데이터융합SW과
김규석 교수

JAVA

기본 프로그래밍 04

Objective

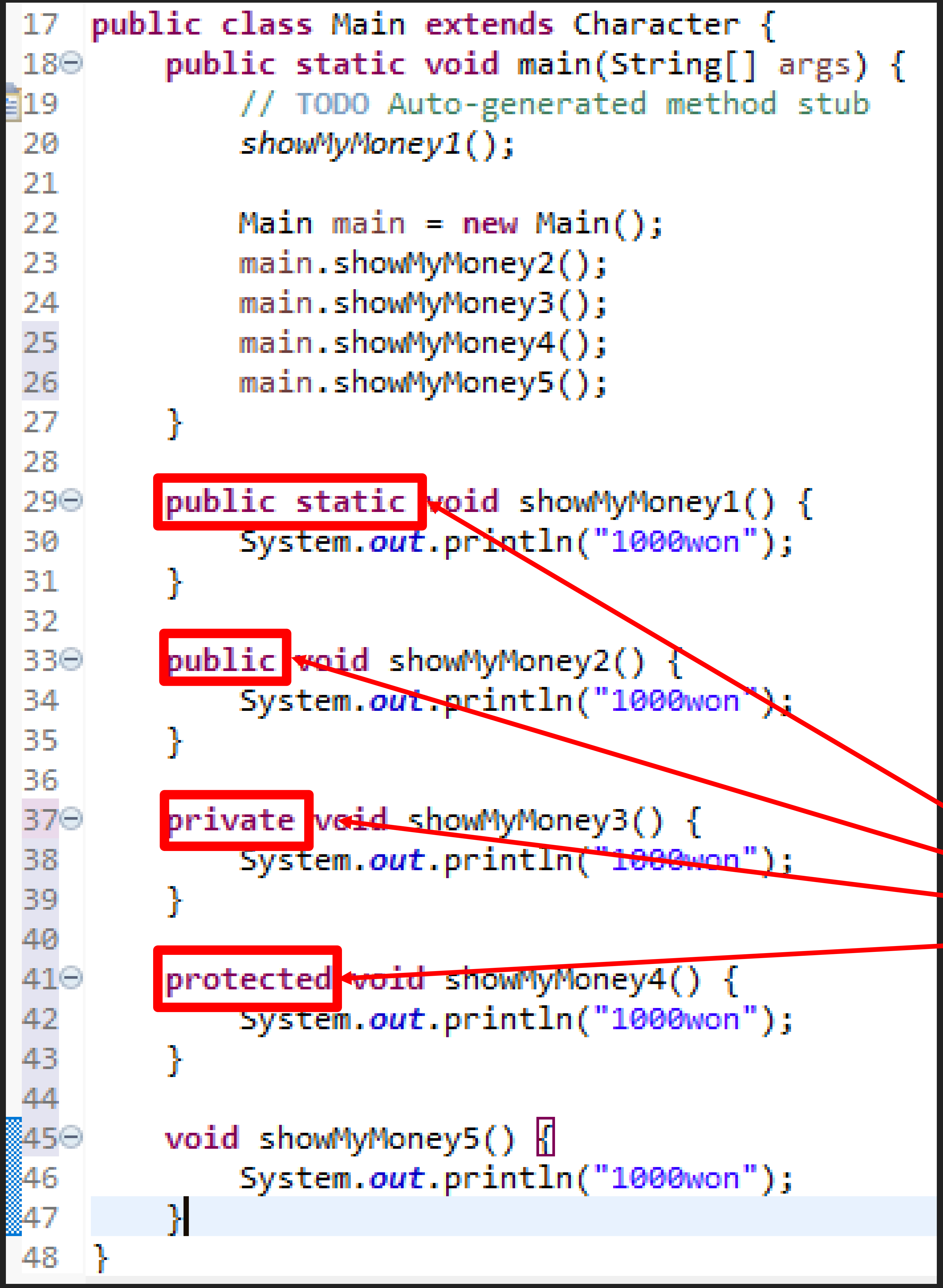
Method

- Calling a Method from the Same Class
- Parameter
- Declaring a static final variable
- Returning a Value
- Overloading
- Recursive Function

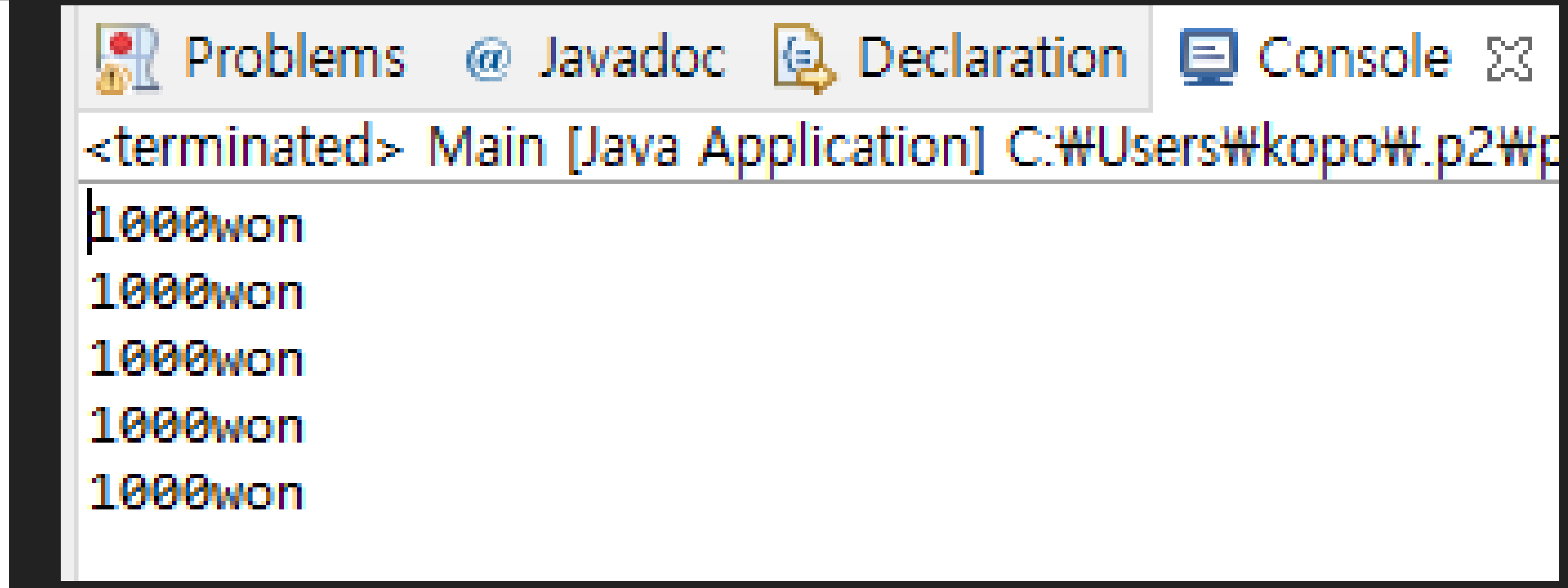
Method(Cont’d)

Calling a Method from the Same Class

- A method is called in different ways as below



```java
public class Main extends Character {
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        showMyMoney1();

        Main main = new Main();
        main.showMyMoney2();
        main.showMyMoney3();
        main.showMyMoney4();
        main.showMyMoney5();
    }

    public static void showMyMoney1() {
        System.out.println("1000won");
    }

    public void showMyMoney2() {
        System.out.println("1000won");
    }

    private void showMyMoney3() {
        System.out.println("1000won");
    }

    protected void showMyMoney4() {
        System.out.println("1000won");
    }

    void showMyMoney5() {
        System.out.println("1000won");
    }
}
```



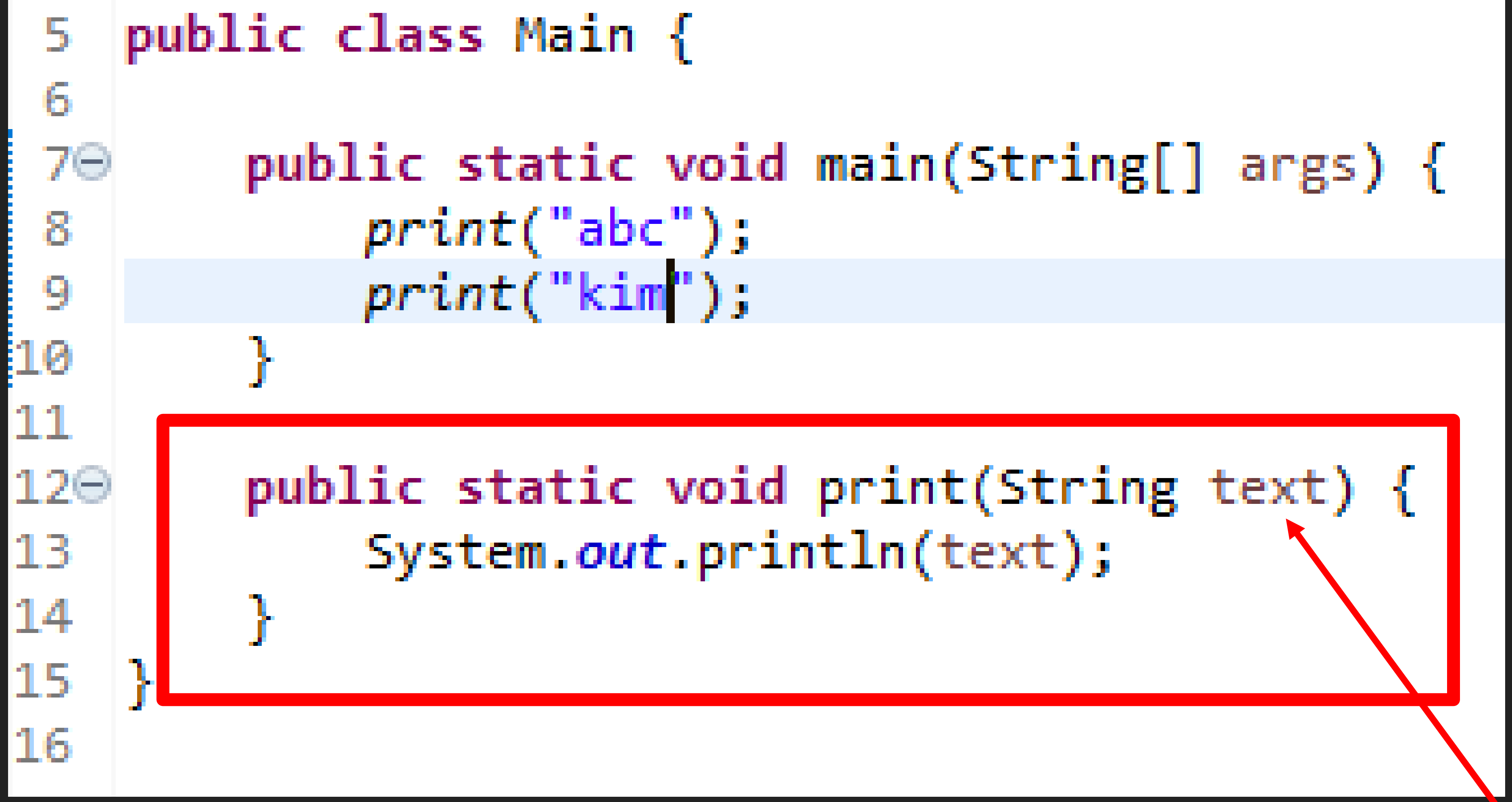
```
<terminated> Main [Java Application] C:₩Users₩kopo₩.p2₩p
1000won
1000won
1000won
1000won
1000won
```

Keyword of Static and Access Modifiers

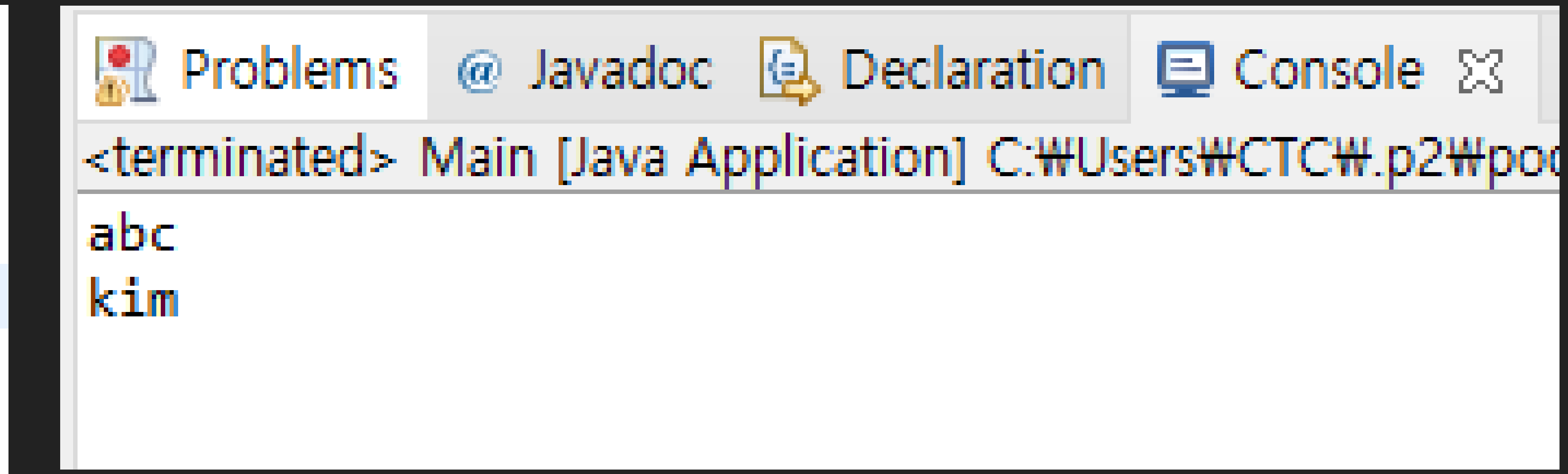
Method(Cont'd)

Method and Parameter

- A method is a block code which runs when it is called.



```
5  public class Main {
6
7      public static void main(String[] args) {
8          print("abc");
9          print("kim");
10     }
11
12     public static void print(String text) {
13         System.out.println(text);
14     }
15 }
16
```



```
<terminated> Main [Java Application] C:₩Users₩CTC₩.p2₩poo
abc
kim
```

Parameter
: Information can be passed to methods by this

Q1* : Explain the meaning of the 12nd line

P1

Printing a Name

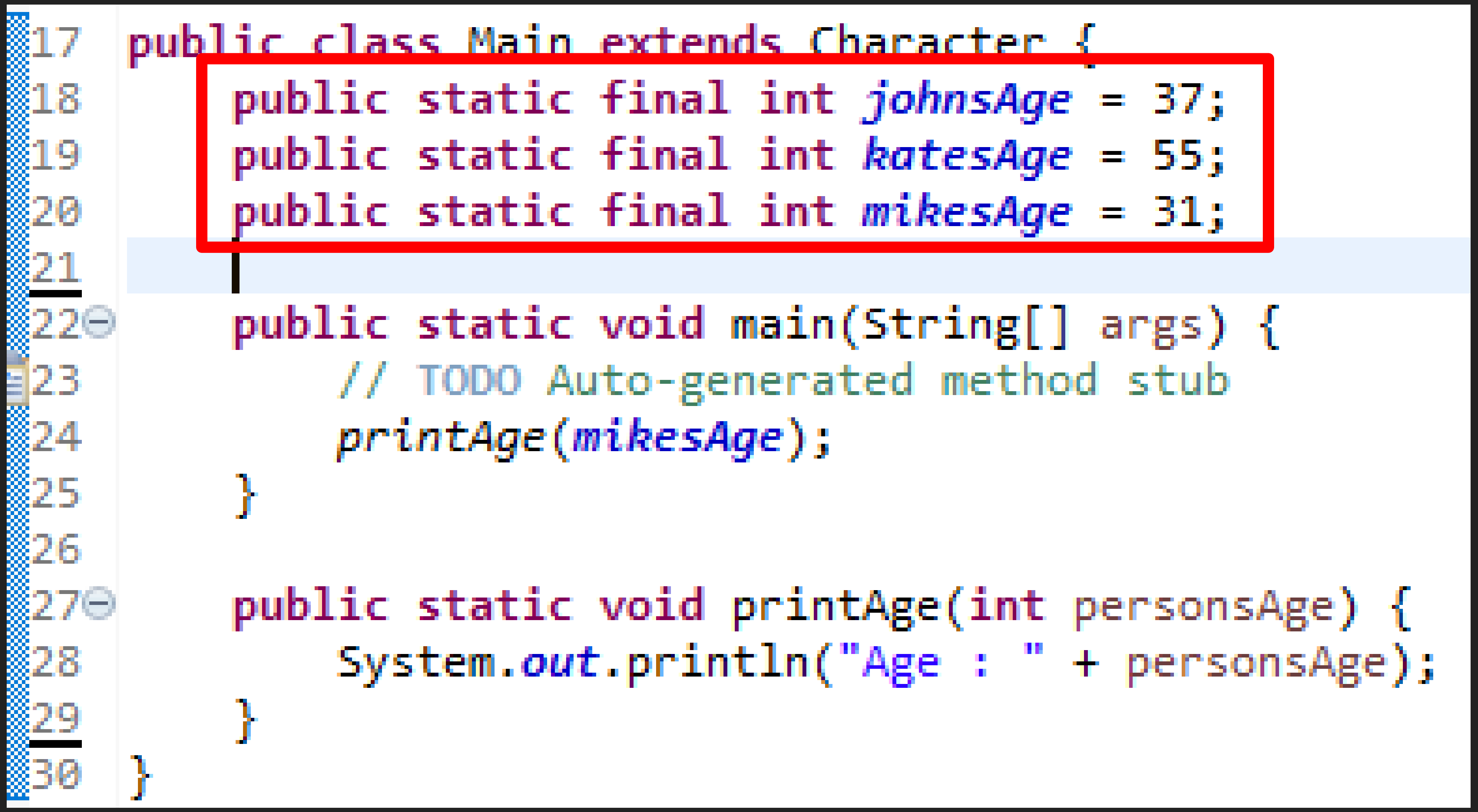
- User selects a number for a family member
- Print the name of the family member
- The program should include the following function
  public void addition(int familyMember)

```
#Printing a Name
1.  Father
2.  Mother
3.  Son
4.  Daughter
-> 3
John
```

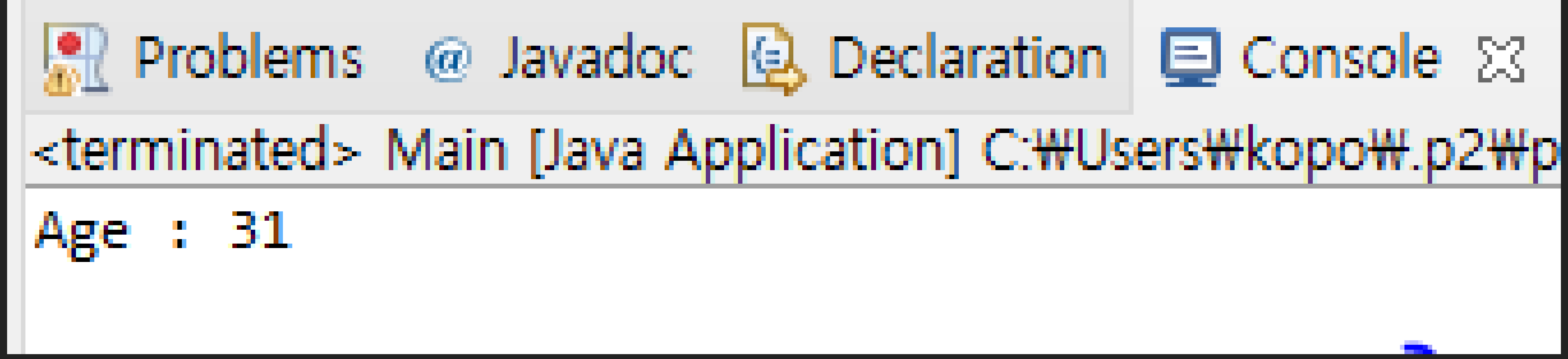
Method(Cont'd)

Declaring a static final Variable

- A static final variable can't be changed



```java
public class Main extends Character {
    public static final int johnsAge = 37;
    public static final int katesAge = 55;
    public static final int mikesAge = 31;

    public static void main(String[] args) {
        // TODO Auto-generated method stub
        printAge(mikesAge);
    }

    public static void printAge(int personsAge) {
        System.out.println("Age : " + personsAge);
    }
}
```



```
<terminated> Main [Java Application] C:₩Users₩kopo₩.p2₩p
Age : 31
```

P2 : Try changing the values for the static final variables in main function

P3

Printing a Name II

- User selects a number for a family member

- Print the name of the family member

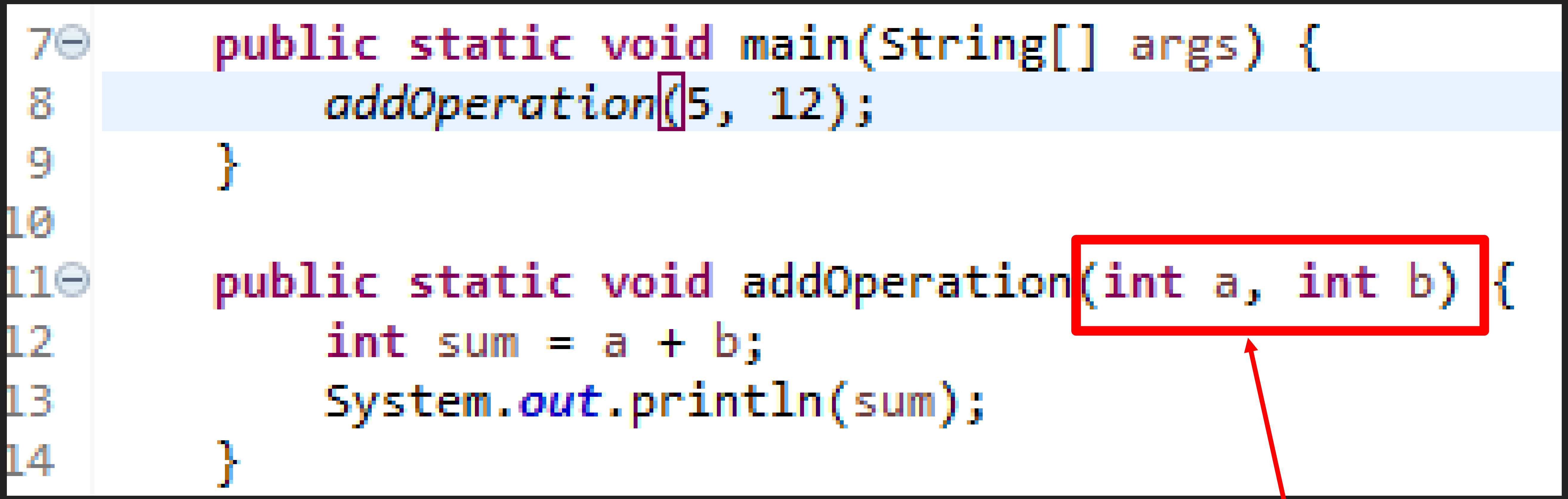
- The program should declare the static final variables for the names and include the function containing a parameter to print the name

```
#Printing a Name
1.  Father
2.  Mother
3.  Son
4.  Daughter
-> 3
John
```

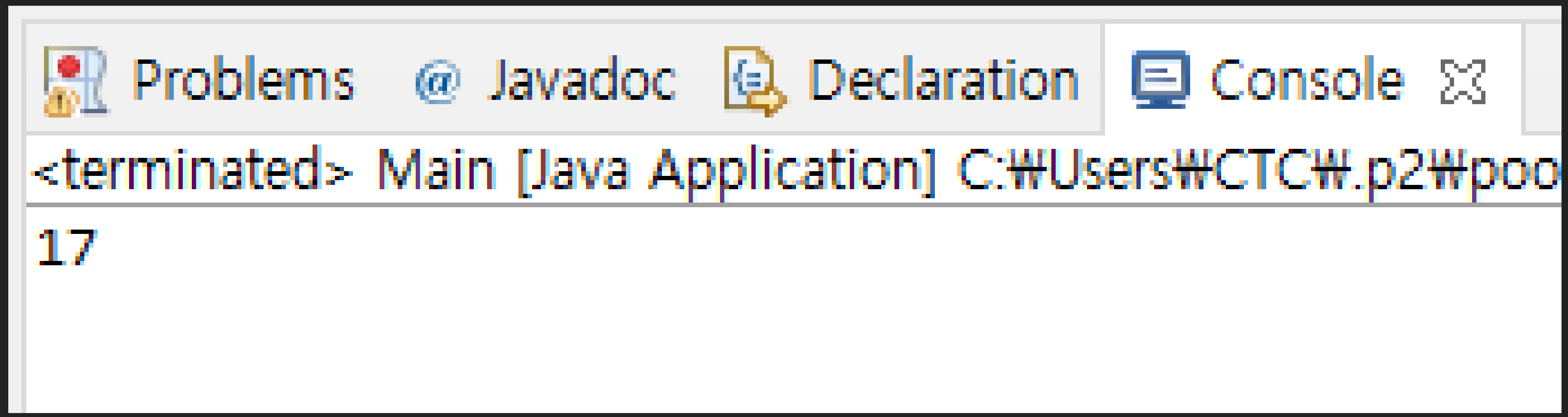
Method(Cont’d)

Multiple Parameters

- Parameters can be separated by comma



```
    public static void main(String[] args) {
        addOperation(5, 12);
    }

    public static void addOperation(int a, int b) {
        int sum = a + b;
        System.out.println(sum);
    }
```



```
<terminated> Main [Java Application] C:₩Users₩CTC₩.p2₩poo
17
```

Multiple Parameters

P4

Calculator

- User inputs a formula including only one operator

- The program calculates the formula by using the following 4 of functions
  1) public static void addition(double num1, double num2)
  2) public static void subtraction(double num1, double num2)
  3) public static void multiplication(double num1, double num2)
  4) public static void division(double num1, double num2)

```
3*7
21
6+9
15
```

P5

Multiplication Table

- User inputs a number N

- Print the N times table

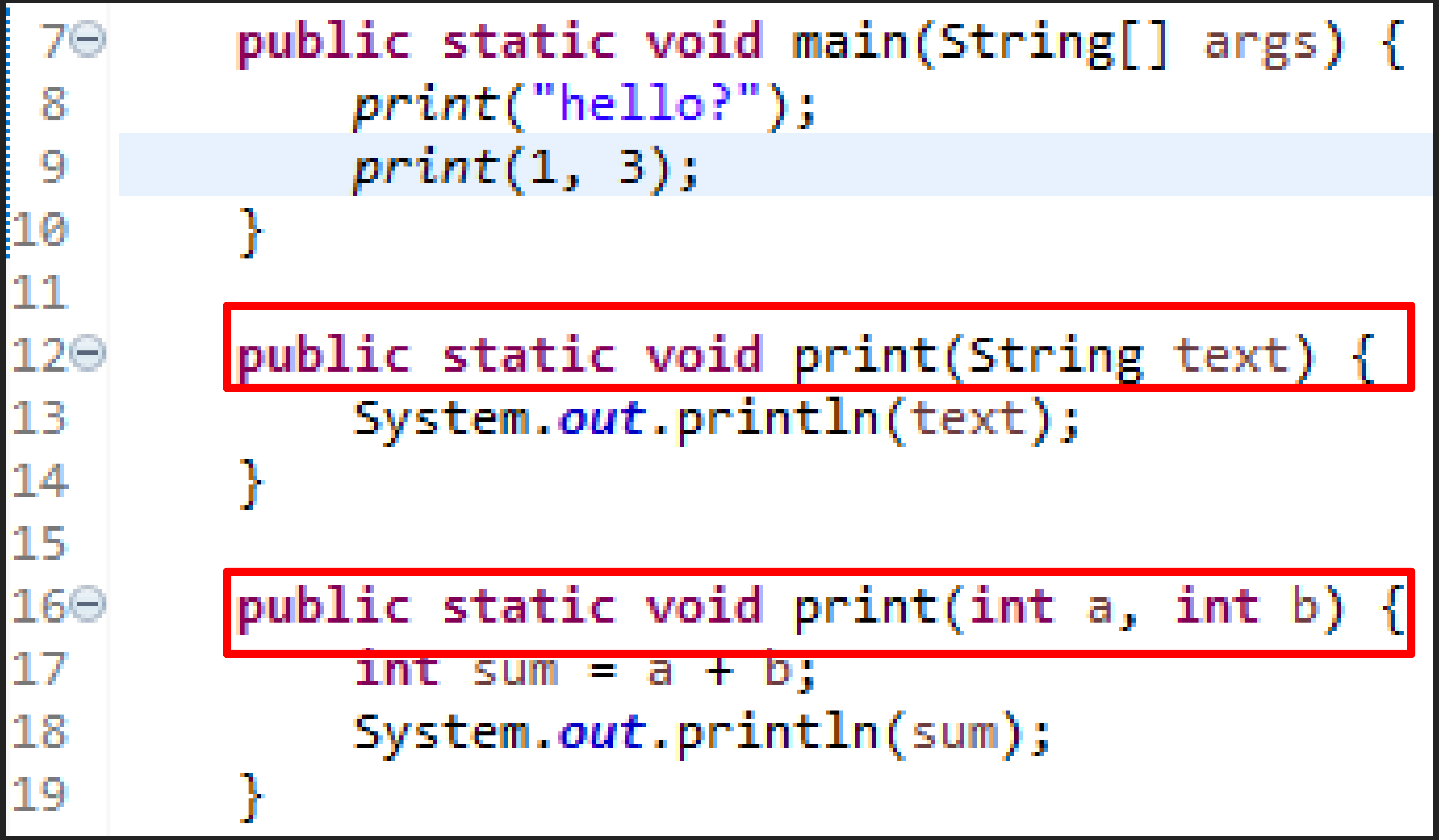
(Use a separate method to print the N times table with a parameter)

Method(Cont'd)

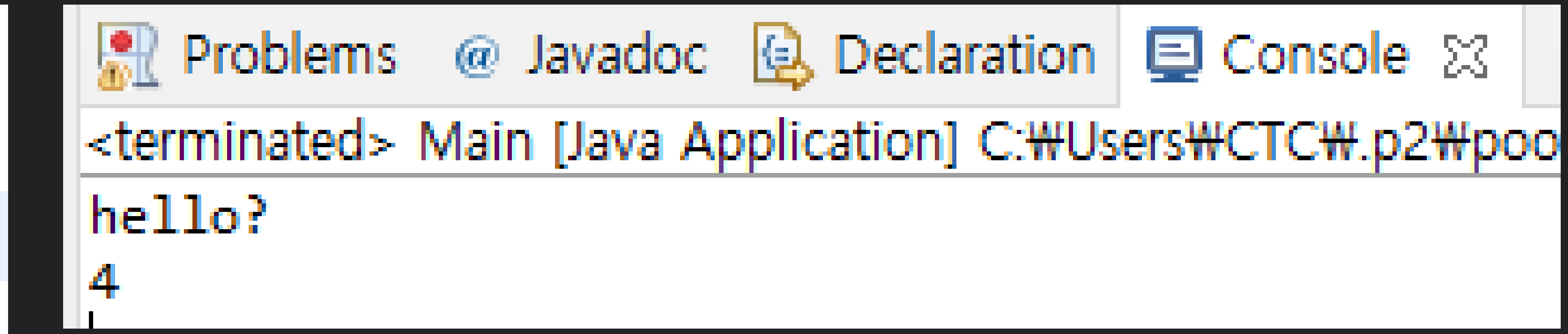
Overloading

- A feature that allows a class to have many methods having the same name

- The parameters are different from one another



```
    public static void main(String[] args) {
        print("hello?");
        print(1, 3);
    }

    public static void print(String text) {
        System.out.println(text);
    }

    public static void print(int a, int b) {
        int sum = a + b;
        System.out.println(sum);
    }
```

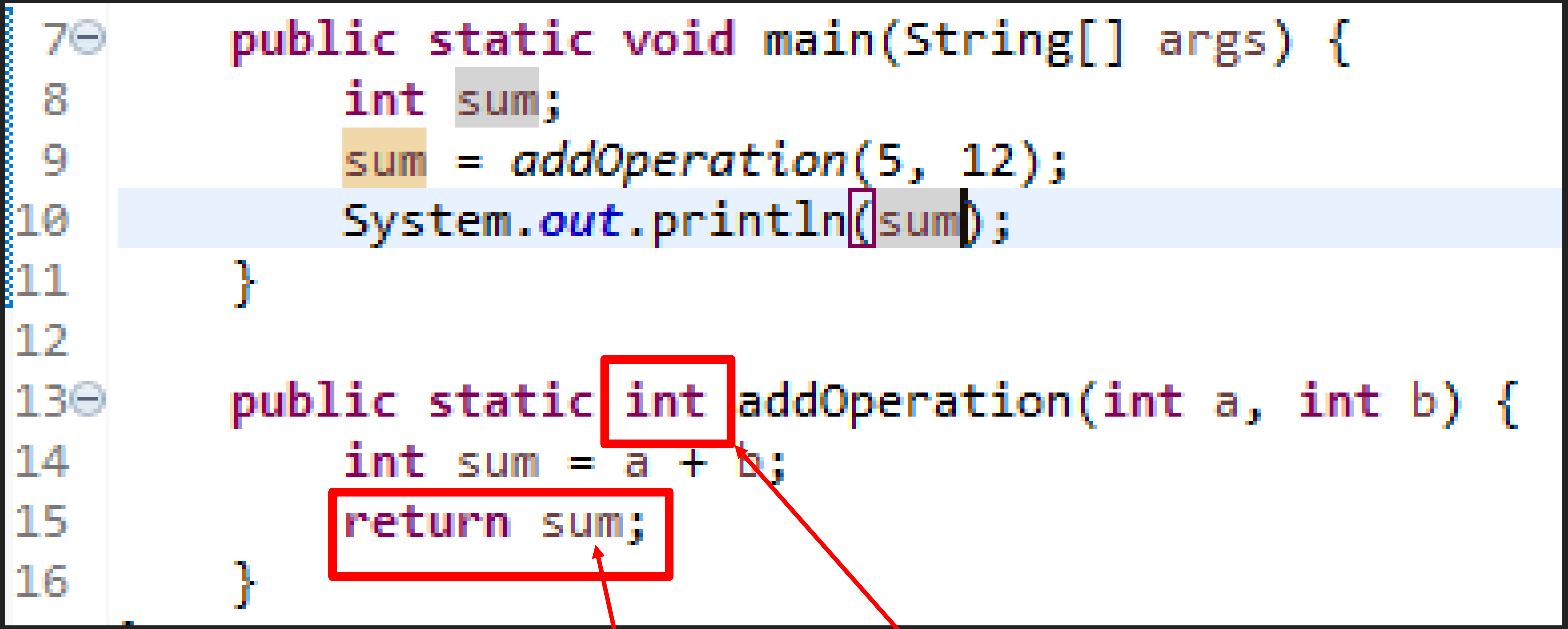


```
<terminated> Main [Java Application] C:₩Users₩CTC₩.p2₩poo
hello?
4
```

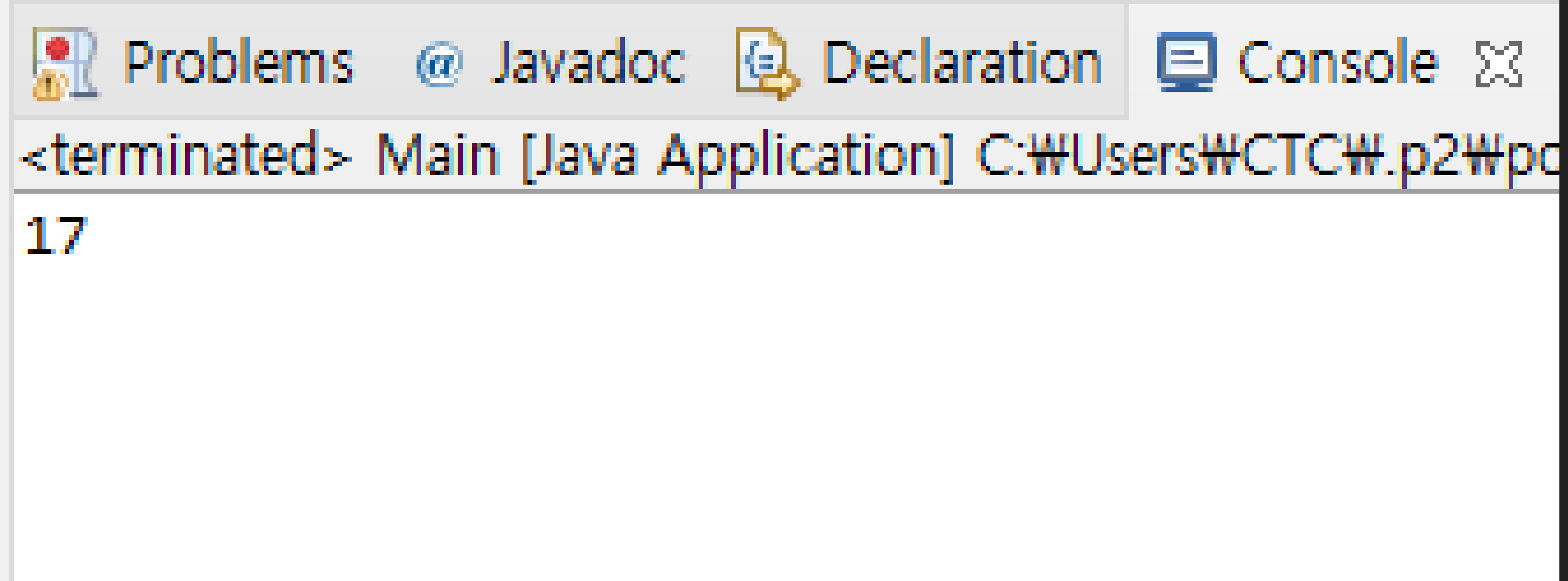
Method

Returning a Value

- A method returns a value when reaching to a return statement, throwing an exception or completing all the statements in the method



```
    public static void main(String[] args) {
        int sum;
        sum = addOperation(5, 12);
        System.out.println(sum);
    }

    public static int addOperation(int a, int b) {
        int sum = a + b;
        return sum;
    }
```



```
<terminated> Main [Java Application] C:₩Users₩CTC₩.p2₩po
17
```

Return Type

Return Value

P6

Calculator

- User inputs a formula including only one operator

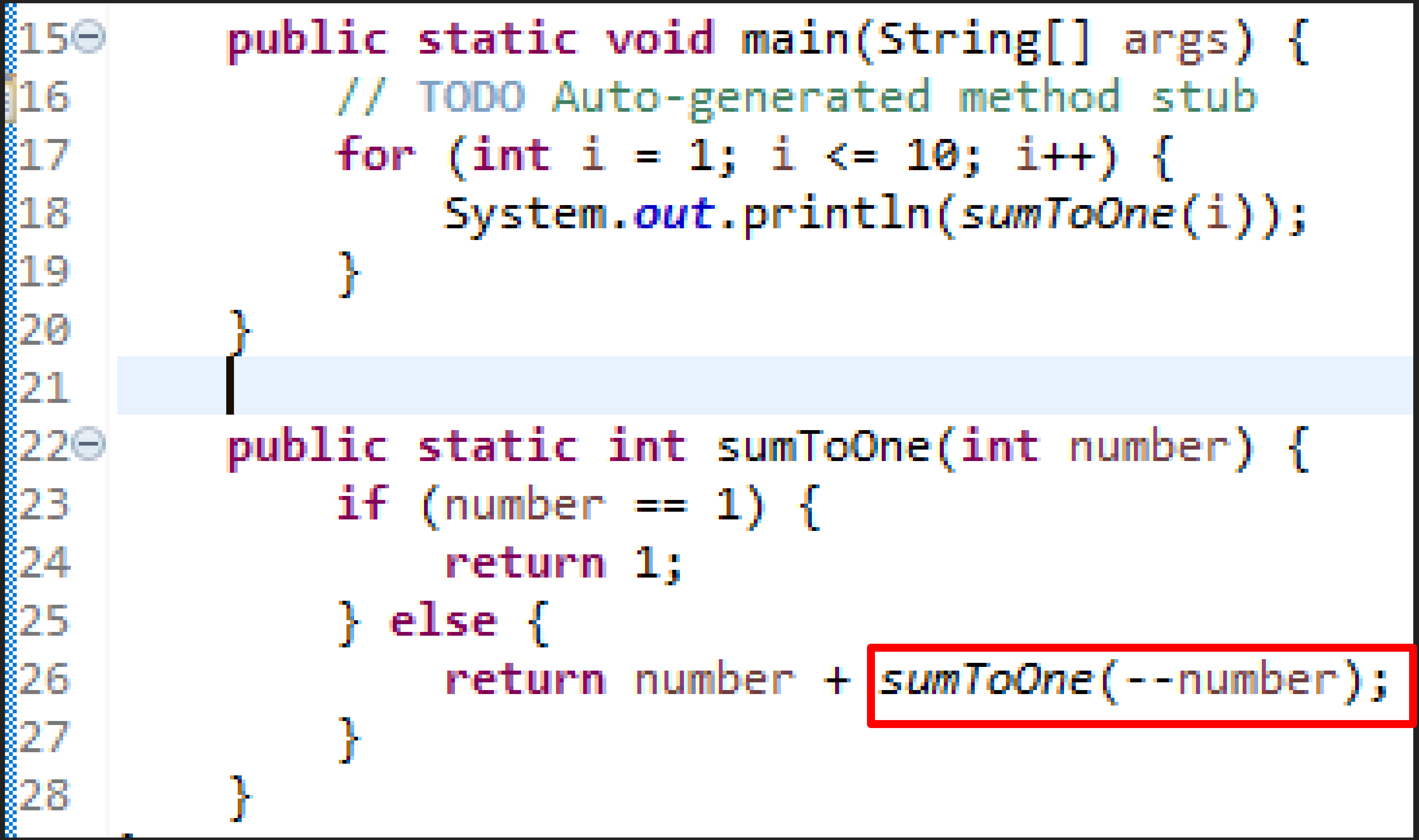
- The program calculates the formula by using the following 5 of functions
  1) public static double addition(double num1, double num2)
  2) public static double subtraction(double num1, double num2)
  3) public static double multiplication(double num1, double num2)
  4) public static double division(double num1, double num2)
  5) public static char getOperator(String formula)

```
3*7
21
6+9
15
```

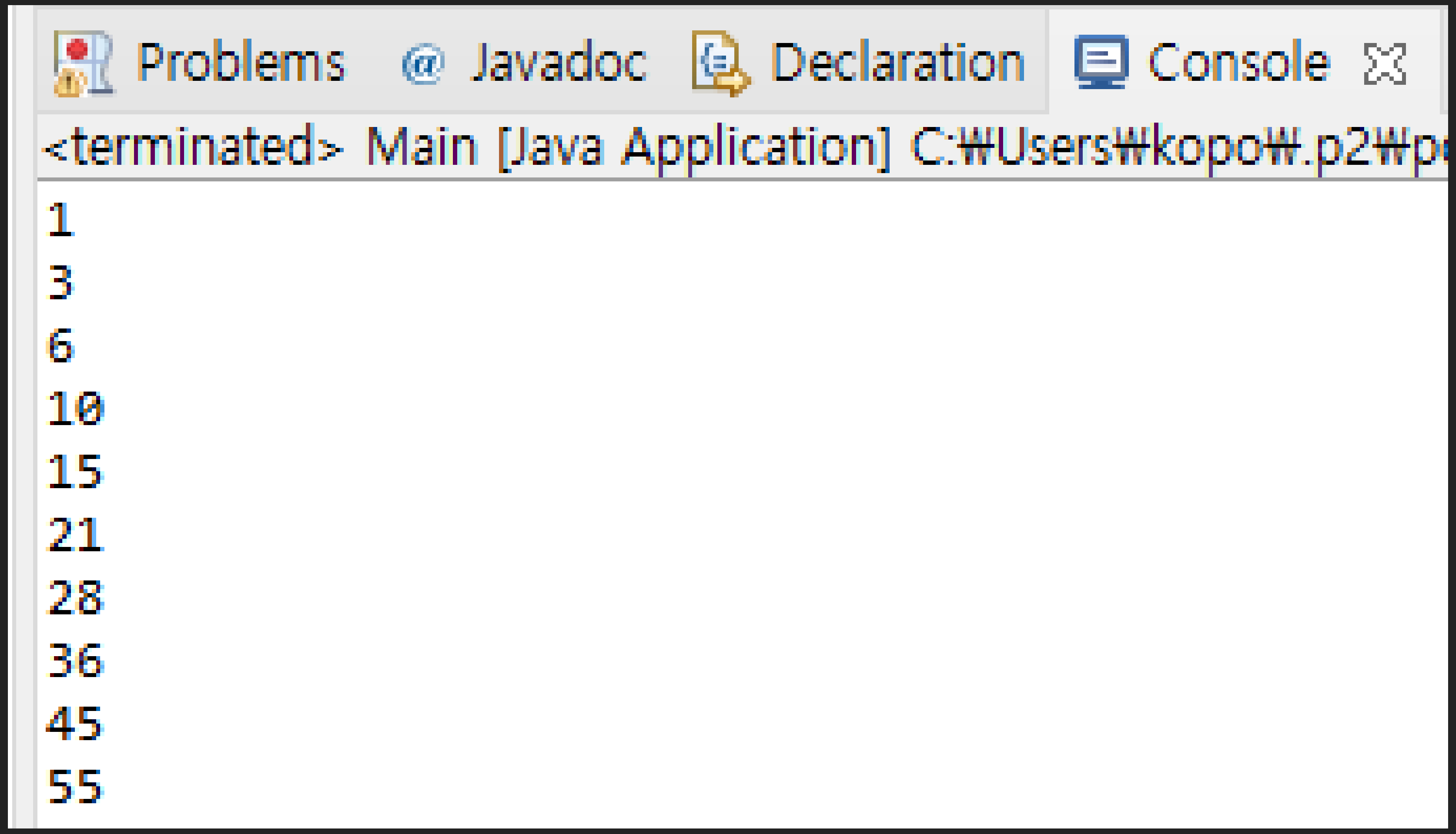
Recursive Function

Recursive Function

- A function which calls itself



```java
    public static void main(String[] args) {
        // TODO Auto-generated method stub
        for (int i = 1; i <= 10; i++) {
            System.out.println(sumToOne(i));
        }
    }

    public static int sumToOne(int number) {
        if (number == 1) {
            return 1;
        } else {
            return number + sumToOne(--number);
        }
    }
```



<terminated> Main [Java Application] C:₩Users₩kopo₩.p2₩p

```
1
3
6
10
15
21
28
36
45
55
```

P7

Factorial

- User inputs a number N
- Print the factorial of N and the calculation process
- Use recursive functions

$$n! = n \times (n-1) \times (n-2) \cdots \times 1$$

```
3
6 = 3 X 2 X 1
```

P8

Permutation

- User inputs the two numbers for n and r
- Print the permutation of N and the calculation process
- Use recursive functions
- $P(n,r) = \frac{n!}{(n-r)!}$

```
5
3

P(5, 3) = 5! / (5 - 3)! = (5 X 4 X 3 X 2 X 1) / (2 X 1) = 120 / 2 = 60
```

P9

Combination

- User inputs the two numbers for n and r
- Print the combination of N and the calculation process
- Use recursive functions
- $C(n,r) = \frac{n!}{r!(n-r)!}$

```
5
3

C(5, 3) = 5! / (5 - 3)! / 3! = (5 X 4 X 3 X 2 X 1) / (2 X 1) / (3 X 2 X 1)
        = 120 / 2 / 6 = 10
```

P10

Multiplication Table

- User inputs a number, N
- Print the N times of multiplication table
- Use recursive functions

```
5

5 X 2 = 10
5 X 3 = 15
5 X 4 = 20
5 X 5 = 25
5 X 6 = 30
5 X 7 = 35
5 X 8 = 40
5 X 9 = 45
```

P11

Fibonacci Sequence

- The formula for this is as below

  $F_0 = 0, F_1 = 1$

  $F_n = F_{n-1} + F_{n-2}$

  0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55 …

- User inputs a number

- Print the number of elements from the Fibonacci sequence

- Use recursive functions

```
7
0 1 1 2 3 5 8
```