

Mục tiêu: Sau bài thực hành này, học viên sẽ biết cách :

- tạo và sử dụng user-defined package
- lưu trữ và truy xuất các đối tượng trong cấu trúc mảng (array)

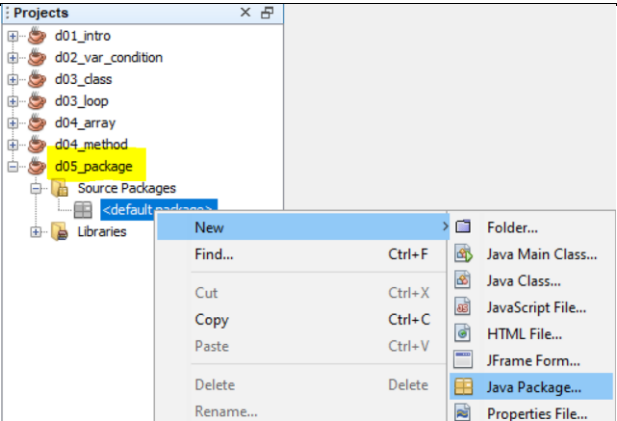
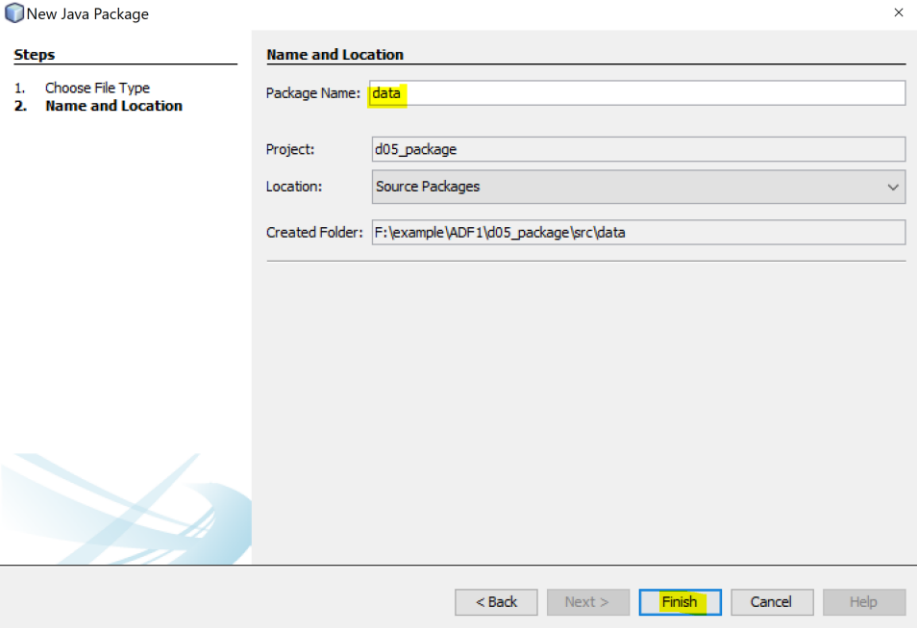
Bài tập:

Viết chương trình quản lý thông tin học viên, bao gồm các chức năng:

1. Thêm sinh viên mới
2. Liệt kê danh sách sinh viên
3. Tìm kiếm sinh viên theo tên
4. Xóa sinh viên ra khỏi danh sách lưu trữ
5. Thoát

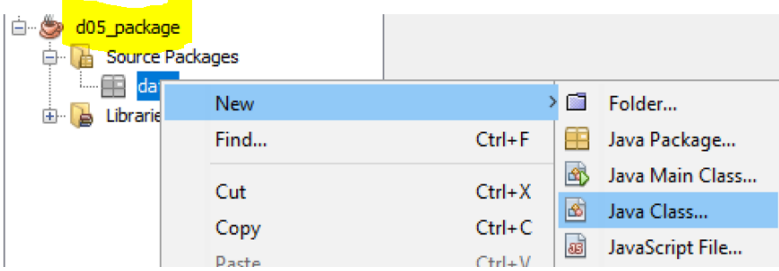
HD thực hiện:

- Mở Netbean, Tạo project Java Application, đặt tên **d05_package** (nhớ bỏ check **Create Main Class**)
- Trong project **d05_package**, tạo package **data**

<p>Trong project d05_package:</p> <ul style="list-style-type: none"> - Nhấp chuột phải trên node "<default Packages>" - chọn New - chọn Java package 	
<p>Trong màn hình New Java Package</p> <ul style="list-style-type: none"> - gõ tên package là data trong ô Package Name - Bấm [Finish] để hoàn tất việc tạo package trong project 	

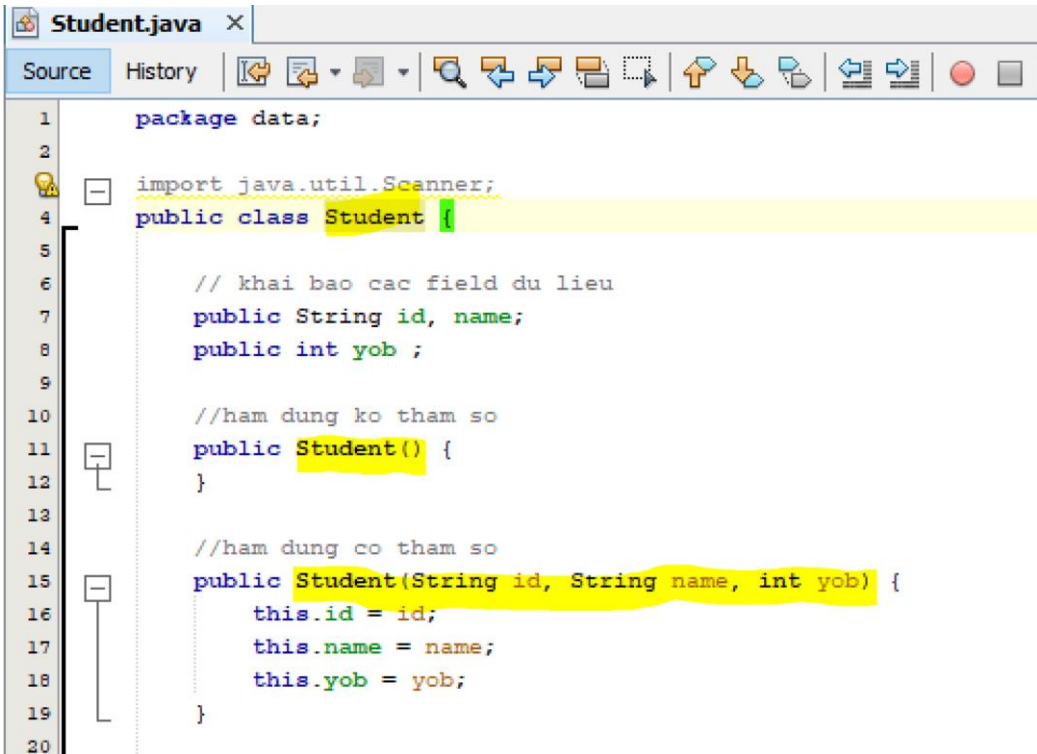
LAB GUIDE 6 – Package

- Định nghĩa class **Student**, trong package **data**, mô tả 1 sinh viên:
 - **Phân tích:**
Cấu trúc một đối tượng sinh viên bao gồm :
 - a. Các fields dữ liệu:
 - i. String **id, name**
 - ii. int **yob**
 - b. Các constructors để khởi tạo giá trị ban đầu cho các field dữ liệu.
 - c. Các method :
 - **input()** để nhập thông tin sinh viên
 - **output()** để xuất thông tin sinh viên ra màn hình
 - **toString()** để trả về 1 chuỗi mô tả thông tin chi tiết sinh viên
 - **Thực hiện:**
Trong package **data**, tạo Java Class **Student**



- Nhấp chuột phải trên package **[data]**, chọn **New**, chọn **Java Class**, gõ tên Class Name: **Student**, bấm **[Finish]**

- Viết code cho class **Student**



LAB GUIDE 6 – Package

```

21 //ham nhap thong tin sinh vien
22 public void input(){
23
24 }
25
26 //ham xuat thong tin sinh vien
27 public void output(){
28
29 }
30
31 @Override
32 public String toString() {
33     return String.format("%s, %s, %d", id, name, yob);
34 }
35

```

Hai hàm **input()**, **output()** sẽ được viết code chi tiết như phần bên dưới

- Code chi tiết của hàm **input()**: nhập thông tin của sinh viên và kiểm tra năm sinh phải ≤ 2006

```

22 //ham nhap thong tin sinh vien
23 public void input() {
24     Scanner sc = new Scanner(System.in);
25
26     System.out.print("Nhap ma so: ");
27     id = sc.nextLine().trim();
28
29     System.out.print("Nhap ho ten: ");
30     name = sc.nextLine().trim();
31
32     while (true) {
33         System.out.print("Nhap nam sinh: ");
34         yob = Integer.parseInt(sc.nextLine().trim());
35         if(yob <= 2006){
36             break; // nam sinh <=2006 => ket thuc vong lap
37         }
38         System.out.println("Nam sinh ko hop le !!!");
39     }
40 }

```

- Code chi tiết của hàm **output**: hiển thị thông tin chi tiết của sinh viên ra màn hình.

```

42 //ham xuat thong tin sinh vien
43 public void output() {
44     System.out.println(" >> Thong tin sinh vien : ");
45     System.out.println("    id: " + id);
46     System.out.println("    ten: " + name);
47     System.out.println("    nam sinh: " + yob);
48 }

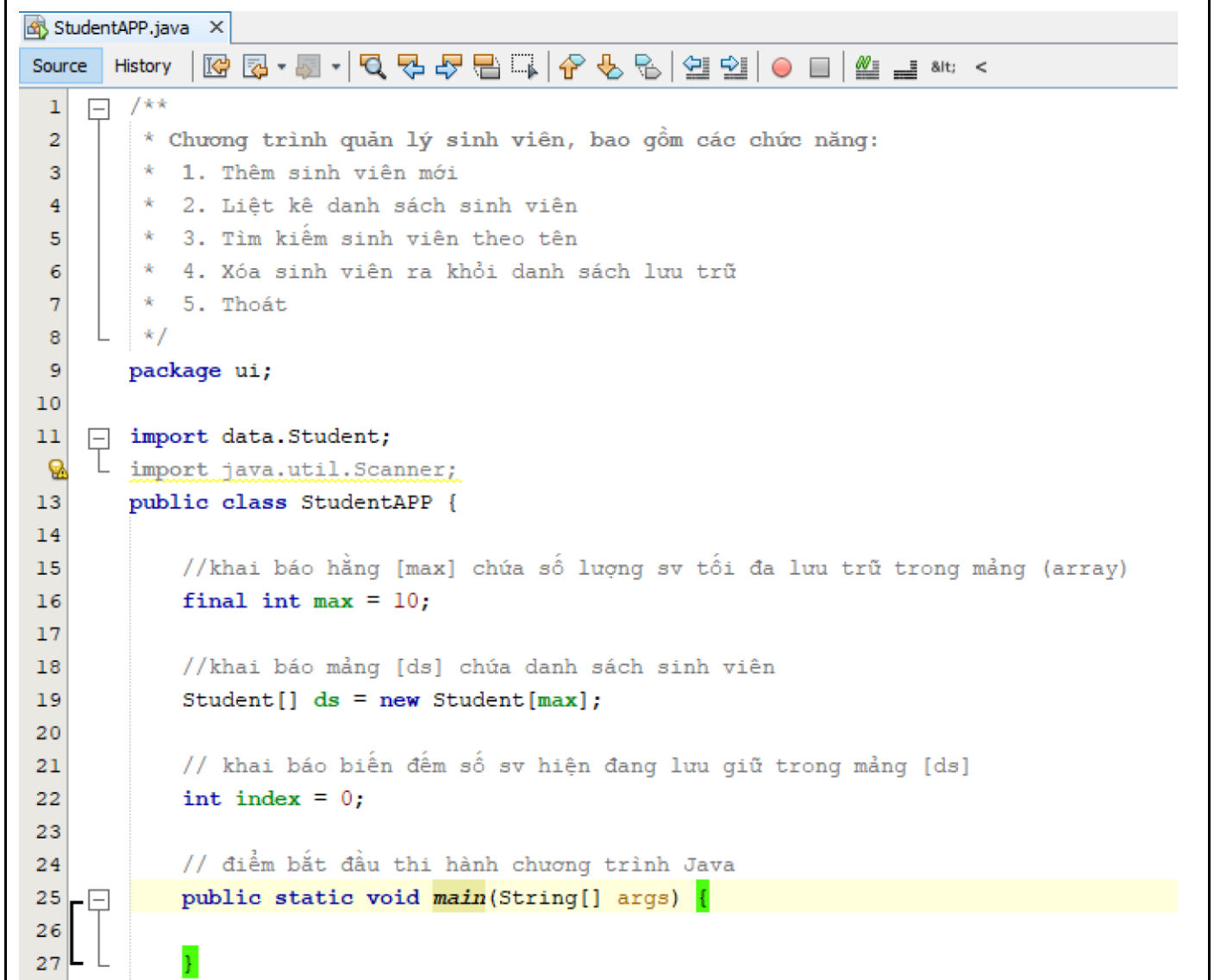
```

- Tạo java main class **StudentAPP**, trong package **ui**, quản lý thông tin học viên bao gồm các chức năng:
 1. Thêm sinh viên mới
 2. Liệt kê danh sách sinh viên
 3. Tìm kiếm sinh viên theo tên
 4. Xóa sinh viên ra khỏi danh sách lưu trữ
 5. Thoát

Thực hiện:

- Trong project **d05_package**, tạo package **ui**
- Trong package **ui**, tạo Java MainClass **StudentAPP**

- Viết code cho class **StudentAPP**



```
1  /**
2   * Chương trình quản lý sinh viên, bao gồm các chức năng:
3   * 1. Thêm sinh viên mới
4   * 2. Liệt kê danh sách sinh viên
5   * 3. Tìm kiếm sinh viên theo tên
6   * 4. Xóa sinh viên ra khỏi danh sách lưu trữ
7   * 5. Thoát
8   */
9  package ui;
10
11  import data.Student;
12  import java.util.Scanner;
13  public class StudentAPP {
14
15      //khai báo hằng [max] chứa số lượng sv tối đa lưu trữ trong mảng (array)
16      final int max = 10;
17
18      //khai báo mảng [ds] chứa danh sách sinh viên
19      Student[] ds = new Student[max];
20
21      // khai báo biến đếm số sv hiện đang lưu giữ trong mảng [ds]
22      int index = 0;
23
24      // điểm bắt đầu thi hành chương trình Java
25      public static void main(String[] args) {
26
27      }
```

```

28
29 //thêm 1 sinh viên mới vào mảng [ds]
30 public void add(){
31
32 }
33 //in danh sách sinh viên
34 public void display(){
35
36 }
37 //tìm và in ra thông tin sinh viên theo tên
38 public void search(String tenSV){
39
40 }
41 // tìm và xóa sinh viên ra khỏi mảng theo mã số
42 public void remove(String ms){
43
44 }
45 }

```

- Code chi tiết cho hàm **add()** : thêm sinh viên mới vào mảng ds

```

29 //thêm 1 sinh viên mới vào mảng [ds]
30 public void add(){
31     //kiểm tra mảng [ds] có còn chỗ trống để thêm sv mới ?
32     if(index == max) {
33         System.out.println(" >> Out of Memory !!!");
34         return;
35     }
36
37     //tạo biến đối tượng [sv] kiểu Student
38     Student sv = new Student();
39     sv.input(); //nhập dữ liệu cho [sv]
40     ds[index] = sv; //lưu [sv] vào mảng [ds] tại vị trí [index]
41     index++; //cập nhập [index] (biến đếm số sv trong [ds])
42 }

```

- Code chi tiết cho hàm **display()** : in toàn bộ danh sách sinh viên trong mảng ds ra màn hình

```

44 //in danh sách sinh viên - mỗi sv trên 1 dòng
45 public void display(){
46     //kiểm tra mảng [ds] có dữ liệu chưa ?
47     if(index == 0) {
48         System.out.println(" >> He thống chưa có dữ liệu !!!");
49         return;
50     }
51
52     System.out.println("\n >> Danh sách Sinh Viên << ");
53     for (int i = 0; i < index; i++) {
54         System.out.println(ds[i]);
55     }
56 }

```

- Code chi tiết cho hàm **search(String tenSV)** :

```

58      //tìm và in ra thông tin sinh viên theo tên
59      public void search(String tenSV) {
60          //kiểm tra mảng [ds] có dữ liệu chưa ?
61          if(index == 0) {
62              System.out.println(" >> He thống chưa có dữ liệu !!!");
63              return;
64          }
65
66          System.out.println("\n >> Danh sách Sinh Viên << ");
67          int cntTen=0;    // biến đếm số sv tìm được - có tên [tenSV] theo yêu cầu
68
69          for (int i = 0; i < index; i++) {
70              if(ds[i].name.contains(tenSV)){
71                  ds[i].output();
72                  cntTen++;
73              }
74          }
75
76          if(cntTen==0){
77              System.out.println(" >> Không tìm thấy sinh viên có tên " + tenSV);
78          }
79      }

```

- Code chi tiết cho hàm **remove(String ms)** : tìm và xóa sinh viên trong mảng ds theo mã số yêu cầu

```

116     // tìm và xóa sinh viên ra khỏi mảng theo mã số
117     public void remove(String ms) {
118         //kiểm tra mảng [ds] có dữ liệu chưa ?
119         if (index == 0) {
120             System.out.println(" >> He thống chưa có dữ liệu !!!");
121             return;
122         }
123
124         for (int i = 0; i < index; i++) {
125             if (ds[i].id.equals(ms)) {
126                 /* tìm thấy sv có mã số muốn xóa :
127                  xóa sv ra khỏi danh sách bằng cách tịnh tiến các sv ở vị trí
128                  sau lên trên 1 dòng
129                 */
130                 for (int k = i; k < index - 1; k++) {
131                     ds[k] = ds[k + 1];
132                 }
133
134                 index--;    //cập nhập [index] (biến đếm số sv trong [ds])
135                 System.out.println(" >> Đã xóa sinh viên có mã số: " + ms);
136                 return;    //kết thúc hàm
137             }
138         }
139
140         System.out.println(" >> Không tìm thấy sinh viên có mã số " + ms);
141     }

```

- Code chi tiết cho hàm **main(String[] args)** :

```
25 // điểm bắt đầu thi hành chương trình Java
26 public static void main(String[] args) {
27     StudentAPP app = new StudentAPP();
28     Scanner sc = new Scanner(System.in);
29     String op = "";
30     while (true) {
31         System.out.println("\n   === MENU === ");
32         System.out.println("1. Them Sinh vien moi");
33         System.out.println("2. Xem danh sach Sinh Vien");
34         System.out.println("3. Tim kiem Sinh Vien theo ten");
35         System.out.println("4. Xoa Sinh Vien theo ma so");
36         System.out.println("5. Thoat");
37
38         System.out.print(" Vui long chon chuc nang [1-5]: ");
39         op = sc.nextLine().trim();
40
41         switch (op) {
42             case "1":
43                 app.add();
44                 break;
45             case "2":
46                 app.display();
47                 break;
48             case "3":
49                 System.out.print(">> nhap ten sinh vien muon tim: ");
50                 String ten = sc.nextLine().trim();
51                 app.search(ten);
52                 break;
53             case "4":
54                 System.out.print(">> nhap ma so sinh vien muon xoa: ");
55                 String maso = sc.nextLine().trim();
56                 app.remove(maso);
57                 break;
58             case "5":
59                 return; //ket thuc ham main -> dung chương trình
60         }
61     }
62 }
```

- Bấm Shift+F6 để biên dịch và chạy thử chương trình

Assignment

Viết chương trình Java quản lý danh sách nhân viên, bao gồm các chức năng thêm mới, tìm kiếm và in toàn bộ danh sách nhân viên trong công ty.

Hướng dẫn :

- a. Tạo project **ass5**
- b. Tạo package **data**
- c. Tạo Java Class **Employee**, trong package **data**, để mô tả cấu trúc 1 nhân viên như sau:
 - i. Dữ liệu (fields – variable): **id, name, position, baseSalary (100-2000USD)**
 - ii. Hàm dựng (constructors):
 1. Không tham số **Employee()**
 2. Có tham số **Employee(String, String, String, int)**
 - iii. Phương thức:
 1. void **input()**: nhập dữ liệu cho các thuộc tính của nhân viên
 2. void **print()**: xuất thông tin chi tiết của đối tượng nhân viên
- d. Tạo Java Main Class **EmployeeTest**, trong package **app**, thực hiện chức năng quản lý danh sách nhân viên :
 - i. Dữ liệu (fields – variable):
empList (là 1 **ArrayList** chứa các đối tượng nhân viên),
 - ii. Hàm dựng không tham số:
khởi tạo empList
 - iii. Phương thức (Methods):
 1. void **add()**: thêm 1 nhân viên mới vào danh sách ds.
Lưu ý: kiểm tra nếu mã số bị trùng sẽ báo lỗi và từ chối thêm mới.
 2. void **display()**: In toàn bộ bảng danh sách nhân viên, và in thông báo lỗi nếu hệ thống chưa có dữ liệu
 3. void **display(int amt)**: In bảng danh sách nhân viên có mức lương trên **amt** USD. Chương trình sẽ thông báo lỗi nếu hệ thống chưa có dữ liệu hoặc không tìm thấy nhân viên nào có mức lương trên **amt** USD
 4. void **display(String empName)**: Tìm và in ra bảng danh sách các nhân viên có tên chứa chuỗi **empName**. Chương trình sẽ thông báo lỗi nếu hệ thống chưa có dữ liệu hoặc không tìm thấy nhân viên nào theo yêu cầu trên.
 5. void **menu()**: xây dựng hệ thống thực đơn cho phép thêm, tìm kiếm và in danh sách các nhân viên theo yêu cầu nói trên.
 6. static void **main ()**: tạo 1 đối tượng kiểu EmployeeTest, gọi phương thức menu() trên đối tượng này để thi hành chương trình.
- e. Biên dịch và thi hành chương trình