Methods

Sungchul Lee

Learning Object

- Class Members
    - Methods
        - Parameters and Arguments
        - Return
    - Four type of methods

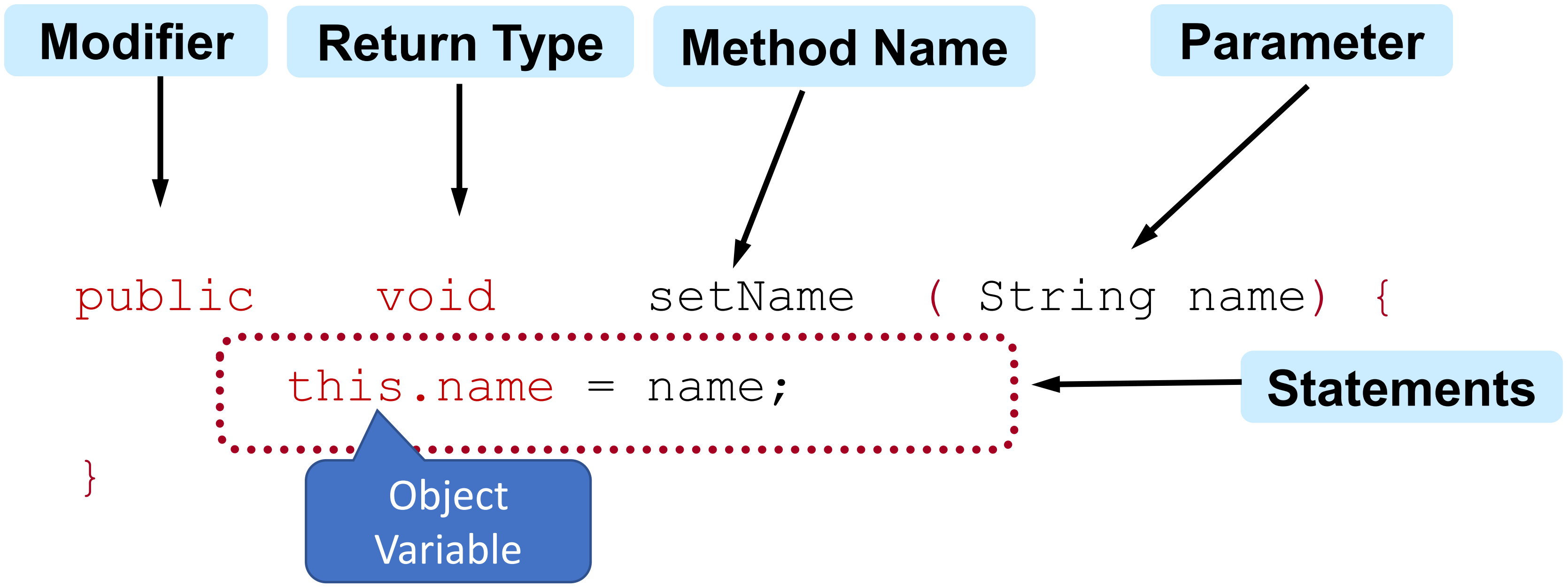
Class Members

- A class have two kinds of members:
  - **Data Member (= *Instance Variable* (*Attributes)***: data variables which determine the status of the class or an object
    - E.g.: HP, moving speed
  - ***Methods***: executable code used to manipulate /change the status of an object or access the value of the data member
    - Similar to functions in procedural languages
    - E.g.: when your character hit an enemy, the enemy’s HP will be reduced based on your hit points.

```
public static void main(String[] args) { …}   //also Method
```

Syntax of Methods

```
<modifier> <return type> <method name>( <parameters>){
       <statements>
}
```



Modifier

Return Type

Method Name

Parameter

```
public    void    setName  ( String name)  {
      this.name = name;
}
```

Statements

Object Variable

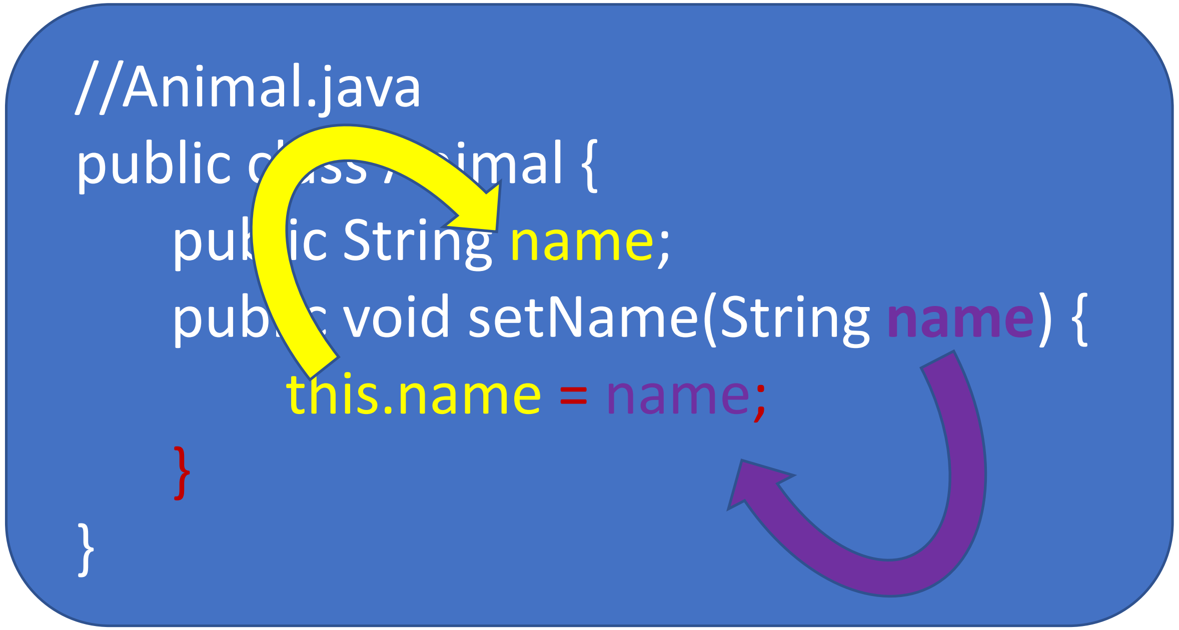
Parameters and Arguments

- A parameter is a local variable in the called method to hold the value of the passed argument
  - **placeholder**
- An argument is a value we pass to a method
  - cat.setName("Pcat");

"Pcat" your input, argument

Methods - Example

- Call the method using dot (.) like instance variable



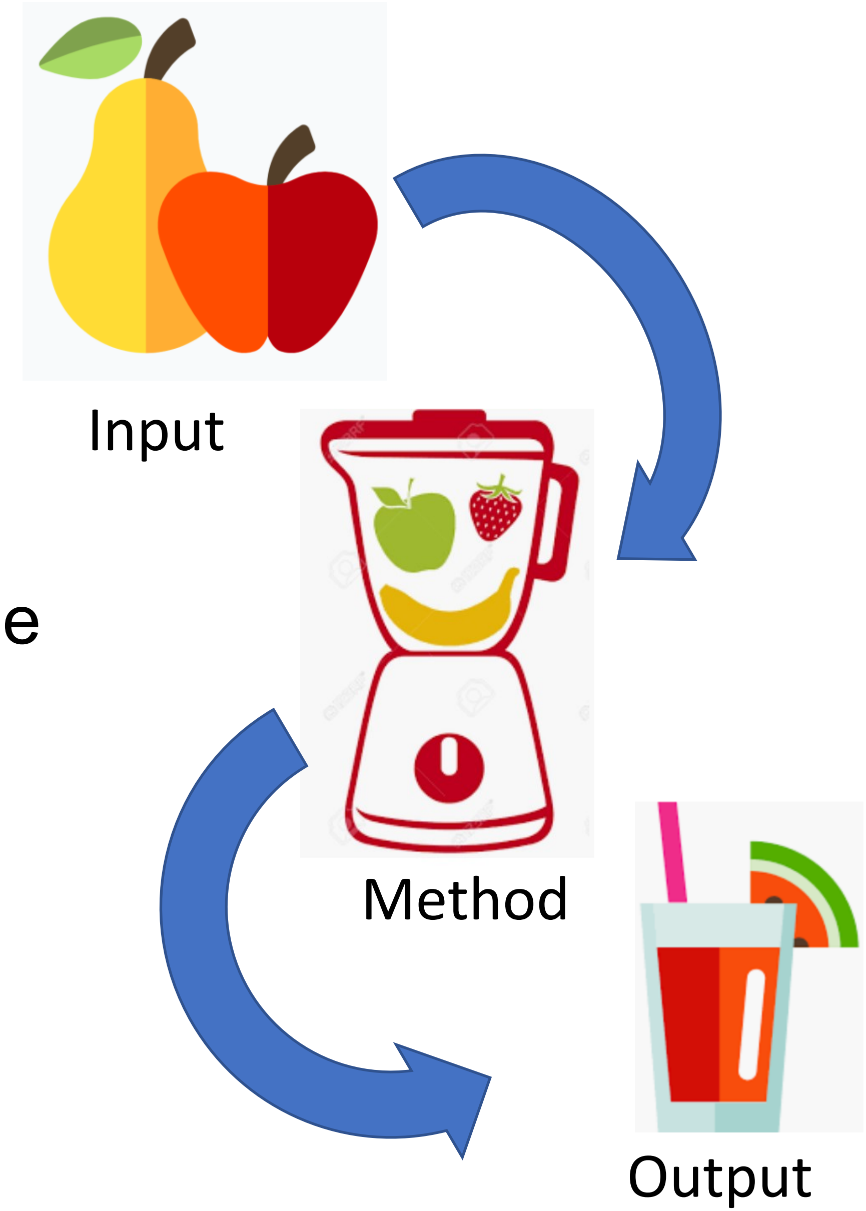
```
//Animal.java
public class Animal {
    public String name;
    public void setName(String name) {
        this.name = name;
    }
}
```

```
//Main.java
public class Main {
    public static void main(String[] args) {
        Animal cat = new Animal();
        cat.setName("Pcat");
        System.out.println(cat.name); // Pcat
    }
}
```

Method return

- ➢Method Input Value
  - ❑Input value using argument/Parameter
- ➢Method Output Value
  - ❑What if we want output from method?
    - ❖Return !!
    - ❖return is the keyword for a return value
- ➢Syntax:

```
public  [Return Type] method() {
    return Value;
}
```



Return Statement

- Purpose of a return statement
    - Tells the Java interpreter to stop executing the current method
        - All sentences after will not be executed
        - The method can be finished in a different way
    - Returns control to the calling method
    - Usually also return a value

```
public void say_nick(String nick) {
   if ("fool".equals(nick)) { return; }
    System.out.println("My Nickname is "+nick);
 }
```

If nick == “fool” , the method is finished

If nick != “fool” , Print out.

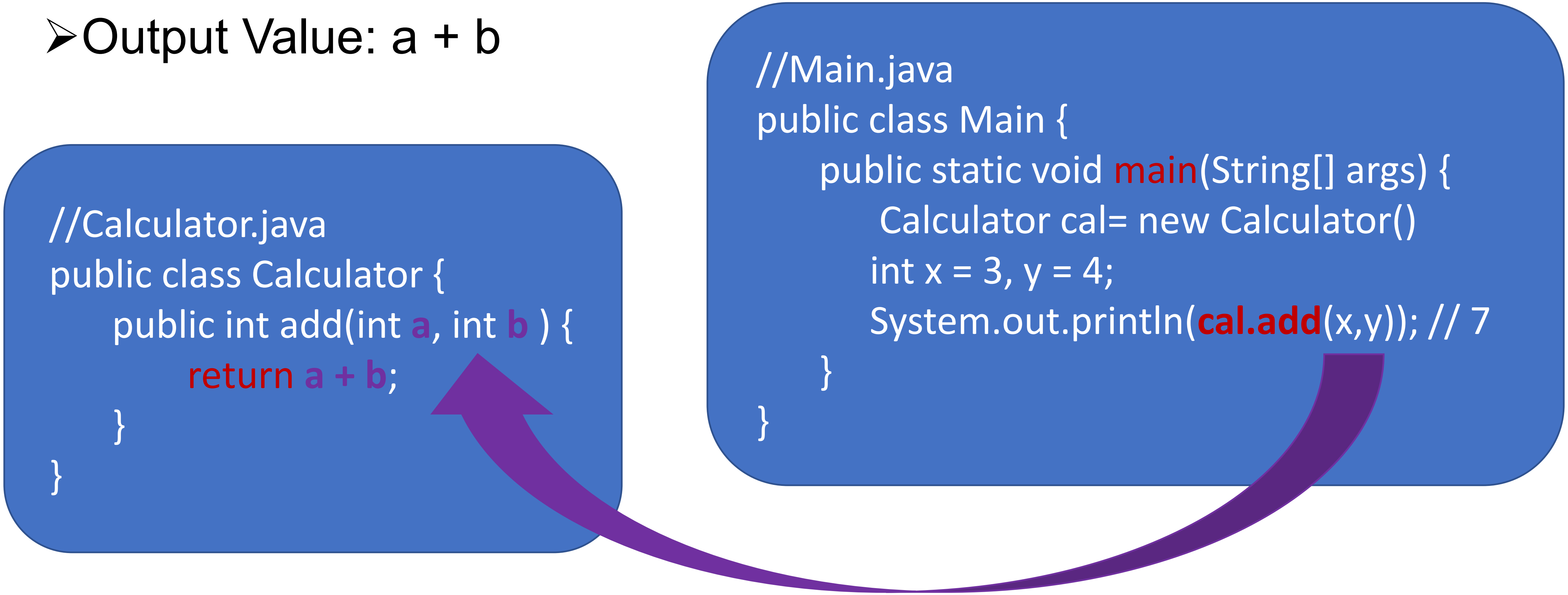
Return Type

- Usually a method will return a value after it is executed
  - This value can be passed to a variable and used for further computations
- **Return type** - the **data type** of the **value returned** by a method
- A special return type – void
  - No value will be returned
  - E.g., just print out the result
    - Or just change the attribute value
- When process meet "return", the method is finished

Important: *the variable returned should have the same data type as the return type*

Return - Example

- Input Value: a, b
- Output Value: a + b



```
//Calculator.java
public class Calculator {
    public int add(int a, int b ) {
        return a + b;
    }
}
```

```
//Main.java
public class Main {
    public static void main(String[] args) {
        Calculator cal= new Calculator()
        int x = 3, y = 4;
        System.out.println(cal.add(x,y)); // 7
    }
}
```

Four type of methods

1. Methods with input value and return value
   ❑Input value: int type a and int type b
   ❑return (output) value : int type value

```
public int add(int a, int b ) {
        return a + b;
}
```

2. Method with input and without return value
   ❑Input value: int type a and int type b
   ❑return (output) value : void (None return value)

```
public void add(int a, int b ) {
        system.out.println(a + " add " + b + "  =  " + ( a + b) );
}
```

Four type of methods – cont.

3. Method without input and with return
   - ❑Input value: No inputs (No Parameters)
   - ❑return (output) value : String type “Hi”

```
public String say(){
    return "Hi";
}
```

4. Method without input value and return
   - ❑Input value: inputs (No Parameters)
   - ❑return (output) value : void (None return value)

```
public void say(){
    system.out.println("Hi");
}
```

Practice

1. Make a new project (Reference: Create Project and Class File)
   - Project name: Methods
2. Create a new Class File
   - Class name: Calculator
     - Add method
     - Subtract method
     - Multiply method
     - Divide method
   - Class name: Main
     - main method for entrance class
       - Start point

Practice - Calculator

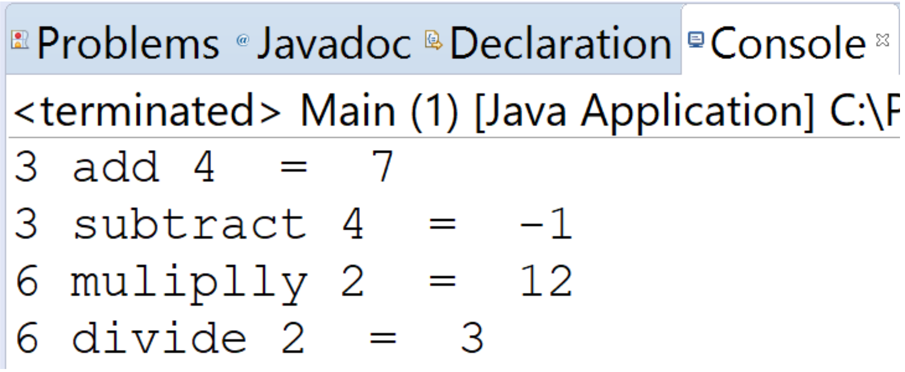
```
//Calculator.java
public class Calculator {
  int a, b; // object variable
  public int add (int a, int b ) {
        return a + b;
  }
  public void sub (int a, int b ) {
        System.out.println(a + " subtract
        " + b + "  =  " + ( a - b) );
  }
```

```
  public int mul () {
        return a * b;
  }
  public void div () {
        System.out.println(a + " divide "
        + b + "  =  " + ( a / b) );
  }
}
```

Practice - Main

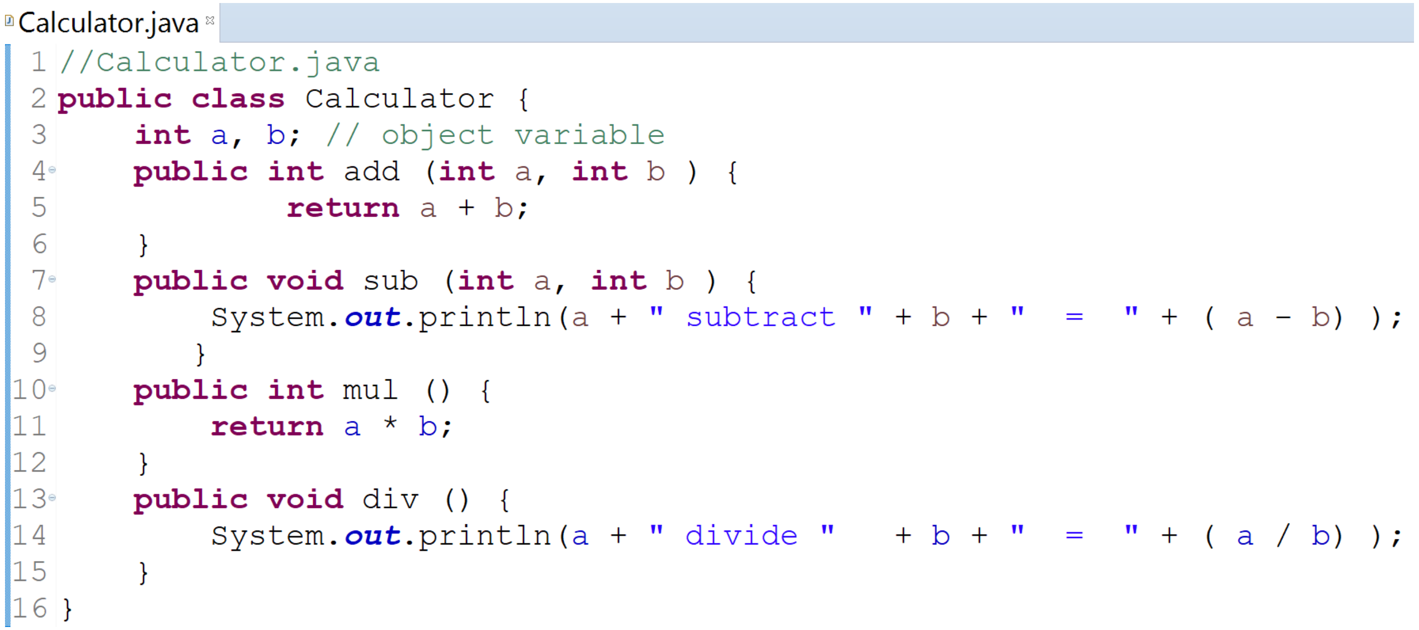
```
//Main.java
public class Main {
  public static void main(String[] args) {
    Calculator cal= new Calculator();
    int x = 3, y = 4;
    cal.a = 6;
    cal.b = 2;
    System.out.println (x + " add " + y + "  =  " + cal.add(x,y));
   cal.sub (x, y);
   System.out.println(cal.a     + " muliplly " + cal.b + "  =  " + cal.mul());
   cal.div ();
 }
}
```

Practice – Code and Result

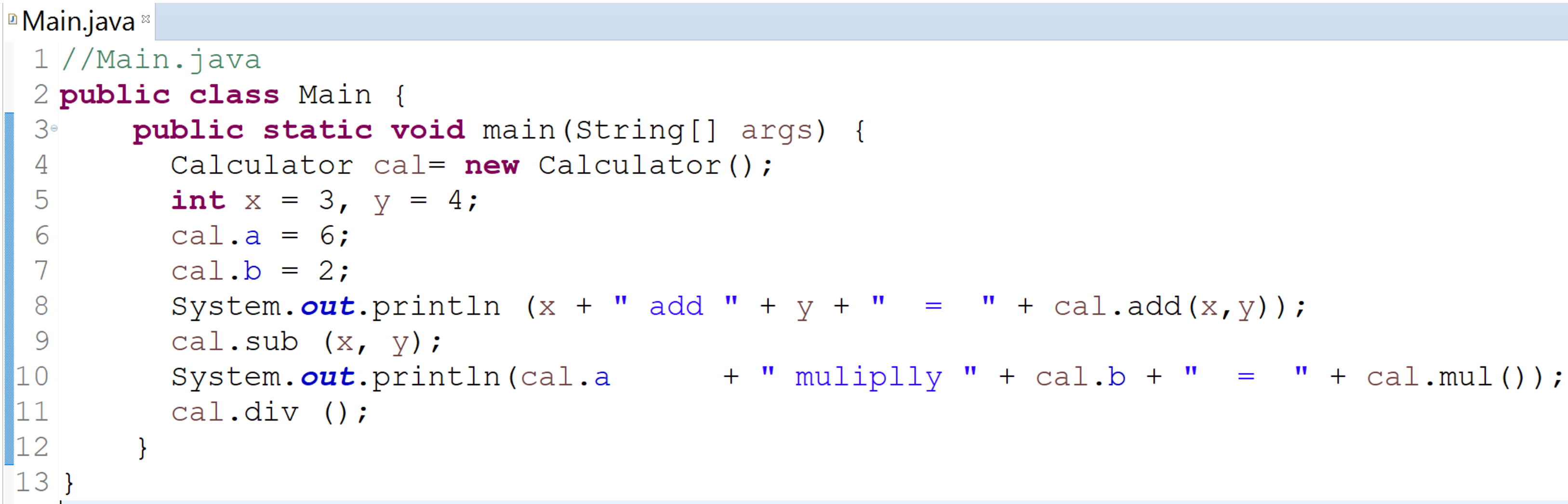


```
<terminated> Main (1) [Java Application] C:\F
3 add 4  =  7
3 subtract 4  =  -1
6 muliplly 2  =  12
6 divide 2  =  3
```

Result



```
//Calculator.java
public class Calculator {
     int a, b; // object variable
     public int add (int a, int b ) {
           return a + b;
     }
     public void sub (int a, int b ) {
          System.out.println(a + " subtract " + b + "  =  " + ( a - b) );
       }
     public int mul () {
          return a * b;
     }
     public void div () {
          System.out.println(a + " divide "   + b + "  =  " + ( a / b) );
     }
}
```



```
//Main.java
public class Main {
    public static void main(String[] args) {
      Calculator cal= new Calculator();
      int x = 3, y = 4;
      cal.a = 6;
      cal.b = 2;
      System.out.println (x + " add " + y + "  =  " + cal.add(x,y));
      cal.sub (x, y);
      System.out.println(cal.a      + " muliplly " + cal.b + "  =  " + cal.mul());
      cal.div ();
    }
}
```

Summary

- Methods
  - Methods Input and Output
    - Parameters and Arguments
    - Return
- Four type of methods
  1. Methods with input value and return value
  2. Method with input and without return value
  3. Method without input and with return
  4. Method without input value and return