

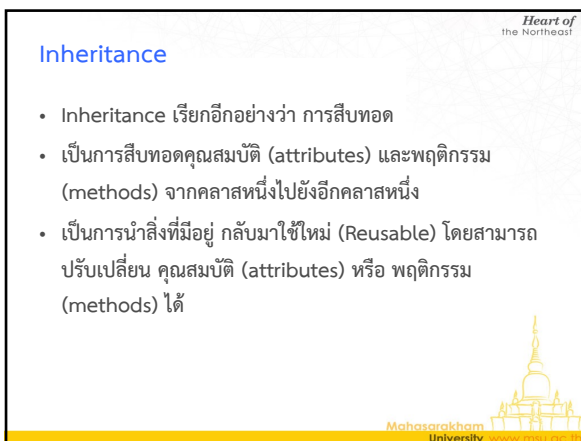


Inheritance & Polymorphism

Mahasarakham University

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

1

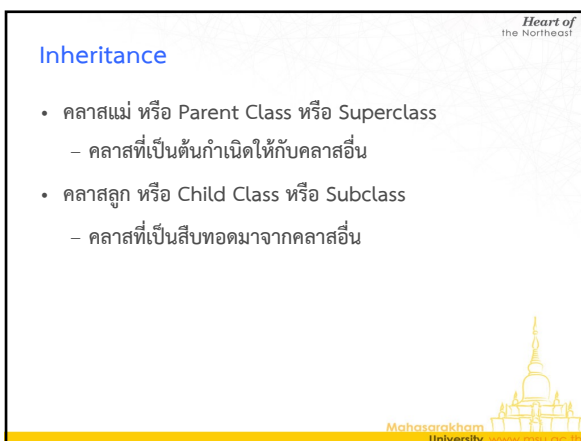


Inheritance

- Inheritance เรียกอีกอย่างว่า การสืบทอด
- เป็นการสืบทอดคุณสมบัติ (attributes) และพฤติกรรม (methods) จากคลาสหนึ่งไปยังอีกคลาสหนึ่ง
- เป็นการนำสิ่งที่มีอยู่ กลับมาใช้ใหม่ (Reusable) โดยสามารถปรับเปลี่ยน คุณสมบัติ (attributes) หรือ พฤติกรรม (methods) ได้

Mahasarakham University

2



Inheritance

- คลาสแม่ หรือ Parent Class หรือ Superclass
 - คลาสที่เป็นต้นกำเนิดให้กับคลาสอื่น
- คลาสลูก หรือ Child Class หรือ Subclass
 - คลาสที่เป็นสืบทอดมาจากคลาสอื่น

Mahasarakham University

3

Inheritance

- keyword ที่ใช้ในการสืบทอดคือ **extends**
 - ใช้ keyword **extends** โดยใช้ต่อท้ายชื่อของ subclass เพื่อเป็นการระบุว่า subclass นั้น สืบทอดมาจาก class ไດ
- ในการสืบทอดนั้น สิ่งที่จะถูกสืบทอดจาก superclass ไปยัง subclass คือสิ่งที่มี access modifier ระดับ package, protected และ public (ทุกระดับ ยกเว้น private)
- ระดับ private จะไม่สืบทอดไปยัง subclass

Mahasarakham University



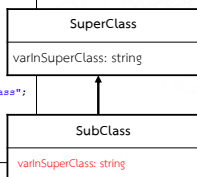
4

Inheritance

```

2 public class Inherit {
3
4     public static void main(String[] args) {
5         SubClass sub = new SubClass();
6         System.out.println(sub.varInSuperClass);
7     }
8 }
9
10 class SuperClass{
11     public String varInSuperClass = "Variable of Superclass";
12 }
13
14 class SubClass extends SuperClass{
15
16 }

```



Problems @ Javadoc Declaration Console
 <terminated> Inherit [Java Application] C:\Program Files\Java\
 Variable of Superclass

Mahasarakham University



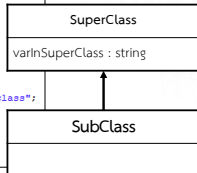
5

Inheritance

```

2 public class Inherit {
3
4     public static void main(String[] args) {
5         SubClass sub = new SubClass();
6         System.out.println(sub.varInSuperClass);
7     }
8 }
9
10 class SuperClass{
11     private String varInSuperClass = "Variable of Superclass";
12 }
13
14 class SubClass extends SuperClass{
15
16 }

```



Problems @ Javadoc Declaration Console
 <terminated> Inherit [Java Application] C:\Program Files\Java\jdk-8.0.60\bin\javaw.exe (7.n.a. 2558 12:18:38)
 Exception in thread "main" java.lang.Error: Unresolved compilation problem:
 The field SuperClass.varInSuperClass is not visible
 at Inherit.main(Inherit.java:6)

Mahasarakham University



6

Inheritance

- ในกรณีที่ superclass และ subclass มีชื่อของ data fields (ทั้ง attributes และ method) เหมือนกัน การเข้าถึง data fields จะเข้าถึง data fields ได้เฉพาะของตัวเองเท่านั้น จะไม่สามารถเข้าถึง data field คลาสอื่นได้
- หรือ การสืบทอดจะไม่สืบทอดมา data fields ของ superclass มาที่ subclass



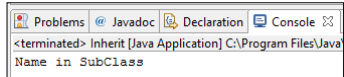
7

Inheritance

```

2 public class Inherit {
3
4     public static void main(String[] args) {
5         SubClass sub = new SubClass();
6         System.out.println(sub.name);
7     }
8 }
9
10 class SuperClass{
11     String name = "Name in SuperClass";
12 }
13
14 class SubClass extends SuperClass{
15     String name = "Name in SubClass";
16 }

```



8

Inheritance

- ในกรณีที่ superclass และ subclass มีชื่อของ data fields (ทั้ง attributes และ method) เหมือนกัน หากต้องการเข้าถึง data fields ของ superclass ต้องทำการแปลงคลาส (class casting) ชนิดให้มันเป็น superclass ก่อน
- การแปลงคลาส (class casting) ทำได้โดยการแปลงแบบ explicit conversion
 - เช่น `castedObject = (superclass)subclassObject;`



9

Class casting

```

1 public class Inherit {
2
3
4     public static void main(String[] args) {
5         // new object of subclass
6         SubClass sub = new SubClass();
7         // explicit conversion from subclass to superclass
8         SuperClass superClass = (SuperClass) sub;
9         System.out.println(superClass.name);
10    }
11 }
12
13 class SuperClass{
14     String name = "Name in SuperClass";
15 }
16
17 class SubClass extends SuperClass{
18     String name = "Name in SubClass";
19 }

```

Maharakham
University



10

Class casting

- ในการสืบทอดคลาสจาก superclass ไปยัง subclass สามารถเพิ่มเติมคุณสมบัติและพฤติกรรมของคลาสได้
- เช่น
 - คลาสของนักศึกษา อาจประกอบได้ รหัสประจำตัว และชื่อ
 - คลาสของนักศึกษาระดับปริญญาโท ประกอบด้วย รหัสประจำตัว ชื่อ และ **ชื่องานวิทยานิพนธ์**

Maharakham
University



11

Inheritance

```

1 public class Inherit2 {
2
3
4     public static void main(String[] args) {
5         Student student = new Student();
6         GradStudent gradStudent = new GradStudent();
7         System.out.println(gradStudent.thesisTitle);
8     }
9 }
10
11 class Student{
12     String id = "123456789";
13     String name = "Petea";
14 }
15
16 class GradStudent extends Student{
17     String thesisTitle = "My Thesis Title";
18 }
19
20
21

```

*** Object ของคลาส Student จะไม่มี thesisTitle เพราะ thesisTitle เพิ่งจะมีการเพิ่มในคลาส GradStudent

```

1 public class Inherit2 {
2
3
4     public static void main(String[] args) {
5         Student student = new Student();
6         GradStudent gradStudent = new GradStudent();
7         System.out.println(gradStudent.thesisTitle);
8     }
9 }
10
11 class Student{
12     String id = "123456789";
13     String name = "Petea";
14 }
15
16 class GradStudent extends Student{
17     String thesisTitle = "My Thesis Title";
18 }
19
20
21

```

12

this()

- คือ method ที่ใช้ในการเรียก Constructor ของตัวเอง
- เพื่อใช้ในการทำงานบางอย่างที่มีขั้นตอนการทำงานที่ซ้ำซ้อนกับ Constructor
- การเรียกใช้ เมธอด this() นั้น จะต้องใช้ parameter ให้ตรงกับ Constructor ของคลาสนั้นๆ

13

this()

```

2 public class TestThis {
3
4     public static void main(String[] args) {
5         Student student1 = new Student("1234", "Peter");
6         Student student2 = new Student("7890", "Michael", 4.00);
7     }
8 }
9
10 class Student {
11     String id = "";
12     String name = "";
13     double gpa = 0.0;
14     public Student(String id, String name) {
15         this.id = id;
16         this.name = name;
17     }
18     public Student(String id, String name, double gpa) {
19         this.id = id;
20         this.name = name;
21         this.gpa = gpa;
22     }
23 }

```

14

this()

```

2 public class TestThis {
3
4     public static void main(String[] args) {
5         Student student1 = new Student("1234", "Peter");
6         Student student2 = new Student("7890", "Michael", 4.00);
7     }
8 }
9
10 class Student {
11     String id = "";
12     String name = "";
13     double gpa = 0.0;
14     public Student(String id, String name) {
15         this.id = id;
16         this.name = name;
17     }
18     public Student(String id, String name, double gpa) {
19         /*
20          * this.id = id;
21          * this.name = name;
22          * this.gpa = gpa;
23          */
24         this(id, name);
25         this.gpa = gpa;
26     }
27 }

```

15

super()

- ในการสืบทอดคลาส สิ่งที่จะไม่ถูกสืบทอดมาคือ constructor ของ superclass
- super() เป็นเมธอดที่ใช้เพื่อเรียก constructor ของ superclass
- การเรียกใช้ เมธอด super() นั้น จะต้องการใช้ parameter ให้ตรงกับ Constructor ของคลาสนั้นๆ



16

ลำดับการทำงานของ constructor

- ในการสืบทอดที่ต้องมีการสร้าง object ของ subclass
- เมื่อสร้าง object ของ subclass จะมีการเรียกใช้งาน constructor ตามลำดับดังนี้
 - ลำดับการเรียก constructor
 1. เรียก constructor ของ superclass
 2. เรียก constructor ของ subclass
 - สรุป การเรียกใช้งาน constructor ของ subclass หากไม่กำหนด super constructor จาบรรทัด super() ใน constructor ของ subclass ให้โดยอัตโนมัติ



17

ลำดับการทำงานของ constructor

```

2 public class Inherit2 {
3
4     public static void main(String[] args) {
5         GradStudent gradStudent = new GradStudent();
6     }
7
8
9     class Student {
10         String id = "123456789";
11         String name = "Peter";
12         public Student() {
13             System.out.println("This is Student constructor");
14         }
15     }
16
17     class GradStudent extends Student {
18         String thesisTitle = "My Thesis Title";
19         public GradStudent() {
20             System.out.println("This is GradStudent constructor");
21         }
22     }
  
```

มี 1 constructor คือ Student ()

จากบรรทัด super constructor เข้ากับบรรทัดแรกของ sub constructor โดยอัตโนมัติ

Problems | Javadoc | Declaration | Console | x
 terminated> Inherit2 [Java Application] C:\Program Files\Java\jdk-11.0.2\bin\java.exe
 This is Student constructor
 This is GradStudent constructor



18

Heart of the Northeast

ลำดับการทำงานของ constructor

ข้อควรระวัง

- หากมีการกำหนด constructor ของ superclass ที่ไม่ใช่ default และไม่มีการเรียก super() จาก constructor ของ subclass จะทำให้โปรแกรมคอมไพล์ไม่ได้

Maharakham University

19

Heart of the Northeast

ลำดับการทำงานของ constructor

```

2 public class Inherit2 {
3
4     public static void main(String[] args) {
5         GradStudent gradStudent = new GradStudent();
6     }
7 }
8
9 class Student {
10     String id = "123456789";
11     String name = "Peter";
12     public Student(String id, String name) {
13         this.id = id;
14         this.name = name;
15         System.out.println("This is Student constructor");
16     }
17 }
18
19 class GradStudent extends Student {
20     String thesisTitle = "My Thesis Title";
21     public GradStudent() {
22         System.out.println("This is GradStudent constructor");
23     }
24 }

```

มี constructor 1 ตัวคือ Student(String id, String name)

ไม่ได้กำหนด super() จาวาจะแทรก super() เข้าครั้งนี้อยู่ในนี้

Maharakham University

20

Heart of the Northeast

ลำดับการทำงานของ constructor

```

2 public class Inherit2 {
3
4     public static void main(String[] args) {
5         GradStudent gradStudent = new GradStudent();
6     }
7 }
8
9 class Student {
10     String id = "123456789";
11     String name = "Peter";
12     public Student(String id, String name) {
13         this.id = id;
14         this.name = name;
15         System.out.println("This is Student constructor");
16     }
17 }
18
19 class GradStudent extends Student {
20     String thesisTitle = "My Thesis Title";
21     public GradStudent() {
22         super();
23         System.out.println("This is GradStudent constructor");
24     }
25 }

```

ไม่เหมือนกัน

จาวาจะแทรก super() เข้าครั้งนี้อยู่ในนี้

Problems | Javadoc | Declaration | Console

Exception in thread "main" java.lang.Error: Unresolved compilation problem: The constructor Student() is undefined

at GradStudent.<init>[Inherit2.java:22]
at Inherit2.main(Inherit2.java:5)

Maharakham University

21

Polymorphism

- **Polymorphism** หรือ การมีหลายรูปแบบ หมายความว่า พฤติกรรมใดๆของ Object สามารถมีได้หลายรูปแบบ
 - เช่น พฤติกรรมการเคลื่อนที่
 - คน ปลา งู หนอน
 - จะต้องเคลื่อนที่ แต่จะเคลื่อนที่ในรูปแบบที่ไม่เหมือนกัน
- ใน OOP นั้น Object สามารถมีกำหนดพฤติกรรมที่เหมือนกันได้ แต่อาจแตกต่างกันในรายละเอียดของพฤติกรรมนั้นๆ

22

Overriding

- Override คือการกำหนดพฤติกรรม (เมธอด) ใน subclass ที่มีชื่อเหมือนเมธอดที่มีอยู่ใน superclass ทำงานได้แตกต่างจากเดิมได้ โดยจะต้องทำตามข้อกำหนดดังนี้
 - จำนวนและชนิดของ argument ต้องเหมือนกัน
 - access modifier จะต้องมียกระดับที่ไม่ต่ำกว่าเมธอดเดิม
- ใน Object ทุกตัวที่ถูกสร้างขึ้นมาจากคลาส จะมี methods ที่เป็น default methods เสมอ เช่น toString() หรือ equals() ซึ่ง methods เหล่านี้จะสามารถถูก override ได้

23

Overriding

```

2 public class Overriden {
3
4     public static void main(String[] args) {
5         SubClass sub = new SubClass();
6         sub.printLetters();
7     }
8 }
9
10 class SuperClass{
11     public void printLetters(){
12         System.out.println("AAAA");
13     }
14 }
15
16 class SubClass extends SuperClass{
17     X
18 }

```

method จะถูกสืบทอดมาดังนี้

Problems Javadoc Declaration Console
 <terminated> Overriden [Java Application] C:\Program Files\Java\jdk-11.0.2\bin\java.exe
 AAAA

24

Heart of the Northeast

Overriding

```

2 public class Overriden {
3
4     public static void main(String[] args) {
5         SubClass sub = new SubClass();
6         sub.printLetters();
7     }
8 }
9
10 class SuperClass{
11     public void printLetters(){
12         System.out.println("AAAA");
13     }
14 }
15
16 class SubClass extends SuperClass{
17     @Override
18     public void printLetters() {
19         System.out.println("BBBB");
20     }
21 }

```

- ชื่อเดียวกัน
- argument เหมือนกัน
- access modifier ไม่ต่ำกว่าของ superclass

Problems | Javadoc | Declaration | Console

<terminated> Overriden [Java Application] C:\Program Files\Java\jre-8\bin\java.exe (T=0.2558145938)

BBBB

ดังนั้นหลักการ Override method

Maharakham University

25

Heart of the Northeast

Overriding

```

2 public class Overriden {
3
4     public static void main(String[] args) {
5         SubClass sub = new SubClass();
6         sub.printLetters();
7     }
8 }
9
10 class SuperClass{
11     public void printLetters(){
12         System.out.println("AAAA");
13     }
14 }
15
16 class SubClass extends SuperClass{
17     @Override
18     private void printLetters() {
19         System.out.println("BBBB");
20     }
21 }

```

Problems | Javadoc | Declaration | Console

<terminated> Overriden [Java Application] C:\Program Files\Java\jre-8\bin\java.exe (T=0.2558145938)

Exception in thread "main" java.lang.Error: Unresolved compilation problem:
The method printLetters() from the type SubClass is not visible
at Overriden.main(Overriden.java:6)

University

26

Heart of the Northeast

Overriding

```

2 public class Overriden {
3
4     public static void main(String[] args) {
5         SubClass sub = new SubClass();
6         sub.printLetters();
7     }
8 }
9
10 class SuperClass{
11     public void printLetters(){
12         System.out.println("AAAA");
13     }
14 }
15
16 class SubClass extends SuperClass{
17     public void printLetters(String letter) {
18         System.out.println("BBBB");
19     }
20 }

```

- ชื่อเดียวกัน
- argument ไม่เหมือนกัน
- access modifier ไม่ต่ำกว่าของ superclass
- ไม่ถือว่าเป็น Override method
- โปรแกรมคอมไพล์ผ่าน รันได้

Problems | Javadoc | Declaration | Console

<terminated> Overriden [Java Application] C:\Program Files\Java\jre-8\bin\java.exe (T=0.2558145938)

AAAA

Maharakham University

27

toString() & equals()

- คลาสใดๆ ในจาวา จะสืบทอดมาจากคลาสชื่อ **Object** เสมอ
- ดังนั้น method หลายตัวในคลาส Object จะถูกสืบทอดมายังคลาสทุกๆ คลาสที่สร้างขึ้น เรียกว่า default methods และสามารถทำ overriding ได้
- Default methods ที่สำคัญ เช่น
 - toString()
 - เมธอดทำงานทันทีเมื่อเรียกชื่อ Object ใดๆ และคืนค่าเป็น String
 - equals()
 - เมธอดใช้เปรียบเทียบ Object ใดๆ กับ Object ตนเอง



28

toString()

- toString()
 - เมธอดทำงานทันทีเมื่อเรียกชื่อ Object ใดๆ และคืนค่าเป็น String

```

2 public class ToStringDemo {
3     public static void main(String[] args) {
4         Person dome = new Person("Dome");
5         System.out.println(dome);
6     }
7 }
8
9 class Person{
10     private String name;
11     public Person(String name) {
12         this.name = name;
13     }
14 }

```

Problems | Javadoc | Declaration | Console
 <terminated> ToStringDemo [Java Application] C:\Program Fil
 Person@2a139a55

แสดงตำแหน่งใน memory ของ Object นั้นๆ



29

toString()

```

2 public class ToStringDemo {
3     public static void main(String[] args) {
4         Person dome = new Person("Dome");
5         System.out.println(dome);
6     }
7 }
8
9 class Person{
10     private String name;
11     public Person(String name) {
12         this.name = name;
13     }
14     @Override
15     public String toString() {
16         return this.name;
17     }
18 }

```

Problems | Javadoc | Declaration | Console
 <terminated> ToStringDemo [Java Application] C:\Program Fil
 Dome

Override toString() จากคลาส Object



30

Heart of the Northeast

equals()

- equals()
- เมธอดใช้เปรียบเทียบ Object ใด ๆ กับ Object ตนเอง

```

2 public class ToStringDemo {
3     public static void main(String[] args) {
4         Person person1 = new Person("Dome");
5         Person person2 = new Person("Dome");
6         System.out.println(person1==person2);
7         System.out.println(person1.equals(person2));
8     }
9 }
10
11
12 class Person{
13     private String name;
14     public Person(String name) {
15         this.name = name;
16     }
17     @Override
18     public boolean equals(Object obj) {
19         return this.name == ((Person)obj).name;
20     }
21 }
  
```

1. เอาตำแหน่งใน memory มาเปรียบเทียบ
2. เอาเนื้อหาของ object มาเปรียบเทียบ

Mahasarakham University

31

Heart of the Northeast

Abstract & Interface

- Abstract แปลตามภาษา Programming แปลว่า ยังไม่เสร็จ, ยังไม่สมบูรณ์
- คลาสหรือเมธอดใดที่ขึ้นต้นด้วย **abstract** หมายถึง เมื่อต้องการใช้งาน จะต้องมีการ เขียนโปรแกรมในส่วนนั้นให้เสร็จก่อน จึงจะคอมไพล์ได้
- Abstract สร้างขึ้นเพื่อบังคับให้ผู้ที่จะใช้งานคลาสหรือเมธอดนั้น ต้องเขียนโปรแกรมในส่วนนั้นใหม่ทุกครั้ง

Mahasarakham University

32

Heart of the Northeast

Abstract

- จะมี keyword คำว่า abstract นำหน้า
- การใช้งาน abstract class ต้องใช้ keyword ในการสืบทอดเป็น extends เสมอ
- abstract class ไม่สามารถ new ได้

Mahasarakham University

33

Heart of the Northeast

Abstract

```

2 public class AbstractDemo {
3
4     public static void main(String[] args) {
5         GradStudent grad = new GradStudent();
6         grad.setName("Peter");
7         System.out.println(grad.name);
8     }
9 }
10
11 abstract class Student{
12     String name;
13     public void setName(String name){
14         this.name = name;
15     }
16 }
17
18     ต้องใช้ keyword extends ในการสืบทอด
19 class GradStudent extends Student{
20     public void setName(String name){
21         this.name = name + " (Graduated student)";
22     }
23 }

```

Problems @ Javadoc Declaration Console

<terminated> AbstractDemo [Java Application] C:\Program Files\Java\jdk-9.0.4\bin\java.exe
 Peter (Graduated student)

34

Heart of the Northeast

Abstract

```

2 public class AbstractDemo {
3
4     public static void main(String[] args) {
5         GradStudent grad = new GradStudent();
6         grad.setName("Peter");
7         System.out.println(grad.name);
8
9         Student student = new Student();
10    }
11 }
12
13 abstract class Student{
14     String name;
15     public void setName(String name){
16         this.name = name;
17     }
18 }
19
20 class GradStudent extends Student{
21     public void setName(String name){
22         this.name = name + " (Graduated student)";
23     }
24 }

```

abstract class จะ new ไม่ได้

Exception in thread "main" java.lang.Error: Unresolved compilation problem:
 Cannot instantiate the type Student
 at AbstractDemo.main(AbstractDemo.java:9)

Mahachulalongkornrajavidyalaya University

35

Heart of the Northeast

Abstract

```

2 public class AbstractDemo {
3
4     public static void main(String[] args) {
5         GradStudent grad = new GradStudent();
6         grad.setName("Peter");
7         System.out.println(grad.name);
8     }
9 }
10
11 abstract class Student{
12     String name;
13     abstract public void setName(String name);
14 }
15
16
17 class GradStudent extends Student{
18     public void setName(String name){
19         this.name = name + " (Graduated student)";
20     }
21 }

```

abstract method จะไม่มี body {}

ถ้ามีการสืบทอดต้องบังคับให้เขียนโปรแกรม ในส่วน
 ของ abstract method เสมอ

University

36

Heart of the Northeast

Abstract

```

2 public class AbstractDemo {
3
4     public static void main(String[] args) {
5         GradStudent grad = new GradStudent();
6         grad.setName("Peter");
7         System.out.println(grad.name);
8     }
9 }
10
11 abstract class Student{
12     String name;
13     abstract public void setName(String name);
14 }
15
16
17 class GradStudent extends Student{
18
19 }

```

ไม่ได้เขียน abstract method

Problems | Javadoc | Declaration | Console

Exception in thread "main" java.lang.Error: Unresolved compilation problem:
The type GradStudent must implement the inherited abstract method Student.setName(String)

at GradStudent.setName(AbstractDemo.java:17)
at AbstractDemo.main(AbstractDemo.java:6)

37

Heart of the Northeast

Interface

Interface คล้ายกับ abstract class แตกต่างตรงที่

- interface ไม่ใช่คลาส ต้องประกาศให้อยู่ในระดับเดียวกับคลาสเท่านั้น
- ใน interface จะมีแต่ เมธอดที่ยังไม่สมบูรณ์เท่านั้น (ไม่มี body นั้นเอง)
- มี keyword คือ **interface** นำหน้า (ไม่มีคำว่า class)
- การใช้งานต้องใช้ keyword **implements** ในการสืบทอดเสมอ

Maharakrakham University

38

Heart of the Northeast

Interface

```

2 public class InterfaceDemo {
3
4     public static void main(String[] args) {
5         GradStudent grad = new GradStudent();
6         grad.setName("Peter");
7         System.out.println(grad.name);
8     }
9 }
10
11 interface Student{
12     public void setName(String name);
13 }
14
15 class GradStudent implements Student
16 {
17     String name;
18     @Override
19     public void setName(String name) {
20         this.name = name;
21     }
22 }

```

interface ไม่ใช่ class ดังนั้น จะไม่มี keyword class ข้างหน้า

มี method อยู่ 1 ตัว บ่งบอกว่าต้องการเขียนโปรแกรมให้กับ method นี้

Maharakrakham University

39

Interface

```
2 public class InterfaceDemo {
3
4     public static void main(String[] args) {
5         GradStudent grad = new GradStudent();
6         grad.setName("Peter");
7         System.out.println(grad.name);
8     }
9 }
10
11 interface Student {
12     public void setName(String name);
13 }
14
15 class GradStudent implements Student {
16     String name;
17 }
18 }
```

ไม่มีการเขียนโปรแกรมในเมธอดที่ interface บังคับ จะคอมไพล์ไม่ได้

Problems | Javadoc | Declaration | Console

<terminated> InterfaceDemo [Java Application] C:\Program Files\Java\jre1.8.0_60\bin\javaw.exe (T.n.a. 2558 16:00:19)

Exception in thread "main" java.lang.Error: Unresolved compilation problem:

at InterfaceDemo.main(InterfaceDemo.java:4)

University

40
