Programming in the Small I: Names and Things (Part II)

188230 Advanced Computer Programming

Asst. Prof. Dr. Kanda Runapongsa Saikaew
(krunapon@kku.ac.th)
Department of Computer Engineering
Khon Kaen University

Agenda

- Arithmetic Operations
- Programming Environments
- Programming Exercises

Numerical Data Types

- byte 8 bits
- short 16 bits
- int 32 bits
- long 64 bits
- float 32 bits
- double 64 bits

Number Literals

- int i = 34;
- long l = 100000; **or** long = 1000000l;
- float f = 100.2f; **or** float f = 100.2F;
- double d = 100.2d or double d = 100.2D;
- int octal = 035;
- int hex = 0xa2;

Constant Variables

- Our own declaration
  - final datatype CONSTANTNAME = value;
  - Examples:
    - final double PI = 3.14159;
    - final int SIZE = 3;
- Constants in Java built-in classes
  - Math.PI
    - Its type is double
    - Its value is 3.14....

Operators

+, -, *, /, and %

Examples

5/2 = ?

5.0/2 = ?

2/4 = ?

5%2 = ?

Shortcut Operators

Operator	Example	Equivalent
+=	i += 8	i = i + 8
-=	i -= 8	i = i – 8
*=	i *= 8	i = i * 8
/=	i /= 8	i = i / 8
%=	i %= 8	i = i % 8

Increment & Decrement Operators

- x = 1;
- a = 1 + x++ = ?
- b = 1 + ++x = ?
- c = 1 + x-- = ?
- d = 1 + --x = ?

Relational Operators

- A == B
- A != B
- A < B
- A > B
- A <= B
- A >= B
- Note that A and B must have numeric types or char type. A and B cannot be String

Boolean Operators

- The boolean operator “and” is &&
  - The result is also a boolean value. The result is true if both of the combined values are true
- The boolean operator ”or” is ||
  - The result is false if either of the combined values is false
- The boolean operator “not” is a unary operator which is !

Short-circuited

- The operators && and || are said to be short-circuited versions of the boolean operators.
- This means that the second operand of && or || is not necessarily evaluated
- Consider the test
- (x != 0) && (y/x > 1)
- If value of x is in fact zero, the computer will never perform the division
- S since when the computer evaluates (x != 0), it finds that the result is false

Boolean Operators Examples

- int a = 2, b = 3;
- boolean c = (a > 2) && (++b > 3)
  - Then a = ? b = ? c = ?
- a = 2; b = 3;
- boolean d = (a > 2) || (++b > 3)
  - Then a = ? b = ? d = ?

Conditional Operators

- Java has the conditional operator
- It's a ternary operator—that is, it has three operands
  - boolean-expression ? expression1 : expression2
- The computer tests the value of boolean-expression
  - If the value is true, it evaluates expression
  - Otherwise, it evaluates expression2

Conditional Operators Example

- int n = 3;
- int next = (n % 2 == 0) ? (n/2) : (3*n+1);
- Then n = ?, next = ?

- n = 2;
- next = (n % 2 == 0) ? (n/2) : (3*n+1);
- Then n = ?, next = ?

Assignment Operators

- Type of the expression on the RHS of an assignment statement must be the same as the type of the variable on the LHS
- But the computer may automatically convert the value computed by the expression to match the type of the variable.
- Bad example:
  - if ((a=b) == 0) System.out.println(”Hello”);
  - else System.out.println(”Bye”);

Type Casting

- int a = 17;
- double x;
- short b;
- x = a; // Legal?
- b = a; // Legal?
- b = (short) a; // Legal?

Type Conversion of String

- How to convert the string "10" into the int value 10?
- How to convert the string "17.42e-2" into the double value 0.1742
- In Java, these conversions are handled by built-in functions.
  - Integer.parseInt(”10”) = 10
  - Integer.parseInt(”a”) = ?

Sample Program

```java
3  public class SimpleCalculator {
4      public static void main(String[] args) {
5          if (args.length != 3) {
6              System.err.println("Usage:SimpleCalculator (+|-) <int1> <int2>")
7              System.exit(1);
8          }
9          int result = 0;
10         String op = args[0];
11         int operand1 = Integer.parseInt(args[1]);
12         int operand2 = Integer.parseInt(args[2]);
13         if (op.equals("+")) {
14             result = operand1 + operand2;
15         } else if (op.equals("-")) {
16             result = operand1 - operand2;
17         } else {
18             System.out.println("Unknown operation");
19         }
20         System.out.println(operand1 + " " + op + " " + operand2 + " = "
21                 + result);
22     }
23 }
```

Problems | Javadoc | Declaration | Console

```
<terminated> SimpleCalculator [Java Application]
2 + 3 = 5
```

Precedence Rules

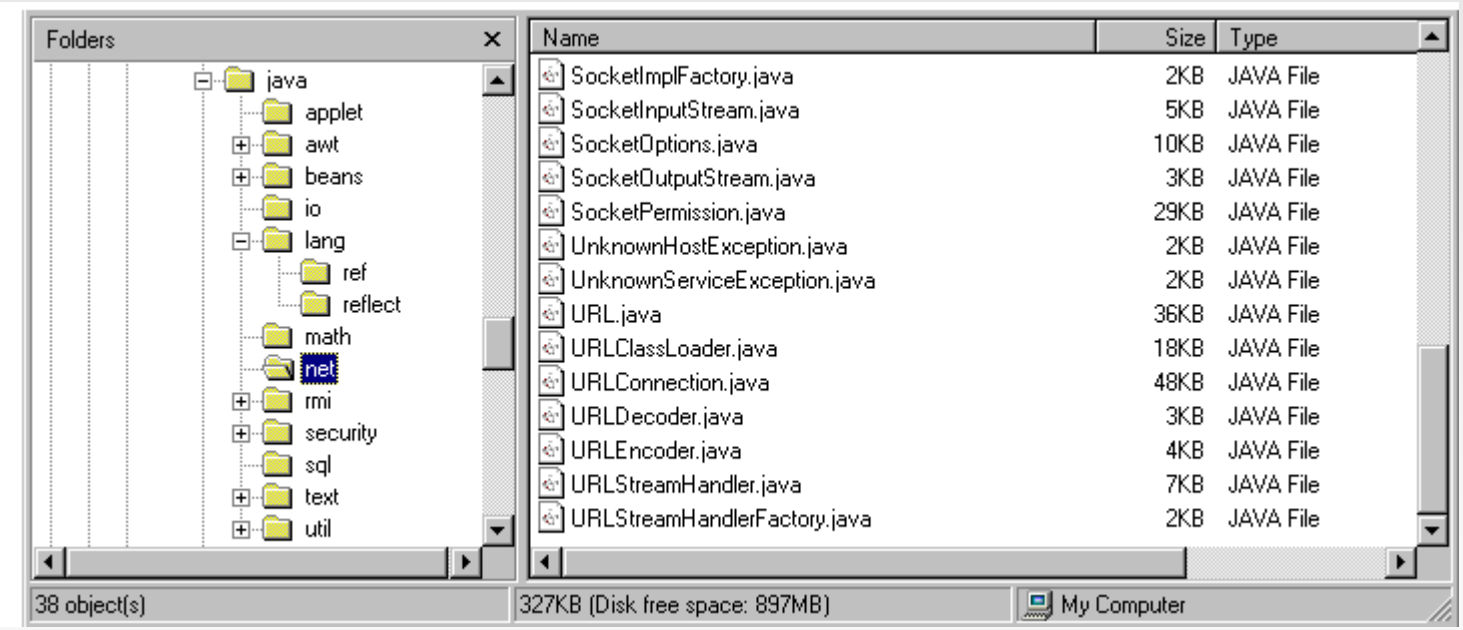
- Unary operators: ++, --, !, unary - and +, type-cast
- Multiplication and division: *, /, %
- Addition and subtraction: +, -
- Relational operators: <, >, <=, >=
- Equality and inequality: ==, !=
- Boolean and: &&
- Boolean or: ||
- Conditional operator: ?:
- Assignment operators: =, +=, -=, *=, /=, %=

Java Packages

- Many times we put all Java files into one single directory
- But the number of files is increasing, putting all these files into the same directory would be difficult to find files
- Packages are nothing more than the way we organize files into different directories according to their functionality, usability category

Java Packages and Directories

- An obvious example of packaging is the JDK package from SUN (java.xxx.yyy) as shown below:



The Purpose of Java Packages

- Basically, files in one directory (or package) would have different functionality from those of another directory
- Files in java.io package do something related to I/O
- Files in java.net package give us the way to deal with the Network.

Packages Solve Class Name Collision

- Packaging also help us to avoid class name collision when we use the same class name as that of others
- If we have a class name called "Vector", its name would crash with the Vector class from JDK
- However, this never happens because JDK use java.util as a package name for the Vector class (java.util.Vector).
- So our Vector class can be named as "Vector" or we can put it into another package like com.mycompany.Vector without fighting with anyone.

Programming Environments

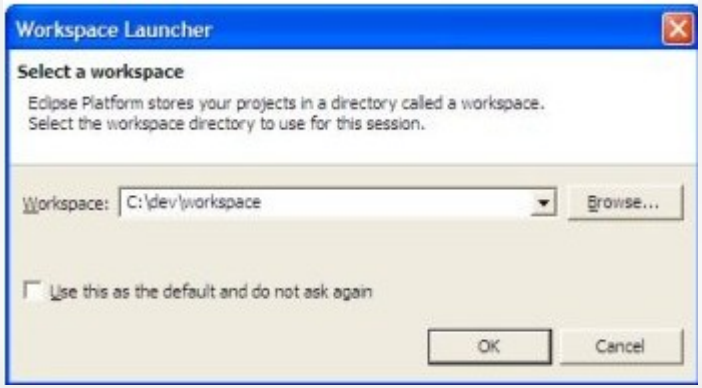
- Two approaches for creating, compiling, and edit Java programs
  - A command line environment
    - The user types commands and the computer responds
    - Example: javac Hello.java
    - java Hello
  - An integrated development environment (IDE)
    - The user uses the keyboard and mouse to interact with a user graphical interface

IDEs and Eclipse

- In an IDE, everything you need to create, compile, and run programs is integrated into a single package, with a graphical user interface
- Eclipse is used by many professional programmers
- Eclipse is probably the most commonly used Java IDE

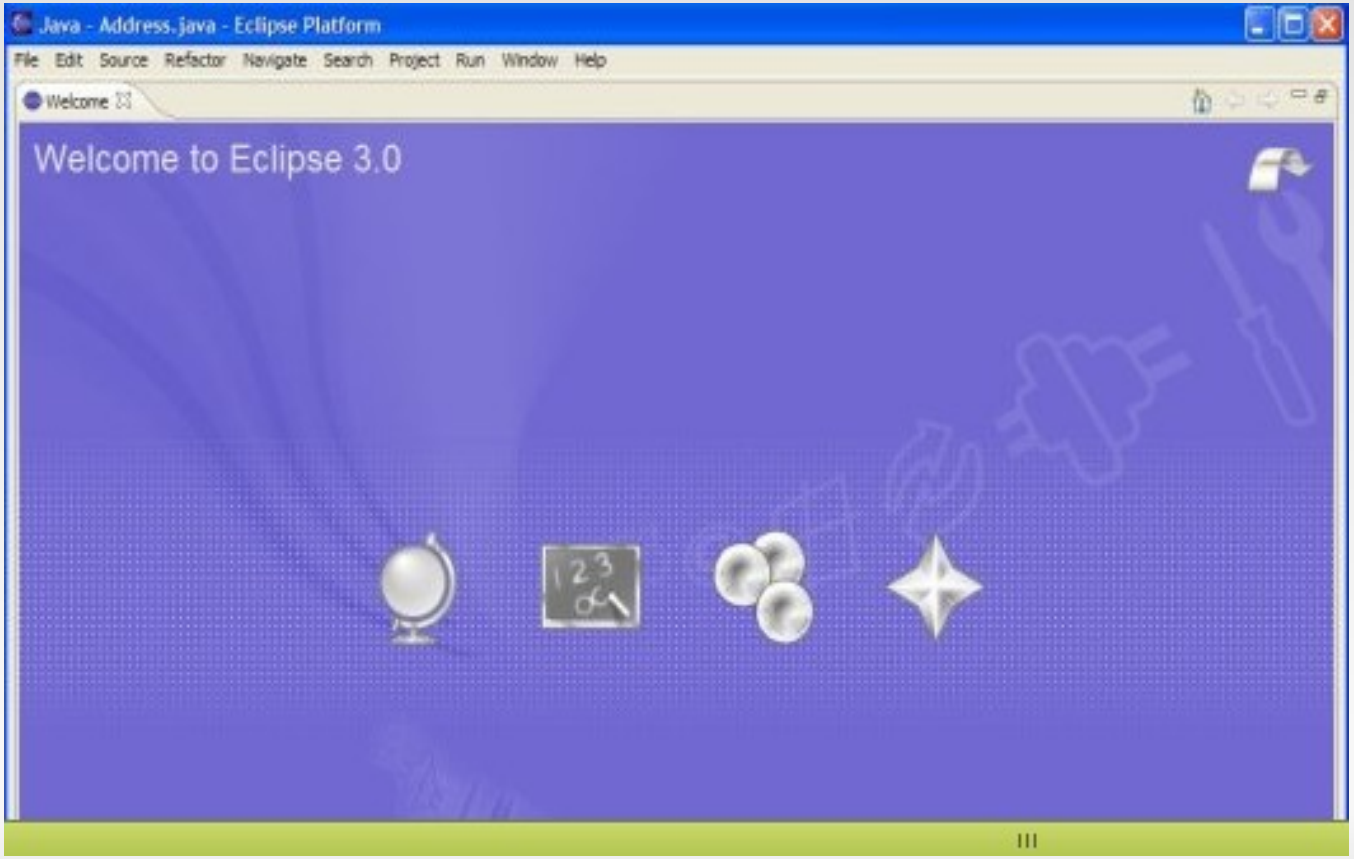
Starting Eclipse

- We'll be working with Eclipse to create and configure a new Java project
- First start up Eclipse and supply a path to a new folder which will serve as your workspace
- The workspace is a folder which Eclipse uses to store your source code



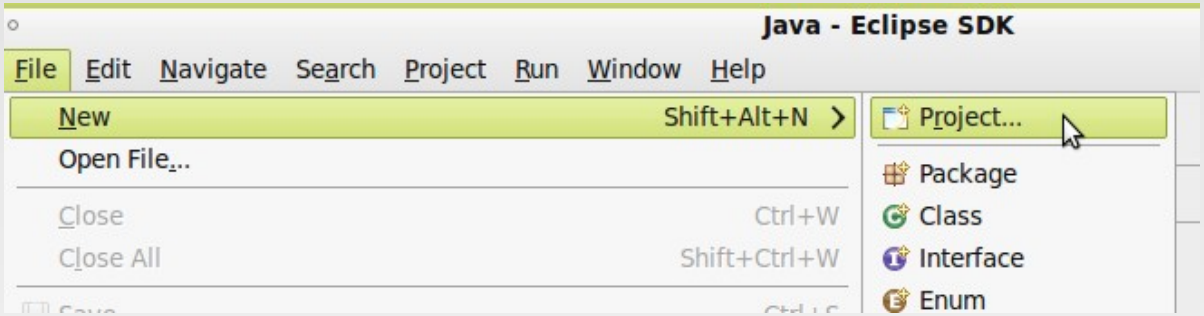
Welcome Page

- When Eclipse starts, you'll see the Welcome page:



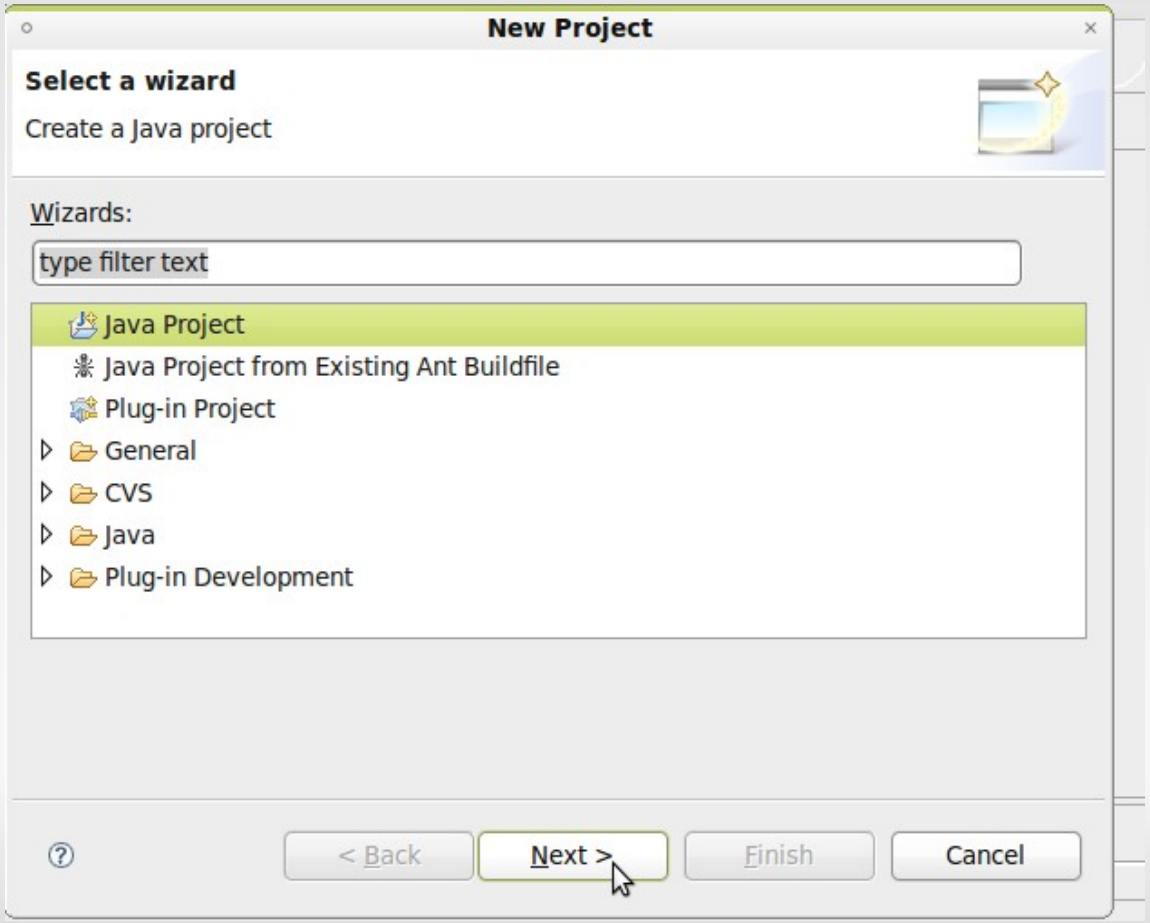
Creating a New Project

- Close the Welcome page. Right-click in the Navigator panel, and select New->Project:



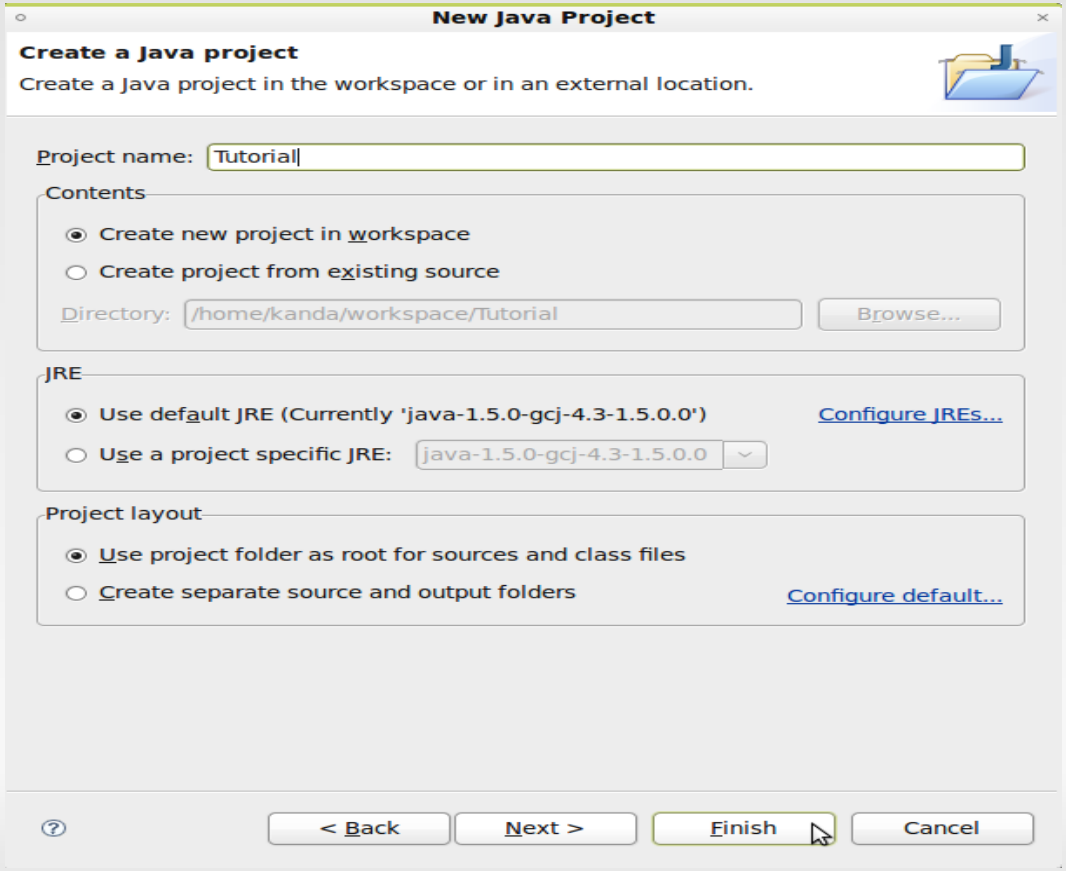
Choose Project Type

- Choose project type as Java project
  - Select Java Project and then Click Next



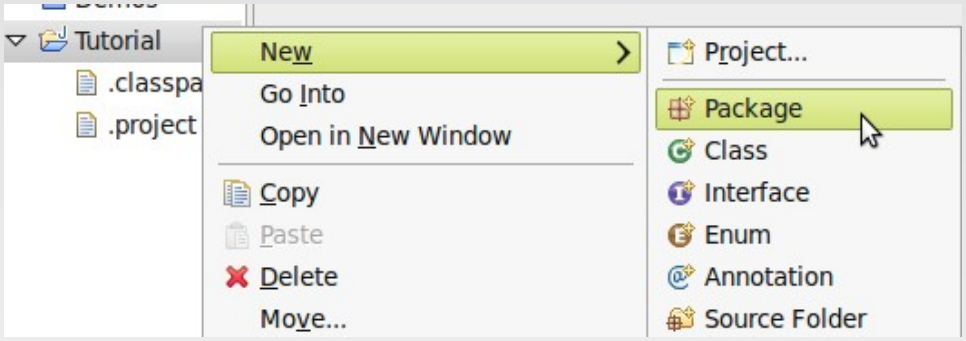
Assign Project Name

- Assign project name
  - Fill the project name field and click Finish



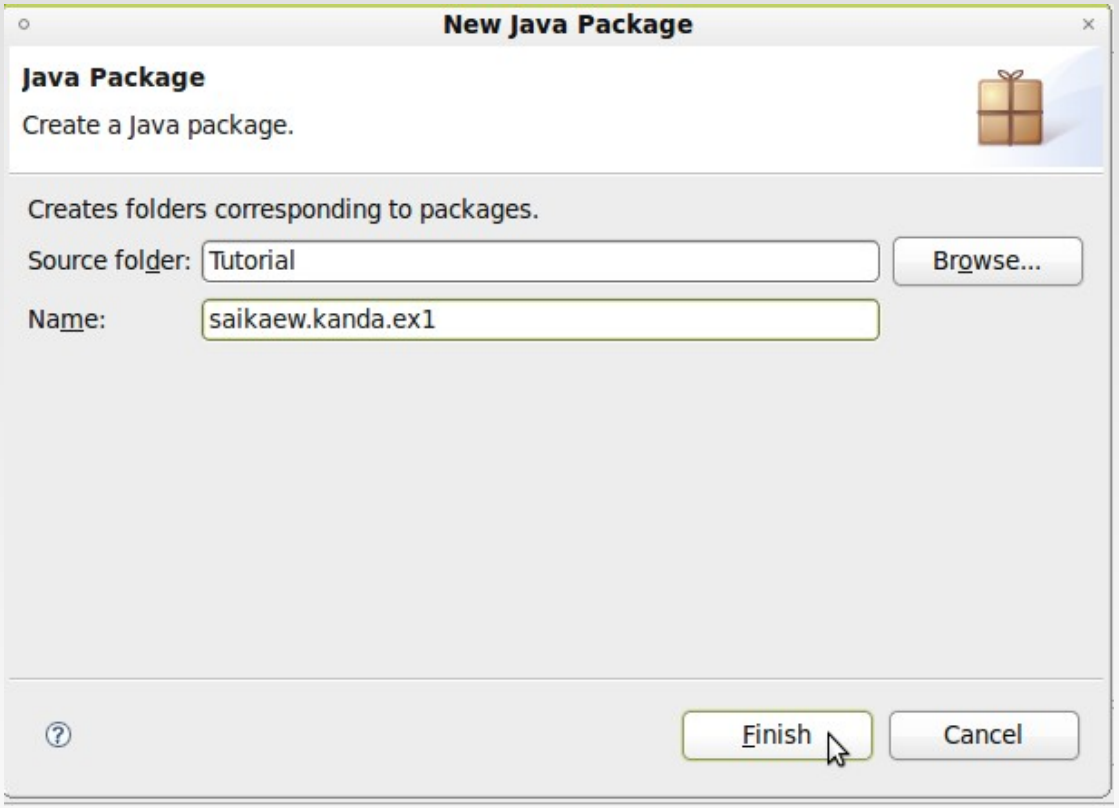
Create a New Package

- Right click Project and choose package



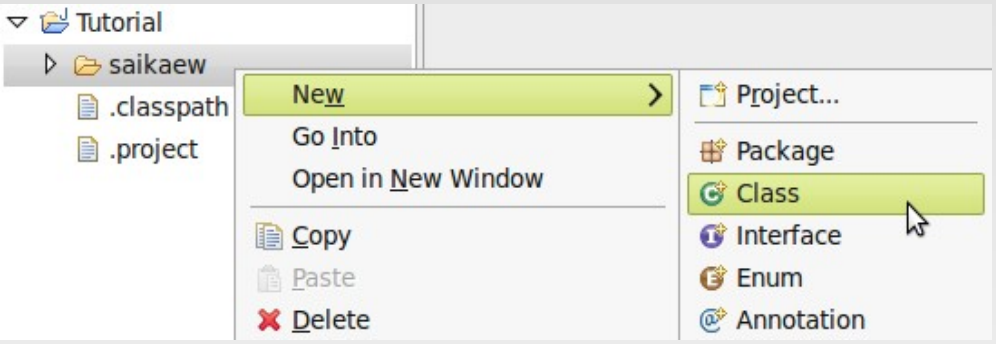
Assign Package Name

- Type package name and then click Finish



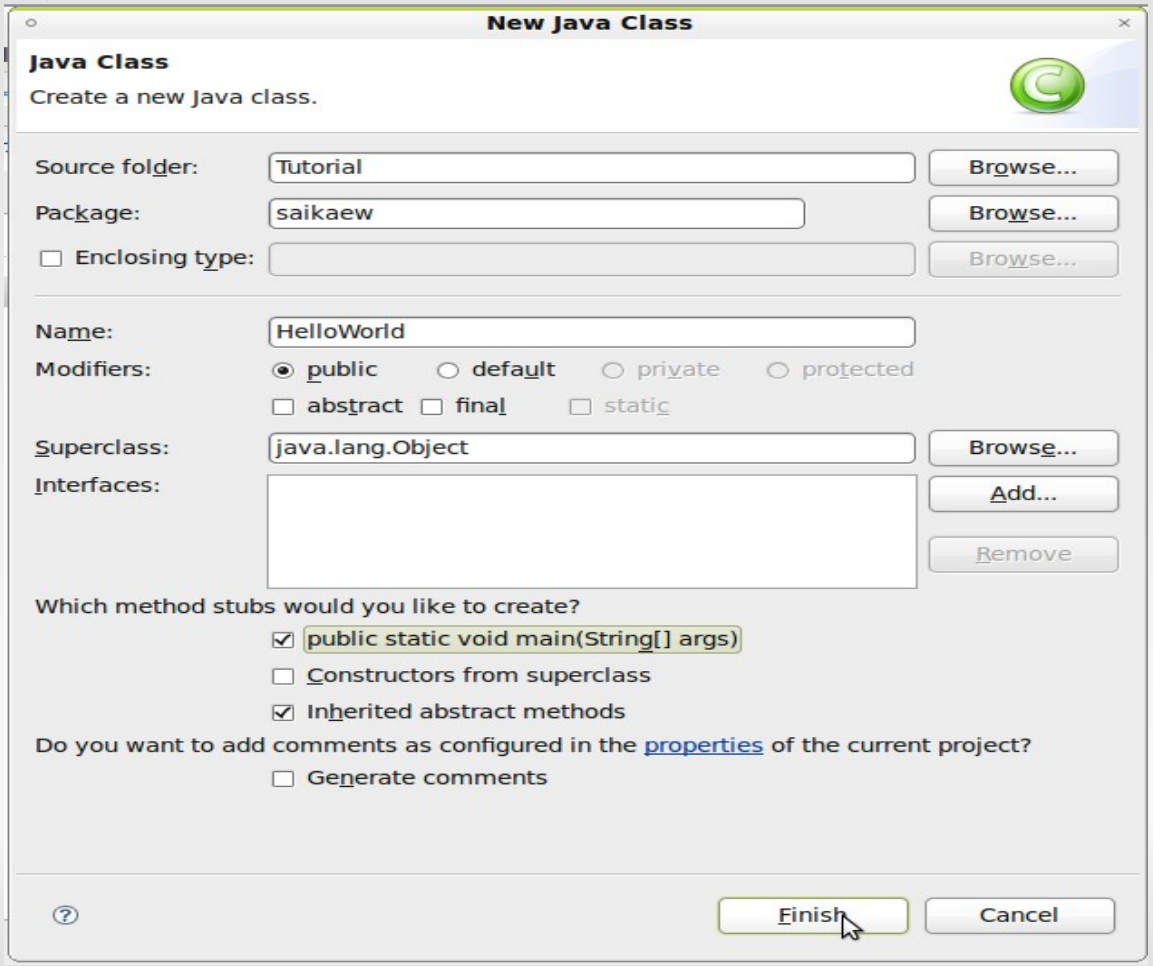
Create a New Class

- Right click at the package and choose Class



Assign Class Name

- Type class name, chose public static void main.., and click Finish



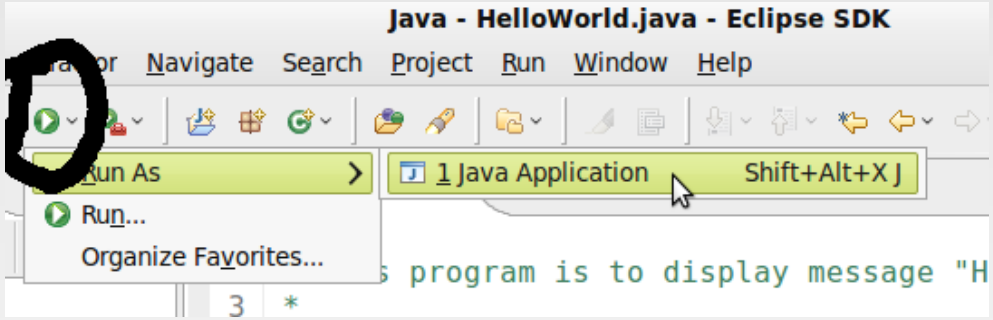
Write the Code with Comments

HelloWorld.java

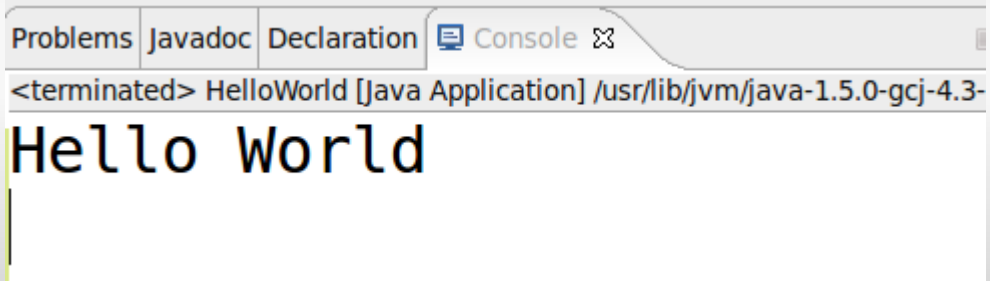
```java
/*
 * This program is to display message "Hello World"
 *
 * It is written by Kanda Saikaew
 * 2009/06/16
 */
package saikaew;

public class HelloWorld {

    /**
     * @param args
     */
    public static void main(String[] args) {
        System.out.println("Hello World");
    }
}
```

Run the Program

- Click at the button that is circled with a black color and then choose Run As > Java Application

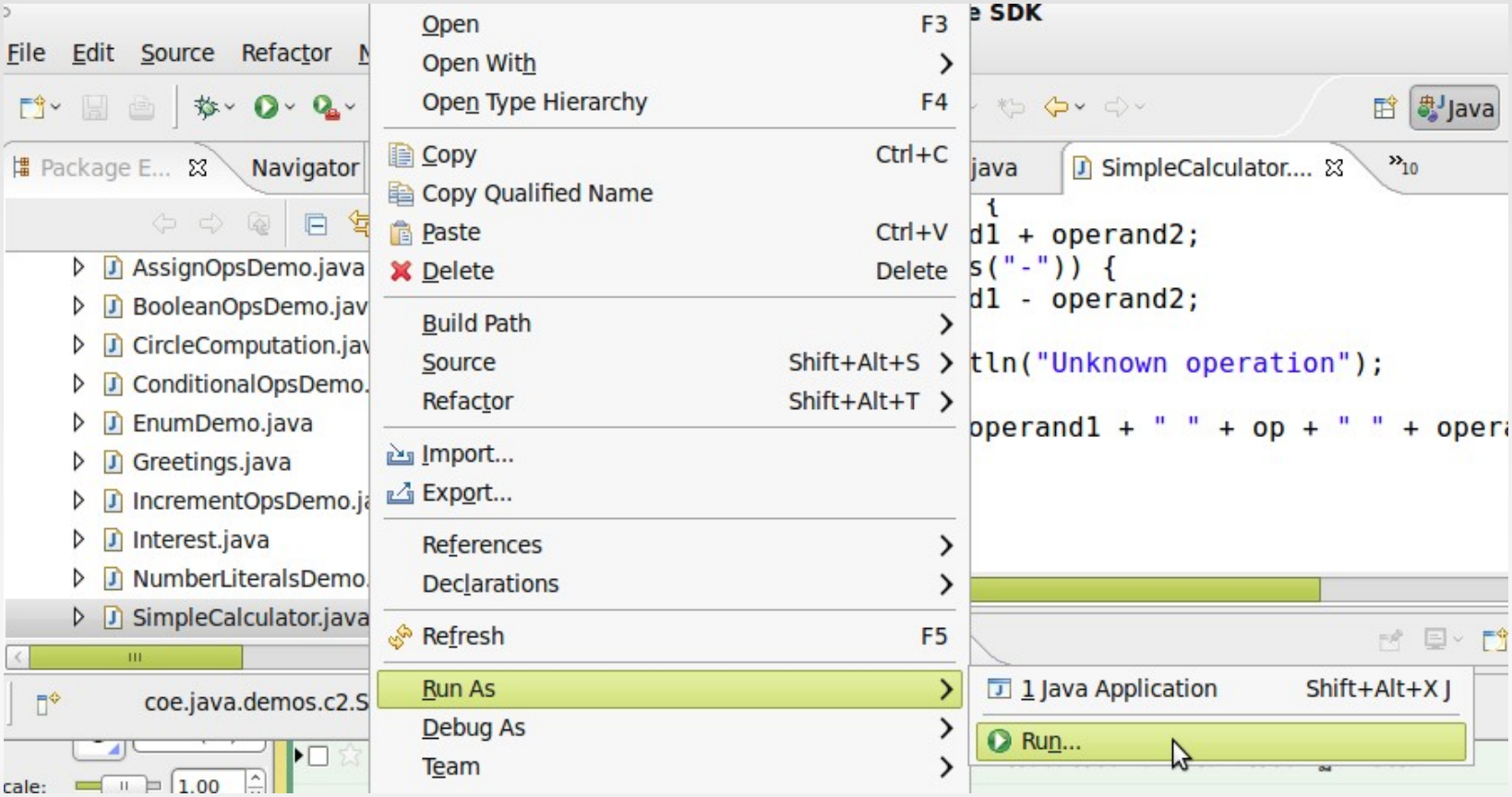


- The program output appears at the console

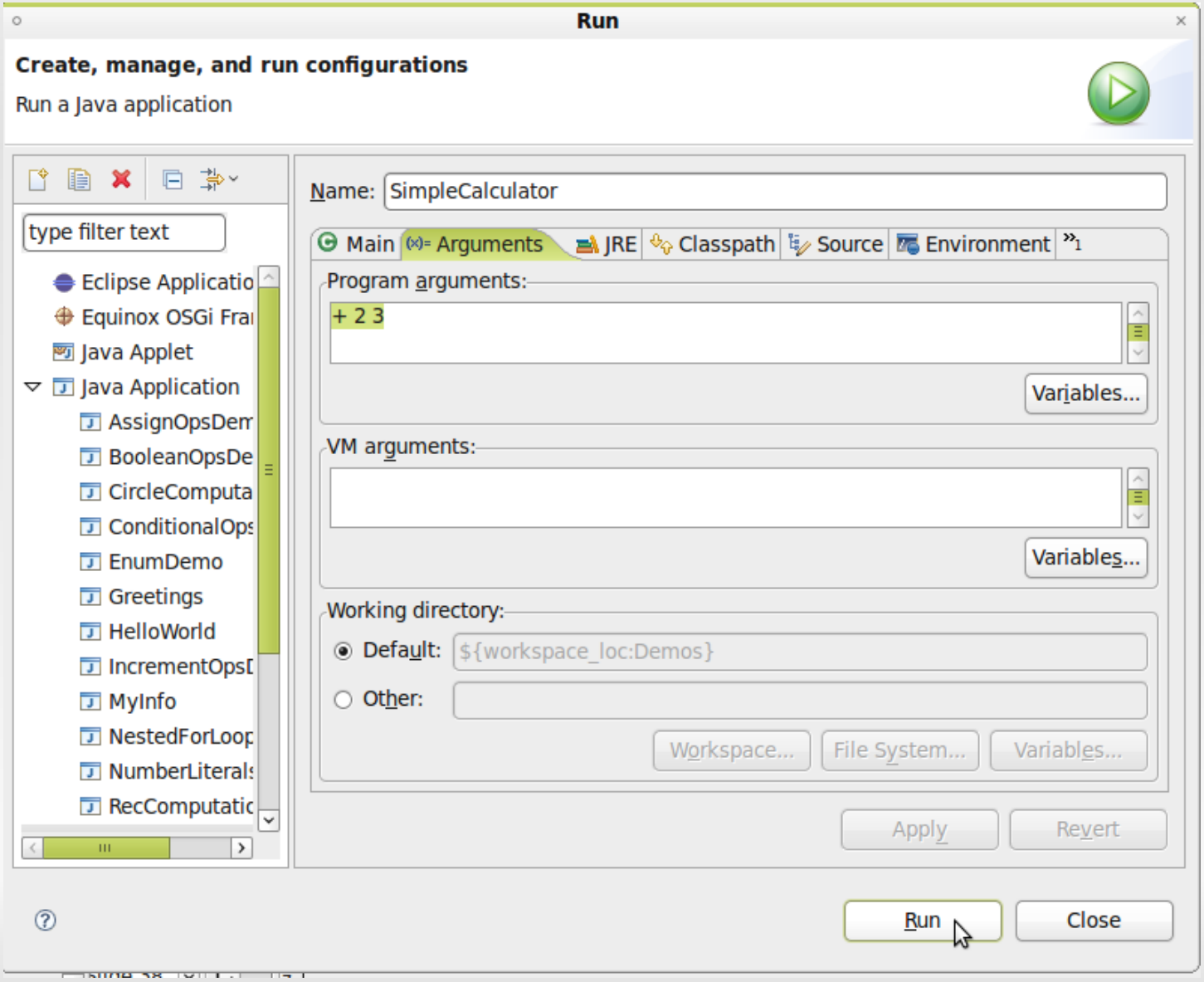


Set the Program Argument

Open the run configuration



Set the Program Argument



Programming Style and Documentation

- Appropriate comments
- Naming conventions
- Proper indentation and spacing lines
- Block styles

Appropriate Comments

- Include a summary at the beginning of the program to explain about the program
  - What the program does and its key features
  - Special techniques it uses
  - Program version
- Include the information about the programmer
  - Your name, class section
  - Date

Naming Conventions

- Choose meaningful and descriptive names
- Capitalize class name
  - public class HelloWorld
- Use lowercase letters for variable and method names
  - int a; void takeClass(String className);
- Use all uppercase letters for a constant
  - final int NUMPROVINCES = 76;

Proper Indentation & Spacing

- Indentation
  - Make an indent codes in the same group indent in the same vertical line
  - Have each statement on each line
- Spacing
  - Use a blank line to separate a group of code

Programming Exercises

- Write a Java program that accepts the command line arguments which include the width and the height of a rectangle. Then, display its circumference and its area
- Sample program output:

java RecComputation 2 3

The circumference of a rectangle with width = 2 and height = 3 is 10 and its area is 6

Group Exercise

1. Form a group of 4 people
2. Develop a Java program in package <member1>.<member2>.<member3>.<member4>

This Java Program is to multiply and divide two double values

It accepts an operator and two operands.

The user needs to choose which operator that he/she will use

References

- David J. Eck, ”Introduction to Programming Using Java”, Version 5.0, December 2006 http://math.hws.edu/javanotes/
- eclipse-tutorial:Developing open source Java applications with java.net and Eclipse, Available at https://eclipse-tutorial.dev.java.net/eclipse-tutorial/part1
- Patrick Bouklee, ”Java Package Tutorial”, Available at
- http://www.jarticles.com/package/package_eng.html