Object-Oriented Language and Theory

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

Lab 2: Java basics and UML

In this lab, we continue with Java basics, IDE, and UML.

1 UML – Use case diagram

1.1 Introduction to UML & Use case Diagram

The Unified Modeling Language (UML) is a family of graphical notations, backed by single metamodel, that help in describing and designing software systems, particularly software systems built using the object-oriented style (Fowler, 2003).

A use case diagram is one of the UML diagrams to captures dynamic behaviors, i.e., the *pattern of state change over time* (https://www.cs.uct.ac.za/mit_notes/software/htmls/ch05s08.html). The use case diagram illustrates the relations among use cases. Use cases work by describing the *typical interactions* between the users of a system and the system itself (Fowler, 2003).

For better understanding, see https://www.uml-diagrams.org/use-case-diagrams.html.

1.2 Astah UML

Astah is a design tool which supports UML. To get Astah UML, go to http://astah.net/student-license-request, fill the form, and send the request. Then follow the 3 steps in the redirected page http://astah.net/student/thank-you. See use case diagram with Astah at http://astah.net/manual/422-usecase-diagram.

1.3 AIMS Project

Draw the usecase diagram for the following system.

There might be a future that Tiki and Sendo be in talks over a potential merger to contend other e-commerce platforms and especially those who have foreign backers. The merger of these two firms would create a Ti-do company, where "Ti" is from Tiki, and "do" is from Sendo, which means a billion-dollar company in Vietnamese. That firm, Ti-do company, would like you to help them create a brand-new system for AIMS project (AIMS stands for An Internet Media Store).

- AIMS system allows user to create a limited number of orders, e.g., 5 orders. When a user creates an order, he or she must provide the information of the media. There are currently three types of media: book, compact disc (CD), and digital video disc (DVD). Also, the system records the date and time of the order creation.
    - When a user adds a book to an order, the system needs supplying with its ID, title, category, list of authors, contents, and the price. The name of an author must be unique in author list of a book.
    - When a user adds a CD to an order, he or she must provide its ID, title, category, artist, director, track list, and the price for the CD. Additionally, each track is unique in a CD with its own title and length, which are also provided by the user. The length of a CD is sum of the lengths of its tracks. When the user adds a track, he or she can choose to play that track, i.e., the system displays track's name and its length. After adding all the tracks of a CD, the user can also choose to play that CD, i.e., the system displays the CD information (i.e., CD title and CD length) and plays all the tracks of the CD.
    - When a user adds a DVD to an order, he or she must provide its ID, title, category, director, length, and the price. After adding a DVD, the user can also choose to play that DVD, i.e., the system displays title and length of the DVD.

- When a user removes an item from the current order, he or she can provide either its ID or title. If the item is found, display information of removed item. Or else, notify the user the item is not found in the current order.
- When a user wants to see the current order, the system display all the information of the items.
    - For books, the system shows their ID, title, category, author list, the content length (i.e., the number of tokens), the token list in alphabet order, and the word frequency of the content. For instance, the content of a book is "*I can can the can, but the can cannot can me*", of which the content length is 6.

Token	*but*	*can*	*cannot*	*i*	*me*	*the*
Frequency	1	5	1	1	1	2

    - For CDs, the system displays CD information (i.e., ID, CD title, category, artist, director, CD length, and the price for the CD) and then plays all the tracks in all CDs in the order. The CDs have the more tracks will be play first. In case CDs have the same number of tracks, the longer CD is the chosen one. To illustrate, the system display in the following pattern.
        1. CD1 information … Total 13.3-minute long
            Information of Track 1 in CD1
            Information of Track 2 in CD1
            Information of Track 3 in CD1
            Information of Track 4 in CD1
        2. CD2 information … Total 14.2-minute long
            Information of Track 1 in CD2
            Information of Track 2 in CD2
            Information of Track 3 in CD2
        3. CD3 information … Total 13.3-minute long
            Information of Track 1 in CD3
            Information of Track 2 in CD3
            Information of Track 3 in CD3
    - For DVDs, the system plays all the DVDs in the alphabet order by title. In case they have the same title, the DVDs have the higher cost will be played first. If the DVDs also share the same cost beside the title, the longer DVD will be put in the higher priority.
- To increase consumer demand for the product and grow sale, users can get an item for free which is randomly picked out in the order by the system.
- If a track or a DVD has the length 0 or less, the system must notify the user that the track, the DVD or the CD of that track cannot be played.

2 Introduction to Eclipse / Netbean

In previous lab, we have written our very first Java applications in a programming text editor such as Visual Studio Code. From this lab forward, we use an integrated development environment, so called IDE, which is like a text editor, but provides various features such as modifying, compiling, and debugging software. Some of the most popular IDEs for Java are JetBrains IntelliJ, NetBeans, and Eclipse. In this course, we use Eclipse for our demonstrations.

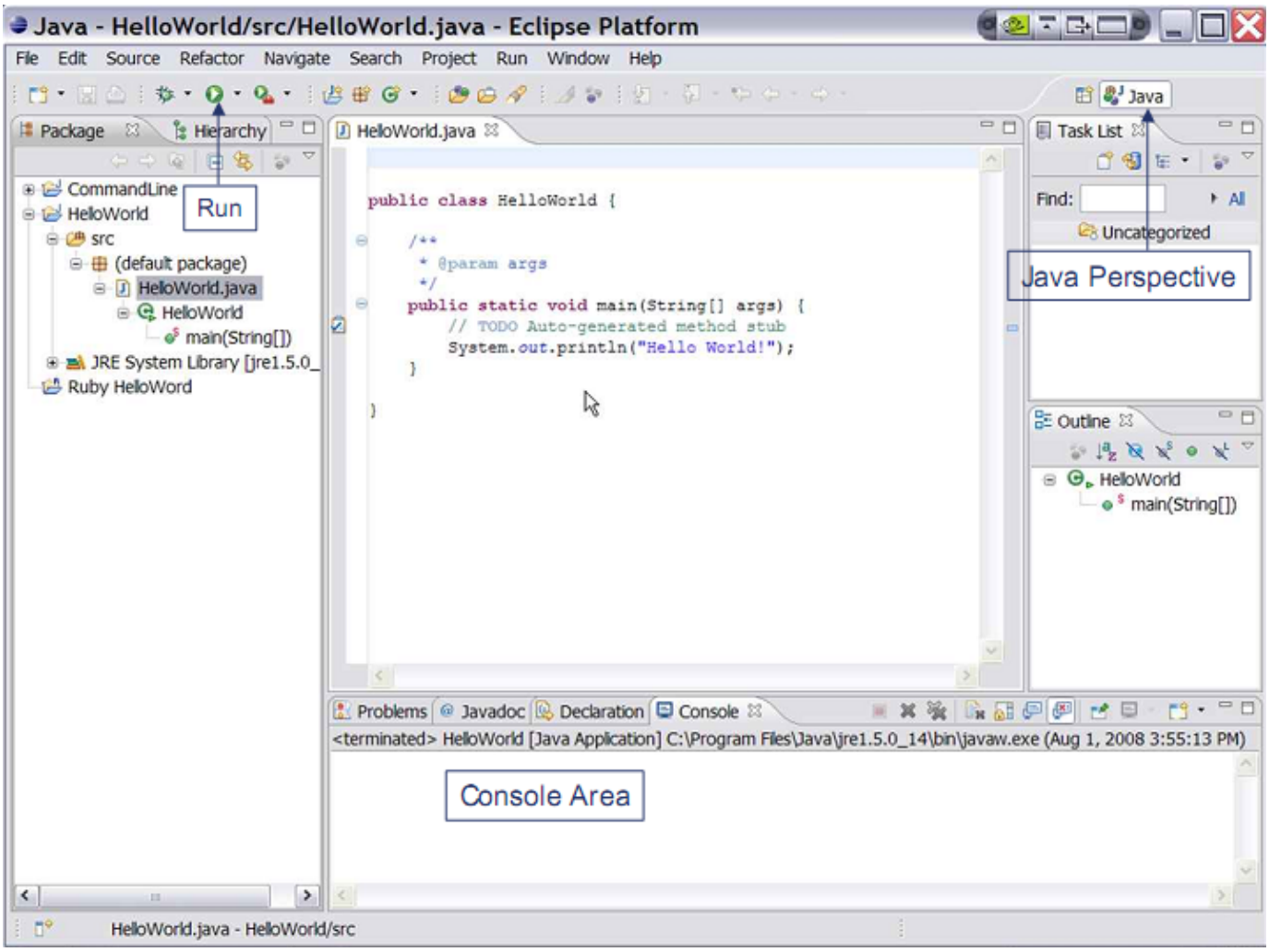


Figure 1-Eclipse IDE

Installation guide:

Note: You should install Java 8 or a later version before installing an IDE.

In this instruction guide, we need no installer; we just download the ZIP file and unzip them.

- Netbeans: Download the binary file at the following link. Read README.html for more details. The application is inside the ***bin*** **directory.**

https://www.apache.org/dyn/closer.cgi/netbeans/netbeans/11.2/netbeans-11.2-bin.zip

If you want to use pre-Apache Netbeans versions, you can see them here (this may not compatible with later Java version).

- Eclipse: We recommend ***Eclipse IDE for Enterprise Java Developers***. Download the suitable binary file at the following link. https://www.eclipse.org/downloads/packages/

3 Javadocs help:

- Open index.html in the docs folder (download from https://www.oracle.com/technetwork/java/javase/documentation/jdk8-doc-downloads-2133158.html)

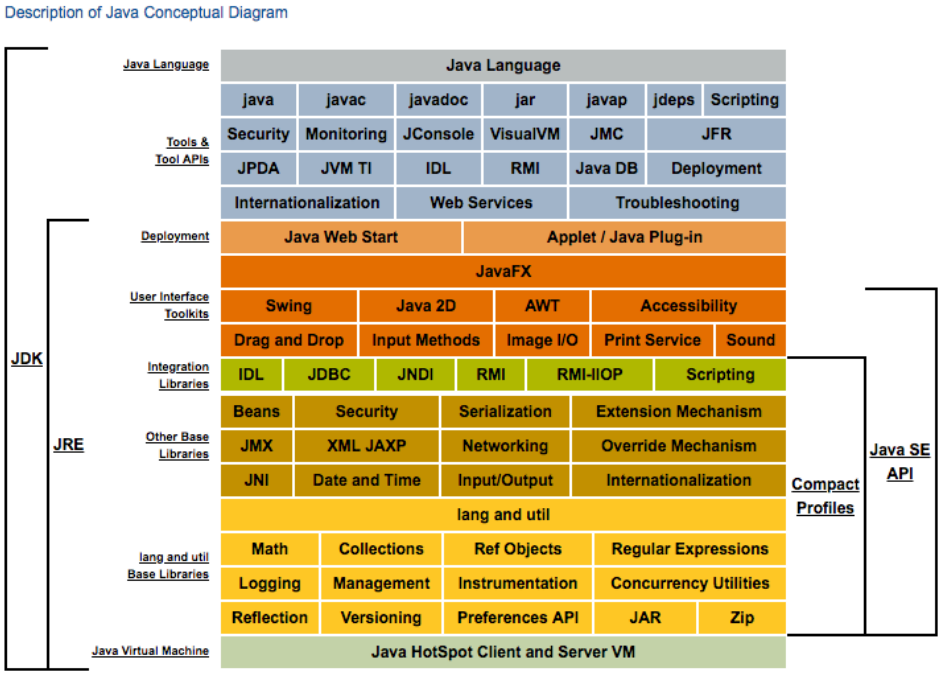


Figure 2-Java Conceptual Diagram

- Click the link Java SE API

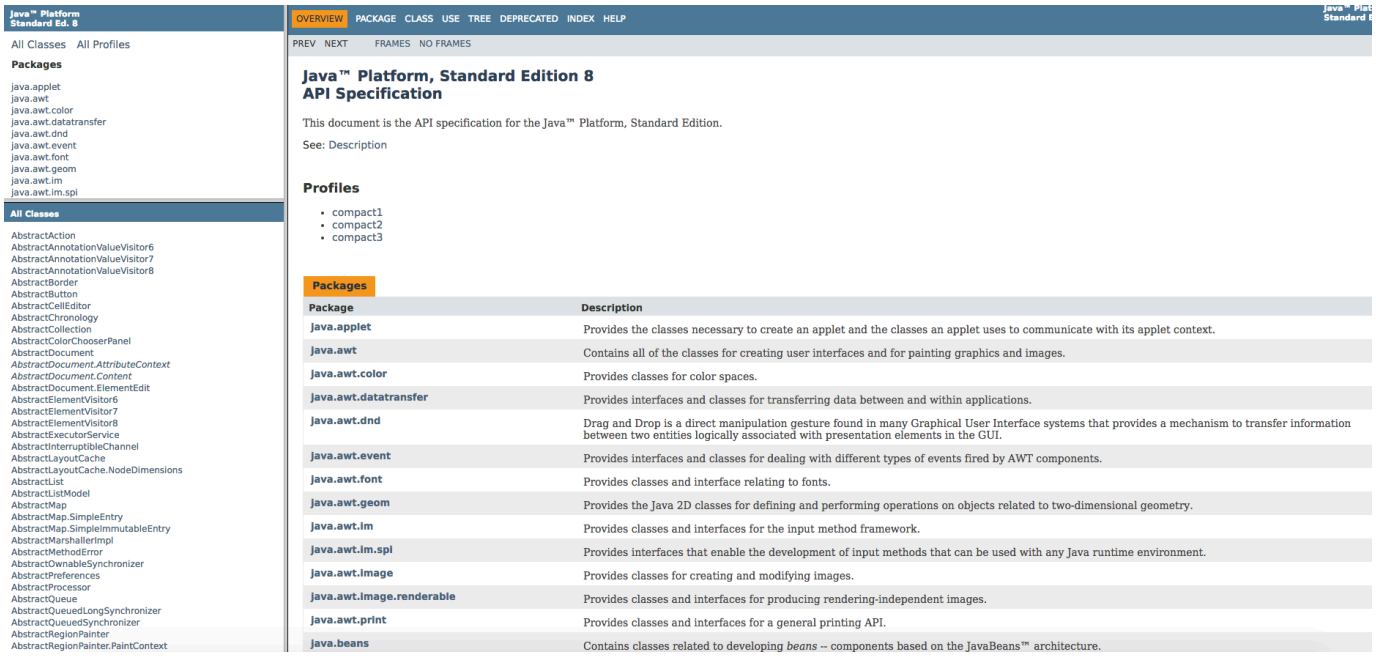


Figure 3-Java SE API

- The top left frame: all packages in Java API
- The bottom left frame: corresponding classes in the chosen above package
- The right frame: Detail information
- Click to a frame, and find the necessary information (Ctrl + F)

4 Your first Java project

1. From the Eclipse install directory, run Eclipse IDE.
2. In Eclipse IDE Launcher window, choose your workspace directory where you want to save the project(s). If you want to use the chosen directory as the default, check the box. Then, click *Launch* button.

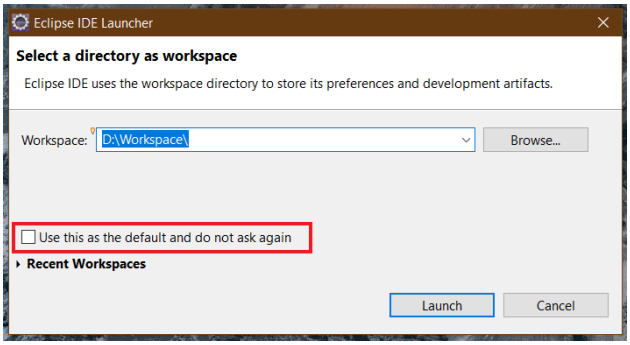


Figure 4-Eclipse Launcher Window

3. To create a new Java project, choose *File → New →Project…*

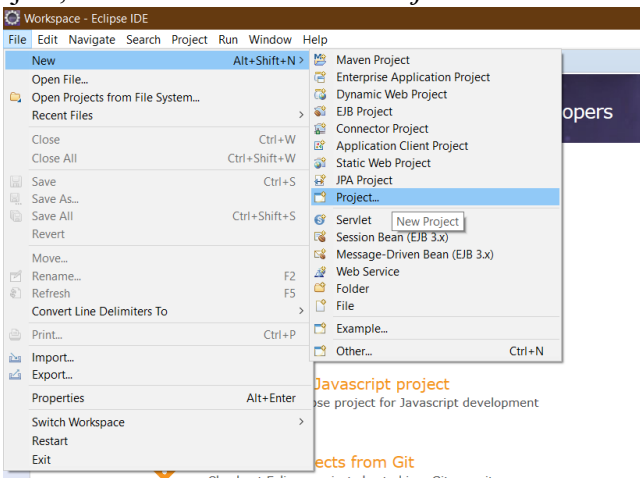


Figure 5-Create new Java project

4. On the pop-up window, choose *Java Project,* then click *Next* > button. If you cannot find it, type the filter text.

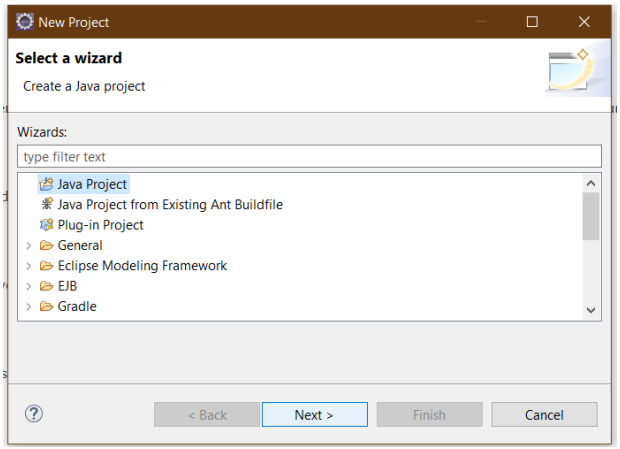


Figure 6-New Project Window

5. On the *New Java Project* window, let the *Project name* be "**JavaBasics**". Then, click *Finish* button.

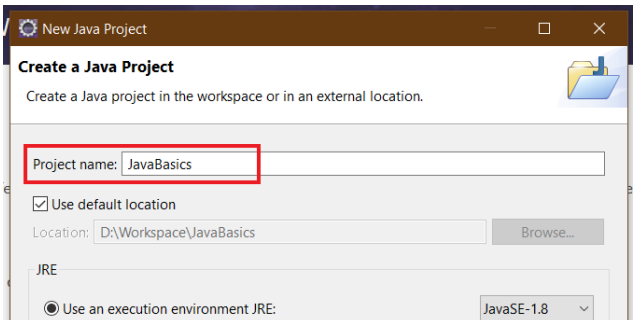


Figure 7-New Java Project Window

6. On the pop-up window, choose *Open Perspective*.

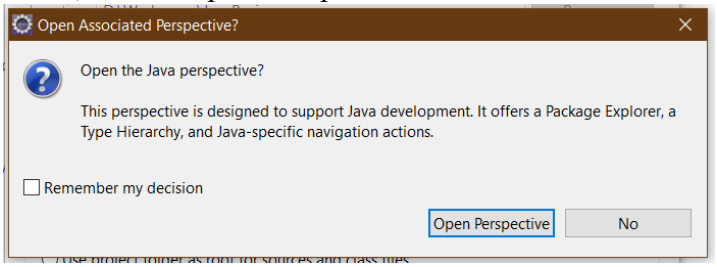


Figure 8-Open Associated Perspective Window

7. Close the Welcome page; then the Java perspective shows up.

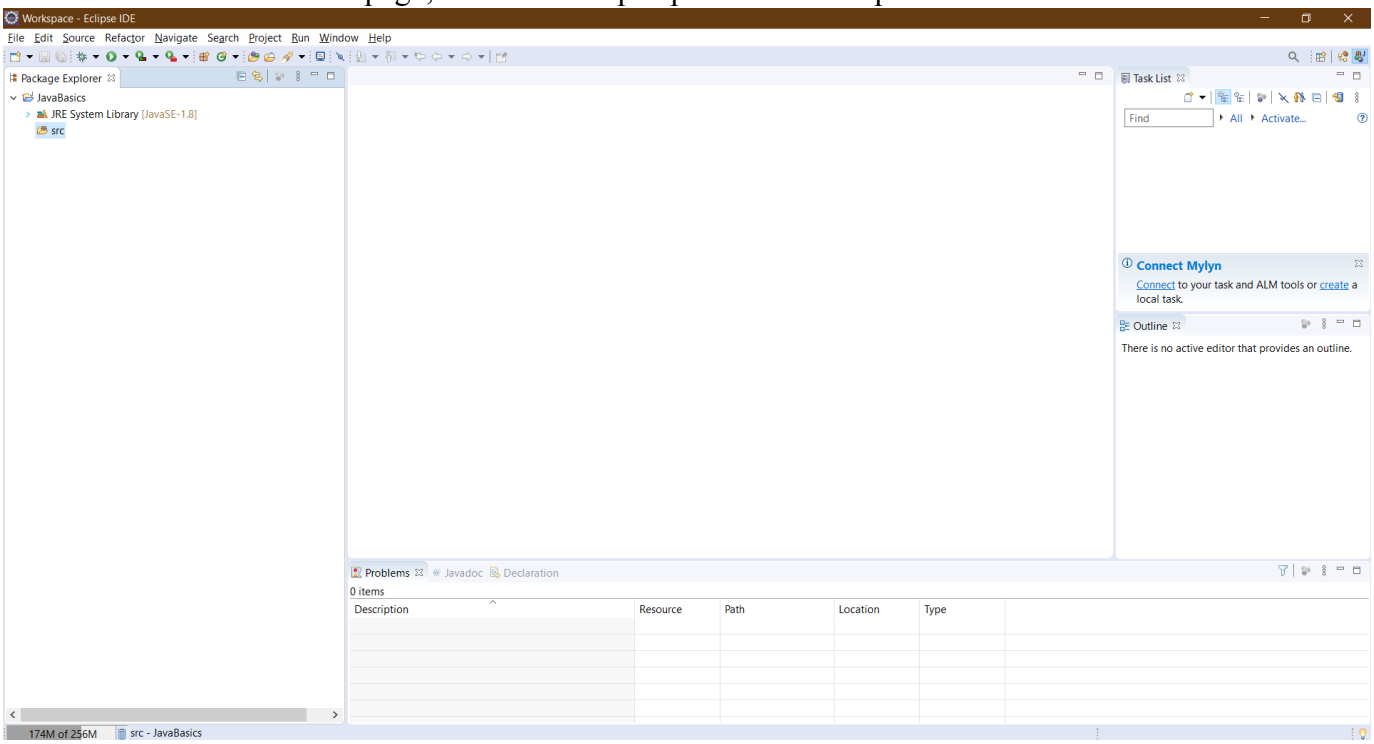


Figure 9-Java Perspective

5 Exercises

5.1 Write, compile and run the ChoosingOption program:

Note: We use JavaBasics project for this exercise.

Step 1: Create a class.

- Choose *File → New → Class*

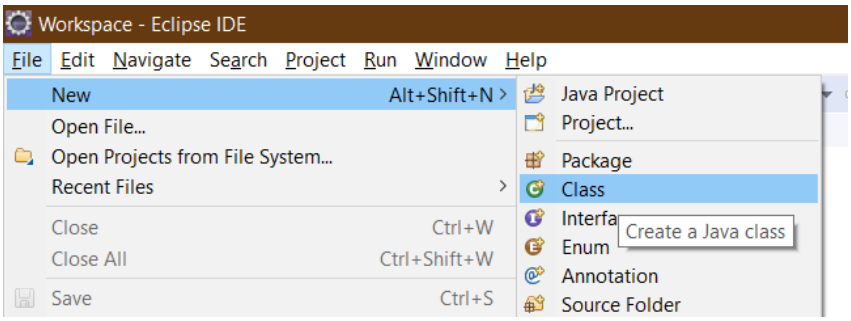


Figure 10-Class creating

- On the pop-up window, set the *Name* same as the class name in the Figure 13, which is "**ChoosingOption**"

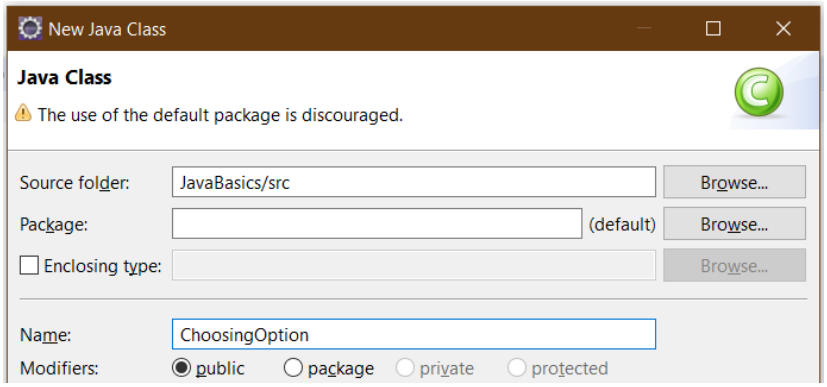


Figure 11-New Java Class Window

We have a new class namely *ChoosingOption* created as shown in the Figure 12.

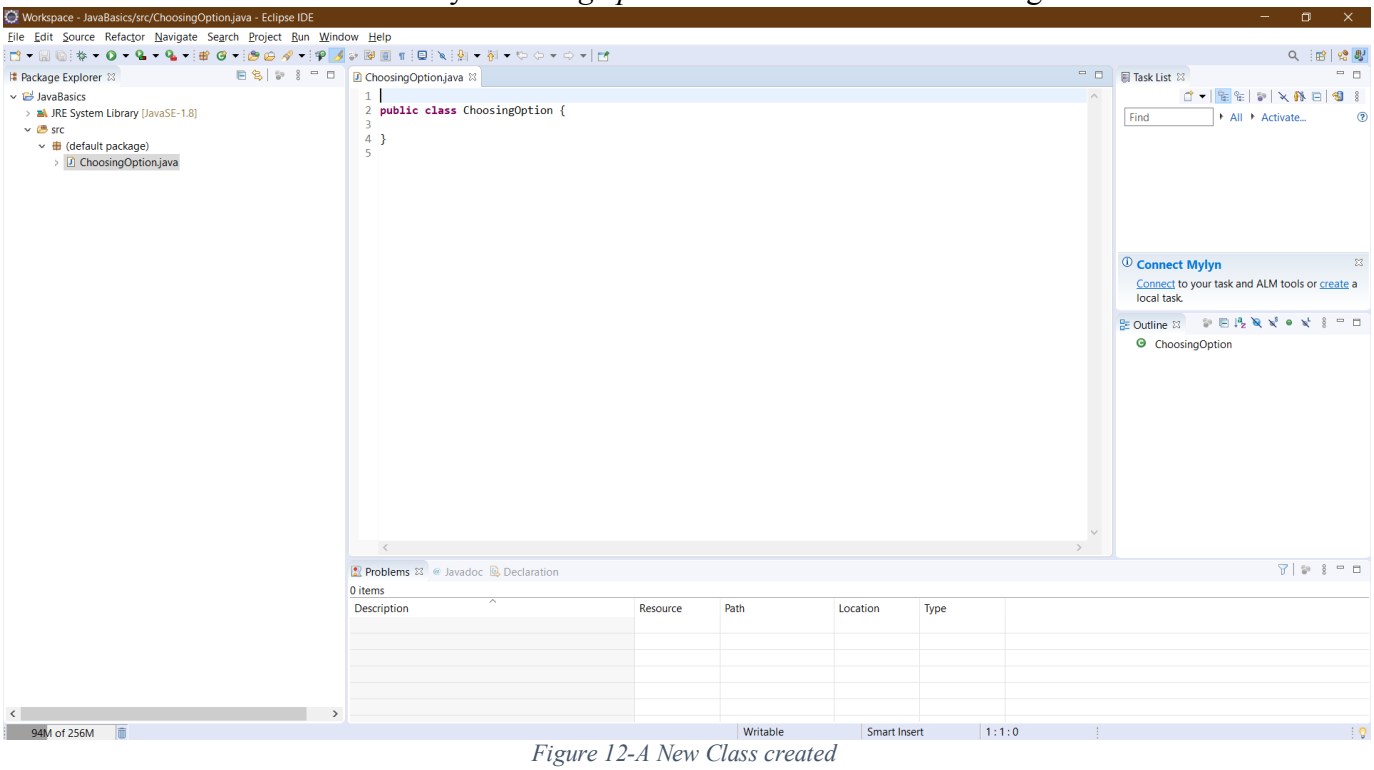


Figure 12-A New Class created

Step 2: Write the program. The source code is illustrated in the Figure 13.

```java
import javax.swing.JOptionPane;
public class ChoosingOption{
  public static void main(String[] args){
    int option = JOptionPane.showConfirmDialog(null,
                "Do you want to change to the first class ticket?");

    JOptionPane.showMessageDialog(null,"You've chosen: "
                    + (option==JOptionPane.YES_OPTION?"Yes":"No"));
    System.exit(0);
  }
}
```

Figure 13-Choosing Option Application

Step 3: Save and Launch.

- Right-click on the *ChoosingOption* class →*Run As* →*Java Application*

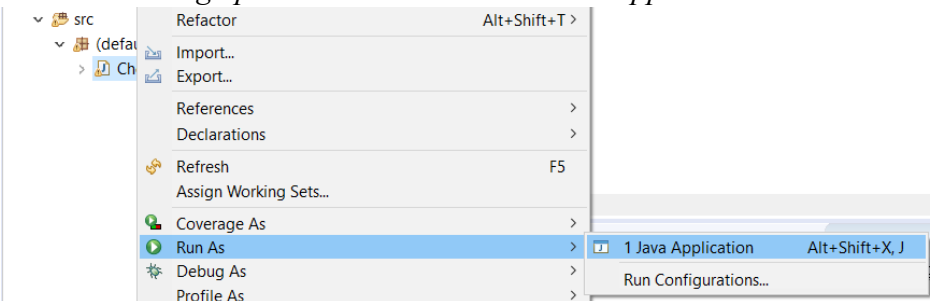


Figure 14-Run Application (1)

- Choose *Always save resources before launching*, then click *OK* button

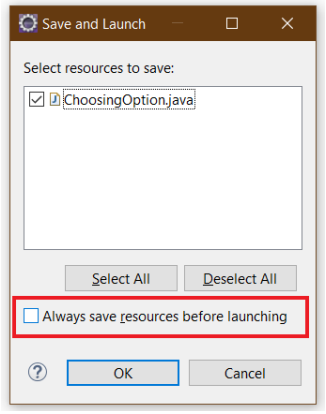


Figure 15-Save and Launch

Questions:

- What happens if users choose "Cancel"?
- How to customize the options to users, e.g. only two options: "Yes" and "No", OR "I do" and "I don't" (Suggestion: Use Javadocs or using Eclipse/Netbean IDE help).

5.2 Write a program for input/output from keyboard

Note: We use JavaBasics project for this exercise.

Step 1: Create a class.

- Choose *File* → *New* → *Class*

- On the pop-up window, set the *Name* same as the class name in the Figure 17Figure 13, which is "**InputFromKeyboard**"

Step 2: Write the program. The source code is illustrated in the Figure 17.

Step 3: Save and Launch.

- Method 1: Right-click on the *InputFromKeyboard* class →*Run As* →*Java Application.*
- Method 2: Click the button and choose the application as shown in the Figure 16

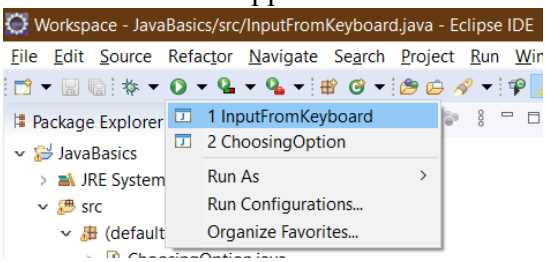


Figure 16-Run Application (2)

```
1  import java.util.Scanner;
2  public class InputFromKeyboard{
3      public static void main(String args[]){
4          Scanner keyboard = new Scanner(System.in);
5
6          System.out.println("What's your name?");
7          String strName = keyboard.nextLine();
8          System.out.println("How old are you?");
9          int iAge = keyboard.nextInt();
10         System.out.println("How tall are you (m)?");
11         double dHeight = keyboard.nextDouble();
12
13         //similar to other data types
14         //nextByte(), nextShort(), nextLong()
15         //nextFloat(), nextBoolean()
16
17         System.out.println("Mrs/Ms. " + strName + ", " + iAge + " years old. "
18                            + "Your height is " + dHeight + ".");
19
20     }
21 }
```

Markers | Properties | Servers | Data Source Explorer | Snippets | Problems | Console | Searc

```
<terminated> InputFromKeyboard [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_171.jdk/Contents/Home/bin/
What's your name?
Trang
How old are you?
35
How tall are you (m)?
1.65
Mrs/Ms. Trang, 35 years old. Your height is 1.65.
```

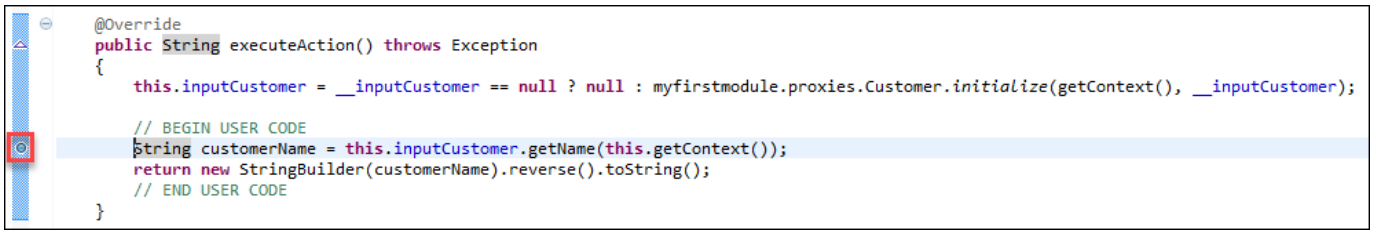
Figure 17-InputFromKeyboard Application

5.3 Use debug to run step by step or go to a checkpoint in a program

Video: https://www.youtube.com/watch?v=9gAjIQc4bPU&t=8s

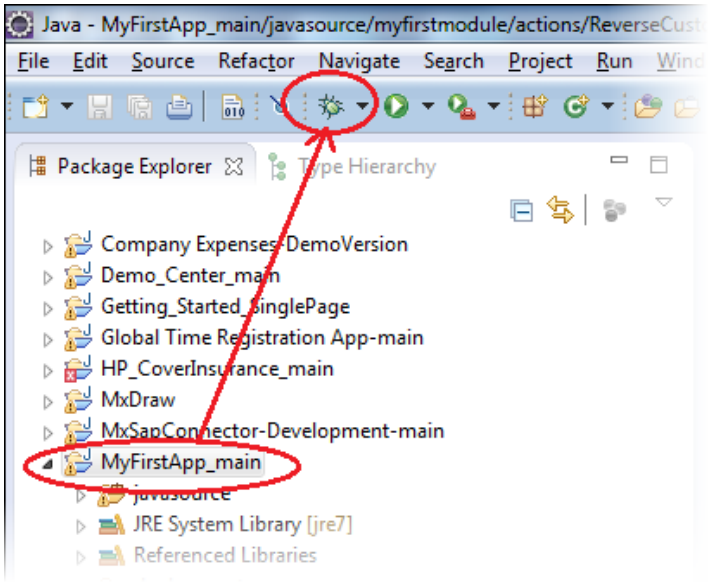
5.3.1 Setting breakpoints:

Place the cursor on the line that needs debugging, hold down Ctrl+Shift, and press B to enable a breakpoint. A blue dot in front of the line will appear.



5.3.2 Debugging in Eclipse:

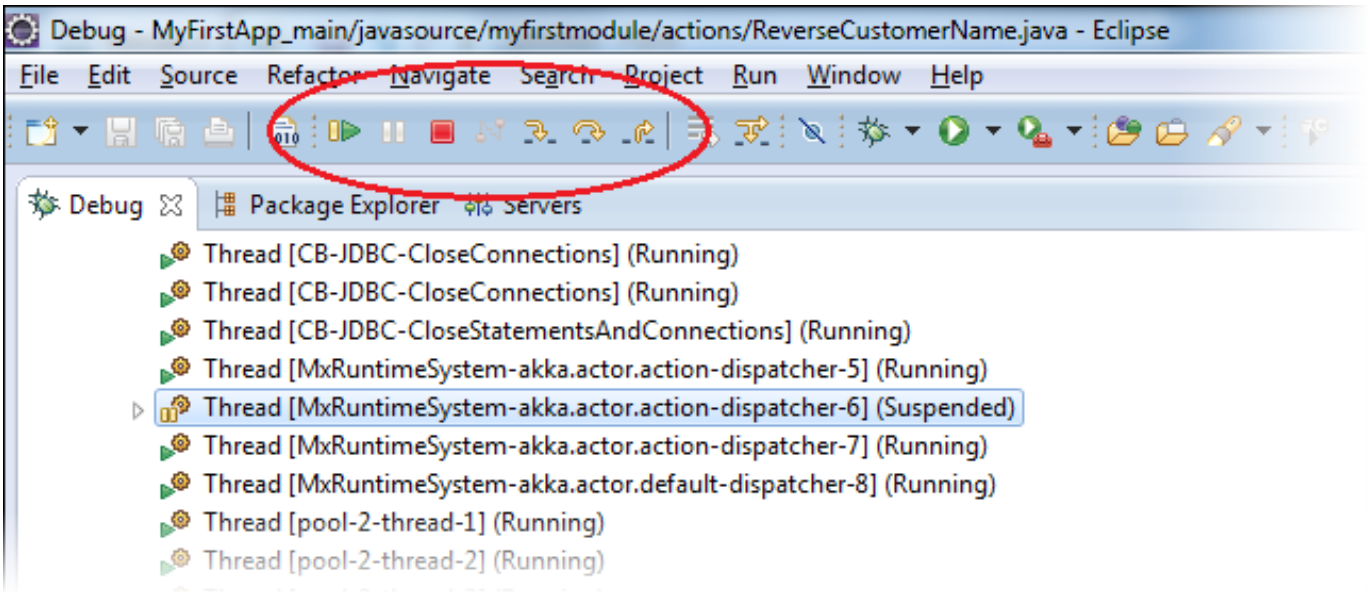
Select the project root node in the package explorer and click the debug icon in the Eclipse toolbar. The application will now be started with Eclipse attached as debugger.



- As soon as the deployment process is ready, open the application in your browser and trigger the Java action:
    - As an end-user of the application, you will see a progress bar on your application
    - As a developer, you will see the Eclipse icon flashing on the Windows task bar
- Open Eclipse. You should now see the “debug” perspective of Eclipse.

5.3.3 Step into or Step over or Step return/Resume

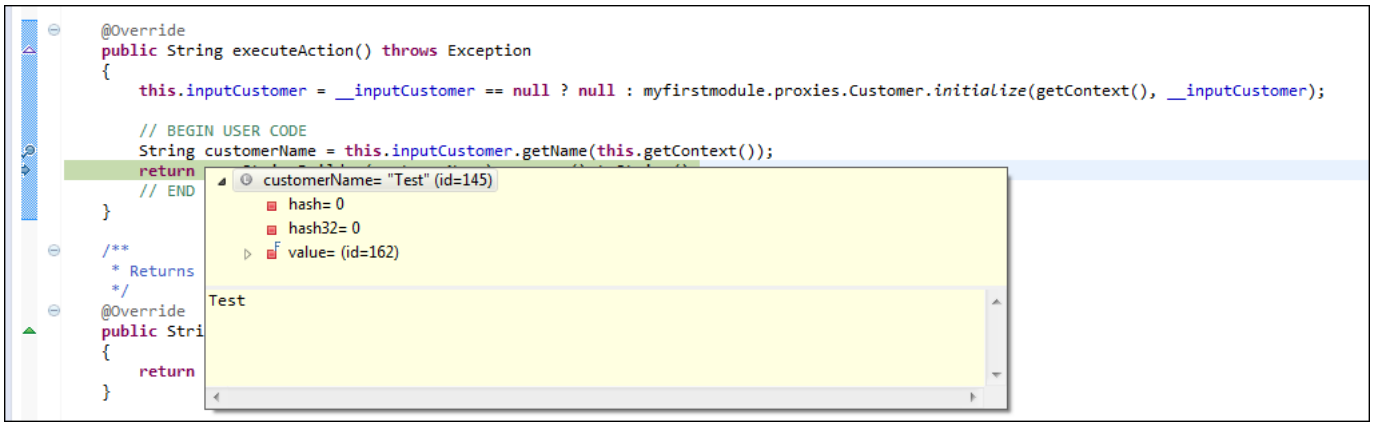
- Click Step into (or press F5) or Step over (or press F6) to move on the next step in the microflow.



- With debugger options, the difference between "Step into" and "Step over" is only noticeable if you run into a function call :
    - "Step into" (F5) means that the debugger steps into the function
    - "Step over" (F6) just moves the debugger to the next line in the same Java action
- With "Step Return" (pressing F7), you can instruct the debugger to leave the function; this is basically the opposite of "Step into."
- Clicking "Resume" (F8) instructs the debugger to continue until it reaches another breakpoint.

5.3.4 Popup window

Place your cursor on any of the variables in the Java action to see its value in a pop-up window.



5.4 Write a program to display a triangle with a height of n stars (*), n is entered by users.

E.g. n=5:

```
    *
   ***
  *****
 *******
*********
```

Note: You must create a new Java project for this exercise.

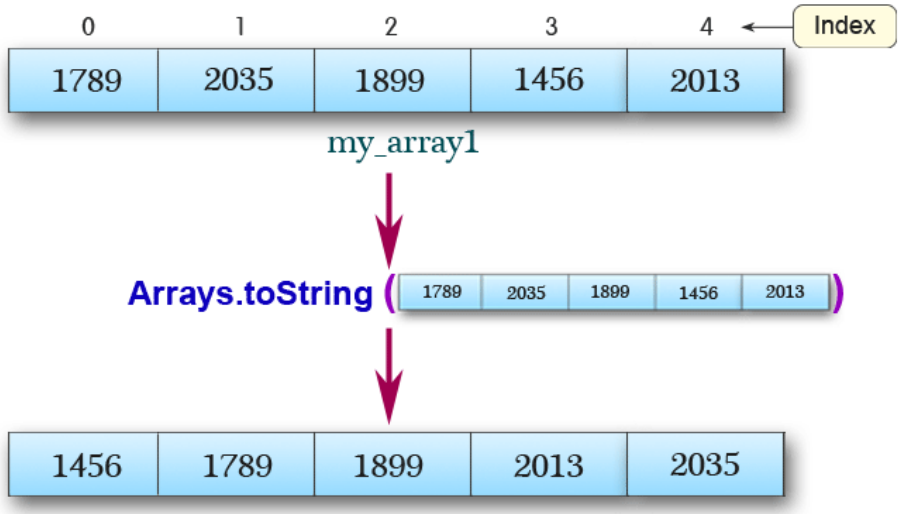
5.5 ***Write a program to display the number of days of a month****, which is entered by users (both month and year). If it is an invalid month/year, ask the user to enter again.*

Note: You must create a new Java project for this exercise.

- The user can either enter a month in its full name, abbreviation, in 3 letters, or in number. To illustrate, the valid inputs of *January* are January, Jan., Jan, and 1.
- The user must enter a year in a non-negative number and enter all the digits. For instance, the valid inputs of year *1999* is only 1999, but not 99, "one thousand nine hundred ninety-nine", or anything else.
- A year is either a common year of 365 days or a leap year of 366 days. Every year that is divisible by 4 is a leap year, except for years that are divisible by 100, but not by 400. For instance, year 1800 is not a leap year, yet year 2000 is a leap year. In a year, there are twelve months, which are listed in order as follows.

Month	January	February	March	April	May	June	July	August	September	October	November	December
Abbreviation	Jan.	Feb.	Mar.	Apr.	May	June	July	Aug.	Sept.	Oct.	Nov.	Dec.
In 3 letters	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
In Number	1	2	3	4	5	6	7	8	9	10	11	12
Days of Month in Common Year	31	28	31	30	31	30	31	31	30	31	30	31
Days of Month in Leap Year	31	**29**	31	30	31	30	31	31	30	31	30	31

5.6 Write a Java program to sort a numeric array, and calculate the sum and average value of array elements.



Note: You must create a new Java project for this exercise.

- The array can be entered by the user or a constant.

5.7 Write a Java program to add two matrices of same size.

Note: You must create a new Java project for this exercise.

- The matrices can be entered by the user or constants.

6 Assignment Submission

You must put the use case diagram (put both the source file and its exported image in the folder namely "AIMS") and all the six applications of this lab (i.e., exercise 5.1, 5.2, 5.4, 5.5, 5.6, 5.7), written by yourself, into a directory namely "Lab02" 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.

7 References

Fowler, M. (2003). *UML Distilled Third Edition: A Brief Guide to the Standard Object Modeling Language.* Addison-Wesley.