

Hiding the Implementation

Vittayasak Rujivorakul

Lecture 5

Hiding the Implementation

Vittayasak Rujivorakul

Lecture 5

package : Library unit

- ช่วยในการรวบรวม library unit เพื่อแจกจ่ายให้ใช้งานร่วมกันได้
- การสั่ง import คือการ include library unit นั้นเข้ามาในโปรแกรม
 - ทำการ import ทั้ง library
`import java.util.*;`
 - ทำการ import เฉพาะ class
`import java.util.Vector;`
- การสร้าง package ทำโดยระบุไว้ในบรรทัดแรกของ class files
`package <package name>;`
 - หากมีมากกว่า 1 files ใน package ทุก file ต้องระบุ package stmt ไว้แถวๆ ว่างๆ ใต้ directory เดียวกัน

การสร้าง package

- เพื่อให้สามารถแยกแต่ละ package ออกจากกันและสามารถใช้ได้
 - การกำหนดชื่อของ package ซึ่งต้อง unique (ใช้ชื่อ Internet domain)
 - การ locate package นั้น (อาศัย classpath)
- การค้นหา package โดย Java Interpreter
 - หาจาก environment variable CLASSPATH
 - รวมกับ import package

```
CLASSPATH = . ; c : \javablab
import edu.logic.mypackage.*;

==> c : \javablab \ edu \ logic \ mypackage \ *.class
```

package

- เมื่อ Java มองหา imported class (l. c., X) ตาม directory ที่ระบุ
 - หากพบ file.class (l. c., X.class) เพียงอย่างเดียว ทำการ import class นั้น
 - หากพบทั้ง file.class และ file.java (l. e., X.java) compiler จะเปรียบเทียบเวลาของ 2 files นั้นหาก .java ใหม่กว่า .class จะทำการ recompile X.java นั้นเพื่อสร้าง X.class ตัวใหม่ให้โดยอัตโนมัติ
- ในกรณีที่ class library ที่อ้างถึงมีชื่อซ้ำกัน ต้องระบุให้ชัดเจนว่าต้องการใช้ตัวใด (เช่นหาก Vector class ที่ใช้มีอยู่ใน 2 package ที่ import เข้ามาต้องระบุให้ชัดว่าจะใช้อันใด)
`java.Util.Vector v;`

การซ่อน implementation (Implementation Hiding)

- Java ให้มีการกำหนด Access Specifiers เพื่อช่วยในการระบุสิ่งที่สามารถอ้างถึงได้ ช่วยกำหนดการ (ซ่อน / ไม่ซ่อน) จากภายนอก
- Access Specifiers : ใช้ได้กับทั้ง data members และ method
 - **public** : อ้างถึงจากภายนอกได้
 - “friendly” (ไม่ระบุ) หรือ Default access : ถือเป็น package access
 - อ้างได้จาก package เดียวกัน หรือ default package (files ใน directory เดียวกัน)
 - **protected** : อ้างถึงโดย class ที่อยู่ใน hierarchy เดียวกัน
 - **private** : ใช้ภายในเท่านั้น

การกำหนด access

- กำหนด member ให้เป็น **public** ในกรณีที่ต้องการให้ทุกคนใช้ได้
- กำหนด member เป็น **default** หากต้องการใช้เพียงใน package
- ในกรณีที่ member ใช้ภายในลูกหลาน (สืบทอด) กำหนดให้เป็น **protected**
- member ที่ใช้เพียงภายในเพื่อซ่อนรายละเอียดจากภายนอกกำหนดให้เป็น **private**
- สร้าง “Accessor/Mutator” methods หรือ (**get/set**) เพื่อใช้ในการอ่านและเปลี่ยนค่า

Class access

- มีเพียง 2 access specifiers สำหรับ class
 - default : class ภายใน package เดียวกัน อ้างถึงได้
 - public : อ้างถึงได้จาก class ใด ๆ
- มีได้เพียง 1 public class ใน 1 file แต่สามารถมี class ผู้ช่วย (default class ที่ใช้ได้เพียงใน package เดียวกัน) หลาย class ใน file เดียวกัน
- ชื่อของ public class ต้องเป็นชื่อเดียวกันกับชื่อ file
- ในกรณีที่ไม่มี public class ใน file สามารถตั้งชื่อ files อย่งไรก็ได้

ตัวอย่าง file : Lunch. java

```
Class Soup {  
    // (1) ไม่อนุญาตให้สร้างผ่าน Constructor  
    private Soup () { }  
    // (2) ยอมให้มีการสร้างผ่าน static method เท่านั้น  
    public static Soup makeSoup() {  
        return new Soup();  
    }  
    // (3) ให้มีการสร้าง object Soup ได้เพียงตัวเดียวเท่านั้น  
    // (singleton pattern) สามารถ access ได้จาก access method  
    private static Soup ps = new Soup();  
    public static Soup access () {  
        return ps;  
    }  
    public void f() { }  
}
```

```
Class Sandwich {  
    void f() { new Lunch(); }  
    public class Lunch {  
        void test() {  
            // (3) สร้าง object Soup โดย constructor ไม่ได้  
            // Soup p = new Soup();  
            // access static method  
            Soup.access().f();  
        }  
    }  
}
```

การเข้าถึงได้ของ Member (Class Member Accessibility)

การเข้าถึง	Member Visibility			
	public	protected	package	private
Class เดียวกัน	ได้	ได้	ได้	ได้
Class ใน package เดียวกัน	ได้	ได้	ได้	ไม่ได้
Subclass ของต่าง package	ได้	ได้	ไม่ได้	ไม่ได้
Non-subclass ของ package	ได้	ไม่ได้	ไม่ได้	ไม่ได้