

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

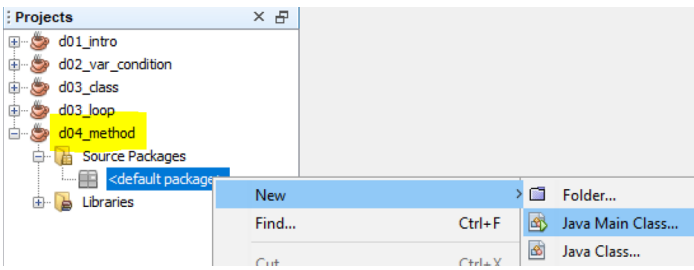
1. Lập trình phương thức, truyền tham số cho phương thức theo phương pháp giá trị, tham chiếu.
2. Sử dụng bộ từ truy xuất : public, protected, private

Bài tập 1:

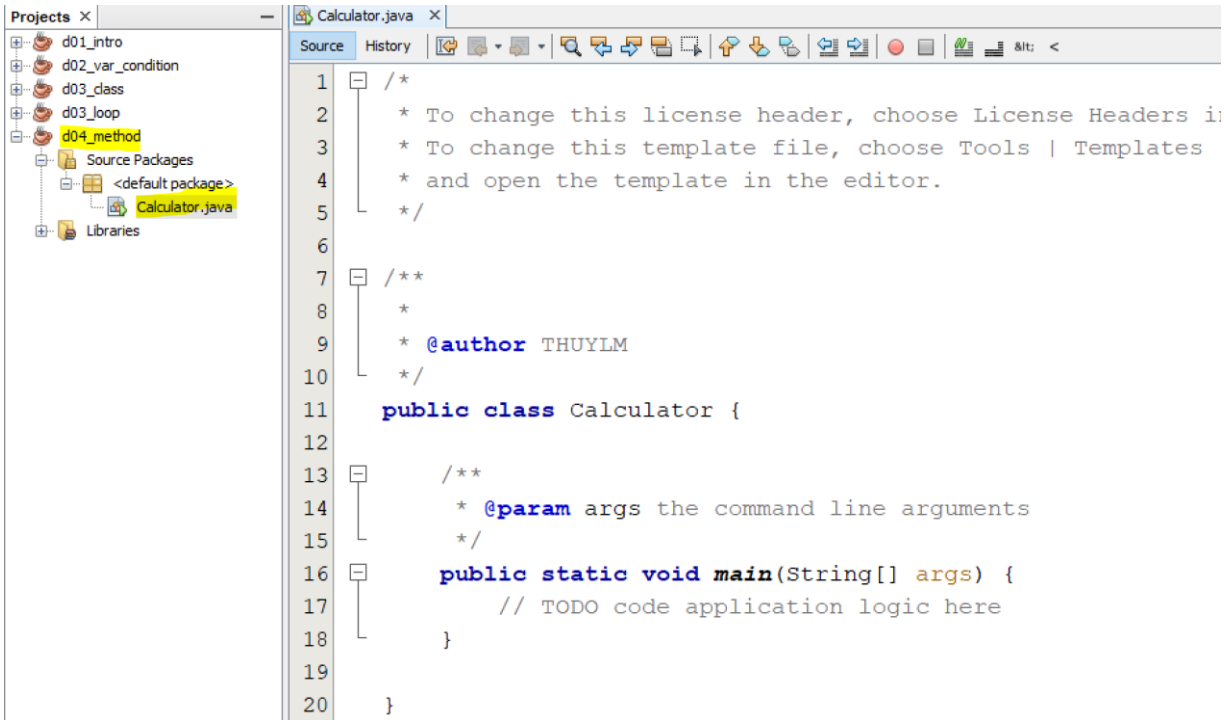
- Lập trình phương thức.
- Viết chương trình in kết quả các phép toán số học: cộng, trừ, nhân ... bằng cách viết các phương thức tương ứng.

HD thực hiện:

- Mở Netbean, Tạo project Java Application, đặt tên **d04_method** (nhớ bỏ check **Create Main Class**)
- Trong project **d04_method**, tạo Java Main Class **Calculator**



- Nhấp chuột phải trên **<default package>**, chọn **New**, chọn **Java Main Class**, gõ tên Class Name: **Calculator**, bấm [Finish]

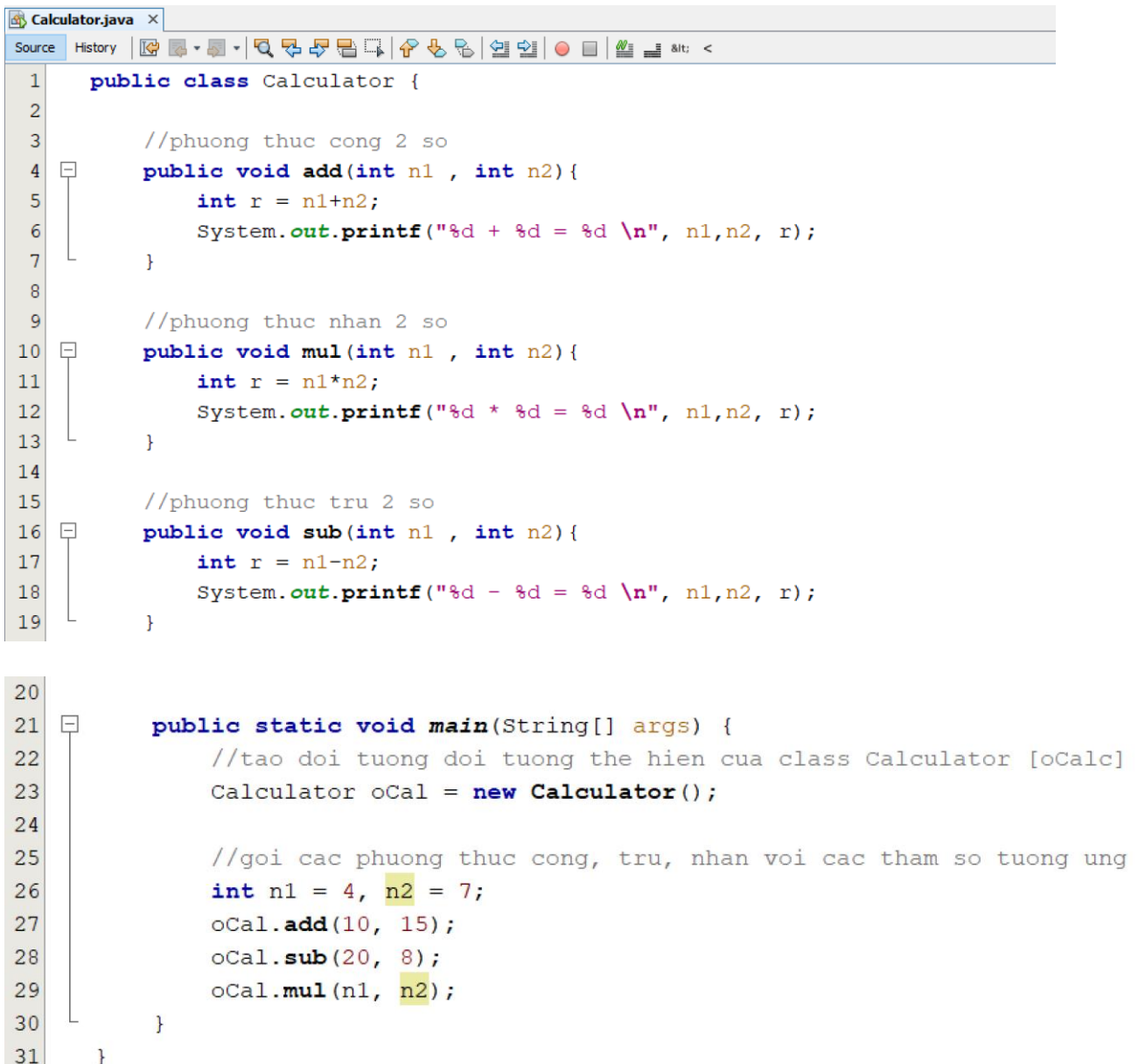


```

1  /*
2   * To change this license header, choose License Headers in
3   * To change this template file, choose Tools | Templates
4   * and open the template in the editor.
5   */
6
7  /**
8   *
9   * @author THUYLM
10  */
11  public class Calculator {
12
13      /**
14       * @param args the command line arguments
15       */
16      public static void main(String[] args) {
17          // TODO code application logic here
18      }
19
20  }

```

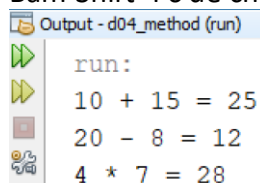
- Viết code cho các phương thức **add()**, **mul()**, **sub()** trong class **Calculator** như sau:



```

1  public class Calculator {
2
3      //phuong thuc cong 2 so
4      public void add(int n1 , int n2){
5          int r = n1+n2;
6          System.out.printf("%d + %d = %d \n", n1,n2, r);
7      }
8
9      //phuong thuc nhan 2 so
10     public void mul(int n1 , int n2){
11         int r = n1*n2;
12         System.out.printf("%d * %d = %d \n", n1,n2, r);
13     }
14
15     //phuong thuc tru 2 so
16     public void sub(int n1 , int n2){
17         int r = n1-n2;
18         System.out.printf("%d - %d = %d \n", n1,n2, r);
19     }
20
21     public static void main(String[] args) {
22         //tao doi tuong doi tuong the hien cua class Calculator [oCal]
23         Calculator oCal = new Calculator();
24
25         //goi cac phuong thuc cong, tru, nhan voi cac tham so tuong ung
26         int n1 = 4, n2 = 7;
27         oCal.add(10, 15);
28         oCal.sub(20, 8);
29         oCal.mul(n1, n2);
30     }
31 }
  
```

- Bấm Shift+F6 để chạy thử chương trình, kết quả sẽ xuất hiện trong màn hình **Output**



```

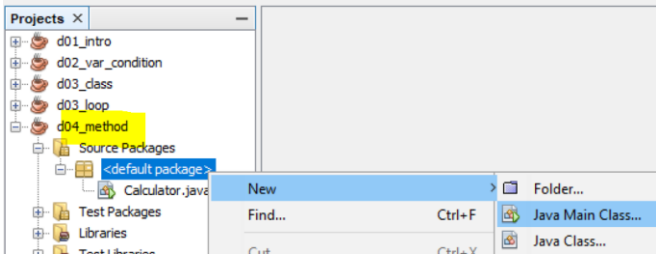
Output - d04_method (run) x
run:
10 + 15 = 25
20 - 8 = 12
4 * 7 = 28
  
```

Bài tập 2:

- Lập trình truyền tham số cho phương thức – **pass by value**

HD thực hiện:

- Trong project **d04_method**, tạo **Java Main Class DemoPassByValue**



- Nhấp chuột phải trên **<default package>**, chọn **New**, chọn **Java Main Class**, gõ tên Class Name: **DemoPassByValue**, bấm **[Finish]**

- Viết code như sau:

```

1  public class DemoPassByValue {
2
3      // p.thức có tham số n - argument by value
4      public void setVal(int n) {
5          n = n + 10;
6      }
7
8      public static void main(String[] args) {
9          // khai báo biến cục bộ
10         int n = 10;
11
12         // khai báo và khởi tạo 1 thể hiện của class [DemoPassByValue]
13         DemoPassByValue obj = new DemoPassByValue();
14
15         // gọi phương thức setVal() với tham số [num1]
16         obj.setVal(n);
17
18         // in và kiểm tra giá trị [n]
19         System.out.println("Value of [n] after invoking setVal() is " + n);
20     }
21 }
22

```

- Bấm Shift+F6 để chạy thử chương trình. Kết quả sẽ xuất hiện trong màn hình **Output**

```

Output - d04_method (run) x
run:
Value of [n] after invoking setVal() is 10
BUILD SUCCESSFUL (total time: 0 seconds)

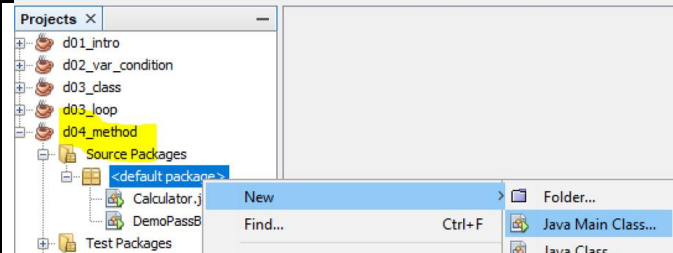
```

Bài tập 3:

- Lập trình truyền tham số cho phương thức – **pass by reference**
- Viết chương trình tính và in diện tích hình tròn

HD thực hiện:

- Trong project **d04_method**, tạo Java Main Class **DemoPassByRef**



- Nhấp chuột phải trên <default package>, chọn **New**, chọn **Java Main Class**, gõ tên Class Name: **DemoPassByRef**, bấm [Finish]

- Viết code như sau:

```

1
2  class Circle {
3      public double radius;    // bán kính
4      public double area;      // diện tích hình tròn
5  }
6
7  public class DemoPassByRef {
8
9      public void calcArea(Circle oCircle, double rad) {
10         oCircle.radius = rad;
11         oCircle.area = Math.PI * rad * rad;
12     }
13
14     public static void main(String[] args) {
15         Circle c = new Circle();
16         DemoPassByRef demo = new DemoPassByRef();
17
18         demo.calcArea(c, 4);
19         System.out.printf("Radius of the circle is %.2f \n", c.radius);
20         System.out.printf("Area of the circle is %.2f \n", c.area);
21     }
22 }
    
```

- Bấm Shift+F6 để chạy thử chương trình

```

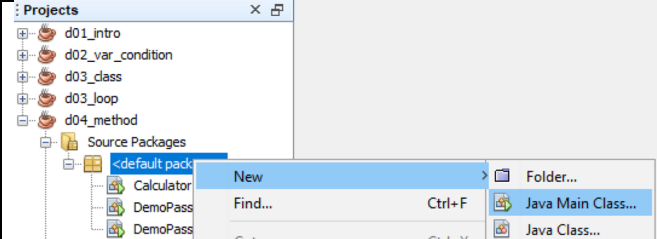
Output - d04_method (run) x
run:
Radius of the circle is 4.00
Area of the circle is 50.27
    
```

Bài tập 4:

- Lập trình truyền tham số cho phương thức – **Variable Argument**
- Viết chương trình in tổng của n-số nguyên.

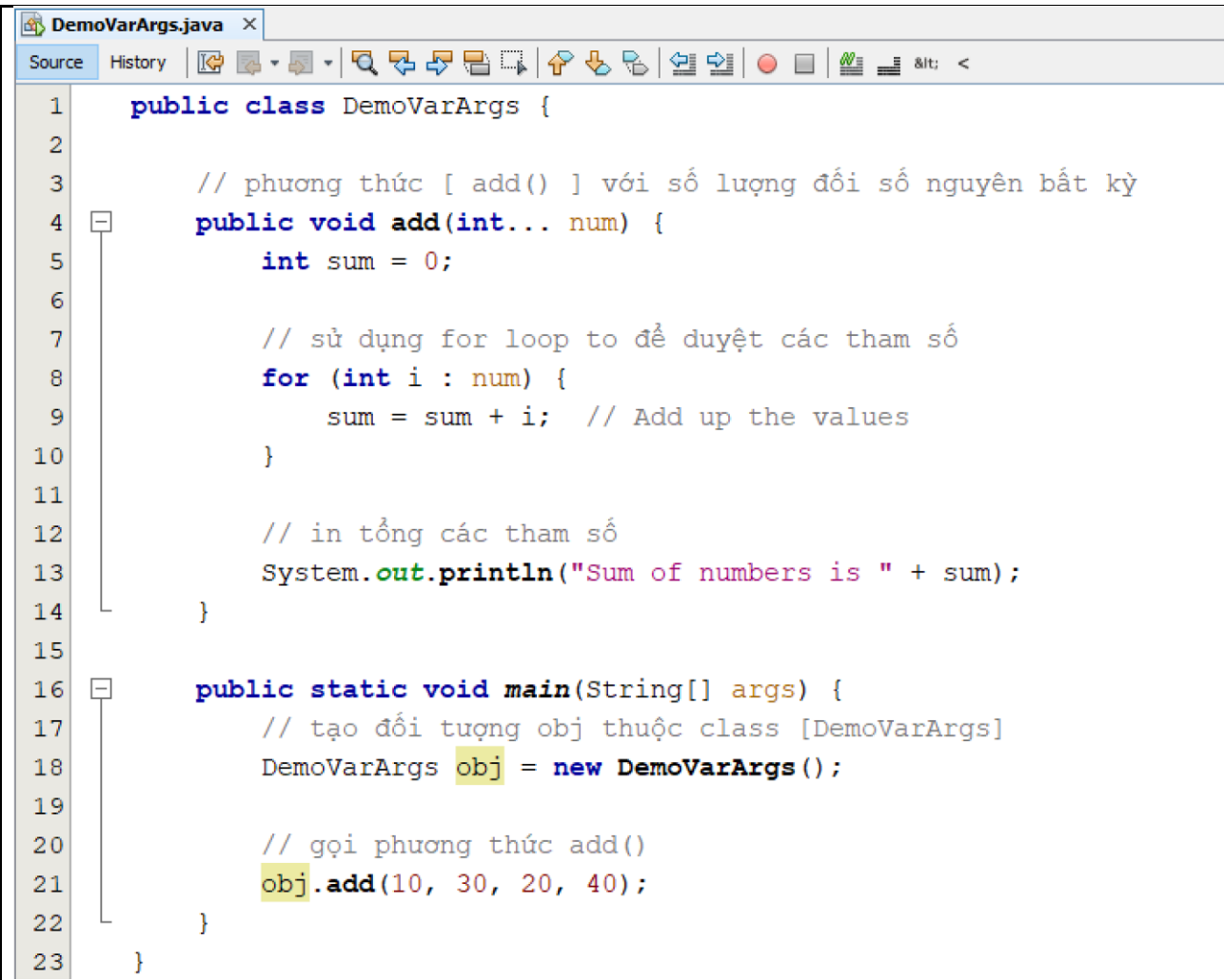
HD thực hiện:

- Trong project **d04_method**, tạo class **DemoVarArgs**



- Nhấp chuột phải trên **<default package>**, chọn **New**, chọn **Java Main Class**, gõ tên Class Name: **DemoVarArgs**, bấm **[Finish]**

- Viết code như sau:

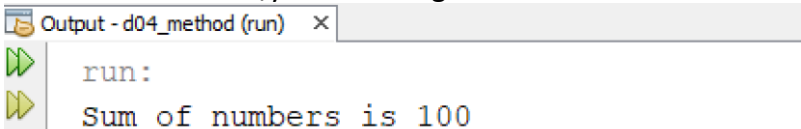


```

1  public class DemoVarArgs {
2
3      // phương thức [ add() ] với số lượng đối số nguyên bất kỳ
4      public void add(int... num) {
5          int sum = 0;
6
7          // sử dụng for loop to để duyệt các tham số
8          for (int i : num) {
9              sum = sum + i; // Add up the values
10         }
11
12         // in tổng các tham số
13         System.out.println("Sum of numbers is " + sum);
14     }
15
16     public static void main(String[] args) {
17         // tạo đối tượng obj thuộc class [DemoVarArgs]
18         DemoVarArgs obj = new DemoVarArgs();
19
20         // gọi phương thức add()
21         obj.add(10, 30, 20, 40);
22     }
23 }

```

- Bấm Shift+F6 để chạy thử chương trình.



```

Output - d04_method (run) x
run:
Sum of numbers is 100

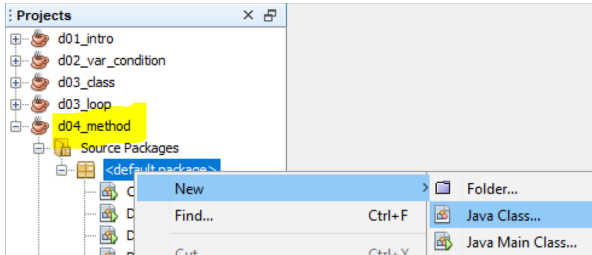
```

Bài tập 5:

- Sử dụng bộ từ truy xuất cho biến và phương thức trong class – **Access Specifier**
- Viết chương trình tạo và in ra thông tin của khách hàng.

HD thực hiện:

- Trong project **d04_method**, tạo class **Customer**



- Nhấp chuột phải trên **<default package>**, chọn **New**, chọn **Java Class**, gõ tên Class Name: **Customer**, bấm **[Finish]**

- Viết code như sau:

Customer.java
Source History

1

public class Customer {

2

//các thuộc tính của khách hàng

3

public String id, name;

4

protected int phone;

5

private int pinCode;

6

7

//hàm dựng ko tham số

8

public Customer() {

9

id = "C01";

10

name = "Nguyen A";

11

}

12

13

//hàm dựng có tham số

14

public Customer(String id, String name, int phone) {

15

this.id = id;

16

this.name = name;

17

this.phone = phone;

18

}

19

20

//lấy mã pin

21

public int getPincode() {

22

return pinCode;

23

}

24

25

//set mã pin

26

public void setPincode(int pinCode) {

27

this.pinCode = pinCode;

28

}

29

30

//lấy số phone

31

public int getPhone() {

32

return phone;

33

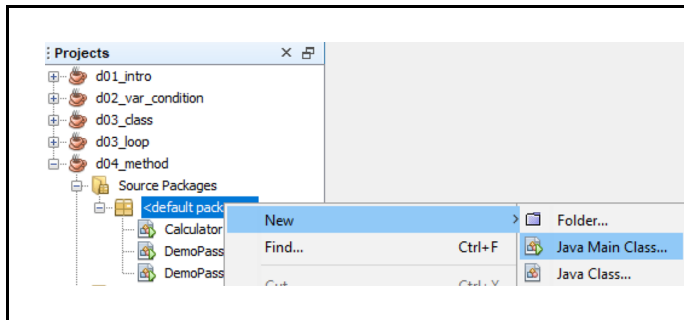
}

34

}

LAB GUIDE 4 – Methods & Access Specifiers

- Trong project **d04_method**, tạo Java Main Class **CustomerTest**, để tạo đối tượng Customer và in thông tin khách hàng



- Nhấp chuột phải trên "<default package>", chọn **New**, chọn **Java Main Class**, gõ tên Class Name: **EmployeeTest**, bấm **[Finish]**

- Viết code như sau:

```
CustomerTest.java
Source History
1  public class CustomerTest {
2
3      public static void main(String[] args) {
4          Customer c = new Customer("C01", "Bean", 8412456);
5          c.setPincode(303);
6
7          System.out.println("Thông tin khách hàng");
8          System.out.printf(" id: %s \n", c.id);
9          System.out.printf(" ten: %s \n", c.name);
10         System.out.printf(" phone: %d \n", c.getPhone());
11         System.out.printf(" pin code: %d \n", c.getPincode());
12     }
13
14 }
```

- Bấm Shift+F6 để chạy thử chương trình.

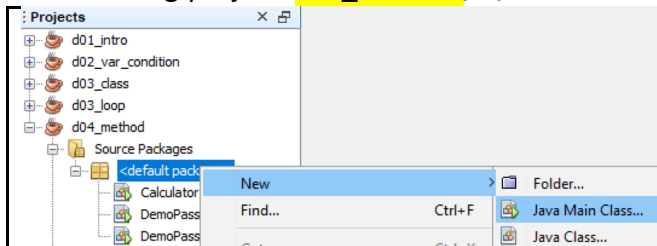
```
Output - d04_method (run)
run:
Thông tin khách hàng
id: C01
ten: Bean
phone: 8412456
pin code: 303
```


Bài tập 6:

- Lập trình phương thức overload.
- Viết chương trình tính tổng của 2 số nguyên, 3 số thực.

HD thực hiện:

- Trong project **d04_method**, tạo class **DemoOverload**



- Nhấp chuột phải trên **<default package>**, chọn **New**, chọn **Java Main Class**, gõ tên Class Name: **DemoOverload**, bấm **[Finish]**

- Viết code như sau:

```

1  public class DemoOverload {
2
3      //cong 2 so nguyen
4      void add(int a, int b){
5          System.out.printf("%d + %d = %d \n", a, b, a+b);
6      }
7
8      //cong 3 so thuc
9      void add(float a, float b, float c){
10         System.out.printf("%.2f + %.2f + %.2f = %.2f \n", a, b, c, a+b+c);
11     }
12
13     public static void main(String[] args) {
14         DemoOverload demo = new DemoOverload();
15         demo.add(10, 20);
16         demo.add(2.5f, 15, 34.83f);
17     }
18
19 }
    
```

- Bấm Shift+F6 để chạy thử chương trình.

```

Output - d04_method (run) x
run:
10 + 20 = 30
2.50 + 15.00 + 34.83 = 52.33
    
```


Assignment

Viết chương trình Java quản lý đối tượng bác sĩ theo hướng dẫn sau

1. Tạo project `ass3`

2. Tạo Java Class, `Doctor` (`Doctor.java`) với cấu trúc như sau :

- a. Fields dữ liệu : `docNo`, `name`, `specification` (string), `exp_years`, `salary` (int)
- b. Hàm dựng: khởi trị ban đầu cho các field dữ liệu, bao gồm hàm dựng
 - i. Không tham số `Doctor()`
 - ii. Có tham số `Doctor(String, String, String, int, int)`
- c. Phương thức:
 - i. void `input()` : nhập dữ liệu chi tiết của 1 bác sĩ . Yêu cầu kiểm tra tính hợp lệ của dữ liệu: `docNo`, `name`, `specification` -ko được để trống, `exp_years`, `salary` phải lớn hơn 0
 - ii. void `print()` : in các thông tin chi tiết của 1 bác sĩ ra màn hình
 - iii. void `print(String id)` : in ra hệ số cấp bậc của bác sĩ có mã số là `id`, dựa vào số năm kinh nghiệm (`exp_years`) như sau :
 - 'D' nếu số năm kn ≥ 15 năm
 - 'C' nếu số năm kn ≥ 10 năm và nhỏ hơn < 15 năm
 - 'B' nếu số năm kn ≥ 5 năm và nhỏ hơn < 10 năm
 - 'A' nếu số năm kn < 5 năm

Lưu ý, chương trình sẽ thông báo lỗi nếu mã số `id` ko đúng.

- iv. int `print(int days)`: tính và trả về thu nhập trong tháng của bác sĩ dựa vào số ngày làm việc (`days`) và số năm kinh nghiệm theo công thức sau :
`income = (salary * days)/24 + allowance` (phụ cấp cấp bậc) ,
với `allowance` = 1000 nếu số năm kn ≥ 15 năm
600 nếu số năm kn ≥ 10 năm và nhỏ hơn < 15 năm
300 nếu số năm kn ≥ 5 năm và nhỏ hơn < 10 năm
100 nếu số năm kn < 5 năm

3. Tạo lớp kiểm thử Java Main Class, `DoctorTest.java` , bao gồm 2 phương thức :

- a. `menu()` : tạo thực đơn để thực hiện các chức năng :
 1. Nhập thông tin bác sĩ,
 2. Xuất thông tin bác sĩ,
 3. In cấp bậc,
 4. Nhập số ngày làm việc và tính thu nhập,
 5. Thoát
- b. `main()` : gọi phương thức `menu()` để thi hành chương trình.

4. Biên dịch và thi hành chương trình.