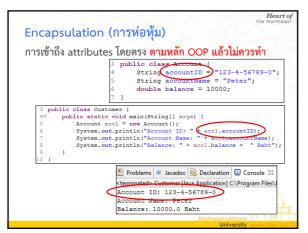
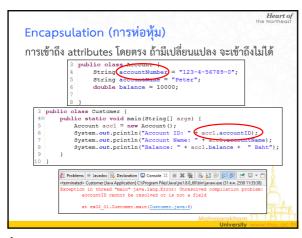


การสื่อสารระหว่าง Object • การเขียนโปรแกรมแบบ OOP นั้น เมื่อมีการสร้าง Object แล้วต้องมีการทำงานร่วมกันระหว่าง Object และต้องมีการ สื่อสารกันระหว่าง Object • message คือ คำสั่งในการ ร้องขอหรือกระทำไปยัง Object อื่น • การรับส่ง message จะทำ โดยการเรียกใช้เมธอด





Encapsulation (การห่อหุ้ม) • Encapsulation หมายถึง การห่อหุ้ม Attributes และ Methods ของ Object เข้าไว้ด้วยกัน เพื่อป้องกันการ มองเห็นหรือการควบคุมการเข้าถึงข้อมูลจาก Object อื่น การ เข้าถึง Attribute จะต้องเข้าถึงผ่าน Methods ที่กำหนด เท่านั้น Object Method 1 Encapsulation หรือการห่อหุ้มข้อมูล Method 3

5

Mutator methods • คือ การเข้าถึง Attributes ที่เป็นถูกป้องกันไว้โดยการเรียก ผ่าน Method เข้าไปเพื่อกำหนดค่า หรือเรียกคืนค่า เรียก Method นี้ว่า Mutator Method • Mutator Methods แบ่งได้เป็น 2 ประเภท คือ - Setter - Getter

Mutator methods

- Method แบบ Setter จะใช้ในการกำหนดค่าของ Attributes โดย ปกติ Method นี้จะไม่มีการส่งค่ากลับคืน ดังนั้นส่วนใหญ่จะมี return type เป็น void และมักขึ้นต้นชื่อ Method ว่า set
- Method แบบ Getter จะใช้ในการเรียกค่าของ Attributes โดยปกติ Method นี้จะมีการส่งค่ากลับคืน ดังนั้นจะมี return type ที่ไม่ใช่ void และมักขึ้นต้นชื่อ Method ว่า get
 - แต่ถ้า Getter Method ที่มี return type เป็น boolean หมายถึง Getter Method ที่ใช้ตรวจสอบค่าใดค่าหนึ่ง มักจะ ขึ้นต้นชื่อ Method ว่า is

Mahasarakham

Heart of

7

8

this

- คำว่า "this" เป็น keyword
- this ในจาวาหมายถึง Object ของตัวเอง มักใช้ this กับสิ่งที่ มีชื่อซ้ำกับ Attribute หรือ Method ของ Object
- การใช้ this อาจทำให้เกิดการสับสนในการในการเขียน
 โปรแกรมได้ เช่นการใช้เพื่อเรียก attributes ที่ชื่อเหมือนกัน
 ดังนั้น ควรตั้งชื่อให้สื่อความหมายและไม่ช้ำกันจะดีกว่า

Mahasarakham

Heart of

11

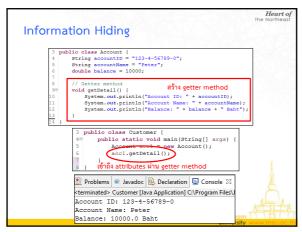
Information Hiding

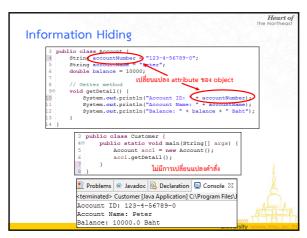
 การซ่อนข้อมูล (Information Hiding) เป็นการปกป้องข้อมูล ของ Object ใด ไม่ให้เข้าถึงได้โดยผิดวิธี และการซ่อนข้อมูล ยังเข้ามาช่วยแก้ปัญหาการเปลี่ยนแปลงข้อมูลที่ไม่สัมพันธ์กัน ได้

Heart of

หลักการสำคัญ

- กำหนดระดับการเข้าถึงให้กับ Attributes และ Methods ของ
 Object ไม่อนุญาตให้เข้าถึงได้โดยตรง
- การใช้ Mutator methods (setter และ getter) เป็นตัวควบคุมการ เข้าถึง attributes





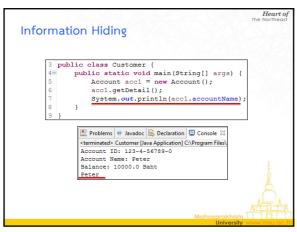
14

ข้อดีของ Encapsulation

 การซ่อนข้อมูล (Information Hiding) ทำให้ Object สามารถติดต่อกับ Object ภายนอก ผ่านทางช่องทางที่ กำหนด (method) เท่านั้น ซึ่งถ้ามีการเปลี่ยนแปลง Attributes หรือ Method ใน Object จะไม่มีผลกระทบต่อ Object ภายนอกที่มาเรียกใช้

Heart of

• ข้<mark>อสังเกต</mark> ถึงแม้ว่าจะเขียนโปรแกรมเพื่อให้เข้าถึงข้อมูลโดย ผ่านเมธอดแล้ว แต่ก็ยังสามารถเข้าถึงข้อมูลได้โดยตรงหากรู้ รายละเอียดของ attribute นั้นๆ



Access Modifier

- จากปัญหาการเข้าถึงข้อมูลที่เกิดขึ้น การเขียนโปรแกรมแบบ OOP จึงมีการออกแบบให้มีการควบคุมการเข้าถึงข้อมูลได้ใน ระดับที่แตกต่างกัน โดยมีการ keyword ที่ใช้อยู่ 3 ตัว (4 ระดับ)
 - Public
 - Protected
 - Private
- ระดับการควบคุมการเข้าถึงข้อมูล เรียกว่า Accessibility หรือ Visibility โดยมีการใช้ Access Modifier เป็นตัวควบคุม

Mahasarakhan Universit

17

Access Modifier

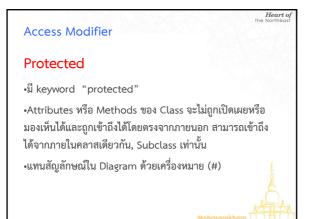
Public

- •มี keyword "public"
- •Attributes หรือ Methods ของ Class จะถูกเปิดเผยหรือ มองเห็นได้และถูกเข้าถึงได้โดยตรงจากภายนอก ไม่มีการปกปิด ใดๆ ทั้งสิ้น
- •แทนสัญลักษณ์ใน Diagram ด้วยเครื่องหมาย (+)



Heart of

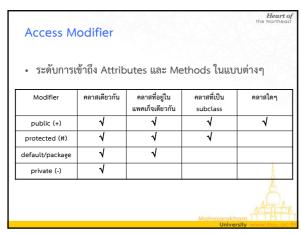
Heart of

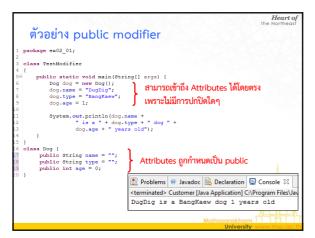


Access Modifier Default หรือ Package •ไม่ต้องมี keyword •Attributes หรือ Methods ของ Class จะไม่ถูกเปิดเผยหรือ มองเห็นได้และถูกเข้าถึงได้โดยตรงจากภายนอก สามารถเข้าถึง ได้จากภายในคลาส และ Package เดียวกัน เท่านั้น •ไม่มีสัญลักษณ์

20

Access Modifier Private •มี keyword "private" •Attributes หรือ Methods ของ Class จะไม่ถูกเปิดเผยหรือ มองเห็นได้และถูกเข้าถึงได้โดยตรงจากภายนอก สามารถเข้าถึง ได้จากภายในคลาสเดียวกันเท่านั้น •แทนสัญลักษณ์ใน Diagram ด้วยเครื่องหมาย (-)

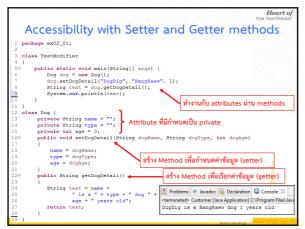


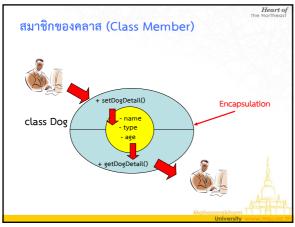


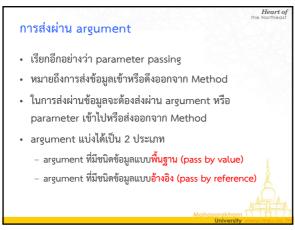
```
## Heart of the Northeast

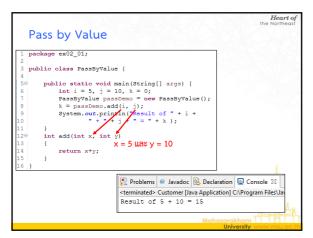
## Package ex02_01;

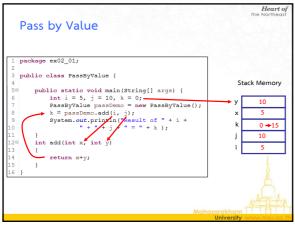
| 2 | 3 | class TestModifier | | |
| 4 | 5 | 5 | 5 |
| 6 | 6 | 6 | 6 | 6 |
| 7 | 8 | 7 | 7 |
| 8 | 9 | 9 | 7 |
| 9 | 9 | 9 | 7 |
| 1 | 9 | 9 | 7 |
| 2 | 9 | 9 | 7 |
| 3 | 9 | 9 | 7 |
| 4 | 9 | 9 | 7 |
| 5 | 9 | 7 | 7 |
| 6 | 7 | 7 | 7 |
| 7 | 7 | 7 |
| 8 | 7 | 7 |
| 9 | 7 | 7 |
| 9 | 8 | 7 | 7 |
| 9 | 9 | 9 | 7 |
| 10 | 1 | 1 | 1 |
| 10 | 1 | 1 |
| 11 | 1 | 1 |
| 12 | 1 | 1 |
| 13 | 1 | 1 |
| 14 | 7 | 7 |
| 15 | 7 | 7 |
| 16 | 7 | 7 |
| 17 | 7 | 7 |
| 18 | 7 | 7 |
| 19 | 7 | 7 |
| 10 | 7 | 7 |
| 10 | 7 | 7 |
| 11 | 7 | 7 |
| 12 | 7 | 7 |
| 13 | 7 | 7 |
| 14 | 7 | 7 |
| 15 | 7 | 7 |
| 16 | 7 | 7 |
| 17 | 7 | 7 |
| 18 | 7 | 7 |
| 19 | 7 | 7 |
| 10 | 7 | 7 |
| 10 | 7 | 7 |
| 11 | 7 | 7 |
| 12 | 7 | 7 |
| 13 | 7 | 7 |
| 14 | 7 | 7 |
| 15 | 7 | 7 |
| 16 | 7 | 7 |
| 17 | 7 |
| 18 | 7 | 7 |
| 19 | 7 |
| 10 | 7 |
| 10 | 7 |
| 10 | 7 |
| 11 | 7 |
| 12 | 7 |
| 13 | 7 |
| 14 | 7 |
| 15 | 7 |
| 16 | 7 |
| 17 | 7 |
| 18 | 7 |
| 18 | 7 |
| 19 | 7 |
| 10 | 7 |
| 10 | 7 |
| 10 | 7 |
| 11 | 7 |
| 12 | 7 |
| 13 | 7 |
| 14 | 7 |
| 15 | 7 |
| 15 | 7 |
| 16 | 7 |
| 17 | 7 |
| 17 | 7 |
| 18 | 7 |
| 18 | 7 |
| 18 | 7 |
| 18 | 7 |
| 18 | 7 |
| 18 | 7 |
| 18 | 7 |
| 18 | 7 |
| 18 | 7 |
| 18 | 7 |
| 18 | 7 |
| 18 | 7 |
| 18 | 7 |
| 18 | 7 |
| 18 | 7 |
| 18 | 7 |
| 18 | 7 |
| 18 | 7 |
| 18 | 7 |
| 18 | 7 |
| 18 | 7 |
| 18 | 7 |
| 18 | 7 |
| 18 | 7 |
| 18 | 7 |
| 18 | 7 |
| 18 | 7 |
| 18 | 7 |
| 18 | 7 |
| 18 | 7 |
| 18 | 7 |
| 18 | 7 |
| 18 | 7 |
| 18 | 7 |
| 18 | 7 |
| 18 | 7 |
| 18 | 7 |
| 18 | 7 |
| 18 | 7 |
| 18 | 7 |
| 18 | 7 |
| 18 | 7 |
| 18 | 7 |
| 18 | 7 |
| 18 | 7 |
| 18 | 7 |
| 18 | 7 |
| 18 | 7 |
| 18 | 7 |
| 18 | 7 |
| 18 | 7 |
| 18 | 7 |
| 18 | 7 |
| 18 | 7 |
| 18 | 7 |
| 18 | 7 |
| 18 | 7 |
| 18 | 7 |
| 18 | 7 |
| 18 | 7 |
| 18 | 7 |
| 18 | 7 |
| 18 | 7 |
| 18 | 7 |
| 18 | 7 |
| 18 | 7 |
| 18 | 7 |
| 18 | 7 |
| 18 | 7 |
| 18 | 7 |
| 18 | 7 |
| 18 | 7 |
| 18 | 7 |
| 18 | 7 |
| 18 | 7 |
| 18 | 7 |
| 18 | 7 |
| 18 | 7 |
| 18 | 7 |
| 18 | 7 |
| 18 | 7 |
| 18 | 7 |
| 18 | 7 |
| 18 |
```

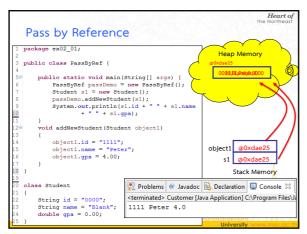




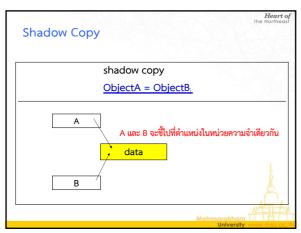








Heart of Object Copy • คือการคัดลอก Object ไปเป็น Object ใหม่โดยใช้ เครื่องหมาย = • เช่น ObjectA = ObjectB; • ผลลัพธ์ที่ได้คือ ObjectA จะเก็บตำแหน่งเดียวกับ ObjectB



การเปรียบเทียบ Object

- Object มีการเก็บข้อมูลแบบอ้างอิง
- การเปรียบเทียบ Object สามารถเปรียบเทียบได้ 2 อย่าง
- เปรียบเทียบ ข้อมูลของ Object
- ต้องใช้ Method equals();
- เปรียบเทียบ ตำแหน่งในหน่วยความจำของ Object
 - -ต้องใช้ เครื่องหมาย ==



Heart of

34

35

Overloading

- Overload ในภาษาจาวาหมายถึงความซ้ำกัน
- Method Overloading คือ การที่ สามารถมี Method ที่มี ชื่อเหมือนกันได้ จาวาจะแยกความแตกต่างของแต่ละ Method โดยดูจาก จำนวน และ ชนิดข้อมูล ของพารามีเตอร์



Heart of

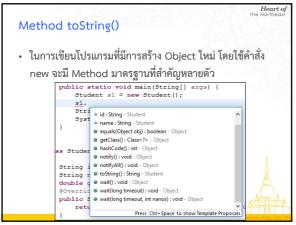
```
Heart of
public static void main(String[] args) {
                                                     public static void main(String[] args) {
    Overloading ov = new Overloading();
    ov.printiine();
    ov.printiine('-');
    ov.printiine('-', 20);
    ov.printline("-");
      for (int i=0;i<20;i++) {
    System.out.print('-');</pre>
      System.out.println();
    id printLine(char ch)
      for (int i=0;i<20;i++) {
    System.out.print(ch);</pre>
      System.out.println();
                                                    🖳 Problems @ Javadoc 🚇 Declaration 📮 Console 🛭
     id printLine(char ch, int num)
                                                      <terminated> Overloading [Java Application] C:\Program File
      for (int i=0;i<num;i++) {
    System.out.print(ch);</pre>
      System.out.println();
     id printLine(String str)
      for (int i=0;i<20;i++) {
    System.out.print(str);</pre>
      System.out.println();
```

Overloading • อย่าเข้าใจผิด!!! • ถ้าจำนวนและชนิดพารามีเตอร์ของ Method Overloading เหมือนกัน – เปลี่ยนชื่อของพารามีเตอร์ ก็ไม่ถือว่าเป็น Overload – เปลี่ยน return type ก็ไม่ถือว่าเป็น Overload

38

```
public static void main(String[] args) {
    Overloading ov = new Overloading();
    ov.printLine();
    ov.printLine('-');
    ov.printLine('-');
    ov.printLine('-');
    ov.printLine("-");
}
                                                                                           public static void main(String[] args) {
    overloading ov = new Overloading();
    ov.printLine();
    ov.printLine('-');
    ov.printLine('-');
    ov.printLine('-');
    ov.printLine("-");
                                                                                              void printLine()
                                                                                                  for (int i=0;i<20;i++) {
    System.out.print('-');
      }
System.out.println();
void printLine(char ch)
                                                                                                  for (int i=0;i<20;i++){
    System.out.print(ch);</pre>
     for (int i=0;i<20;i++) {
    System.out.print(ch);</pre>
                                                                                                   }
System.out.println();
      }
System.out.println();
                                                                                                oid printLine(char ch, int num)
 void printLine(char ch, int num)
                                                                                                  for (int i=0;i<num;i++) {
    System.out.print(ch);</pre>
     for (int i=0;i<num;i++) {
    System.out.print(ch);</pre>
                                                                                                    System.out.println();
                                                                                               oid printLine(String text)
void printLine(String str)
                                                                                                  for (int i=0;i<20;i++) {
    System.out.print(text);
      for (int i=0;i<20;i++) {
    System.out.print(str);
                                                                                                   )
System.out.println();
      }
System.out.println();
       เปลี่ยนชื่อตัวแปรไม่ได้หมายถึงการทำ overloading <sub>hasarakham</sub>
```





Heart of the Northeast Method toString() • Method toString(); • เป็น Method มาตรฐานที่ใช้ในการแปลงค่าของ Object ให้ เป็นข้อมูลที่มีชนิดเป็น String • หากมีการเรียก Object ตรงๆ Object จะทำงานที่ method toString(); • สามารถเปลี่ยนแปลงหรือแก้ไข method มาตรฐาน ได้โดย การทำ overriding (สัปดาห์หน้า)

