Object-Oriented Language and Theory

Lecturer: NGUYEN Thi Thu Trang, trangntt@soict.hust.edu.vn
Teaching Assistant: DO Minh Hieu, hieudominh@hotmail.com

Lab 1: Environment Preparation and First Programs

Introduction

In this lab, we prepare for the development environment, then we see some simple examples writen in the prepared enviroment. We compile and run the programs on the command line with Java JDK.
Keywords: jdk, java installation, programming text editor

1 Getting Started

1.1 Java Development Kit

Java Platform, Standard Edition Development Kit (JDK) is a development environment for building applications, applets, and components using the Java programming language. Of all the releases, Oracle JDK 8 is recommended.

The installation steps are illustrated as follows.

Step 1: Check if JDK has been pre-installed

1. Open a Command Prompt (on Windows. Press Windows+R to open "Run" box. Type "cmd" and then click "OK" to open a regular Command Prompt) or a Terminal (on Linux or macOS).
2. Issue the following command.
```
$ javac -version
```
3. In case a JDK version number is returned (e.g., JDK x.x.x), then JDK has already been installed. When the JDK version is prior to 1.8, a message "*Command 'javac' not found*", or a message "*'javac' is not recognized as an internal or external command, operable program or batch file.*", proceed to **step 2** to install Oracle JDK 8. Otherwise, proceed to 1.2.
Note: Linux usually chooses OpenJDK as its default JDK since OpenJDK is open source. However, Oracle JDK is not completely compatible with OpenJDK and it is recommended to use Oracle JDK.

Step 2: Download Oracle JDK 8

1. Go to Java SE Development Kit (JDK) 8 download site at the following link.
https://www.oracle.com/java/technologies/javase/javase-jdk8-downloads.html
2. Download the installation file, depended on the operating system, under "*Java SE Development Kit 8u241*" section. The recommended file for Linux is Compressed Archive file. We may need an Oracle Account to download for the Oracle JDK License has changed for releases since April 16, 2019.

Step 3: Install and Configure

- **Windows.**
  1. Install Oracle JDK 8. Run the downloaded installer and follow the instructions.
  2. Configure. Launch Control Panel → System and Security → System → Advanced system settings → Environment Variables in Advanced tab. Then edit or create variable JAVA_HOME to point to where the JDK software is located, e.g., "*C:\Program Files\Java\jdk1.8.0_241*").
- **Linux.**
  1. Create installation directory. We shall install Oracle JDK 8 under "*/usr/local/java*" directory.
```
$ cd /usr/local
```

```
$ sudo mkdir java
```

2. Extract the downloaded package (e.g., jdk-8u241-linux-x64.tar.gz) to the installation directory.

```
$ cd /usr/local/java
$ sudo tar xzvf ~/Downloads/jdk-8u241-linux-x64.tar.gz
// x: extract, z: for unzipping gz, v: verbose, f: filename
```

3. Inform the Linux to use this JDK/JRE

```
// Setup the location of java, javac and javaws
$ sudo update-alternatives --install "/usr/bin/java" "java"
"/usr/local/java/jdk1.8.0_241/bin/java" 1
        // --install symlink name path priority
$ sudo update-alternatives --install "/usr/bin/javac" "javac"
"/usr/local/java/jdk1.8.0_241/bin/javac" 1
$ sudo update-alternatives --install "/usr/bin/javaws" "javaws"
"/usr/local/java/jdk1.8.0_241/bin/javaws" 1


// Use this Oracle JDK/JRE as the default
$ sudo update-alternatives --set java /usr/local/java/jdk1.8.0_241/bin/java
        // --set name path
$ sudo update-alternatives --set javac /usr/local/java/jdk1.8.0_241/bin/javac
$ sudo update-alternatives --set javaws /usr/local/java/jdk1.8.0_241/bin/javaws
```

- **macOS.** Double-click the DMG file and follow the instructions.

Step 4: Verify the JDK Installation. Issue the following command.

```
$ javac -version
```

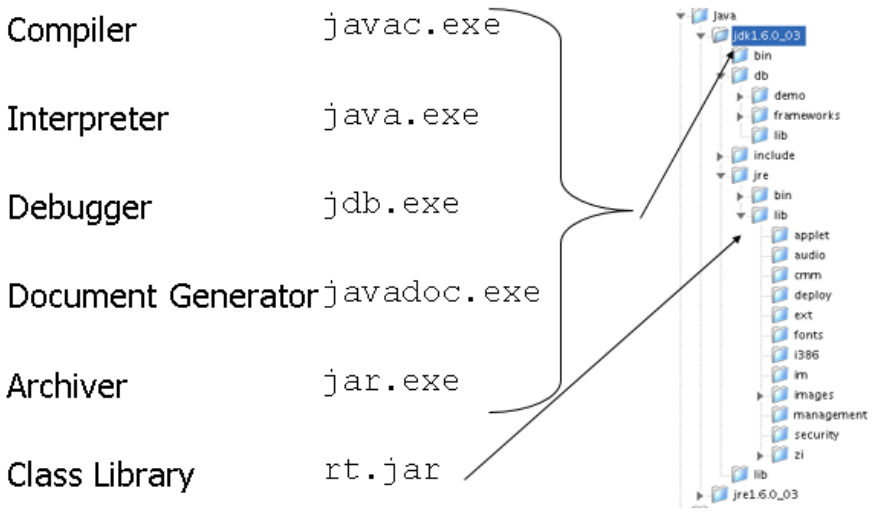

Figure 1-Development Environment

1.2 Programming Text Editor

In this lab, Visual Studio Code works as our development environment. The installation steps are illustrated as follows.

Step 1: Download installer. Go to VS Code download site at the following link and download the suitable installer. https://code.visualstudio.com/download

Step 2: Install Visual Studio Code. Run the downloaded installer and follow the installation instruction.

2 First Programs

2.1 Java Programming Steps

The steps in writing a Java program are illustrated in the following steps and in Figure 2.

Step 1: Write the source code such as the code shown in Figure 3. and save in, e.g., "HelloWorld.java" file.

Step 2: Compile the source code into Java portable bytecode (or machine code) using the JDK's Java compiler by issuing the following command.

```
$ javac HelloWorld.java
```

Step 3: Run the compiled bytecode using the JDK's Java Runtime by issuing the following command.

```
$ java HelloWorld
```

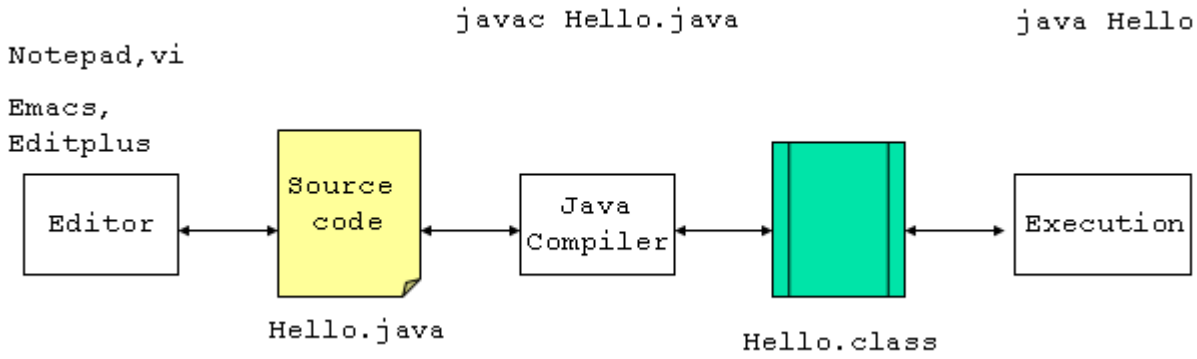

Figure 2-Compile a Java application by command line

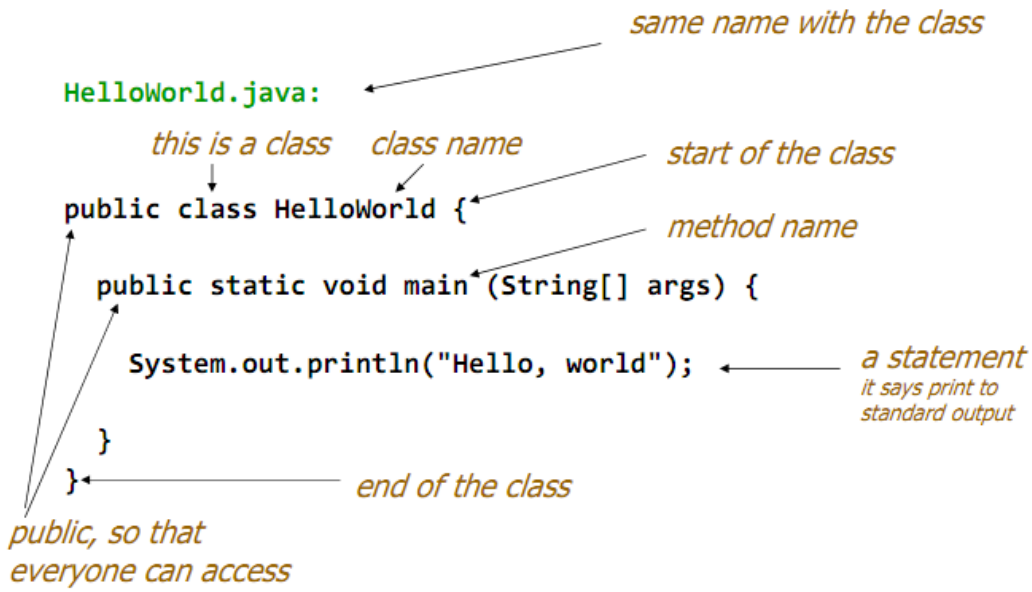

Figure 3-The First Java Application

The result is shown in Figure 4.

```
%> javac HelloWorld.java
%> java HelloWorld
Hello, world
```

Figure 4-Result

For the better illustration, we can watch the following demo videos.
https://www.youtube.com/watch?v=G1ubVOl9IBw
https://www.youtube.com/watch?v=2Xa3Y4xz8_s

2.2 The Very First Java Programs

2.2.1 Write, compile the first Java application:

Step 1: Create a new file. From the Visual Studio Code interface, choose File → New File.

Step 2: Save the file. From the Visual Studio Code interface, choose File → Save. Browse the desired directory, change the file name to "*HelloWorld.java,"* and click "*Save*" button.

Step 3: Write the source code. The source code is shown in Figure 5.

```
1  //Example 1: HelloWorld.java
2  //Text-printing program
3  public class HelloWorld {
4
5     public static void main(String args[]){
6           System.out.println("Xin chao \n cac ban!");
7           System.out.println("Hello \t world!");
8
9     } // end of method main
10 }
```

Figure 5-The First Java Application

Step 4: Compile. On a Command Prompt or a Terminal, change the current working directory[1] into the directory where we have saved the source code. Then issue the following commands**.**

```
$ javac HelloWorld.java
$ java HelloWorld
```

[1] In various operating system, the *cd <desired directory name>* command (*cd* stands for ***c****hange* ***d****irectory*) allows us to change the current working directory to the desired directory. Besides, in Windows 10, to access another drive, we type the drive's letter, followed by ":". For instance, to change the current working drive to drive D, we issue the command "*d:*"

2.2.2 Write, compile the first dialog Java program

Step 1: Create a new file. From the Visual Studio Code interface, choose File → New File.

Step 2: Save the file. From the Visual Studio Code interface, choose File → Save. Browse the desired directory, change the file name to "*FirstDialog.java,"* and click "*Save*" button.

Step 3: Write the source code. The source code is shown in Figure 6

```
1 // Example 2: FirstDialog.java
2 import javax.swing.JOptionPane;
3 public class FirstDialog{
4     public static void main(String[] args){
5         JOptionPane.showMessageDialog(null,"Hello world! How are you?");
6         System.exit(0);
7     }
8 }
```

Figure 6- The First Dialog Java Application

Step 4: Compile. On a Command Prompt or a Terminal, change the current working directory into the directory where we have saved the source code. Issue the following commands.

```
$ javac FirstDialog.java
$ java FirstDialog
```

2.2.3 Write, compile the first input dialog Java application

Step 1: Create a new file. From the Visual Studio Code interface, choose File → New File.

Step 2: Save the file. From the Visual Studio Code interface, choose File → Save. Browse the desired directory, change the file name to "*HelloNameDialog.java,"* and click "*Save*" button.

Step 3: Write the source code. The source code is shown in Figure 7

```
1  // Example 3: HelloNameDialog.java
2  import javax.swing.JOptionPane;
3  public class HelloNameDialog{
4    public static void main(String[] args){
5      String result;
6      result = JOptionPane.showInputDialog("Please enter your name:");
7      JOptionPane.showMessageDialog(null, "Hi "+ result + "!");
8      System.exit(0);
9    }
10 }
```

Figure 7- The First Input Dialog Java Application

Step 4: Compile. On a Command Prompt or a Terminal, change the current working directory into the directory where we have saved the source code. Issue the following commands.

```
$ javac HelloNameDialog.java
$ java HelloNameDialog
```

2.2.4 Write, compile, and run the following example:

Step 1: Create a new file. From the Visual Studio Code interface, choose File → New File.

Step 2: Save the file. From the Visual Studio Code interface, choose File → Save. Browse the desired directory, change the file name to "*ShowTwoNumbers.java,"* and click "*Save*" button.

Step 3: Write the source code. The source code is shown in Figure 8

```
1  // Example 5: ShowTwoNumbers.java
2  import javax.swing.JOptionPane;
3  public class ShowTwoNumbers {
4    public static void main(String[] args){
5      String strNum1, strNum2;
6      String strNotification = "You've just entered: ";
7
8      strNum1 = JOptionPane.showInputDialog(null,
9                  "Please input the first number: ","Input the first number",
10                 JOptionPane.INFORMATION_MESSAGE);
11     strNotification += strNum1 + " and ";
12
13     strNum2 = JOptionPane.showInputDialog(null,
14                 "Please input the second number: ","Input the second number",
15                 JOptionPane.INFORMATION_MESSAGE);
16     strNotification += strNum2;
17
18     JOptionPane.showMessageDialog(null,strNotification,
19                 "Show two numbers", JOptionPane.INFORMATION_MESSAGE);
20     System.exit(0);
21   }
22 }
```

Figure 8-Java Application showing two entered numbers and their sum

Step 4: Compile. On a Command Prompt or a Terminal, change the current working directory into the directory where we have saved the source code. Issue the following commands.

```
$ javac ShowTwoNumbers.java
$ java ShowTwoNumbers
```

2.2.5 Write a program to calculate sum, difference, product, and quotient of 2 double numbers which are entered by users.

Notes

- To convert from String to double, you can use

  ```
  double num1 = Double.parseDouble(strNum1)
  ```

- Check the divisor of the division

2.2.6 Write a program to solve:

- The first-degree equation (linear equation) with one variable

Note: A first-degree equation with one variable can have a form such as $ax + b = 0$ $(a \neq 0)$, where x is the variable, and a and b are coefficients. Given a and b, the equation has a unique solution $x = -\frac{b}{a}$.

Additionally, we only consider real number in this task for simplicity.

- The system of first-degree equations (linear system) with two variables

Note: A system of first-degree equations with two variables x_1 and x_2 can be written as follows.

$$\begin{cases} a_{11}x_1 + a_{12}x_2 = b_1 \\ a_{21}x_1 + a_{22}x_2 = b_2 \end{cases}$$

Given the coefficients a_{11}, a_{12}, a_{21}, a_{22}, b_1, and b_2, we derive the following determinants.

$$\mathrm{D} = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} = a_{11}a_{22} - a_{21}a_{12}$$

$$\mathrm{D}_1 = \begin{vmatrix} b_1 & a_{12} \\ b_2 & a_{22} \end{vmatrix} = b_1a_{22} - b_2a_{12}$$

$$\mathrm{D}_2 = \begin{vmatrix} a_{11} & b_1 \\ a_{21} & b_2 \end{vmatrix} = a_{11}b_2 - a_{21}b_1$$

In case $D \neq 0$, the system has a unique solution $(x_1, x_2) = \left(\frac{D_1}{D}, \frac{D_2}{D}\right)$.

In case $D = 0$, the system has infinitely many solutions if $D_x = D_y = D = 0$.

Otherwise, the system has no solution.

Additionally, we only consider real number in this task for simplicity.

- The second-degree equation with one variable

Note: A second-degree equation with one variable (i.e., quadratic equation) can have a form such as $ax^2 + bx + c = 0$, where x is the variable, and a, b, and c are coefficients $(a \neq 0)$. Given a, b, and c, the equation has the discriminant $\Delta = b^2 - 4ac$. In case $\Delta = 0$, the equation has double root $-\frac{b}{2a}$. If $\Delta > 0$, the equation has two distinct roots $\frac{-b+\sqrt{\Delta}}{2a}$ and $\frac{-b-\sqrt{\Delta}}{2a}$. Otherwise, the equation has no solution.

Additionally, we only consider real number in this task for simplicity.

3 Assignment Submission

You must put **all** the six programs of this lab, written by yourself, into a directory namely "*Lab01*" and push it to your master branch of the valid repository before the deadline announced in the class.

Each student is expected to turn in his or her own work and not give or receive unpermitted aid. Otherwise, we would apply extreme methods for measurement to prevent cheating.

4 References

Hock-Chuan, C. (2020, January). *How to Install JDK 13 (on Windows, macOS & Ubuntu) and Get Started with Java Programming*. Retrieved from Nanyang Technological University: https://www3.ntu.edu.sg/home/ehchua/programming/howto/JDK_HowTo.html