

Session 7**Methods and Access Specifiers****Phần I - Thực hiện trong 120 phút****1.1 Mục tiêu**

- ✓ Hiểu vững cú pháp khai báo phương thức.
- ✓ Hiểu rõ các bộ từ truy cập, khai báo và phạm vi sử dụng.
- ✓ Hiểu vững cách truyền tham số, những kiểu tham số trong khai báo phương thức.
- ✓ Viết phương thức có chứa tham số trả về.
- ✓ Xây dựng thói quen viết java-doc cho phương thức để giải thích ý nghĩa, tham số sử dụng, giá trị trả về.
- ✓ Hiểu và nắm vững ý nghĩa hàm nạp chồng (overloading).

1.2 Thực hiện

Bài thực hành 1: Viết một chương trình Java chuyển đổi nhiệt độ qua lại giữa độ C và độ F theo công thức:

$$^{\circ}\text{F} = (^{\circ}\text{C} \times 1.8) + 32$$

$$^{\circ}\text{C} = (^{\circ}\text{F} - 32) / 1.8$$

Yêu cầu: viết 1 class ConvertTemperature có 1 thuộc tính private là temperNum kiểu float và

- 1 hàm private tên convertC2F() trả về kiểu float.
- 1 hàm private tên convertF2C() trả về kiểu float.
- 1 hàm public tên convert() nhận vào 2 tham số:
 - nhietDo - float: nhiệt độ cần đổi
 - isToC - boolean: true-chuyển từ C sang F, false()-chuyển từ F sang C.

Phương thức convert() cần viết chi tiết java-doc. Viết class BaiThucHanh01 có hàm main() để kiểm thử.

Bước 1: Viết class ConvertTemperature có thuộc tính và các hàm như yêu cầu.

```
6 package demo.jp1.lab05;
7
8 /**
9  *
10  * @author minhvt
11  */
12 public class ConvertTemperature {
13     private float temperNum;
14
15     private float convertC2F() {
16         return 0;
17     }
18
19     private float convertF2C() {
20         return 0;
21     }
22
23     public void convert(float nhietDo, boolean isToC) {
24     }
25 }
26
27
28
```

Bước 2: Viết nội dung cho 2 hàm chuyển đổi nhiệt độ theo công thức.

```
11 /**
12  */
13 public class ConvertTemperature {
14     private float temperNum;
15
16     private float convertC2F() {
17         return (float) (temperNum * 1.8 + 32);
18     }
19
20     private float convertF2C() {
21         return (float) ((temperNum - 32) / 1.8);
22     }
23
24     public void convert(float nhietDo, boolean isToC) {
25     }
26 }
27
28
```

Bước 3: Viết nội dung cho hàm convert, viết chi tiết java-doc cho hàm.

```

15
16     private float convertC2F() {
17         return (float) (temperNum * 1.8 + 32);
18     }
19
20     private float convertF2C() {
21         return (float) ((temperNum - 32) / 1.8);
22     }
23
24     /**
25      * @param nhietDo chỉ số nhiệt độ muốn chuyển đổi
26      * @param isToC True nếu muốn chuyển đổi từ F sang C. False nếu muốn từ C
27      * sang F
28      * @return void
29      * @see Lệnh in ra màn hình kết quả chuyển đổi
30      */
31     public void convert(float nhietDo, boolean isToC) {
32         this.temperNum = nhietDo;
33         if (isToC) {
34             System.out.println("Nhiệt độ " + nhietDo + " C = " + convertC2F() + " F");
35         } else {
36             System.out.println("Nhiệt độ " + nhietDo + " F = " + convertF2C() + " C");
37         }
38     }
39 }
40

```

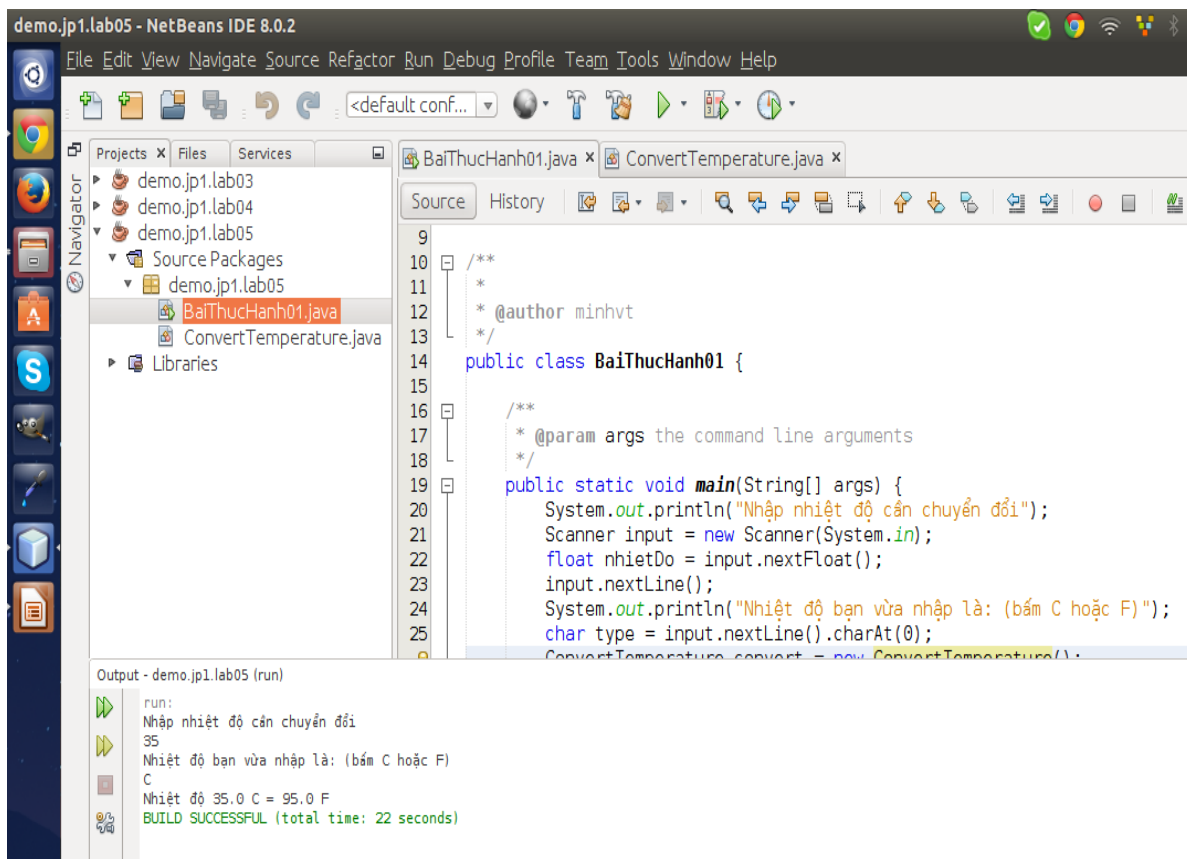
Bước 4: Trong hàm main() của class BaiThucHanh01 khởi tạo đối tượng ConvertTemperature và gọi hàm chuyển đổi.

```

14     public class BaiThucHanh01 {
15
16         /**
17          * @param args the command line arguments
18          */
19         public static void main(String[] args) {
20             System.out.println("Nhập nhiệt độ cần chuyển đổi");
21             Scanner input = new Scanner(System.in);
22             float nhietDo = input.nextFloat();
23             input.nextLine();
24             System.out.println("Nhiệt độ bạn vừa nhập là: (bấm C hoặc F)");
25             char type = input.nextLine().charAt(0);
26             ConvertTemperature convert = new ConvertTemperature();
27             switch (type) {
28                 case 'C':
29                     convert.convert(nhietDo, true);
30                     break;
31                 case 'F':
32                     convert.convert(nhietDo, false);
33                     break;
34             }
35         }
36     }
37 }
38

```

Bước 5: Chạy chương trình để kiểm thử.



Bài thực hành 2: Viết một chương trình Java tính toán diện tích, chu vi của các hình chữ nhật, tròn, tam giác. Các hàm chức năng tính toán sau đó trả về dữ liệu. Viết hàm cùng tên nhưng khác dữ liệu truyền vào tương ứng với các hình để hiểu rõ tính nạp chồng - overloading.

Bước 1: Viết class Shape có các thuộc tính static kiểu int (để xác nhận hình muốn xử lý), một thuộc tính kiểu int khác tên là shapeCode để lưu trữ lựa chọn của người dùng.

```
6 package demo.jp1.lab05;
7
8 /**
9  *
10  * @author minhvt
11  */
12 public class Shape {
13
14     public static final int SHAPE_RECTANGLE = 1;
15     public static final int SHAPE_CIRCLE = 2;
16     public static final int SHAPE_TRIANGLE = 3;
17
18     public int shareCode = 0;
19 }
20
```

Bước 2: Viết các hàm `tinhDienTich()`, `tinhChuVi()` cho các hình. Viết Java-doc cho các phương thức. (Copy code vào hết 1 file và thêm chú thích vào đó)

```
20 public void tinhDienTich(float width, float height) {
21 }
22
23
24 public void tinhDienTich(float radius) {
25 }
26
27
28 public void tinhDienTich(float height, double bottom) {
29 }
30
31
32 public void tinhChuVi(float width, float height) {
33 }
34
35
36 public void tinhChuVi(float radius) {
37 }
38
39
40 public void tinhChuVi(float edge1, float edge2, float edge3) {
41 }
42 }
43
```

demo.jp1.lab05.Shape >

```

20  /**
21  * @param width chiều rộng
22  * @param height chiều dài
23  * @return void
24  * @see Diện tích hình chữ nhật - vuông
25  */
26  public void tinhDienTich(float width, float height) {
27  }
28
29
30  /**
31  * @param radius Bán kính của hình tròn
32  * @return void
33  * @see Diện tích hình tròn
34  */
35  public void tinhDienTich(float radius) {
36  }
37
38
39  /**
40  * @param height chiều cao tam giác
41  * @param bottom chiều dài cạnh đáy
42  * @see Diện tích hình tam giác
43  */
44  public void tinhDienTich(float height, double bottom) {
45  }
46
47
48  /**

```

Find: | Previous Next |

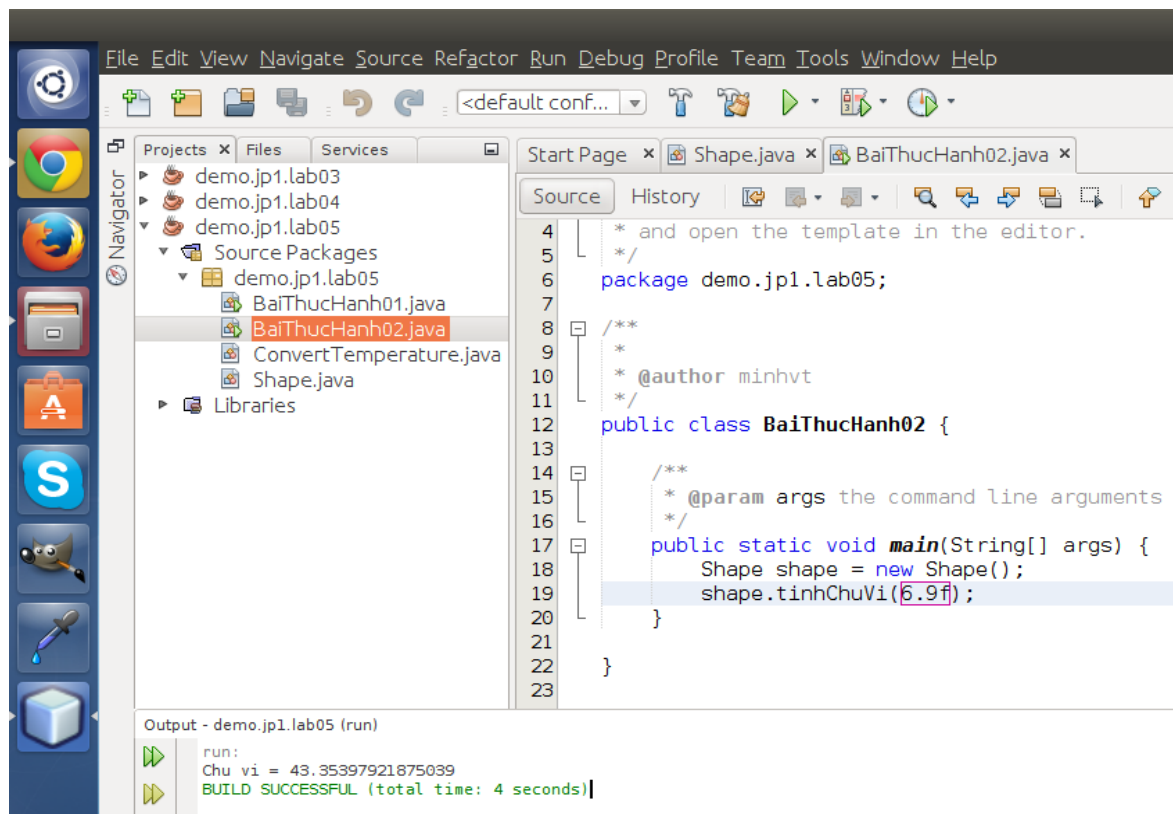
demo.jp1.lab05.Shape >

```

48  /**
49  * @param width Chiều dài
50  * @param height Chiều rộng
51  * @see Chu vi của hình chữ nhật - vuông
52  */
53  public void tinhChuVi(float width, float height) {
54  }
55
56
57  /**
58  * @param radius Bán kính
59  * @see Chu vi của hình tròn
60  */
61  public void tinhChuVi(float radius) {
62  }
63
64
65  /**
66  * @param edge1 Cạnh thứ 1
67  * @param edge2 Cạnh thứ 2
68  * @param edge3 Cạnh thứ 3
69  * @see Chu vi của tam giác
70  */
71  public void tinhChuVi(float edge1, float edge2, float edge3) {
72  }
73
74

```

Bước 3: viết class BaiThucHanh02 có hàm main(), khởi tạo đối tượng và kiểm thử.



Bài thực hành 3: Viết một class tên là SinhVien để nhập thông tin cơ bản (private) của sinh viên gồm:

```
int rollNo;
String name;
String address;
float mark;
```

Viết các hàm get-set cho các thuộc tính trên. Viết 4 constructor cho class này (mặc định và có tham số) lần lượt như sau:

1. Mặc định
2. rollNo + name
3. rollNo + name + address
4. rollNo + name + address + mark

Các constructor cần viết java-doc. Tạo class BaiThucHanh03 có hàm main() để kiểm thử.

Bước 1: Viết class SinhVien với các thuộc tính như yêu cầu.

```

5  L  */
6      package demo.jp1.lab05;
7
8  L  /**
9      *
10     * @author minhvt
11     */
12     public class SinhVien {
13
14         private int rollNo;
15         private String name;
16         private String address;
17         private float mark;
18     }
19

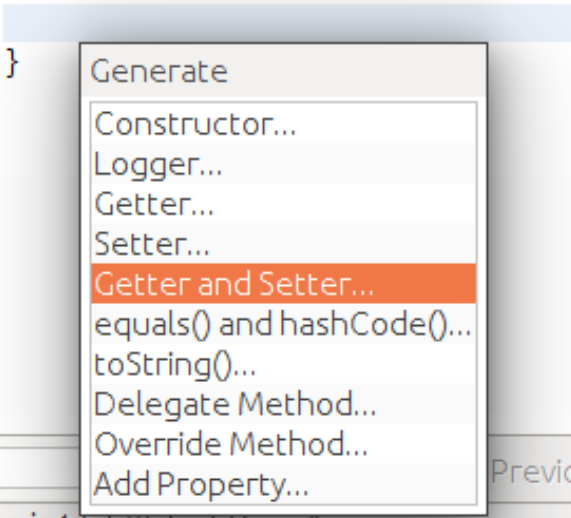
```

Bước 2:Viết các hàm get-set

```

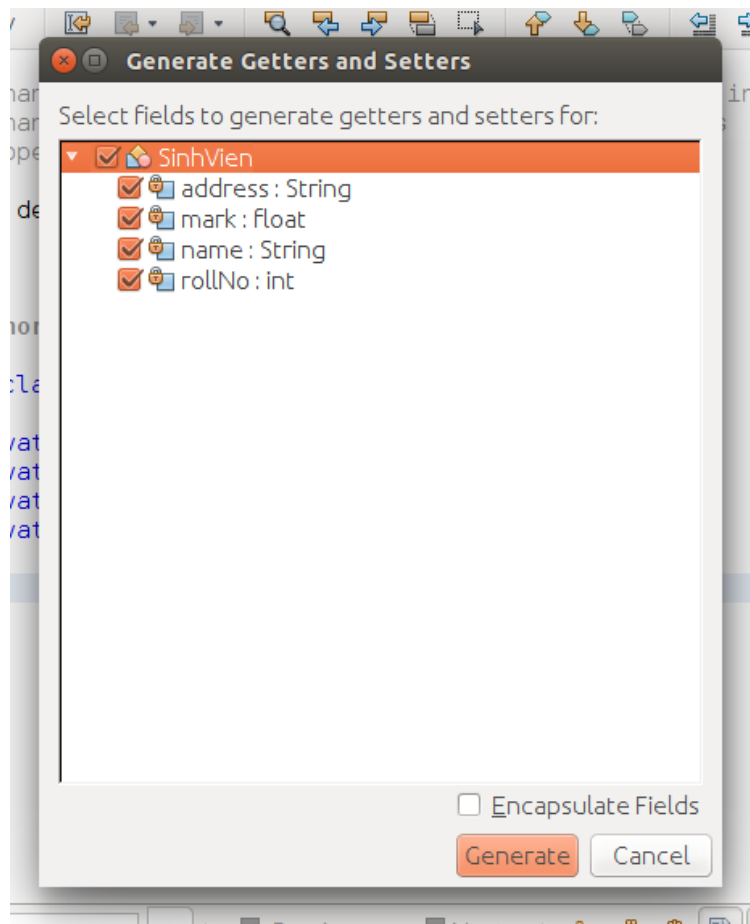
11  L  */
12     public class SinhVien {
13
14         private int rollNo;
15         private String name;
16         private String address;
17         private float mark;
18
19
20     }
21

```



Find: Previous

demo.jp1.lab05.SinhVien >



```
18
19 public int getRollNo() {
20     return rollNo;
21 }
22
23 public void setRollNo(int rollNo) {
24     this.rollNo = rollNo;
25 }
26
27 public String getName() {
28     return name;
29 }
30
31 public void setName(String name) {
32     this.name = name;
33 }
34
35 public String getAddress() {
36     return address;
37 }
38
39 public void setAddress(String address) {
40     this.address = address;
41 }
42
43 public float getMark() {
```

Find: Previous Next

demo.jp1.lab05.SinhVien > getRollNo >

Bước 3: Viết các constructor

```

46
47     public void setMark(float mark) {
48         this.mark = mark;
49     }
50
51     public SinhVien() {
52     }
53
54     public SinhVien(int rollNo, String name) {
55         this.rollNo = rollNo;
56         this.name = name;
57     }
58
59     public SinhVien(int rollNo, String name, String address) {
60         this.rollNo = rollNo;
61         this.name = name;
62         this.address = address;
63     }
64
65     public SinhVien(int rollNo, String name, String address, float mark) {
66         this.rollNo = rollNo;
67         this.name = name;
68         this.address = address;
69         this.mark = mark;
70     }
71
72 }
73

```

Find: Previous Next

demo.jp1.lab05.SinhVien

Bước 4: Viết java-doc cho các constructor

```

56
57     /**
58      * @param rollNo Mã sinh viên
59      * @param name Tên sinh viên
60      */
61     public SinhVien(int rollNo, String name) {
62         this.rollNo = rollNo;
63         this.name = name;
64     }
65
66     /**
67      * @param rollNo Mã sinh viên
68      * @param name Tên sinh viên
69      * @param address Địa chỉ của sinh viên
70      */
71     public SinhVien(int rollNo, String name, String address) {
72         this.rollNo = rollNo;
73         this.name = name;
74         this.address = address;
75     }
76
77     /**
78      * @param rollNo Mã sinh viên
79      * @param name Tên sinh viên
80      * @param address Địa chỉ của sinh viên
81      * @param mark Điểm của sinh viên
82      */
83     public SinhVien(int rollNo, String name, String address, float mark) {
84         this.rollNo = rollNo;
85         this.name = name;
86         this.address = address;
87         this.mark = mark;
88     }
89

```

Find: Previous Next

demo.jp1.lab05.SinhVien

Bước 5: Viết class BaiThucHanh03 có hàm main() để kiểm thử.

Phần II - Bài tập tự làm

Ghi chú: Giảng viên hướng dẫn cơ bản một cách sơ lược về cách khai báo mảng đối tượng để sinh viên có thể code được bài tập, chưa yêu cầu làm chi tiết truy cập hay thao tác tới đối tượng, chỉ cần vận dụng kiến thức của buổi này là được.

I. Tạo lớp Product trong gói product và có các thuộc tính:

Field	Description
private String prodId;	Mã sản phẩm
private String prodName;	Tên sản phẩm
private String manufacturer;	Nhà sản xuất
private float producerPrice;	Giá sản xuất

Tạo 2 constructors, các phương thức set/get cho các thuộc tính của lớp.

Cài đặt các phương thức:

```
public void input();  
public void display();  
public float caculateSalePrice();
```

Phương thức caculateSalePrice() tính giá bán cho sản phẩm theo cách tính:

$\text{sale_price} = \text{producerPrice} + 5\% * \text{producerPrice}$ (giá bán bằng giá sản xuất cộng với 5 phần trăm giá sản xuất – trong đó 5 phần trăm là tiền lãi của nhà sản xuất).

II. Tạo lớp VietNamImportPrice trong gói vietnam, kế thừa lớp Product và thêm vào các thuộc tính:

```
private float taxImported;    -> Thuế nhập khẩu
```

Tạo 2 constructors, các phương thức set/get cho các thuộc tính thêm vào trong lớp.

Cài đặt tất cả các phương thức trong lớp Product. Trong đó phương thức caculateSalePrice() được sử dụng để tính giá bán cho sản phẩm được bán ở Việt Nam:

*(Giá bán bằng giá nhập khẩu (giá này bằng giá mà sale_price = imported_price + taxImported*imported_price + 10% * (imported_price + taxImported*imported_price) nhà sản xuất đưa ra) cộng với thuế nhập khẩu + 5 phần trăm giá nhập và thuế cộng gộp (đây sẽ là tiền lãi cho công ty nhập hàng hóa về Việt Nam)).*

III. Tạo lớp ThailandImportPrice trong gói thailand, kế thừa lớp Product và thêm vào các thuộc tính:

private float taxImported; -> Thuế nhập khẩu

private float importPriceSupport; -> Giá hỗ trợ nhập khẩu

Tạo 2 constructors, các phương thức set/get cho các thuộc tính được thêm vào.

Cài đặt tất cả các phương thức trong lớp Product. Trong đó phương thức caculateSalePrice() được sử dụng để tính giá bán cho sản phẩm được bán ở Thailand:

sale_price = imported_price + taxImported*imported_price -
importPriceSupport*imported_price + 10% * (imported_price +
taxImported*imported_price - importPriceSupport*imported_price)

(Giá bán bằng giá nhập khẩu (giá này bằng giá mà nhà sản xuất đưa ra) cộng với thuế nhập khẩu trừ đi giá hỗ trợ nhập khẩu do chính phủ đưa ra + 5 phần trăm giá cộng gộp các khoản trên (đây sẽ là tiền lãi cho công ty nhập hàng hóa về Thái Lan)).

IV. Tạo lớp GeneralManager trong gói vietnam và có menu sau:

1. Nhập thông tin cho n sản phẩm được nhập khẩu vào VietNam
2. Nhập thông tin cho n sản phẩm được nhập khẩu vào Thailand
3. Hiển thị tất cả thông tin các sản phẩm ở Việt Nam cũng như ở Thái Lan
4. Sắp xếp các sản phẩm tăng dần theo tên sản phẩm và hiển thị danh sách sau khi sắp xếp.
5. Tính toán giá bán sản phẩm ở Việt Nam và ở Thái Lan sau đó hiển thị danh sách sản phẩm với thông tin và cả giá bán.
6. Tìm kiếm sản phẩm theo giá bán và hiển thị kết quả tìm kiếm
7. Kết thúc.

Xây dựng tất cả phương thức cho các hàm ở trên.