

Search Engines - Recitation 1

QryEval 101

-
- Java OOP Properties
 - QryEval Class Structure
 - Implementation Tips
 - Environment Setup (Eclipse)
 - Debugging in Editors (IntelliJ)
 - Q&A
-

- Class is a prototype from which objects are created.
 - Class has properties and methods

```
public class QrySopOR
extends QrySop{
    private String term;
    public double getScore(){

    }
}
```

- In Java, one class may inherit the features(fields and methods) of another class using **extends** keyword.

```
class A {  
    int a;  
    void print_value(){  
        System.out.println(a);  
    }  
  
    class B extends A {  
        int b;  
        void print_value(){  
            temp = a + b; // a is inherited  
            System.out.println(temp);  
        }  
    }
```

Base	Derived
Qry	QrySop, QryloP
QrySop	QrySopOR, QrySopAND
QryloP	QrylopTerm, QryloPSyn

- An abstract class is a class that is declared with **abstract** keyword
- An abstract method is a method that is declared without an implementation
 - Typically for a subclass to have its own implementation

Example : QryClass

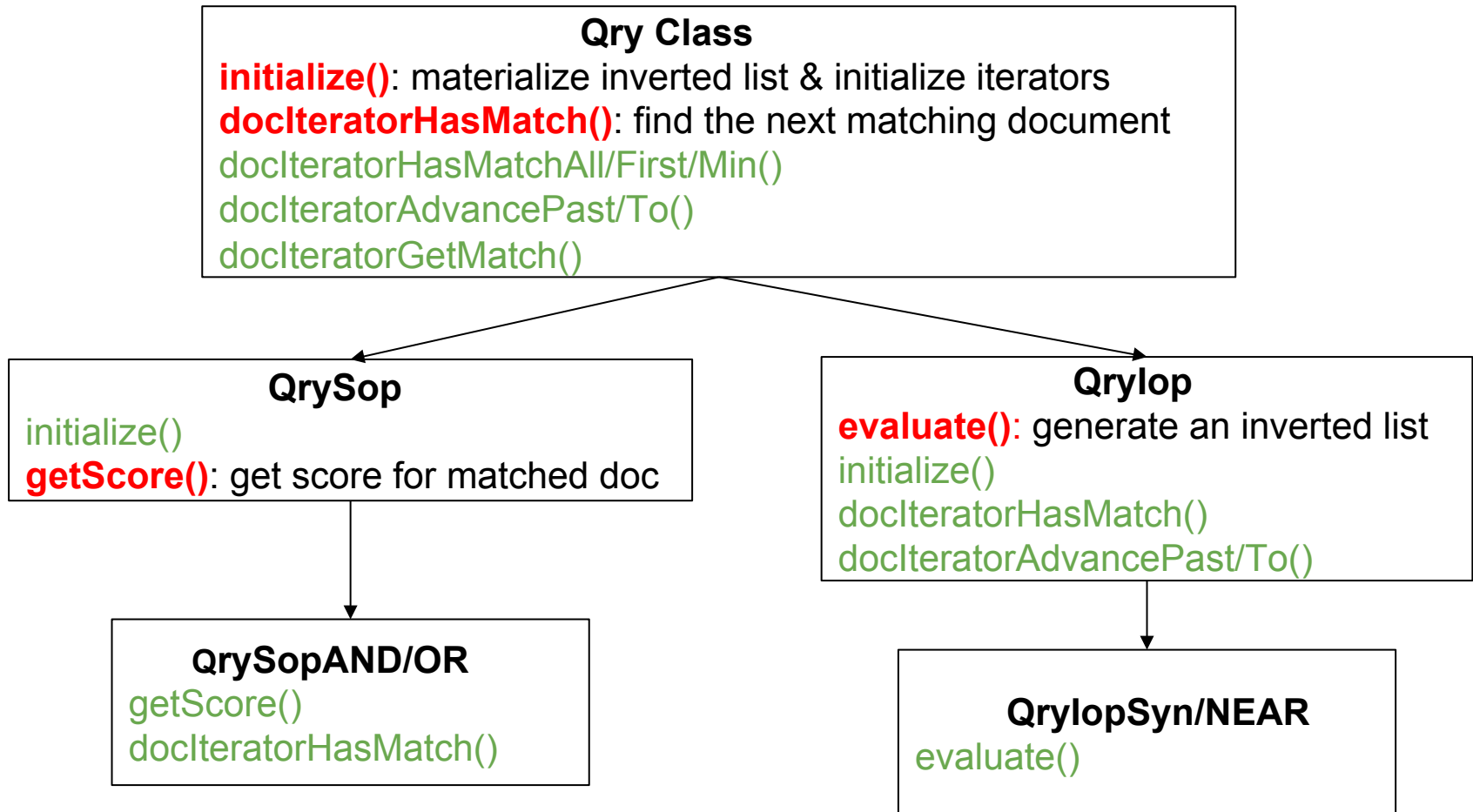
```
public class Qry{
```

```
    public abstract boolean docIteratorHasMatch (RetrievalModel r);
```

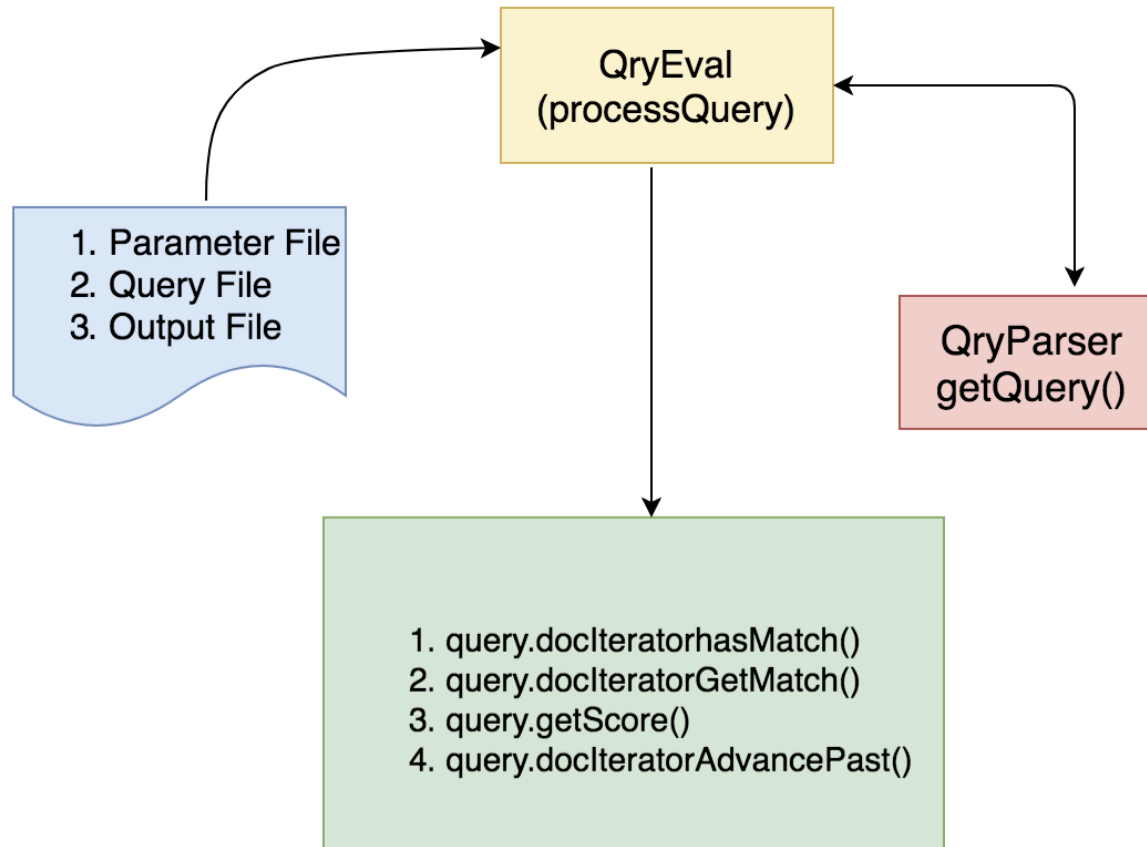
```
}
```

You have to implement
this method

- There are two RetrievalModels in the scope of this assignment.
 1. Unranked Boolean
 2. Ranked Boolean



* ■ - Abstract Methods, ■ - Implemented Methods

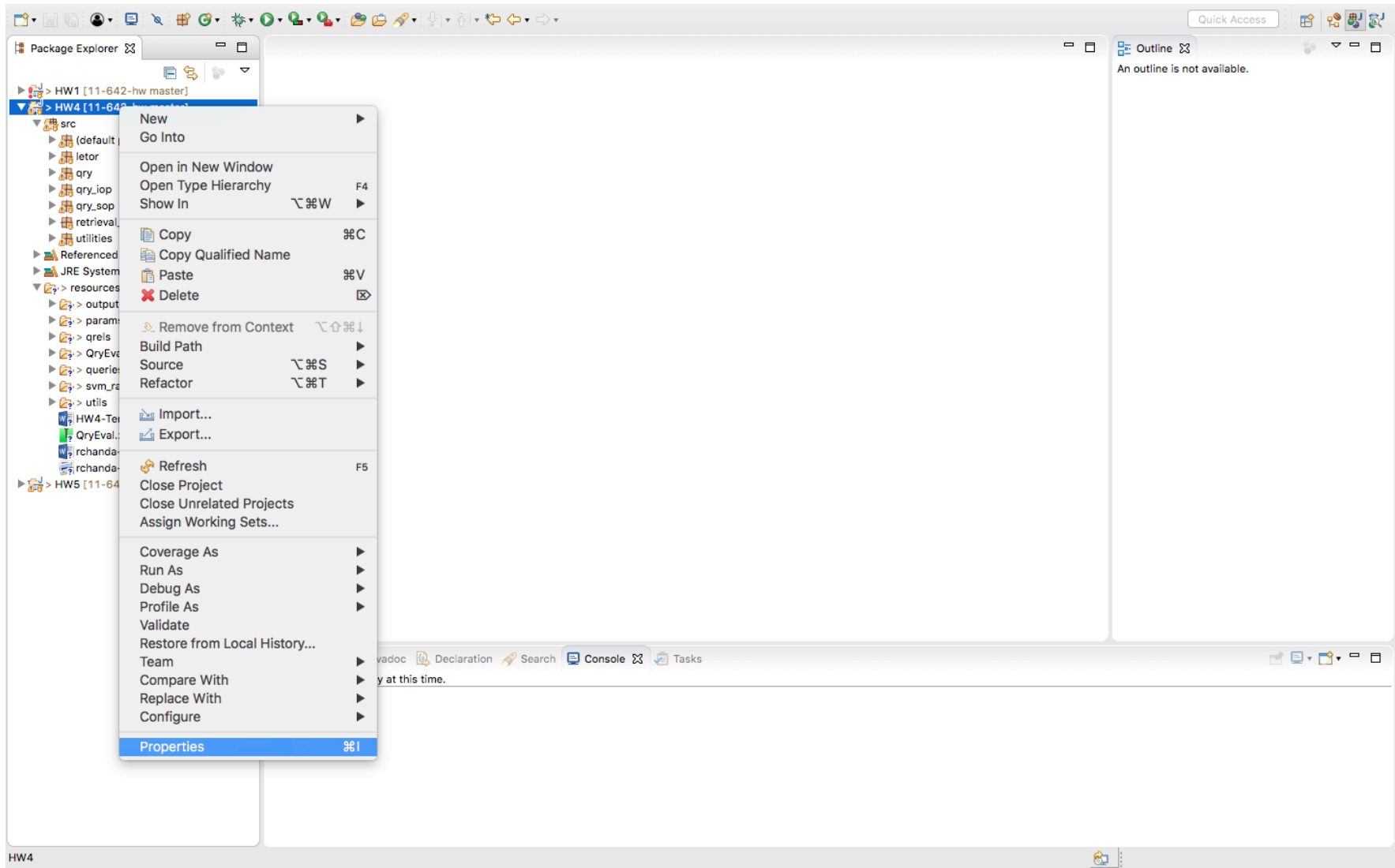


- QryEval.java contains main(), so it gets executed first
- 3 phases
 - **Read** parameter file
 - **Initialize** retrieval model
 - **Process** Query file
- Control transfers to your retrieval model following a long chain of function calls

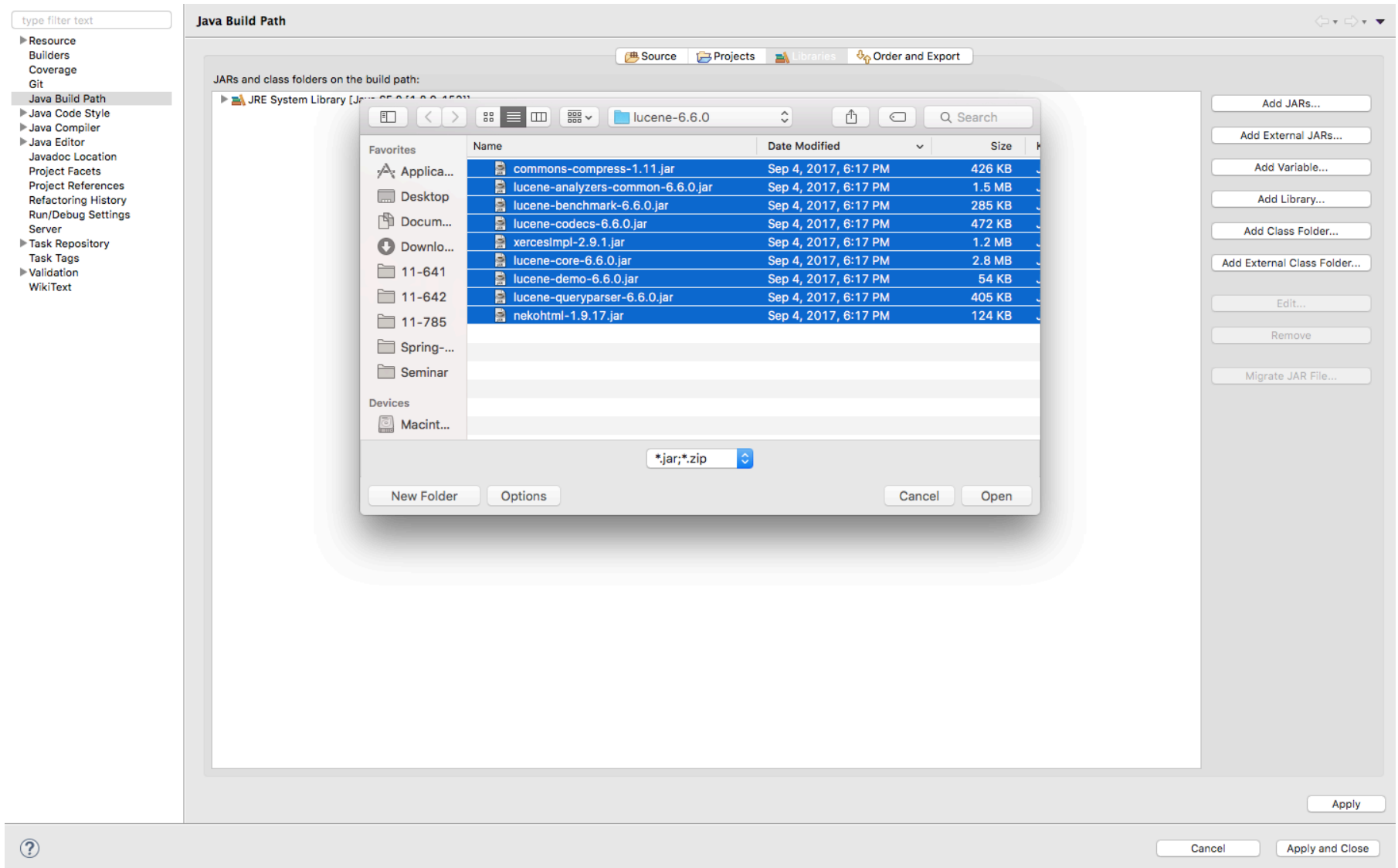
- The index is large. Don't wait till the last minute to download it.
- Parts of an object may be defined by different parts of the class hierarchy. If you are unable to find the implementation of a function, look at superclasses/subclasses.
- You may find it helpful to use **Project -> Generate Javadoc** tool to generate Java documentation of your program.
- For debugging, it'll be a good starting point to find the queries where your results disagree with Jamie's to a large degree.
- There are limited servers hosting the homework testing service. You might want to debug/test on your local laptop/PC first.
- Take a look at **InspectIndex** class. It may be helpful.
- Look at **FAQ** on the HW page.

1. Import the source code as Java Project in Eclipse
2. Add lucene jars into the classpath of the project
Properties -> Java Build Path -> Add External Jars
1. Run the project as Java Application. Choose QryEval as the main class.
2. Edit the path variables in the param files to point to your local files.
3. Edit the run configuration of the project by providing the location of params file in the Arguments tab.
4. Copy all the files in src folder to new folder named QryEval, zip the folder and submit for evaluation.
5. Upload the zip file in test service (located in the right side) to test it against the given test cases
6. You can also upload the output files teln to get the results (located in the left side of the test service)

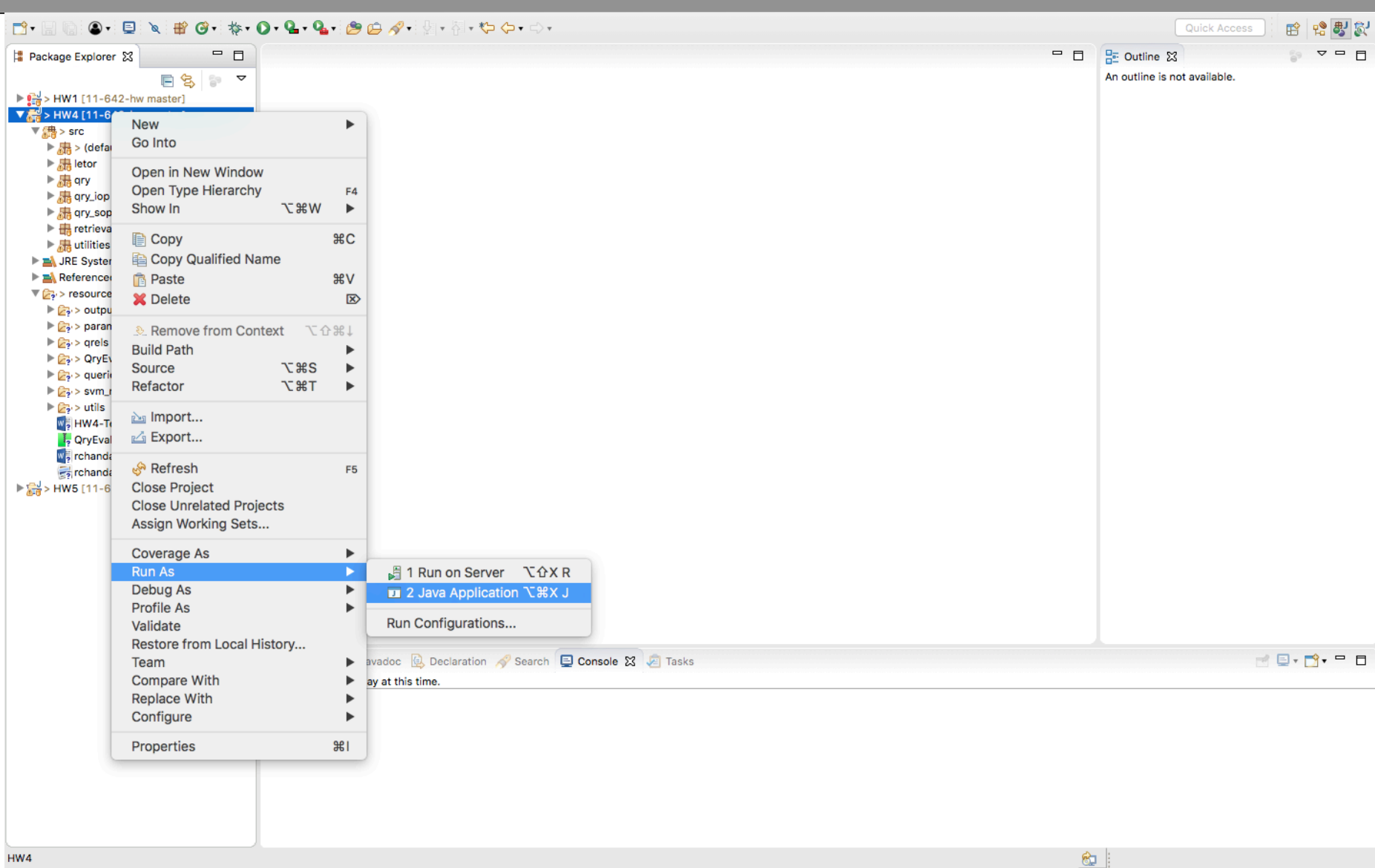
Environment Setup (Eclipse) - Step 2



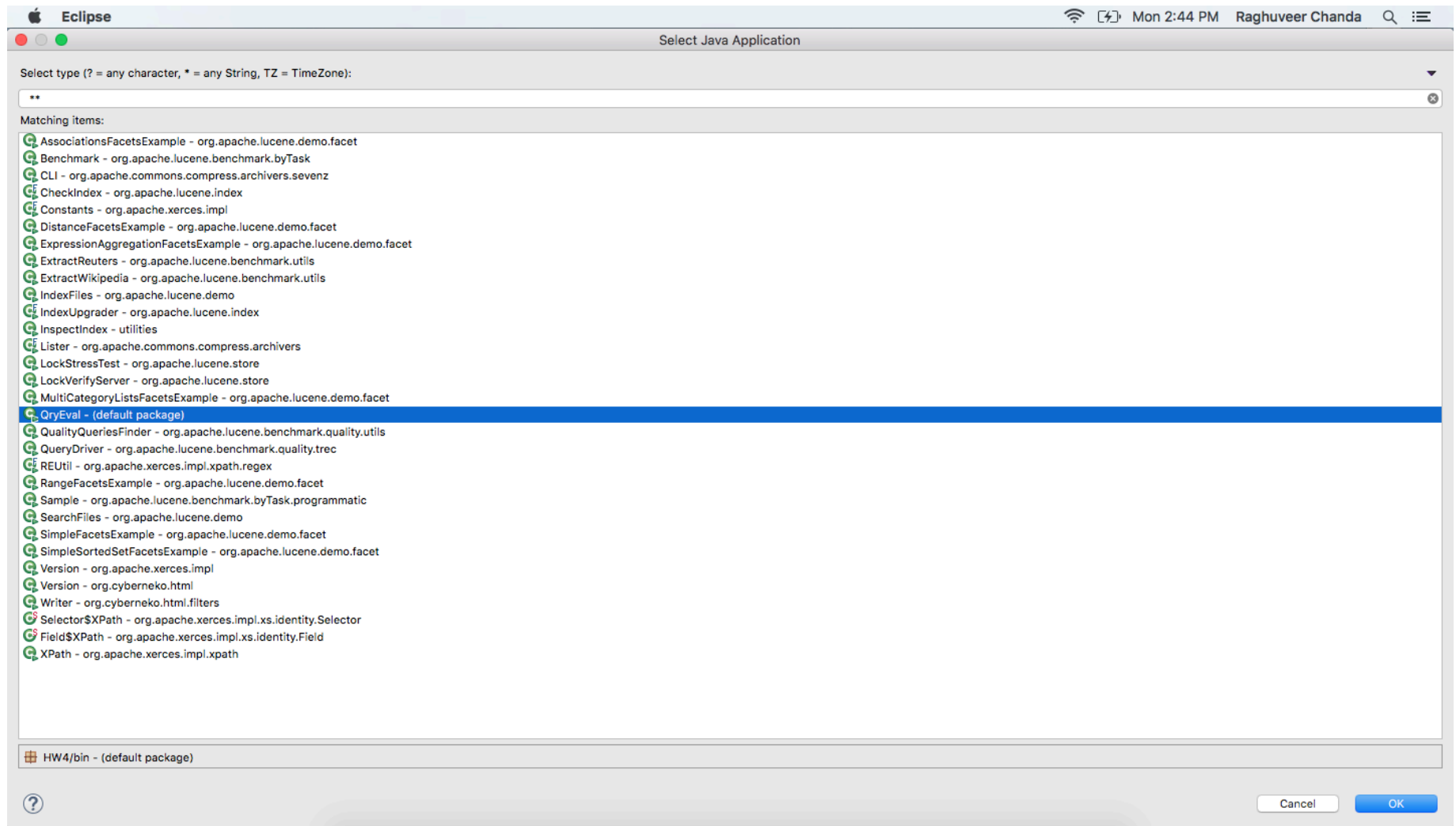
Environment Setup (Eclipse) - Step 2



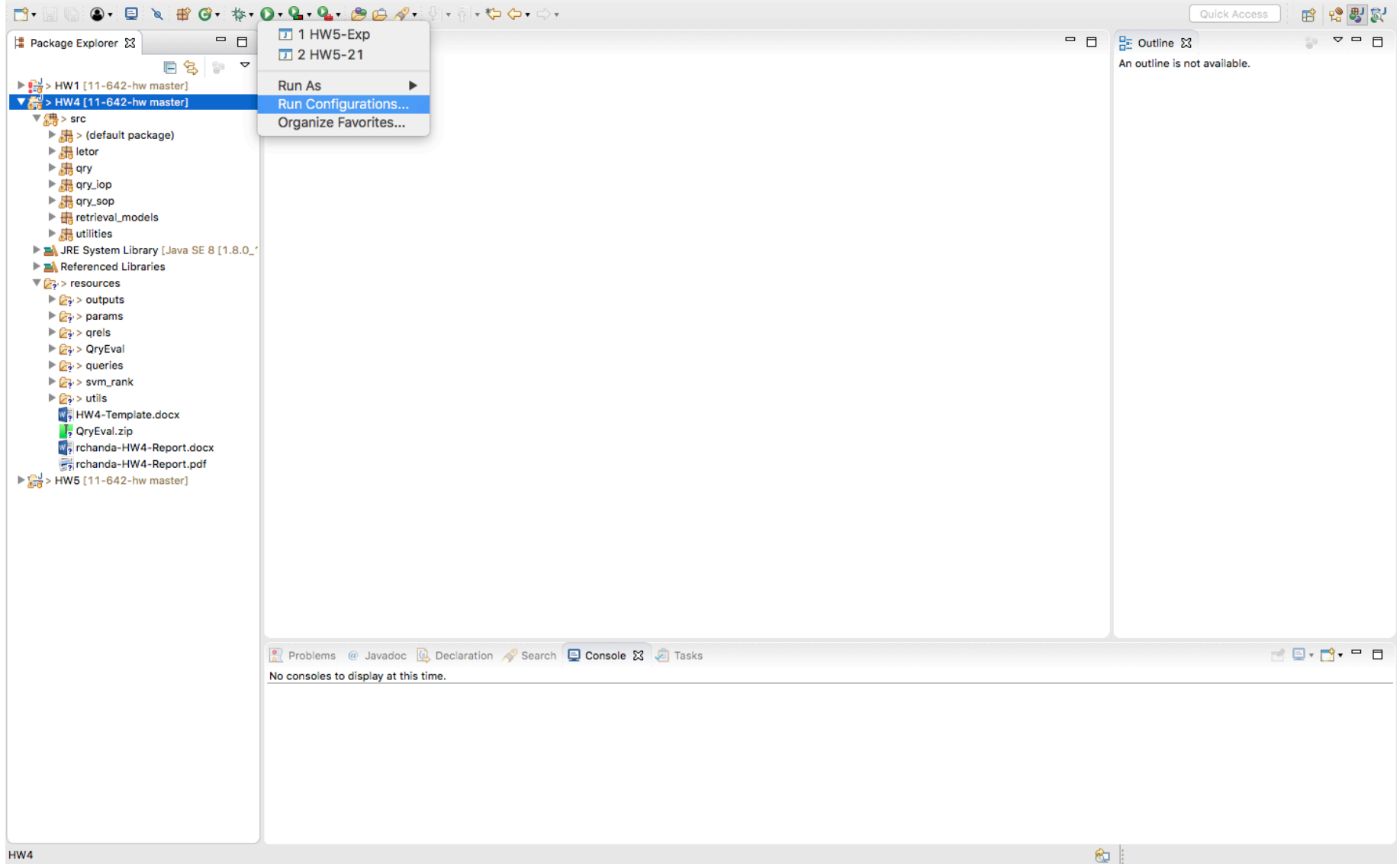
Environment Setup (Eclipse) - Step 3



Environment Setup (Eclipse) - Step 3



Environment Setup (Eclipse) - Step 5



Environment Setup (Eclipse) - Step 5

Carnegie Mellon

Create, manage, and run configurations

Run a Java application



The screenshot shows the Eclipse IDE's Run Configuration dialog for a project named "HW5-Exp". The left sidebar displays a list of launch configurations, with "HW5-Exp" selected. The main panel shows the configuration details for this launch configuration.

Launch Configuration List (Left Sidebar):

- type filter text
- Apache Tomcat
- Eclipse Application
- Eclipse Data Tools
- Generic Server
- Generic Server(External Launch)
- Gradle Project
- Grunt
- Gulp
- HTTP Preview
- J2EE Preview
- Java Applet
- Java Application
 - HW1
 - HW4-0
 - HW4-9
 - HW4-Test
 - HW5-21
 - HW5-Exp**
- JUnit
- JUnit Plug-in Test
- Launch Group
- Maven Build
- Node.js Application
- OSGi Framework
- Task Context Test
- XSL

Run Configuration Dialog (Main Panel):

Name: HW5-Exp

Tabs: Main | Arguments | JRE | Classpath | Source | Environment | Common

Program arguments:

```
resources/params/HW5-Exp-26.param
```

VM arguments:

☒ Use the -XstartOnFirstThread argument when launching with SWT

Working directory:

☒ Default: \${workspace_loc:HW5}

☐ Other:

Buttons: Workspace..., File System..., Variables..., Revert, Apply, Close, Run

