

BÀI 3

TÍNH BAO GÓI

TRONG LẬP TRÌNH JAVA

TÌNH HUỐNG DẪN NHẬP

Bài toán: Hệ thống quản lý chuột máy tính

Chủ cửa hàng Nam Việt chuyên cung cấp các loại chuột máy tính muốn xây dựng một phần mềm để tiện cho việc quản lý. Qua tìm hiểu ta thấy có 2 loại chuột cơ bản là chuột không dây và chuột có dây chúng thuộc loại chuột quang hoặc chuột laser.



Một yêu cầu khác là chủ cửa hàng muốn sau khi xây dựng phần mềm này có thể tái sử dụng lại một số lớp đối tượng để mình tự viết một số chương trình khác.




Vậy theo Anh Chị làm thế nào để có thể phân loại để quản lý được thông tin về các loại chuột? Và làm sao để có thể tái sử dụng lại một số chương trình đã được xây dựng?

MỤC TIÊU BÀI HỌC


Trình bày về tính bao gói trong lập trình hướng đối tượng.





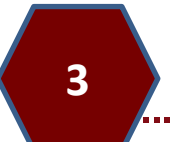
Mô tả cách tổ chức lớp sử dụng package và cách thức kiểm soát truy xuất vào các thành phần của đối tượng thông qua các từ khóa (private, public, protected, default).



Xây dựng chương trình gồm các lớp với các thành phần sử dụng các từ khóa truy xuất được tổ chức trong package.



NỘI DUNG

-  1 Giới thiệu về tính bao gói trong lập trình hướng đối tượng.
-  2 Cách thức tổ chức lớp sử dụng Package.
-  3 Cách thức kiểm soát truy xuất vào các thành phần của đối tượng.

3.1. TÍNH BAO GÓI TRONG LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

Khả năng truy xuất vào các thành phần của một đối tượng trong khi vẫn đảm bảo các thông tin riêng tư bên trong đối tượng được gọi là tính che giấu thông tin (hay còn gọi là tính bao gói).



Chương trình quản lý chuột được đóng gói trong đĩa CD.

Người sử dụng có thể sử dụng dữ liệu được đóng gói.

3.2. TỔ CHỨC LỚP TRONG CÁC PACKAGE



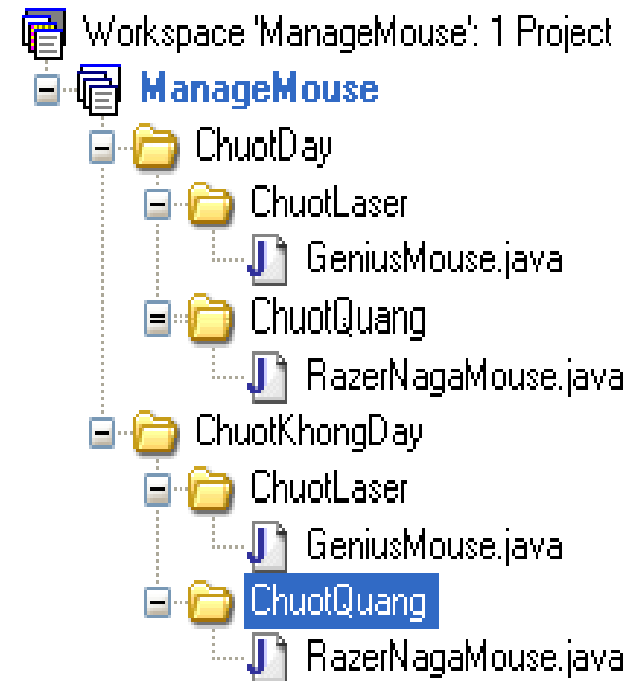
Package và cách thức tổ chức các lớp trong Java.



Truy xuất vào các lớp trong package.

3.2.1. PACKAGE VÀ CÁCH THỨC TỔ CHỨC LỚP VÀO PACKAGE

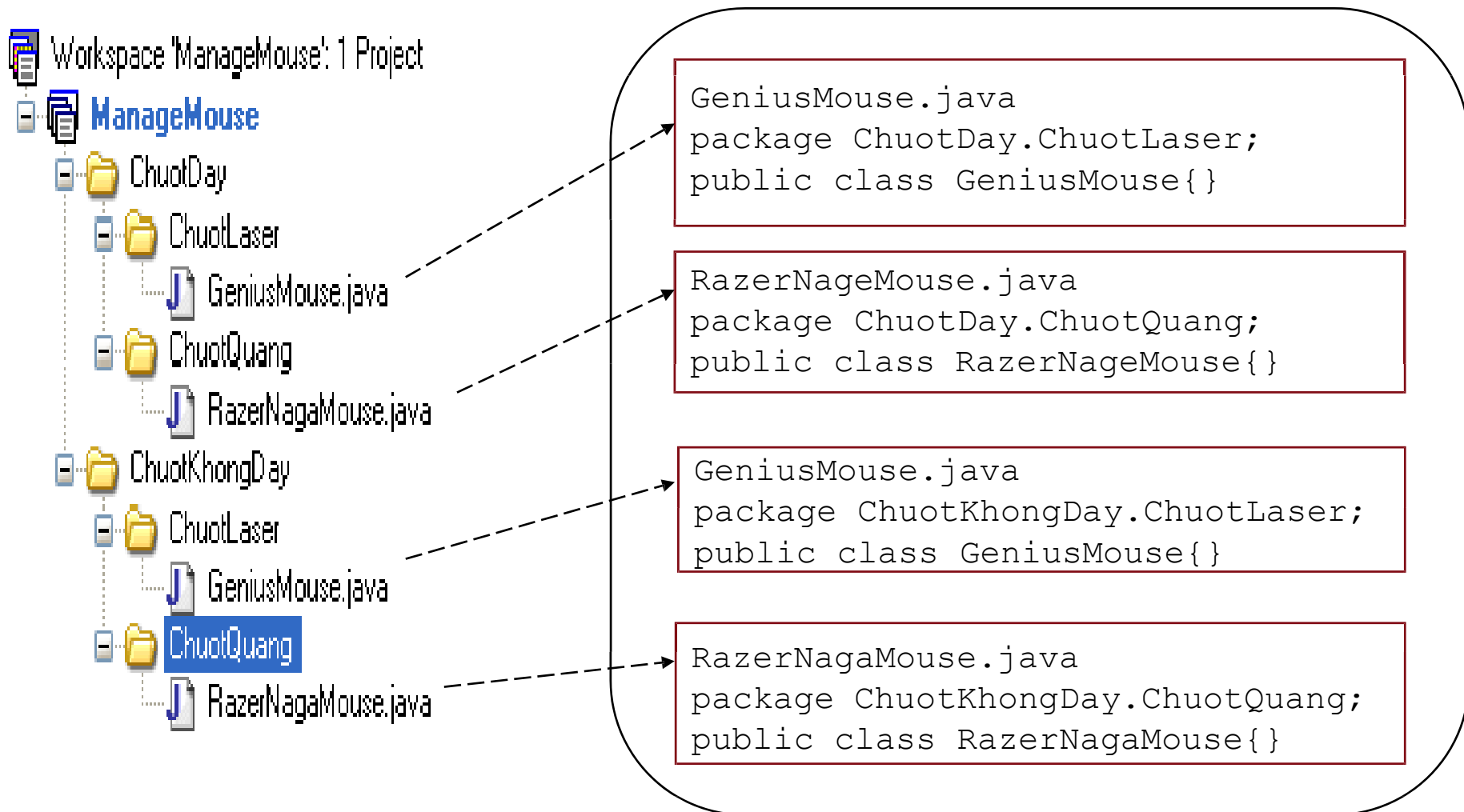
- Khi số lượng lớp có trong một ứng dụng tăng lên có thể nảy sinh các vấn đề:
 - Tìm kiếm;
 - Tên lớp dài;
 - Vấn đề lớp trùng tên;
 - Chia sẻ, tái sử dụng và tích hợp.
- Java cho phép tổ chức các lớp vào các package. Package là cơ chế tổ chức lớp giống như tổ chức cây thư mục trong máy tính.



Với bài tập tình huống có thể tổ chức quản lý loại chuột như trên

3.2.1. PACKAGE VÀ CÁCH THỨC TỔ CHỨC LỚP VÀO PACKAGE (tiếp theo)

Sử dụng từ khóa package ở dòng đầu tiên của file mã nguồn java, theo sau là tên package.



Tổ chức file trong thư mục

Tổ chức lớp trong package

3.2.2. TRUY XUẤT VÀO CÁC LỚP TRONG PACKAGE

- Sử dụng từ khóa import:

```
package demo;
```

```
import ChuotDay.ChuotLaser.GeniusMouse;
```

```
public class PackageDemo{
```

```
    public void runDemo {
```

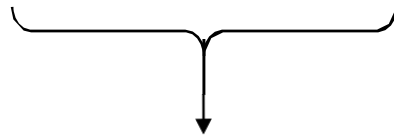
```
        GeniusMouse s=new GeniusMouse();
```

```
        ChuotKhongDay.ChuotLaser.GeniusMouse gen
```

```
            = new ChuotKhongDay.ChuotLaser.GeniusMouse();
```

```
    }
```

```
}
```



- Sử dụng tên đầy đủ của một lớp khi có sự xung đột về tên lớp.

3.2.3. CÁC PACKAGE XÂY DỰNG SẴN

- Nền tảng java cung cấp cho lập trình viên một tập các Java API cơ bản để lập trình ứng dụng.
- Các gói cơ bản:
 - java: chứa các gói thư viện chuẩn của java.
 - java.lang: các lớp thư viện mặc định được **import** trong chương trình java.
 - java.util: các lớp tiện ích. Ví dụ lớp Scanner giới thiệu trong bài 1.
 - javax: chứa các gói thư viện chuẩn mở rộng
- Các gói này được nén vào trong file jar đặt trong thư mục lib của JDK. Khi sử dụng cần cung cấp CLASSPATH dẫn tới.

Sử dụng công cụ jar của Java người dùng có thể tự nén các lớp trong chương trình thành một file jar duy nhất tiện cho việc chia sẻ và cài đặt.

3.2.3. CÁC PACKAGE XÂY DỰNG SẴN (tiếp theo)

```
/**
 * Write a description of class Client here.
 * Client.java
 */
import java.util.Scanner;
import ChuotDay.ChuotLaser.GeniusMouse;

public class Client {
    public void inputMouse() {
        Scanner scan = new Scanner(System.in);
        System.out.println("Name Mouse: ");
        String nameMouse = scan.next();

        GeniusMouse gen = new GeniusMouse(nameMouse);
        System.out.println("Name of genius mouse: " +
gen.toString());
    }
}
```

Đáp án tham khảo

3.3. KIỂM SOÁT TRUY XUẤT

Các từ khóa truy xuất:

- Sử dụng từ khóa truy xuất `private`
- Sử dụng từ khóa truy xuất `public`
- Sử dụng từ khóa truy xuất `<default>`
- Bảng tổng kết.
- Các từ khóa truy xuất được dùng để điều khiển cách thức truy cập vào các thành phần trong một lớp.
- Các từ khóa truy xuất cũng quyết định xem một thành phần của lớp này có được truy xuất bởi một thành phần của lớp khác hay không.
- Java cung cấp các mức truy xuất tương ứng với các từ khóa sau:
 - `public`
 - `private`
 - `<default>`: không chỉ định từ khóa truy xuất

3.3.1. SỬ DỤNG TỪ KHÓA PUBLIC

- Từ khóa `public` tạo ra mức độ truy xuất lớn nhất.
- Từ khóa `public` được áp dụng cho lớp, interface, trường, hàm tạo và phương thức.
 - Áp dụng cho lớp hoặc interface: Có thể truy xuất vào lớp, interface này từ các gói khác.
 - Áp dụng cho trường, hàm tạo và phương thức: Có thể truy xuất vào các thành phần này từ các lớp khác.

```
package mouse.wiredmouse;

public class GeniusMouse {
    public String name;
    private String publisher;
    String description;
    public GeniusMouse() {}

    public GeniusMouse(String
name, String publisher, String
description) {
        this.name = name;
        this.publisher =
publisher;
        this.description =
description;
    }

    @Override
    public String toString() {
        return "Name: " +
this.name;
    }
}
```

3.3.2. SỬ DỤNG TỪ KHÓA PRIVATE

- Từ khóa tạo ra mức độ truy xuất hẹp nhất.
- Từ khóa `private` có thể áp dụng cho lớp, interface, trường, hàm tạo và phương thức.
 - Áp dụng cho lớp và interface: Không thể truy xuất vào các lớp, interface này từ các gói khác.
 - Áp dụng cho trường, hàm tạo và phương thức: Không thể truy xuất vào được các thành phần này từ các lớp khác.

```
package mouse.wiredmouse;

public class GeniusMouse {
    public String name;
    private String publisher;
    String description;
    public GeniusMouse() {}

    public GeniusMouse(String
name, String publisher, String
description) {
        this.name = name;
        this.publisher =
publisher;
        this.description =
description;
    }

    @Override
    public String toString() {
        return "Name: " +
this.name;
    }
}
```

3.3.2. SỬ DỤNG CHỈ THỊ TRUY XUẤT MẶC ĐỊNH <DEFAULT>

- Khi không chỉ định từ khóa truy xuất thì java sẽ áp dụng chỉ thị truy xuất mặc định default.
- Chỉ thị truy xuất mặc định có thể áp dụng cho lớp, interface, trường, hàm tạo và phương thức.
 - Áp dụng cho lớp, interface: Chỉ truy xuất được các lớp, interface này khi cùng một package.
 - Áp dụng cho trường, hàm tạo, phương thức: Chỉ truy xuất được từ các lớp trong cùng một package.

```
package mouse.wiredmouse;

public class GeniusMouse {
    public String name;
    private String publisher;
    String description;
    public GeniusMouse() {}

    public GeniusMouse(String
name, String publisher, String
description) {
        this.name = name;
        this.publisher =
publisher;
        this.description =
description;
    }

    @Override
    public String toString() {
        return "Name: " +
this.name;
    }
}
```

VÍ DỤ

```
/**
 * Write a description of class Client here.
 * Client.java
 * @author (your name)
 * @version (a version number or a date)
 */
public class Client {
    public static void demoStudent() {
        CD objCD = new CD("MAXELL", "DVD-R", 10);
        System.out.println("speed: " + objCD.speed);
        System.out.println("publisher: " + objCD.publisher);
    }
}
```

speed: 10

Exception in thread "main" java.lang.RuntimeException:
Uncompilable source code - publisher has private access in CD
at Client.main(Client.java:5)

Java Result: 1

CÂU HỎI THẢO LUẬN

Bổ từ public và protected có gì khác nhau.

CASE STUDY

Mô phỏng sự hoạt động của đồng hồ điện tử:

Lớp các đối tượng đồng hồ lấy tên là DigitalClock, với những đặc điểm được mô tả như sau:

- Có khả năng hiển thị giờ, phút, giây theo định dạng: hh:mm:ss
- Có khả năng thay đổi trạng thái giờ, phút, giây sau mỗi lần chạy.
- Một đối tượng đầu cuối sẽ sử dụng đối tượng đồng hồ và chạy trong khoảng 1000s.

BẢNG TÓM TẮT

Access modifier	Elements Visible in			
	Class	Package	Subclass	Outside package
public	yes	yes	yes	yes
protected	yes	yes	yes	no
private	yes	no	no	no
no modifier	yes	yes	no	no

CÂU HỎI THẢO LUẬN

Vì sao cần sử dụng đến package?

TÓM LƯỢC CUỐI BÀI

Sau khi học xong bài này chúng ta đã nắm được các kiến thức sau:

- Hiểu được cách tạo gói.
- Sử dụng được gói đã tạo và các gói được cung cấp sẵn.
- Hiểu cách sử dụng các bộ từ truy xuất: private, public, protected, default.
- Xây dựng ứng dụng phân chia trong các gói khác nhau.

CÂU HỎI TRẮC NGHIỆM

Câu hỏi 1 trên 8 ▼

Điểm: 10

Bạn đang viết một đoạn mã cho 1 lớp, lớp này nằm trong package test và sử dụng các thành phần trong gói java.util. Lựa chọn đoạn mã để khởi đầu cho file dữ liệu nguồn.

- ☐ A. Import java.util.*;
- ☐ B. Package test;
import java.util.*;
- ☐ C. Import java.util.*;
package test;
- ☐ D. Package test;

PROPERTIES

On passing, 'Finish' button:

On failing, 'Finish' button:

Allow user to leave quiz:

User may view slides after quiz:

User may attempt quiz:

Goes to Next Slide

Goes to Next Slide

At any time

At any time

Unlimited times



Properties...



Edit in Quizmaker

Search

Thuộc tính của lớp khi không xác định chính xác với bộ từ truy xuất,...?

Thuộc tính của lớp khi không xác định chính xác với bộ từ truy xuất, thì sẽ lấy bộ từ nào là mặc định?

Gợi ý:

Thuộc bộ từ mặc định default.

Bộ từ public và protected có gì khác nhau?

Tại sao không nên sử dụng `java.util.*`; mà nên sử dụng `java.util.Scanner`;

PROPERTIES

Allow user to leave interaction:

[Anytime](#)

Show 'Next Slide' Button:

[Don't show](#)

Completion Button Label:

[Next Slide](#)



Properties...



Edit in Engage

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

Select a term:

Access modifier

Default

File jar

Java.lang

Java.util

Javax

Package default

Từ khóa import

Từ khóa package

Từ khóa private

Từ khóa protected

Từ khóa public

Access modifier

Bổ từ truy xuất gồm các chỉ định truy xuất: private, public, protected.

PROPERTIES

Allow user to leave interaction:

Show 'Next Slide' Button:

Completion Button Label:

Anytime

Don't show

Next Slide

