- Object สามารถถูกใช้งานได้โดยตรง
- Object จะมีคุณสมบัติและความสามารถตามที่กำหนดไว้ในต้นแบบหรือกำหนดไว้ใน Class นั้นเอง

Maharakham University



5

คลาส (Class) vs ออปเจ็ค (Object)

สรุป

- Class คือ ต้นแบบของวัตถุสิ่งใดสิ่งหนึ่ง
- Object คือ วัตถุหรือสิ่งที่สร้างมาจากต้นแบบ
- Class ถูกเรียกใช้งานโดยตรง **ไม่ได้**
- Object ถูกเรียกใช้งานโดยตรง **ได้**

Maharakham University



6

ตัวอย่าง

Class = แบบแปลนยานบิน

Object =

Class = แบบแปลนยานยนต์

Object =

Class = สัตว์

Object =

Mahasarakham
University

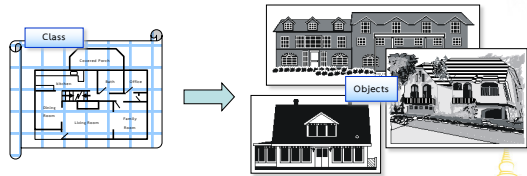
Heart of
the Northeast



7

คลาส (Class) vs ออปเจ็ค (Object)

คลาส สามารถนำไปเป็นต้นแบบในการสร้างออปเจ็คได้
หลากหลายรูปแบบ แต่ละออปเจ็คอาจเหมือนหรือไม่เหมือนกันก็ได้



Mahasarakham
University

Heart of
the Northeast



8

สมาชิกของคลาส (Class Members)

- ออปเจ็ค แม้จะสร้างมาจากคลาสเดียวกัน แต่ละออปเจ็คก็มีความแตกต่างกันได้ สิ่งที่ทำให้ออปเจ็คแตกต่างกัน คือ **คุณลักษณะ (Attributes บางครั้งเรียกว่า Property) และ เมธอด (Method)**

ตัวอย่าง

– Class : คน

– Attributes : อาชีพ



Mahasarakham
University

Heart of
the Northeast



9

Heart of the Northeast

คุณลักษณะ (Attributes)

- คุณลักษณะ คือ ข้อมูลหรือรายละเอียดของออปเจ็ค
- แบ่งออกได้เป็น 2 ประเภทคือ
 - ตัวแปร (variable) เปลี่ยนแปลงค่าได้
 - ค่าคงที่ (constant) เปลี่ยนแปลงค่าไม่ได้
- ตัวอย่าง ออปเจ็ค นักศึกษา
- คุณลักษณะที่เปลี่ยนค่าได้ (variable) เช่น คะแนน, ชั้นปี
- คุณลักษณะที่เปลี่ยนค่าไม่ได้ (constant) เช่น รหัสประจำตัว

Maharakham University

www.mru.ac.th

10

Heart of the Northeast

คุณลักษณะ (Attributes)

Maharakham University

www.mru.ac.th

11

Heart of the Northeast

คุณลักษณะ (Attributes)

Maharakham University

www.mru.ac.th

12

เมธอด (Method)

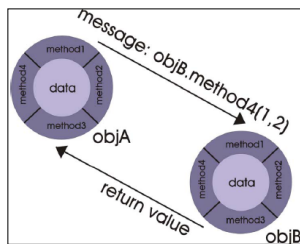
- เมธอด คือ พฤติกรรมของออปเจ็ค เรียกอีกอย่างว่า behavior
- เป็นตัวกำหนดว่า ออปเจ็คมีความสามารถอะไรบ้าง
- เป็นตัวที่ใช้ในการสื่อสารกันระหว่างออปเจ็ค
- การกระทำใดๆก็ตาม จะต้องสั่งการผ่าน เมธอด เสมอ



15

เมธอด (Method)

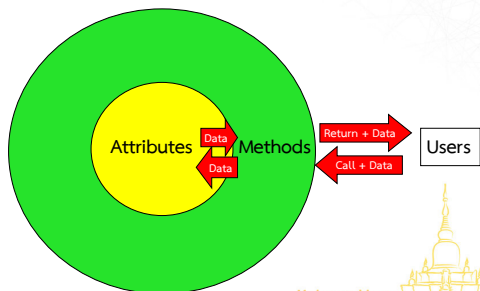
- การสื่อสารกันระหว่างออปเจ็ค



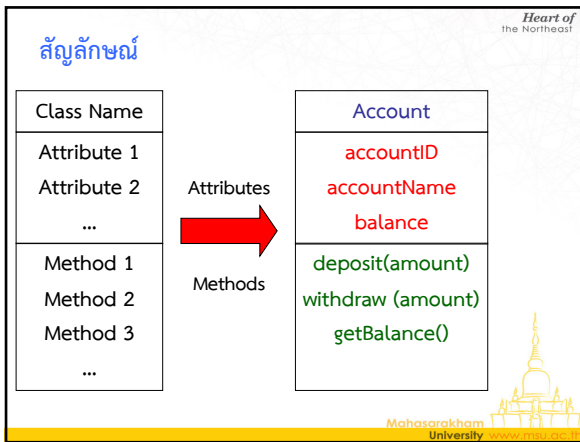
16

ภาพรวมส่วนประกอบของออปเจ็ค

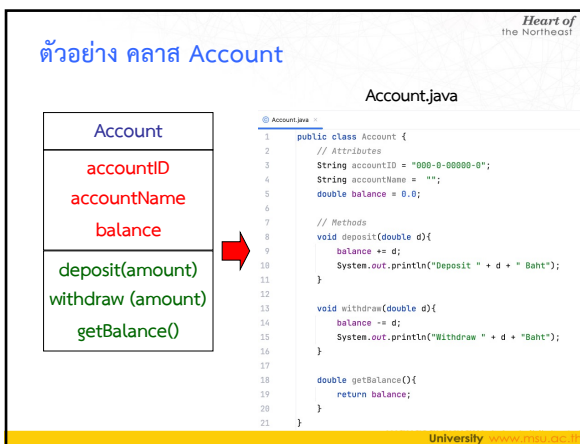
- การสื่อสารกันระหว่างออปเจ็ค



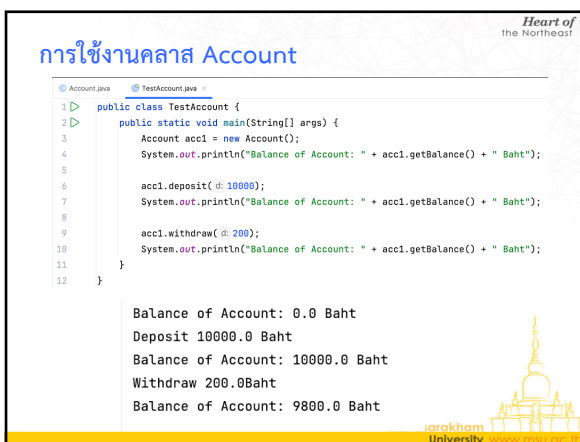
17



18



19



21

คอนสตรัคเตอร์ (Constructor)

- คือ เมธอดที่ทำงานทันทีที่มีการสร้างออปเจ็ค
- คอนสตรัคเตอร์มีลักษณะสำคัญคือ
 - มีชื่อเหมือนกับ ชื่อคลาส
 - ในแต่ละคลาสสามารถมีคอนสตรัคเตอร์ได้หลายตัว โดยพิจารณาความแตกต่างโดยดูที่ พารามิเตอร์
 - คอนสตรัคเตอร์ ไม่มี Return Type



22

คอนสตรัคเตอร์ (Constructor)

```
public class AccountCon {
    // Attributes
    String accountID;
    String accountName;
    double balance;
    // Constructor 1
    AccountCon() {
        accountID = "000-0-00000-0";
        accountName = "";
        balance = 0.0;
    }
    // Constructor 2
    AccountCon(String id, String name, double money){
        accountID = id;
        accountName = name;
        balance = money;
    }
    // Methods
    void deposit(double d){
        balance += d;
        System.out.println("Deposit " + d + " Baht");
    }
    void withdraw(double d){
        balance -= d;
        System.out.println("Withdraw " + d + " Baht");
    }
    double getBalance() {return balance;}
}
```

} Constructor ตัวที่ 1

} Constructor ตัวที่ 2



23

คอนสตรัคเตอร์ (Constructor)

```
1> public class TestAccountCon {
2>     public static void main(String[] args) {
3>         AccountCon acc1 = new AccountCon();
4>         System.out.println("Account ID: " + acc1.accountID);
5>         System.out.println("Account Name: " + acc1.accountName);
6>         System.out.println("Account Balance: " + acc1.getBalance() + " Baht");
7>
8>         AccountCon acc2 = new AccountCon("111-1-11111-1", "Peter", 30000);
9>         System.out.println("Account ID: " + acc2.accountID);
10>        System.out.println("Account Name: " + acc2.accountName);
11>        System.out.println("Account Balance: " + acc2.getBalance() + " Baht");
12>    }
13> }
```

ทำงานที่ Constructor ตัวที่ 1

ทำงานที่ Constructor ตัวที่ 2

Account ID: 000-0-00000-0
Account Name:
Account Balance: 0.0 Baht
Account ID: 111-1-11111-1
Account Name: Peter
Account Balance: 30000.0 Baht



24

Heart of the Northeast

คอนสตรัคเตอร์ (Constructor)

```

1 public class AccountCon {
2     // Attributes
3     String accountID;
4     String accountName;
5     double balance;
6     // Constructor 1
7     AccountCon() {
8         accountID = "000-0-00000-0";
9         accountName = "";
10        balance = 0.0;
11    }
12    //...
13    void deposit(double d){
14        balance += d;
15        System.out.println("Deposit " + d + " Baht");
16    }
17    void withdraw(double d){
18        balance -= d;
19        System.out.println("Withdraw " + d +
20        " Baht");
21    }
22    double getBalance() { return balance; }
23 }
24
25
26
27
28
29
30
31

```

มี Constructor ตัวเดียว

เรียก Constructor AccountCon ตัวที่ 2

```

1 public class TestAccountCon {
2     public static void main(String[] args) {
3         AccountCon acc1 = new AccountCon();
4         System.out.println("Account ID: " + acc1.accountID);
5         System.out.println("Account Name: " + acc1.accountName);
6         System.out.println("Account Balance: " + acc1.getBalance() + " Baht");
7
8         AccountCon acc2 = new AccountCon("111-1-11111-1", "Thana", 30000);
9         System.out.println("Account ID: " + acc2.accountID);
10        System.out.println("Account Name: " + acc2.accountName);
11        System.out.println("Account Balance: " + acc2.getBalance() + " Baht");
12    }
13 }

```

AccountCon acc2 = new AccountCon("111-1-11111-1", "Thana", 30000);

Account ID: 111-1-11111-1
Account Name: Thana
Account Balance: 30000

Mahasarakham University

25

Heart of the Northeast

คอนสตรัคเตอร์ (Constructor)

- ไม่สามารถคอมไพล์ได้เนื่องจากไม่พบ Constructor
AccountCon(String, String, int)

java: constructor AccountCon in class AccountCon cannot be applied to given types;
required: no arguments
found: java.lang.String,java.lang.String,int
reason: actual and formal argument lists differ in length

- ข้อสังเกต ถ้าในคลาสไม่มีคอนสตรัคเตอร์ใดๆเลย จาวาจะสร้าง Constructor ให้เองโดยอัตโนมัติ ซึ่งเป็น Constructor ที่ไม่มีพารามิเตอร์ และไม่มีการทำงานใดๆ แต่ถ้ามีคอนสตรัคเตอร์แม้เพียง 1 ตัว จาวาจะไม่สร้างให้โดยอัตโนมัติ

Mahasarakham University

26

Heart of the Northeast

ทดสอบคอนสตรัคเตอร์

```

1 public class AccountConTest {
2     String accountID;
3     String accountName;
4     double balance;
5     void AccountConTest() {
6         accountID = "000-0-00000-0";
7         accountName = "";
8         balance = 0.0;
9     }
10    AccountConTest(String id, String name, double money){
11        accountID = id;
12        accountName = name;
13        balance = money;
14    }
15 }

```

มี Constructor กี่ตัว ?

Mahasarakham University

27

Heart of the Northeast

คำสั่ง new

- **new** คือ คำสั่งในการสร้างออบเจ็ค
- มีขั้นตอนการทำงานดังนี้
 - กำหนดพื้นที่หน่วยความจำ
 - กำหนดค่าเริ่มต้นให้กับ Attribute
 - กำหนดค่าที่ของ Attribute ตามกำหนด
 - เรียกใช้งาน คอนสตรัคเตอร์

Maharakham University

www.mru.ac.th

28

Heart of the Northeast

คำสั่ง new

```

1 public class MyDate {
2     int day = 1;
3     int month = 1;
4     int year = 2000;
5     MyDate(int d, int m, int y)
6     {
7         day = d;
8         month = m;
9         year = y;
10    }
11 }
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

ขั้นตอนที่หนึ่ง

ขั้นตอนที่สอง

ขั้นตอนที่สาม

ขั้นตอนที่สี่

MyDate d1 = new MyDate(16,8,1972);

ขั้นตอนที่หนึ่ง

ขั้นตอนที่สอง

ขั้นตอนที่สาม

ขั้นตอนที่สี่

Maharakham University

www.mru.ac.th

29
