Lab 02
Reusing Projects

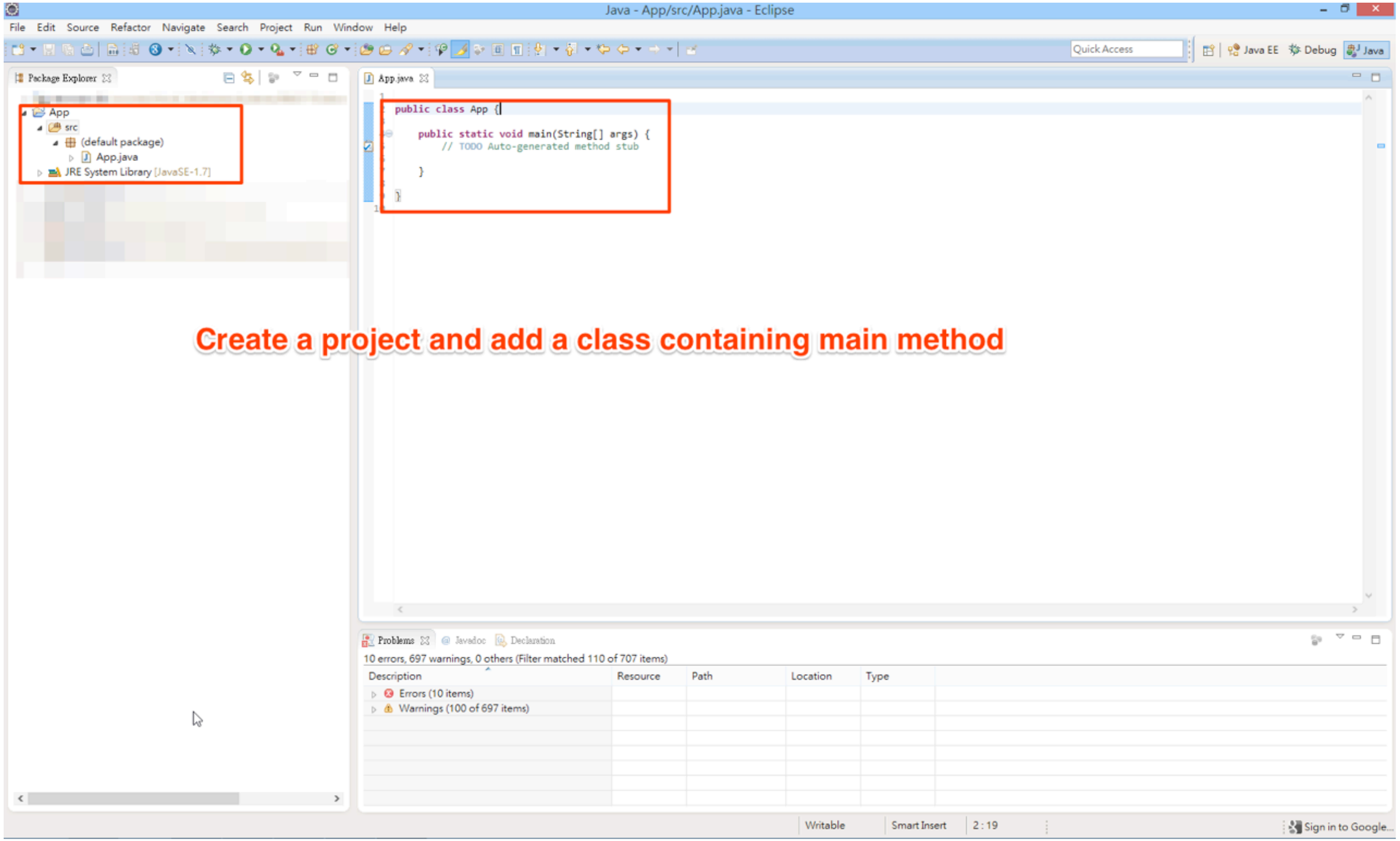
Software Studio
NetDB, CS, NTHU
Fall, 2013

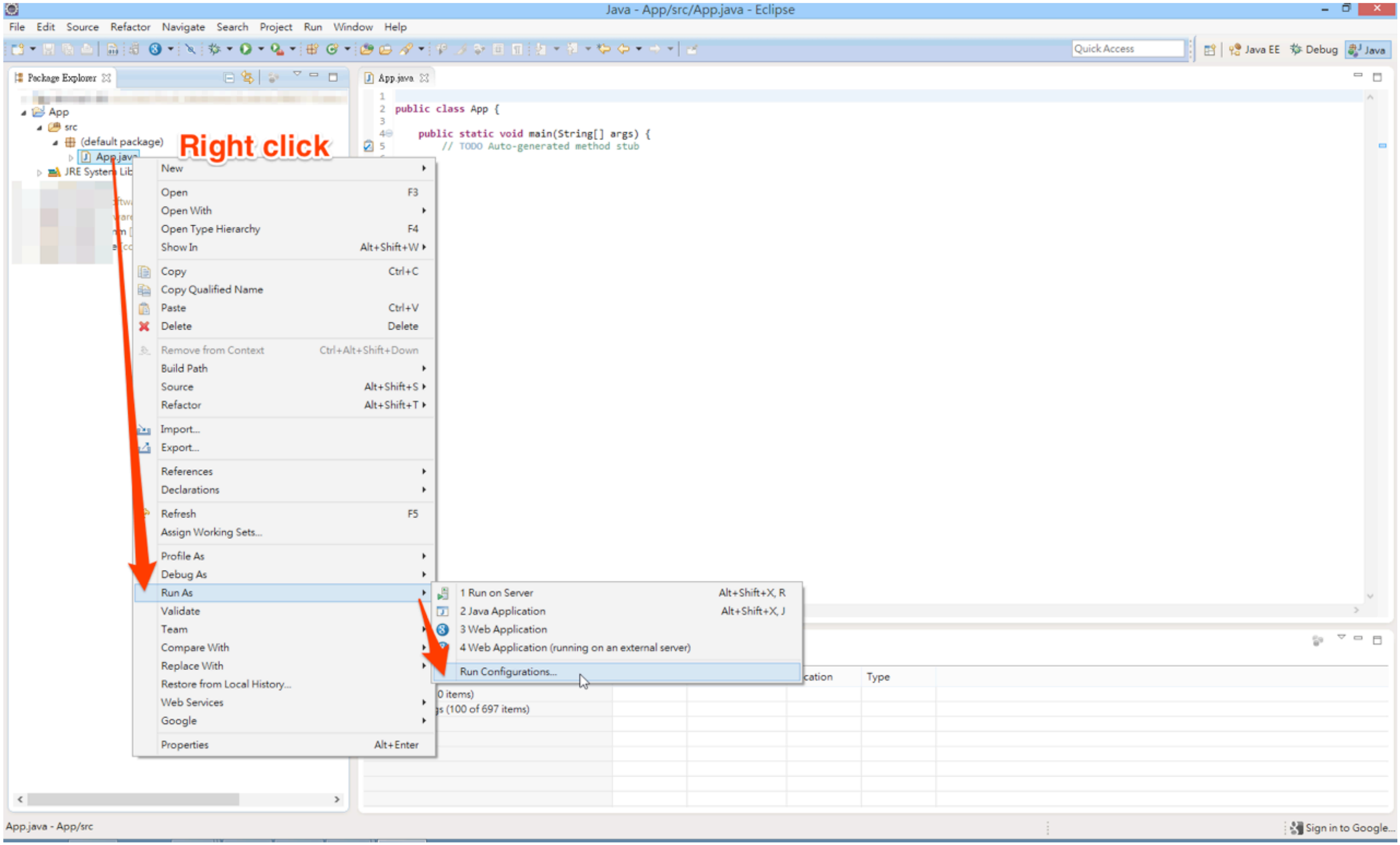
Outline

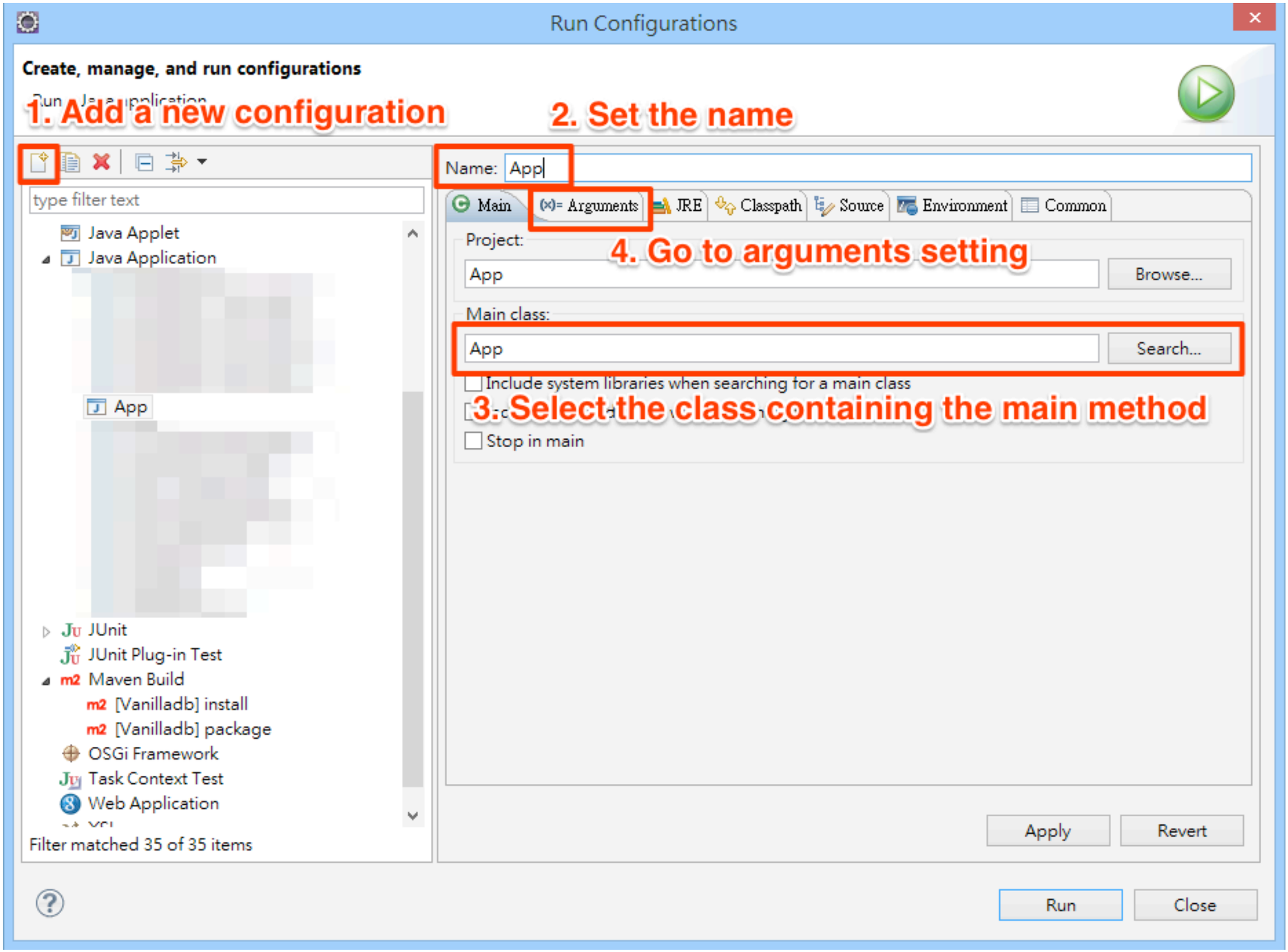
- Argument Passing in Eclipse
- Reusing Project
- Today’s Mission

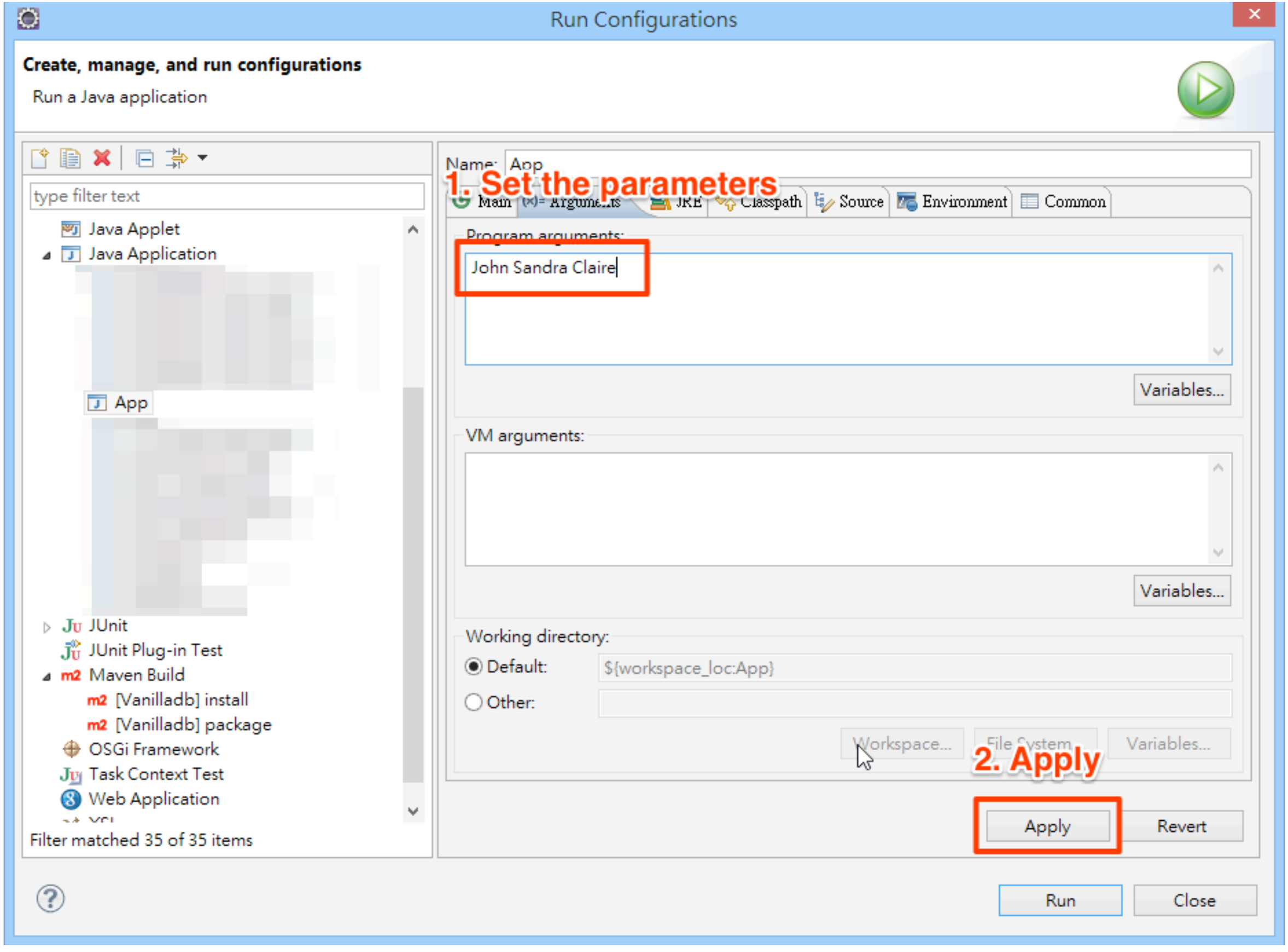
Outline

- Argument Passing in Eclipse
- Reusing Project
- Today’s Mission


Java - App/src/App.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Java EE
Debug
Java
Package Explorer
App
src
(default package)
App.java
JRE System Library [JavaSE-1.7]
App.java
public class App {
public static void main(String[] args) {
// TODO Auto-generated method stub
}
}
Create a project and add a class containing main method
Problems
Javadoc
Declaration
10 errors, 697 warnings, 0 others (Filter matched 110 of 707 items)
Description
Resource
Path
Location
Type
Errors (10 items)
Warnings (100 of 697 items)
Writable
Smart Insert
2 : 19
Sign in to Google...


Java - App/src/App.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Java EE
Debug
Java
Package Explorer
App
src
(default package)
App.java
JRE System Lib
Right click
App.java
1
2 public class App {
3
4 public static void main(String[] args) {
5 // TODO Auto-generated method stub
New
Open F3
Open With
Open Type Hierarchy F4
Show In Alt+Shift+W
Copy Ctrl+C
Copy Qualified Name
Paste Ctrl+V
Delete Delete
Remove from Context Ctrl+Alt+Shift+Down
Build Path
Source Alt+Shift+S
Refactor Alt+Shift+T
Import...
Export...
References
Declarations
Refresh F5
Assign Working Sets...
Profile As
Debug As
Run As
Validate
Team
Compare With
Replace With
Restore from Local History...
Web Services
Google
Properties Alt+Enter
1 Run on Server Alt+Shift+X, R
2 Java Application Alt+Shift+X, J
3 Web Application
4 Web Application (running on an external server)
Run Configurations...
Type
0 items)
s (100 of 697 items)
App.java - App/src
Sign in to Google...


Run Configurations
Create, manage, and run configurations
1. Add a new configuration
2. Set the name
type filter text
Java Applet
Java Application
App
JUnit
JUnit Plug-in Test
Maven Build
[Vanilladb] install
[Vanilladb] package
OSGi Framework
Task Context Test
Web Application
Filter matched 35 of 35 items
Name: App
Main
Arguments
JRE
Classpath
Source
Environment
Common
4. Go to arguments setting
Project:
App
Browse...
Main class:
App
Search...
Include system libraries when searching for a main class
3. Select the class containing the main method
Stop in main
Apply
Revert
Run
Close


Run Configurations
Create, manage, and run configurations
Run a Java application
type filter text
Java Applet
Java Application
App
JUnit
JUnit Plug-in Test
Maven Build
[Vanilladb] install
[Vanilladb] package
OSGi Framework
Task Context Test
Web Application
Filter matched 35 of 35 items
Name: App
1. Set the parameters
Main
Arguments
JRE
Classpath
Source
Environment
Common
Program arguments:
John Sandra Claire
Variables...
VM arguments:
Variables...
Working directory:
Default:
${workspace_loc:App}
Other:
Workspace...
File System...
Variables...
2. Apply
Apply
Revert
Run
Close

```java
package netdb.Hello2;

/**
 * Hello world!
 *
 */
public class App {
    public static void main(String[] args) {
        for (int i = 0; i < args.length; i++) {
            System.out.println("Hello " + args[i]);
        }
    }
}

```

Markers Properties Servers Data Source Explorer Snippets Console

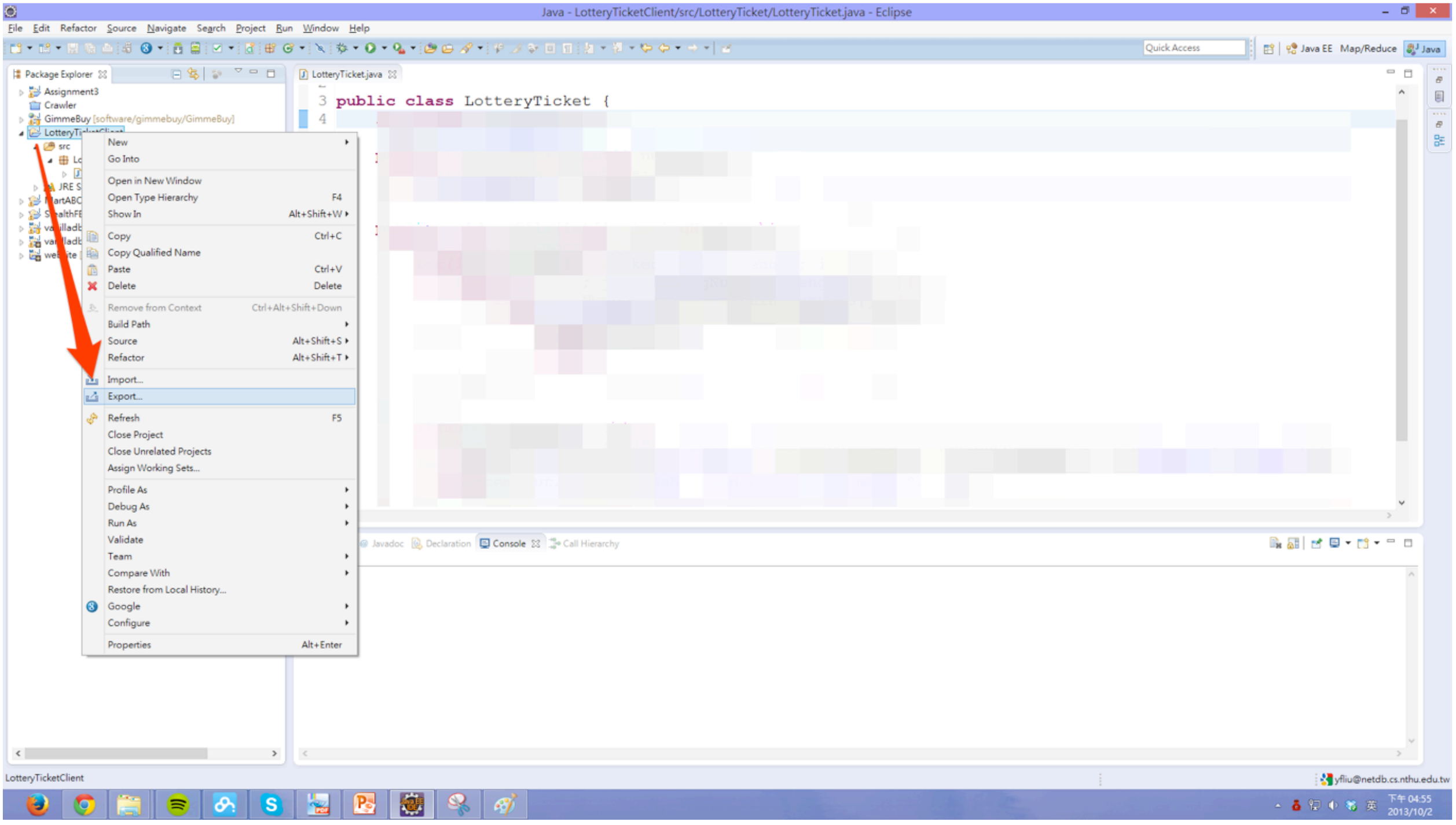
<terminated> App [Java Application] C:\Program Files\Java\jdk1.7.0_03\bin\javaw.exe (Sep 2

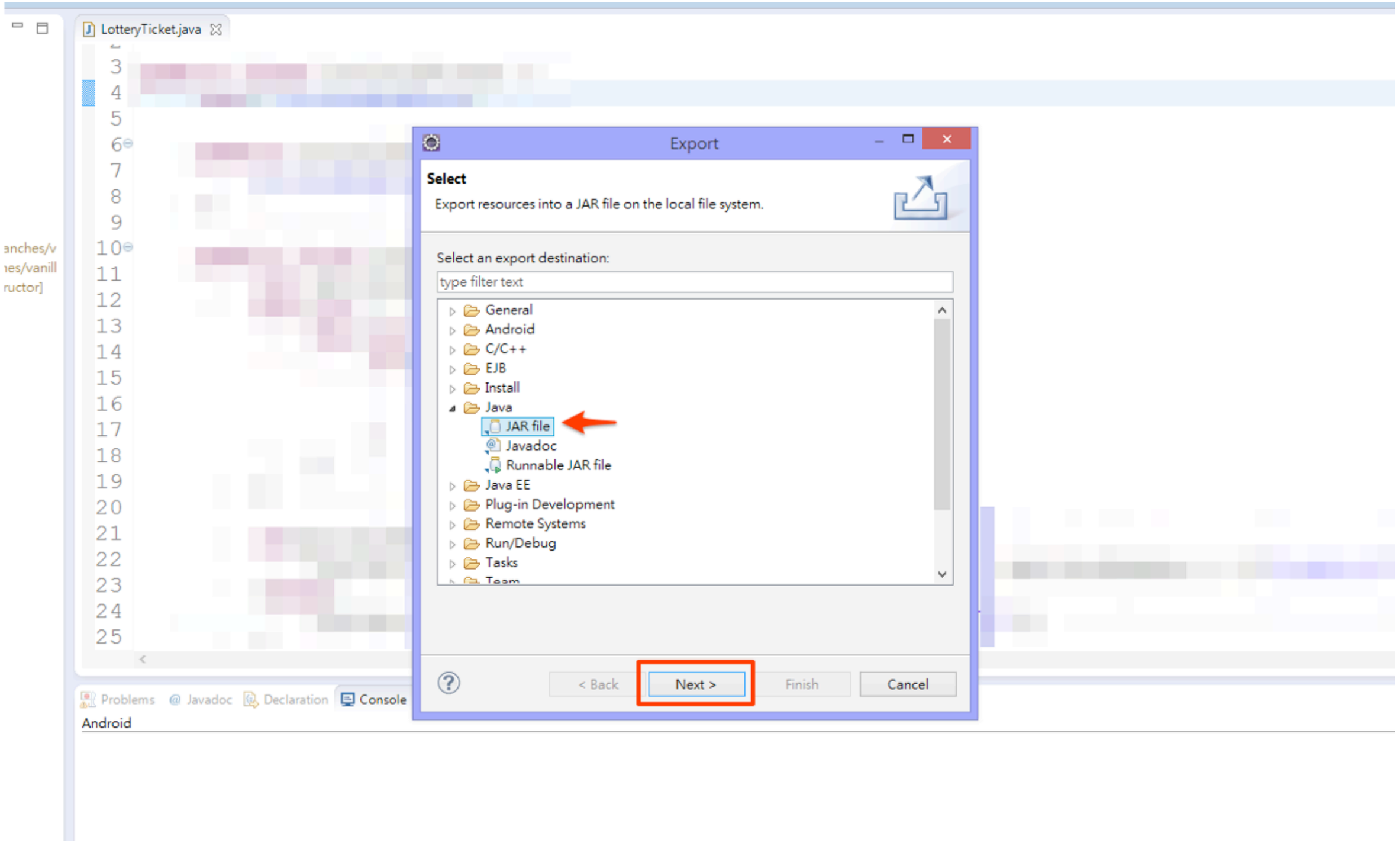
```
Hello John
Hello Sandra
Hello Claire
```

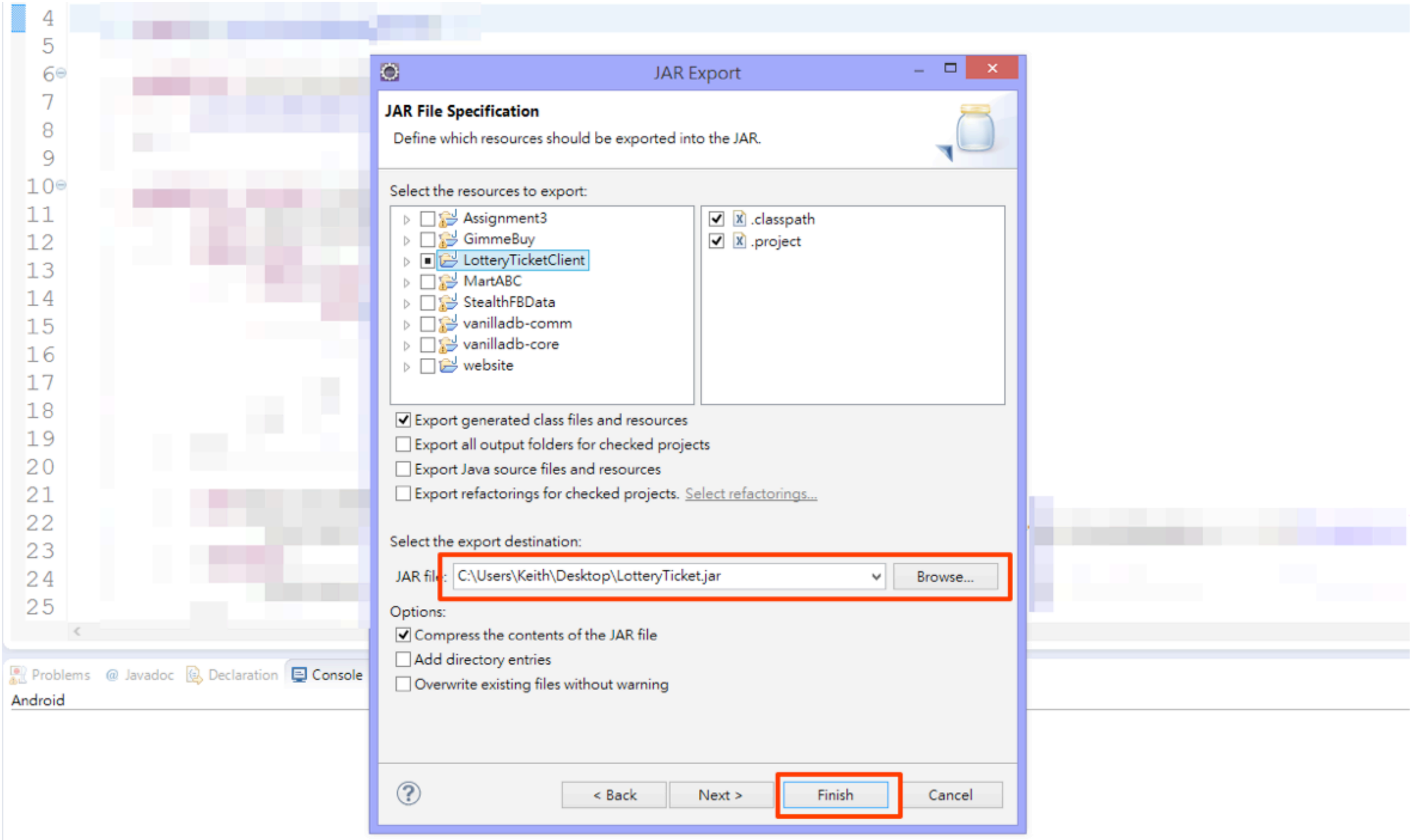
Outline

- Argument Passing in Eclipse
- Reusing Project
- Today’s Mission

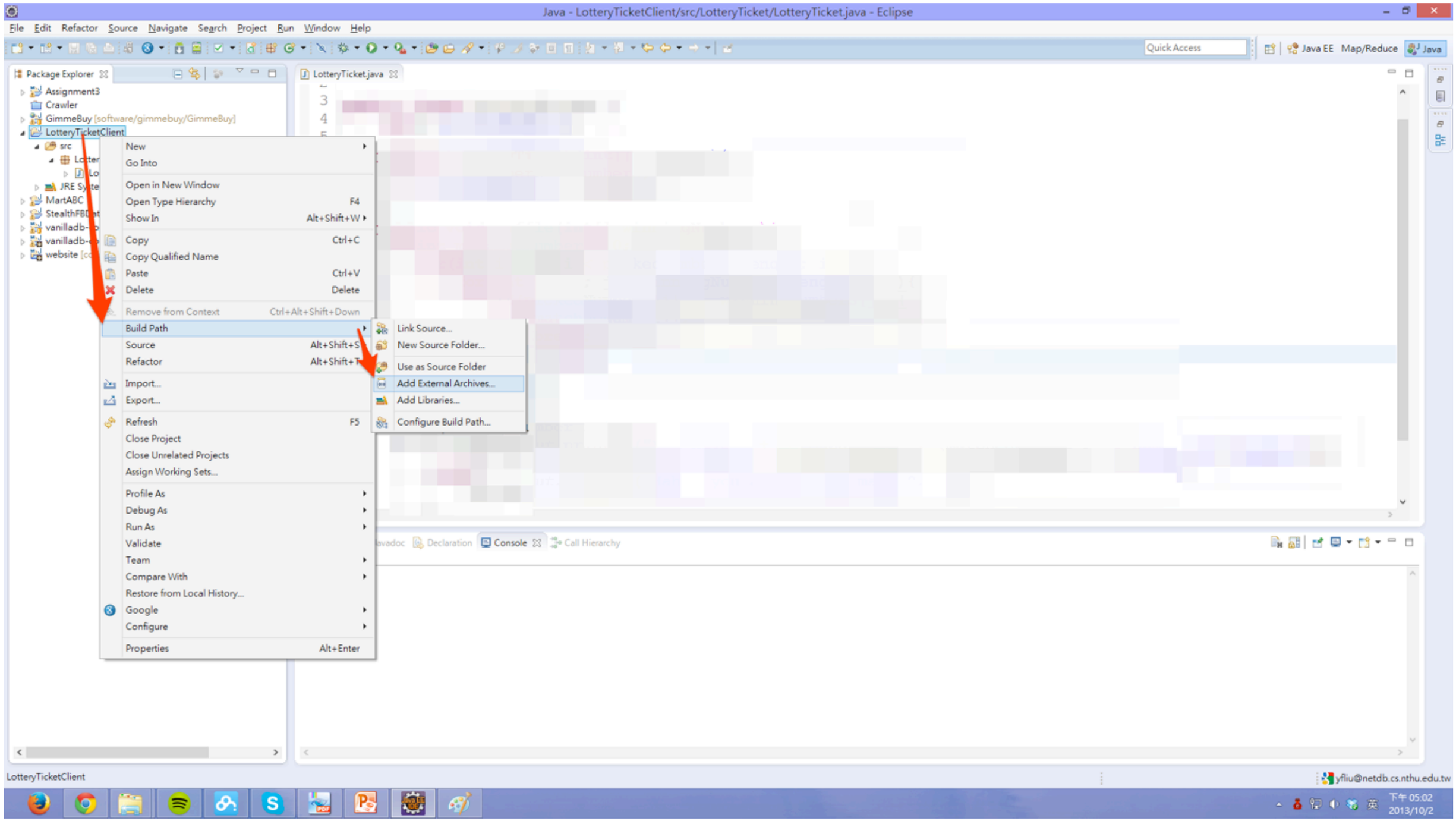
Export a .jar file




LotteryTicket.java
Export
Select
Export resources into a JAR file on the local file system.
Select an export destination:
type filter text
General
Android
C/C++
EJB
Install
Java
JAR file
Javadoc
Runnable JAR file
Java EE
Plug-in Development
Remote Systems
Run/Debug
Tasks
< Back
Next >
Finish
Cancel
Problems
Javadoc
Declaration
Console
Android


JAR Export
JAR File Specification
Define which resources should be exported into the JAR.
Select the resources to export:
Assignment3
GimmeBuy
LotteryTicketClient
MartABC
StealthFBData
vanilladb-comm
vanilladb-core
website
.classpath
.project
Export generated class files and resources
Export all output folders for checked projects
Export Java source files and resources
Export refactorings for checked projects. Select refactorings...
Select the export destination:
JAR file:
C:\Users\Keith\Desktop\LotteryTicket.jar
Browse...
Options:
Compress the contents of the JAR file
Add directory entries
Overwrite existing files without warning
< Back
Next >
Finish
Cancel
Problems
Javadoc
Declaration
Console
Android

Add a .jar file

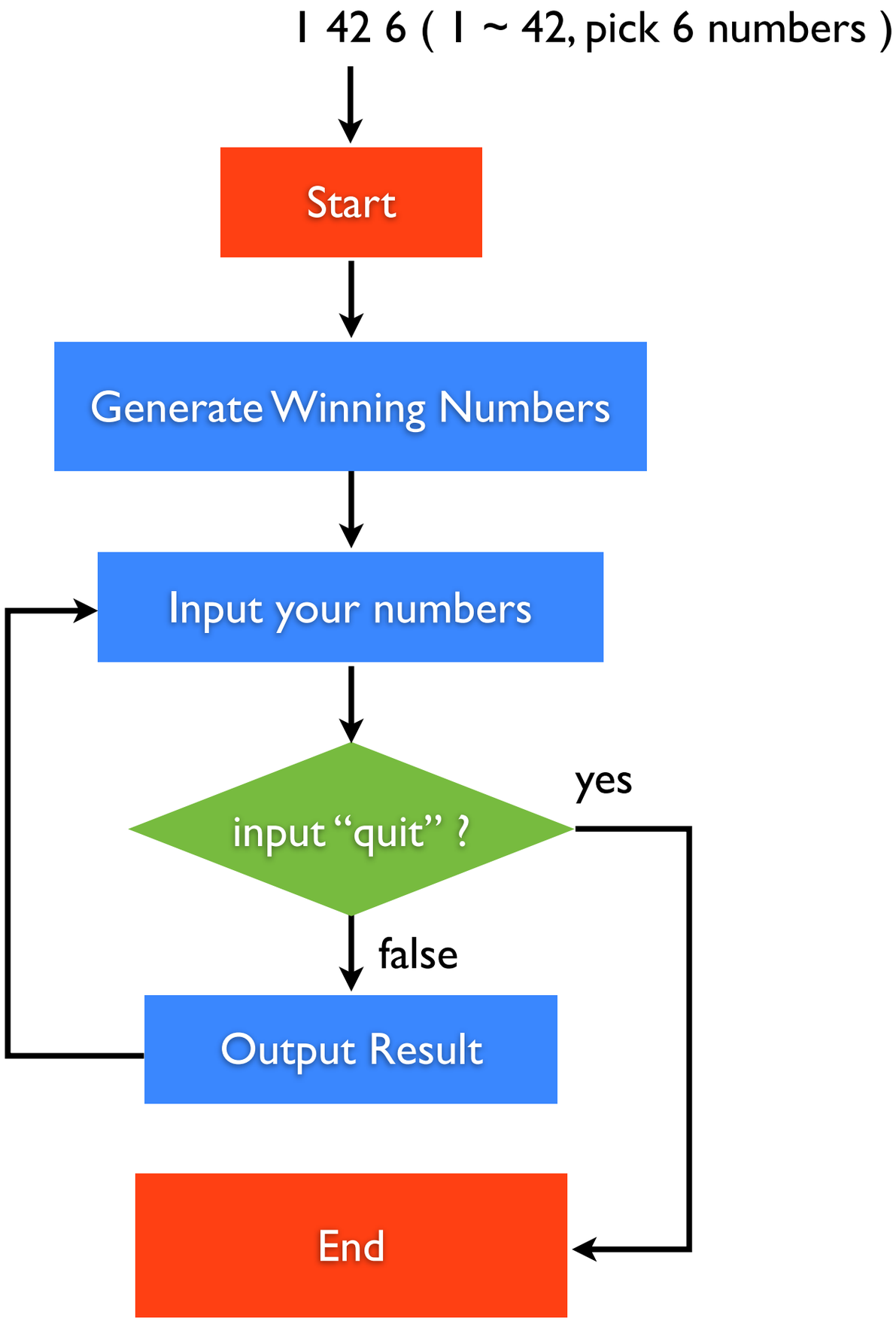


Outline

- Argument Passing in Eclipse
- Reusing Project
- Today’s Mission

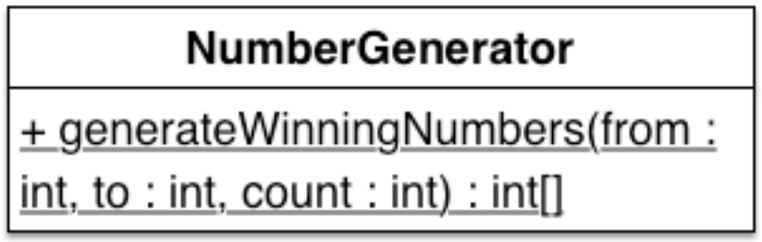
Today's Mission

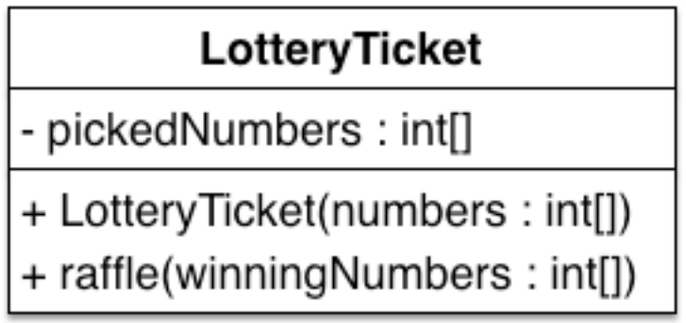
- Implementing a lottery system, with following steps:
  - Input 3 arguments to specify 'from', 'to' and 'count'(ex: 1 42 6)
  - Generate <count> **random non-repeat** integers as the winning numbers
  - Input <count> numbers and create a lottery ticket
  - Match your numbers with winning numbers and show the result (how many numbers are matched)


1 42 6 (1 ~ 42, pick 6 numbers)
Start
Generate Winning Numbers
Input your numbers
input “quit” ?
yes
false
Output Result
End

Rules

- 2 people per group
- You need to create 2 projects, NumberGenerator and LotteryTicket, and export them into .jar files
- Import your .jar files and modify the provided template to complete this task


NumberGenerator
+ generateWinningNumbers(from : int, to : int, count : int) : int[]


LotteryTicket
- pickedNumbers : int[]
+ LotteryTicket(numbers : int[])
+ raffle(winningNumbers : int[])

Hints

- To parse an integer from String:

```
int i = Integer.parseInt(“1”);
```

Hints

- To create an integer array:

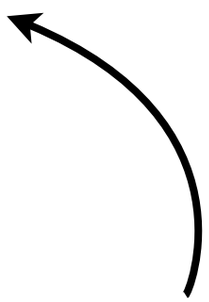
```
int[] intArray = new int[ARRAY_LENGTH];
```

For more information about array in Java, you can read this article.

Hints

- To split a String into a String array:

```
String[] tokens
= “Java made easy”.split(“ ”);
```

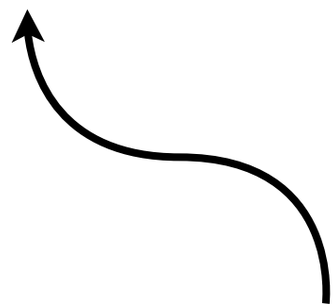


It will return an array containing {“Java”,“made”,“easy”}

Hints

- To generate random numbers

```
double randNum = Math.random();
```



It will return an random value 0.0 ~ 1.0

Hints

- Remember to import the classes

```
import xxx.xxx.xxx;
```

- Using Eclipse, you can use the hot key

```
Ctrl + Shift + O
```