Lesson 1: Introduction to Java And the Eclipse Development Environment

Pulling the Arrow Back to Hit the Target

Wholeness of the Lesson

- Java is an object-oriented highly portable programming language that arose as an easy alternative to the once dominant, but error-prone, C++ language. Eclipse is one of many open source, powerful but easy-to-use integrated development environments for use with Java and related technologies. Transcendental meditation is a highly portable and powerful technique working from deeper levels of intelligence allows one to accomplish more with less effort.

Outline of Topics

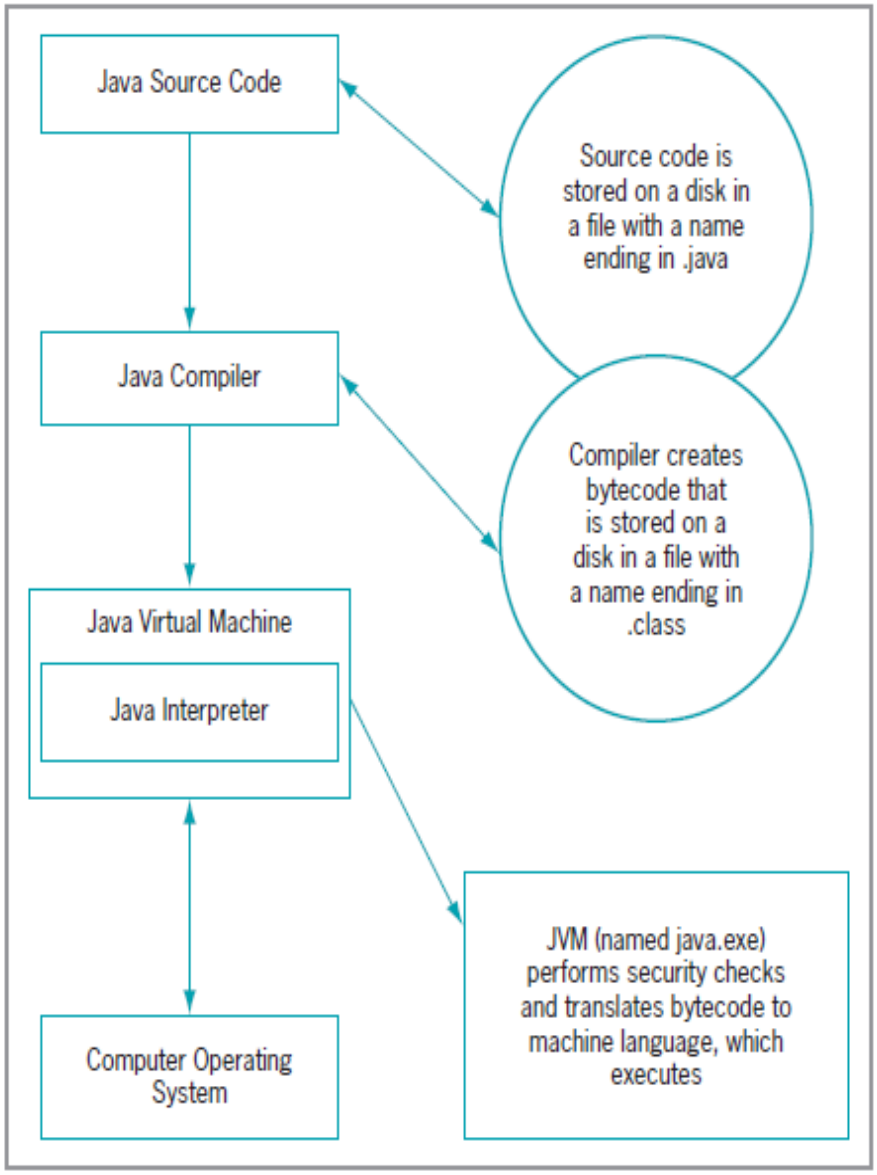
- About Java
- Java 10 API Docs
- JDK Comment line tools
- Java First Program
- Working with IDE
- Viewing Java API Source Code

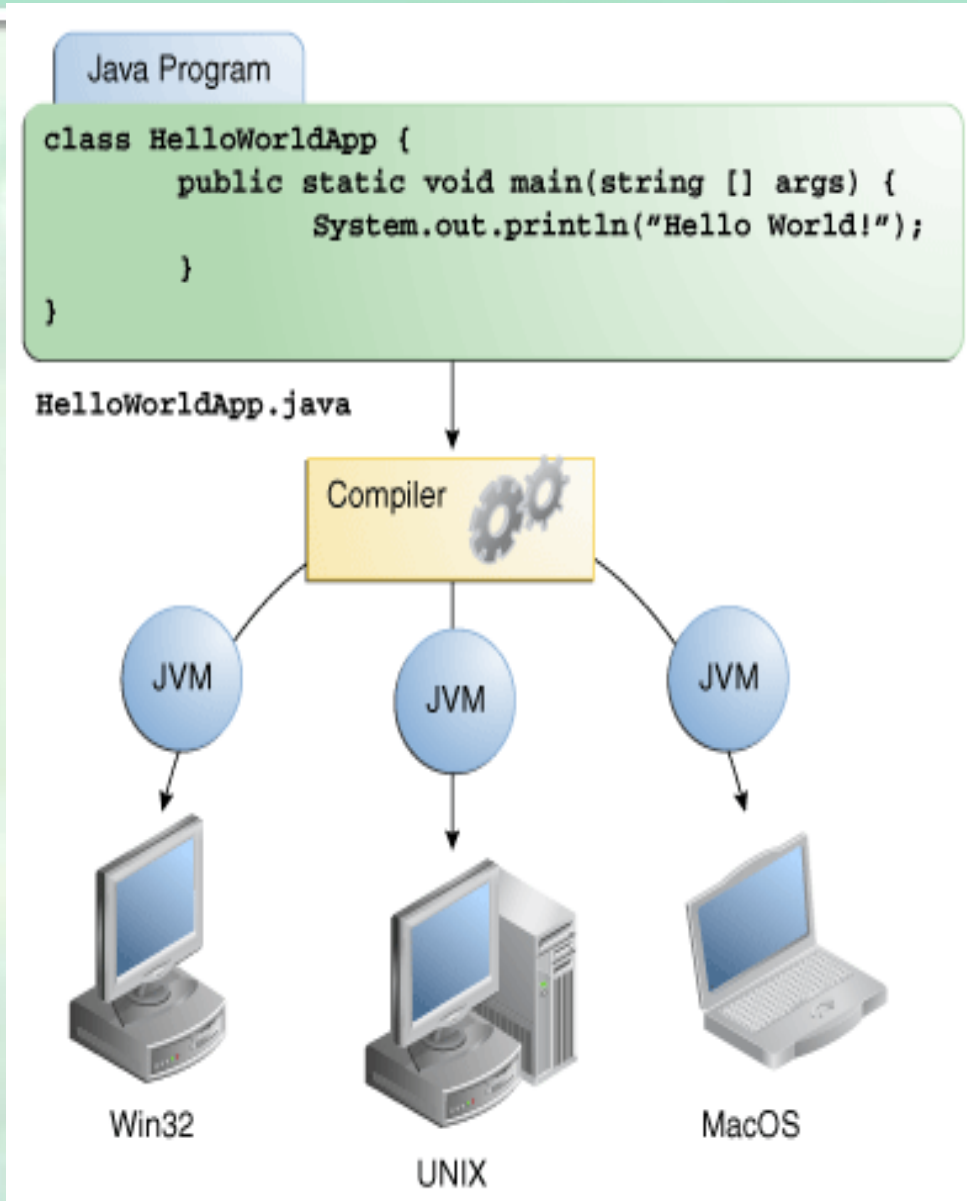
About Java

- ***Brief History.*** The Java language began as a language for programming electronic devices, though the original project was never completed. Its creator was James Gosling, of Sun Microsystems. The language was developed privately starting in 1991, and was made publicly available in 1994. In 2009, Oracle bought the rights to Java from Sun Microsystems.

- ***Interpreted Language.*** When you "compile" Java code, the result is not executable binary code, targeted to a particular machine; instead, the result is **bytecode**, having a portable intermediate code format. The bytecode is then executed by running an **interpreter,** called the **Java Virtual Machine** (JVM). This approach makes Java code highly portable; Java will run on any platform for which a JVM has been created.

Java Environment





About Java

- ***JIT.*** Interpreters run much more slowly than native binary executable code. Java performance is boosted considerably in modern-day JVMs because of the inclusion of a **just-in-time compiler**; this feature compiles frequently occurring bytecode sequences into native binary code and caches the results, re-using the cache as needed. The result is that Java runs almost as fast as C++ in typical cases.
- ***No Pointers.*** Unlike C and C++, Java does not make use of pointers; developers are not required to manage the memory usage of their applications. Memory management is handled automatically in Java by means of a **garbage collector.** The garbage collector is run by the JVM at different times during program execution to return to memory all unusued object references.

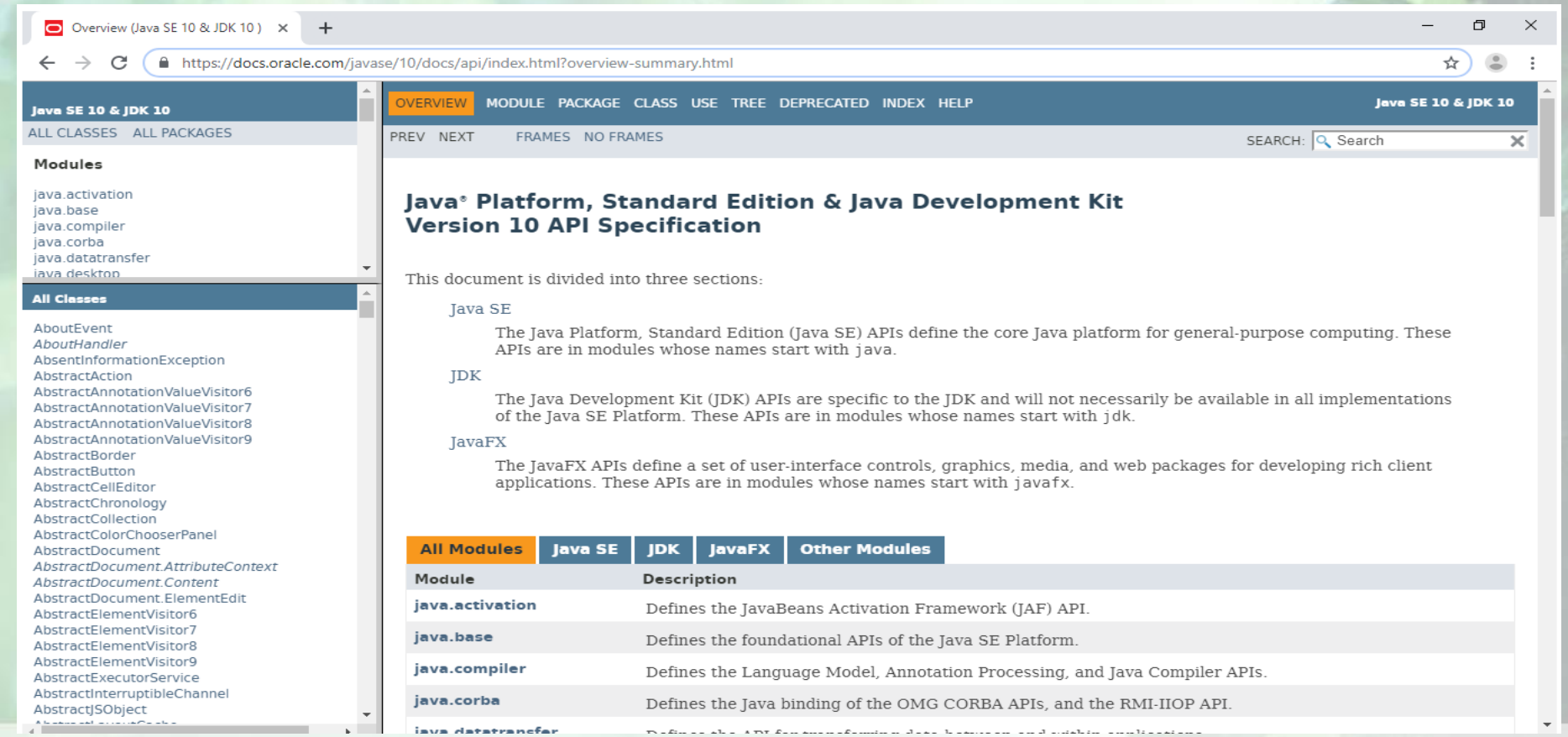
About Java

- ***Java Is an OOP Language.*** Java is an OO programming language, and succeeds in this better than C++, which (for the sake of backwards compatibility with C) supports a non-OO programming style (in C++, OO programming is "optional").
- ***Convenient Libraries.*** Java provides convenient libraries for handling files and streams, networking, http, gui development, and database connection. Compared to languages like C and C++, the increased ease of use is significant.
- ***Good Security Model.*** Java has a relatively good security model to support use over networks and distributed environments. Though security holes are still found sometimes, these are rare and quickly patched.

The Java 10 API Docs

- Oracle provides online documentation of all the Java library classes. Full documentation of each class in the Java libraries is provided. For Java 10, the link is

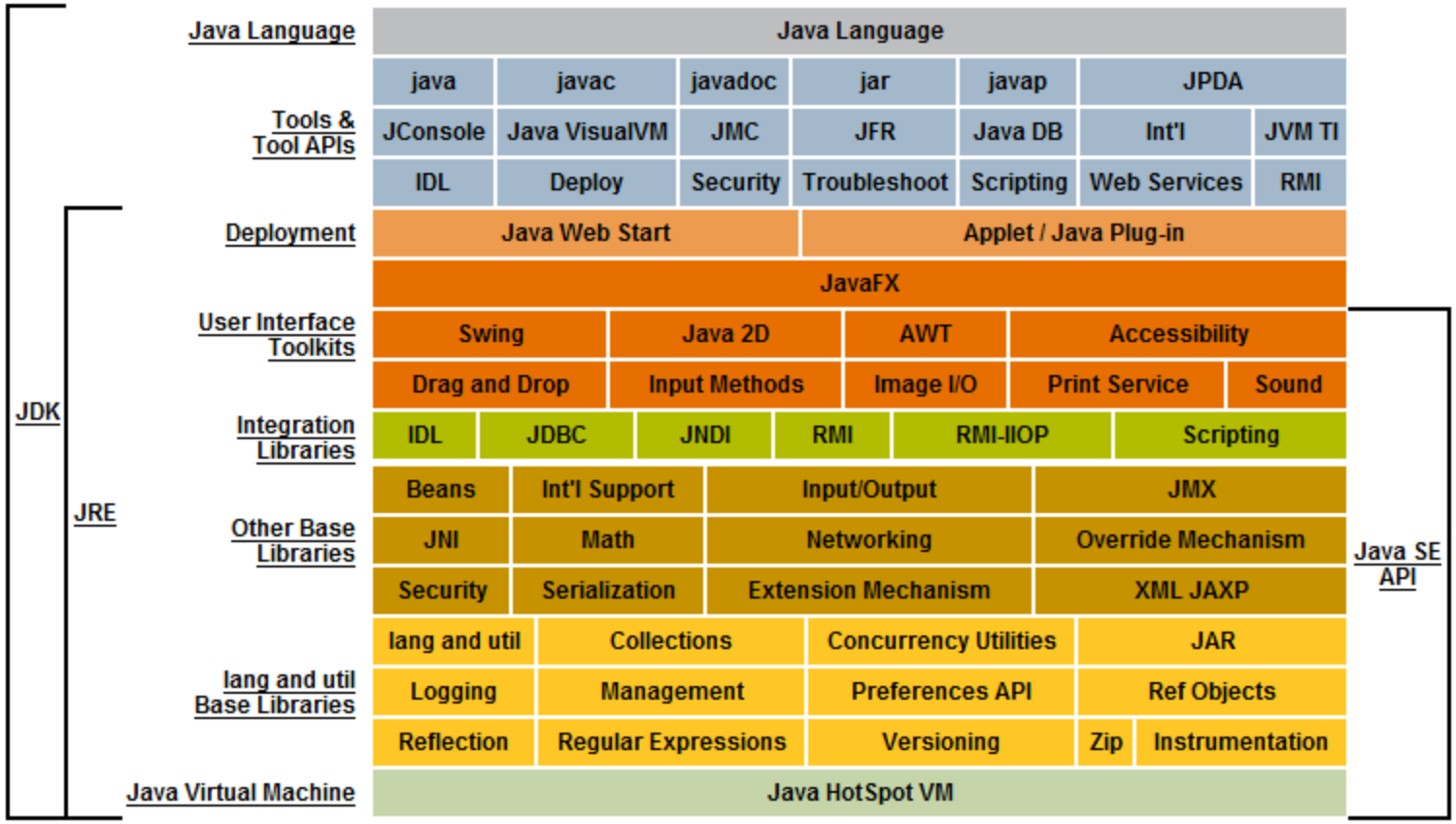
https://docs.oracle.com/javase/10/docs/api/index.html?overview-summary.html



Java Development Kit

The following conceptual diagram illustrates the components of Oracle's Java SE products:

Description of Java Conceptual Diagram



The JDK Commandline Tools

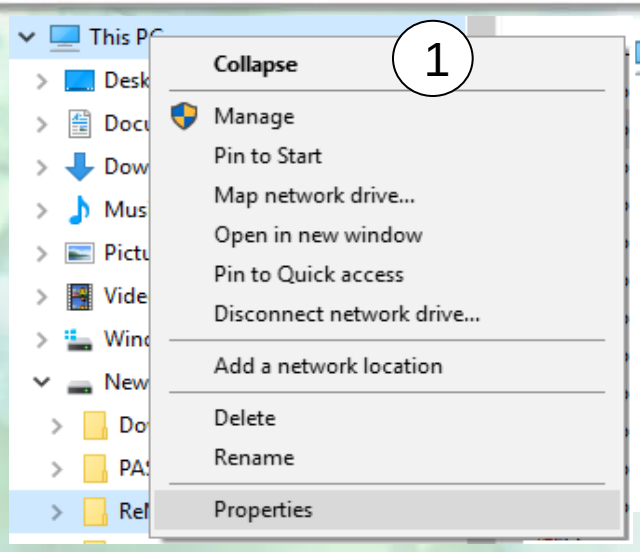
- To compile and run Java from a command window, you will use the executables javac.exe and java.exe, located in the bin directory of the Java distribution. The bin directory contains many additional tools for commandline processing.

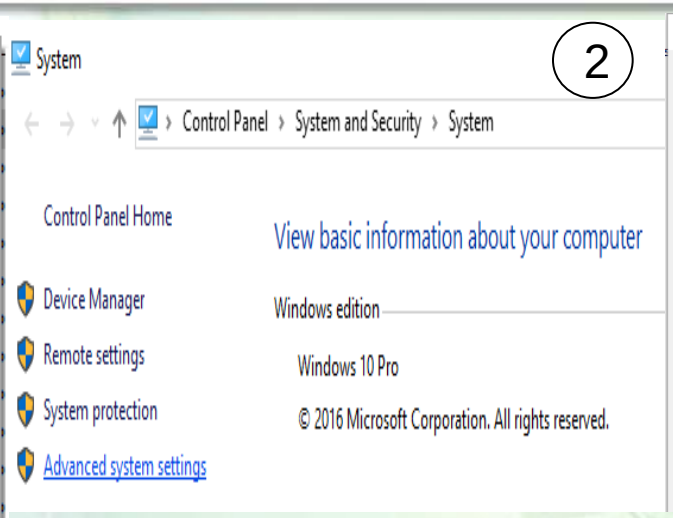
- *TIP:* If, to your PATH environment variable, you add the path to `javac.exe` and `java.exe`, compiling and running Java programs from the console is much easier. Typical path to these executables is the following (modify the jdk version number as necessary)

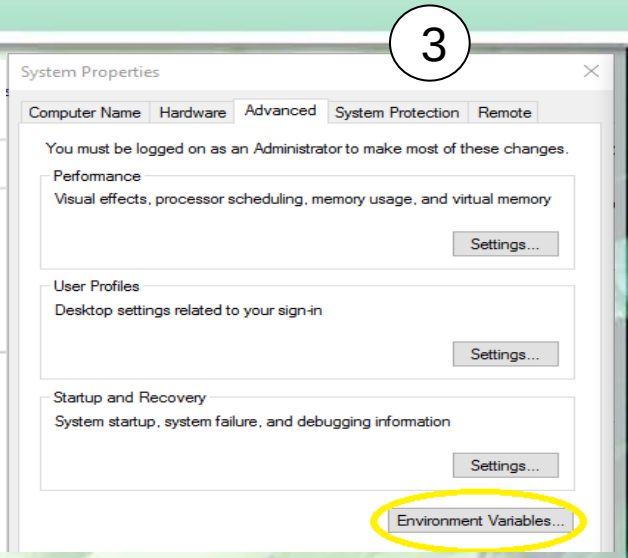
  ```
  C:\Program Files\Java\jdk-10.0.4\bin
  ```

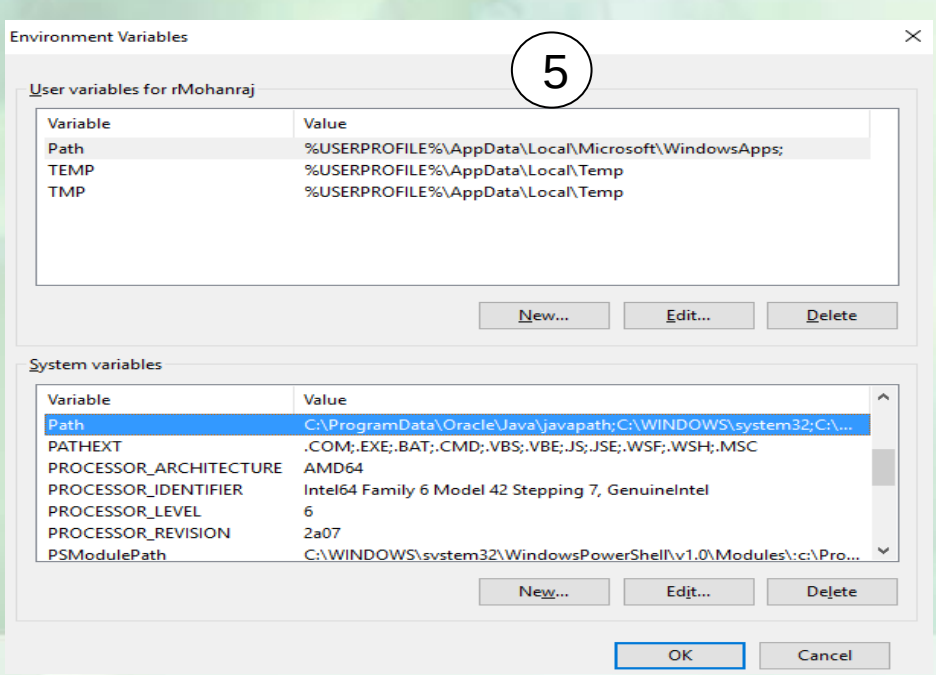
  Add this string to the end of the PATH string in your system environment variables. See the next slide for pictorial representation.

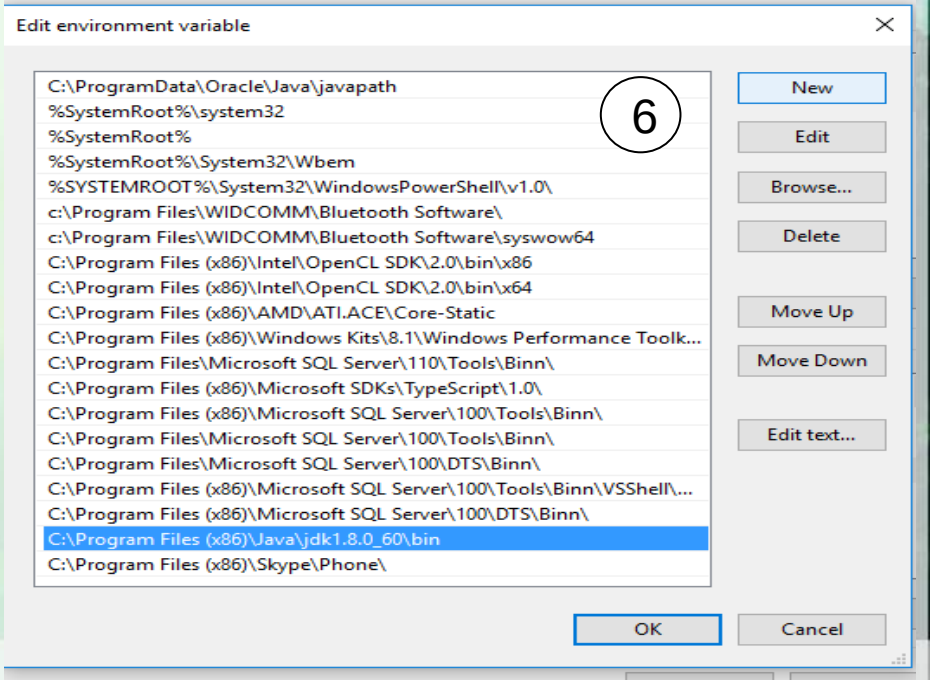
Steps to set the Path Variable – Method 1











Setting Path Variable

- **Method 1 :** To **change** the system environment variables, follow these steps: In the left panel of windows explorer, right-click My Computer and click Properties. In the System Properties **window**, click Advanced System Settings > Environment Variables. Under “System variables” select the path and select “New…” button and enter JAVA_HOME as “Variable name” and the path to your Java JDK directory under “Variable value”
- **Method 2 :** Set Environment Variable in your machine as follows in the command prompt of your working directory if the method1 fails.

  set path=%path%;C:\Program Files (x86)\Java\jdk1.10.0_60\bin

Compiling and Running from a Command Window

- Create a "hello world" example by typing the following lines of code in WordPad (or Notepad), and save in a convenient location as Hello.java.

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

- Open a command window, change directory to the directory that contains Hello.java, and then compile by typing the following command in the console:

```
javac Hello.java
```

- If a compiler error occurs, the compiler attempts to indicate clearly where the mistake occurred.

- When compilation is successful, a new file `Hello.class` has been created. Execute the code with the command

```
java Hello
```

```
//output:  Hello World!
```

Introducing Java Things to understand:

- public
- class
- static
- void
- main
- String[]
- System, System.out, System.out.println (vs System.out.print)
- delimiters: ;, }, { (“blocks”)
- capitalization conventions

Quiz

public static void main(String[] a) { }. This statement is _____

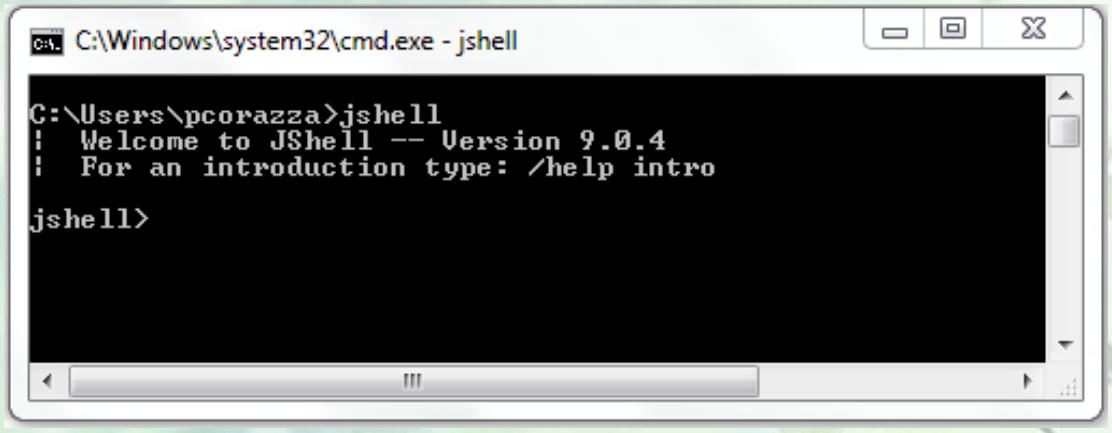
a. True b. False

Byte code is Platform dependent.

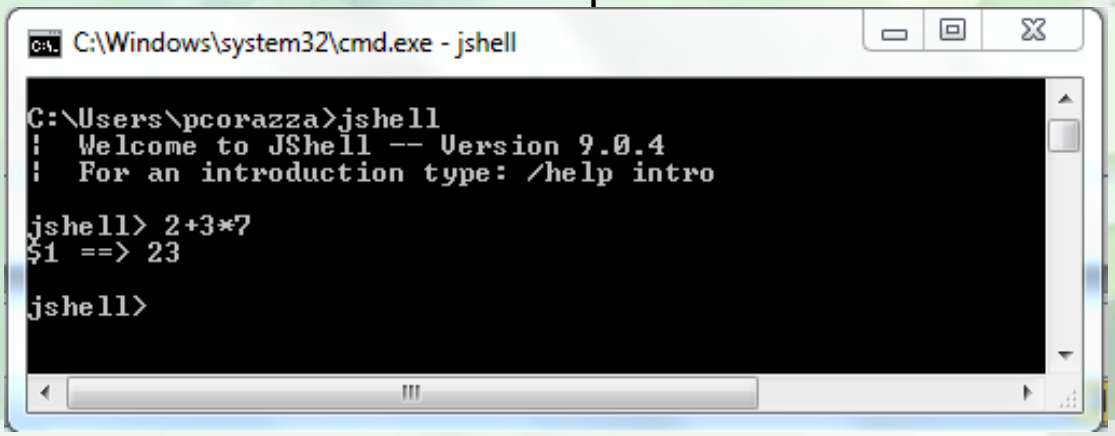
a. True b. False

Trying Out Code with JShell

- JShell is a Java 9 feature that lets you try out code interactively at the command line.
- To start JShell, type jshell at the command line

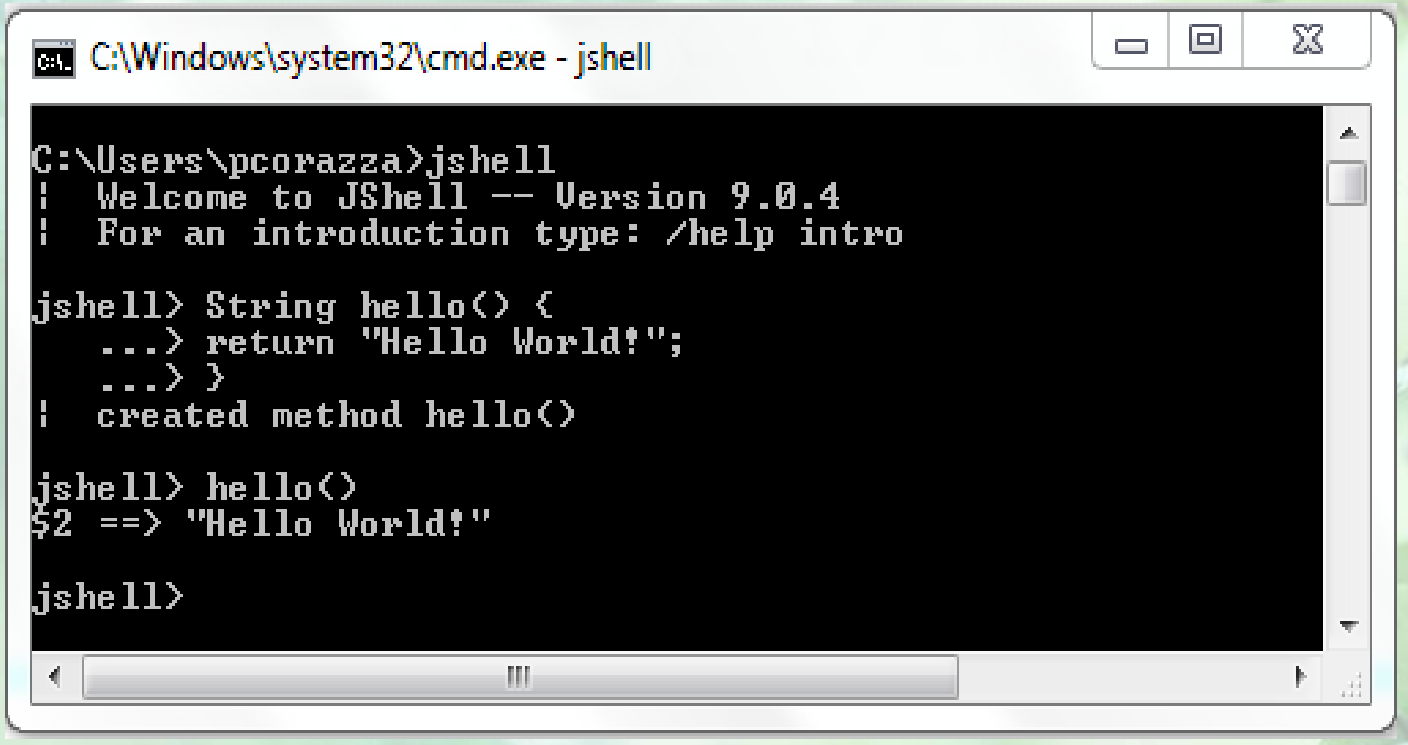


- You can use JShell to evaluate expressions



(continued)

- You can define a Java method in JShell and then run it



- Exit JShell by typing /exit at the jshell prompt

Main Point 1

- Java is an object-oriented language that is easier to use, similar to C++, but simplified to eliminate language features that cause common programming errors.
- *Science of Consciousness* :Transcendental Meditation is an easy and effortless technique to make a mind clear and more alert so that we can achieve anything in our life without error.

Integrated Development Environments

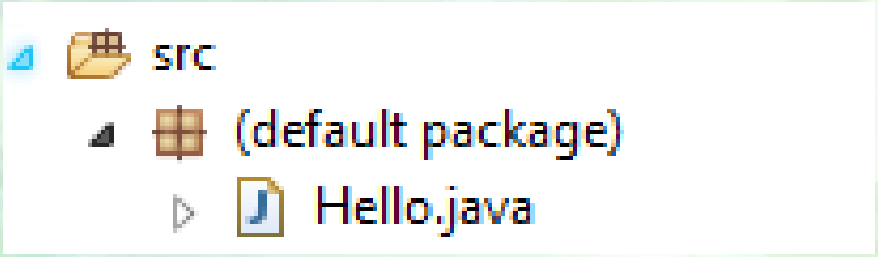
- A good IDE supports compiling, running, and debugging code with tools that are integrated and typically easy to use. For a large Java project, an IDE is indispensable.

- Good choices of IDE are NetBeans, Sun Forte, IBM Rational Application Developer (formerly WebSphere Application Developer), Borland's JBuilder, JetBrains' IntelliJ

- Another excellent choice, which has become an industry standard, is the open-source IDE Eclipse, written entirely in Java. Among IDEs for Java, in recent years, Eclipse is the most widely used. We will use Eclipse in this course.
  - Install JDK
  - Install Eclipse IDE
  - Getting started. To use Java 10, you need Eclipse Oxygen (or later); earlier versions of Eclipse do not support Java 10.
    www.vogella.com/articles/Eclipse/article.html

"Hello World" in Eclipse

- When you create a class Hello in Eclipse, it creates a default package. Later we will see how to place a class in a different package.



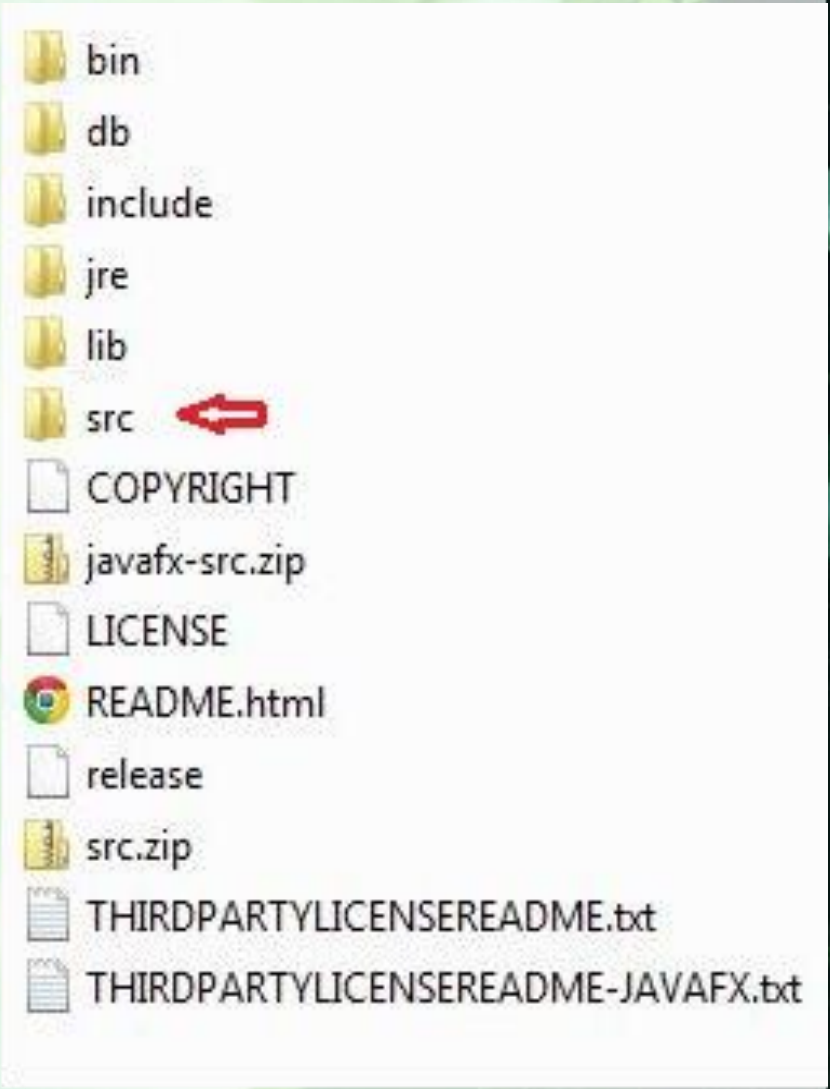
- We can directly print "Hello World", or we can ask a function to return the String "Hello World" and then print that returned String

```
public class HelloSimple {
    public static void main(String[] args) {
        System.out.println("Hello World!");
    }
}
```

```
public class Hello {
    public static void main(String[] args) {
        System.out.println(hello());
    }
    static String hello() {
        return "Hello World!";
    }
}
```

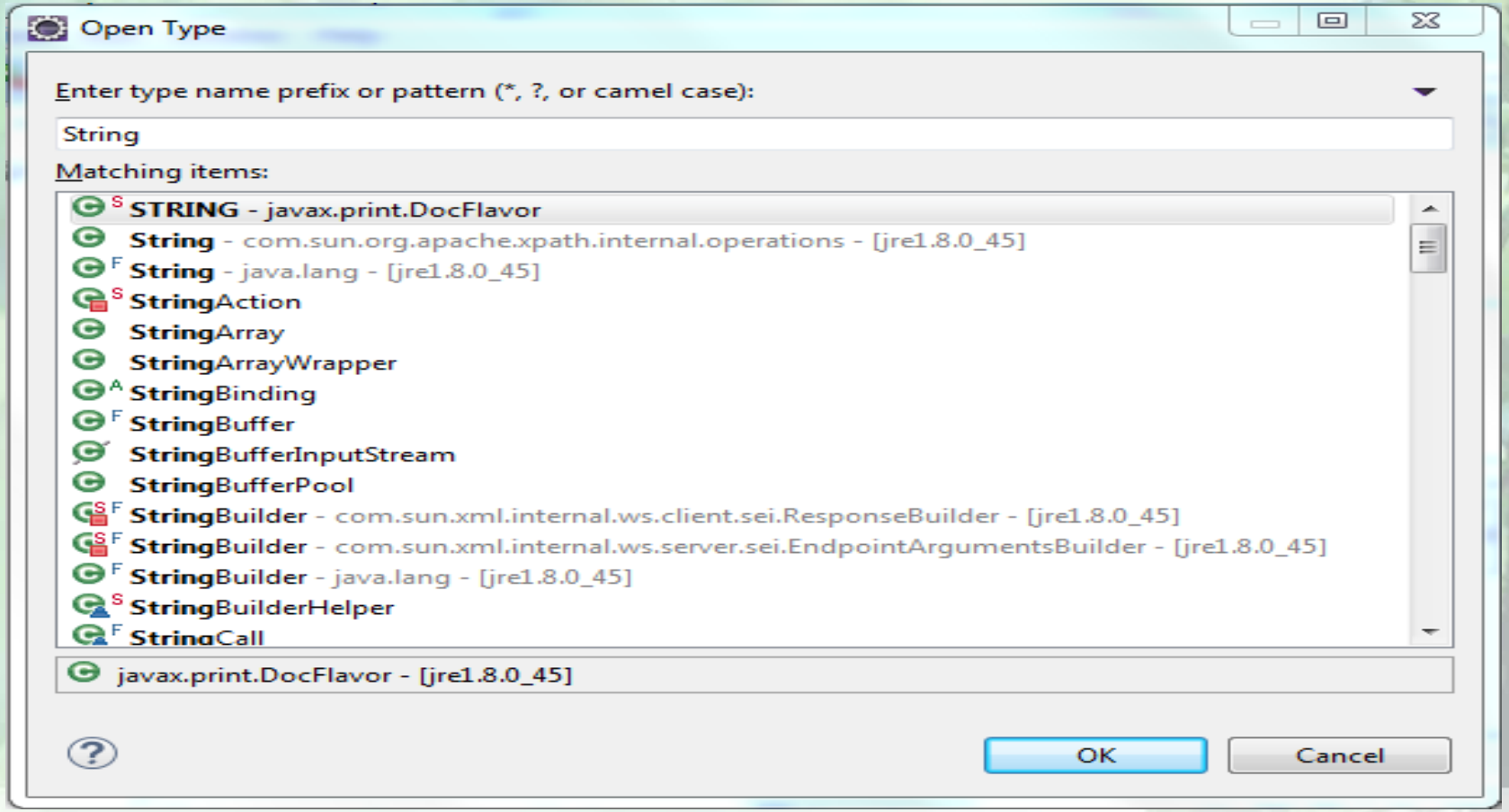
The JDK Distribution & Viewing Source Code

- It's a good idea to unzip src.zip and place in a source folder (called “src” in the screen capture above) – this will allow you to see how Oracle has implemented its library classes. You can also unzip javafx-src.zip to look at JavaFX source code.

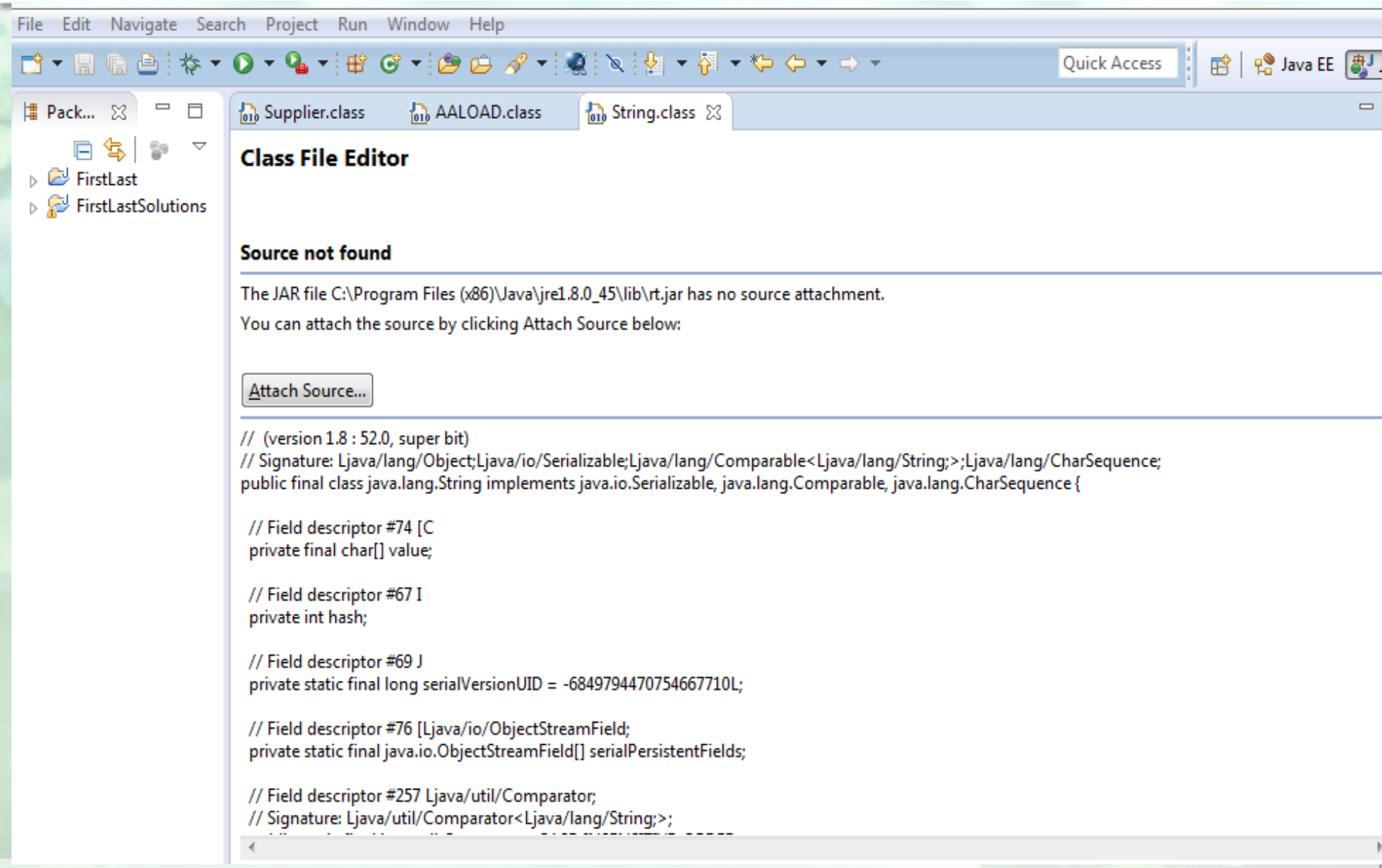


View of Attaching Java Source Code from API

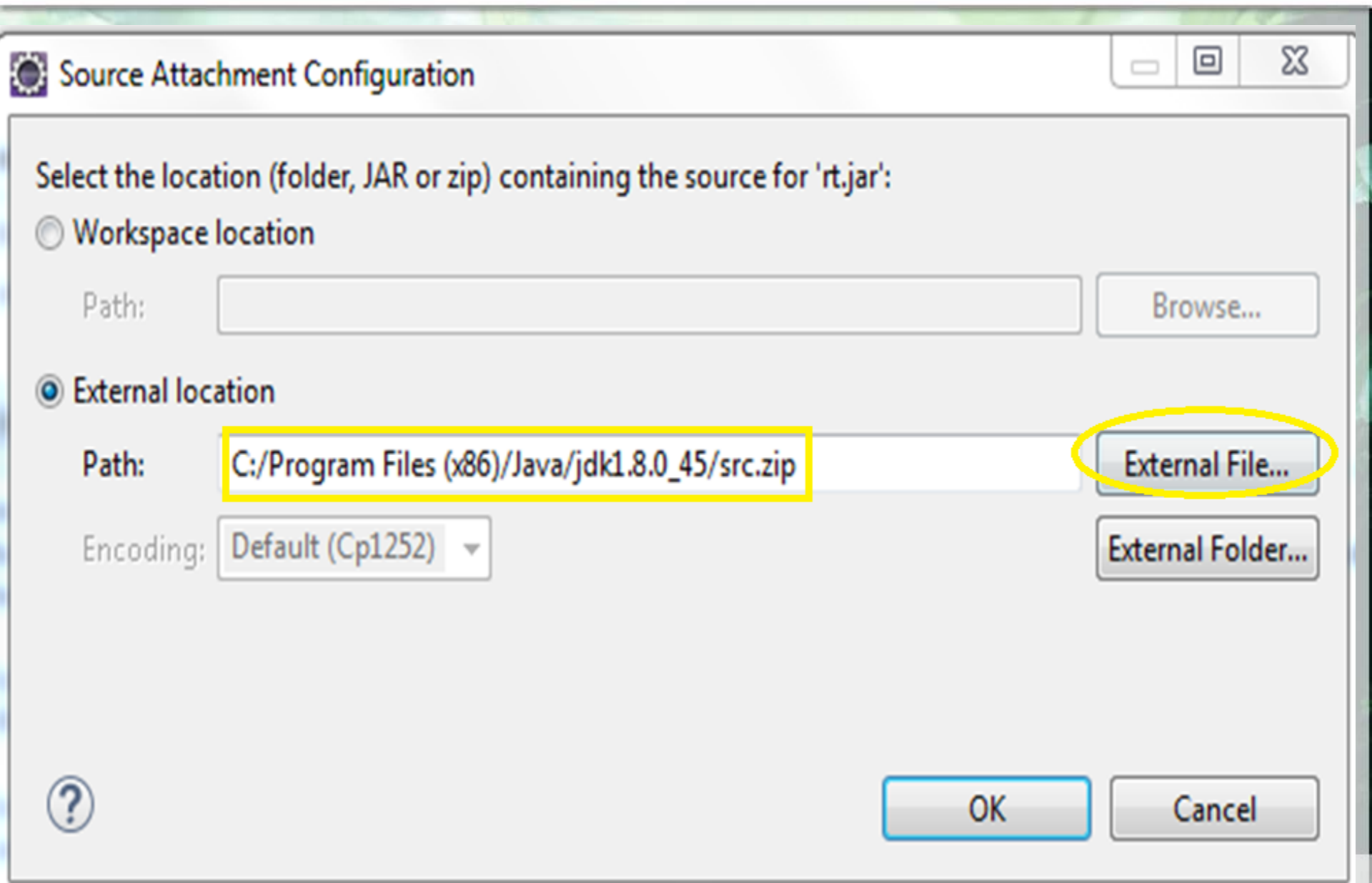
- Allows you to see how Java has implemented their library code.
- Steps:
  1. Ctrl – Shift – T
     - Brings up this screen: (Here, "String" was typed into search window")
     - Click OK.



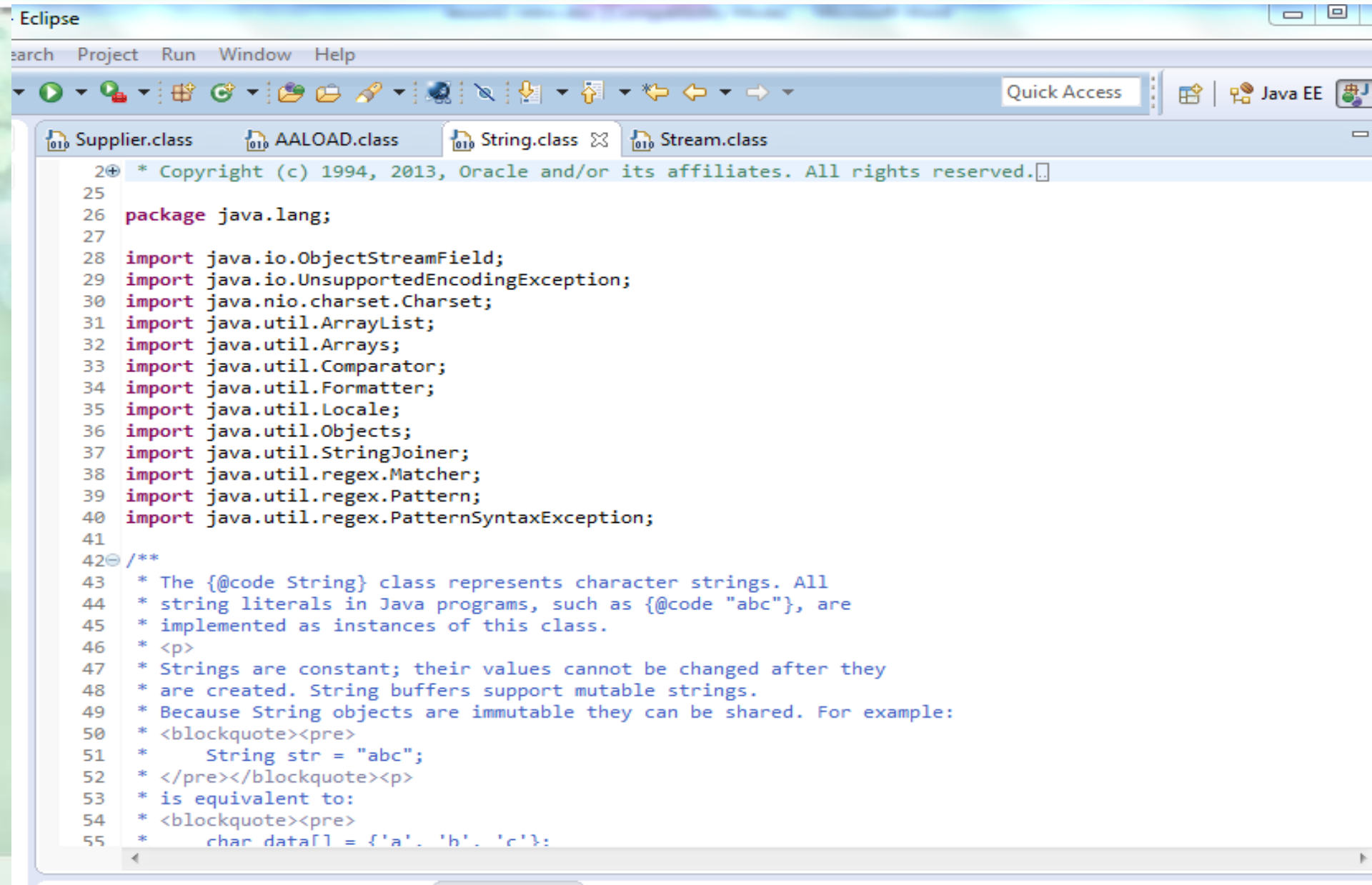
2. After Clicking OK from previous screen lets you see incomplete info of the class file (of String in this case), Click Attach Source



3. After Clicking Attach Source and navigate to src.zip in your jdk distribution (do only once – next time just do steps A and B) by clicking External File**. In JDK 10 src.zip is available in lib directory.**

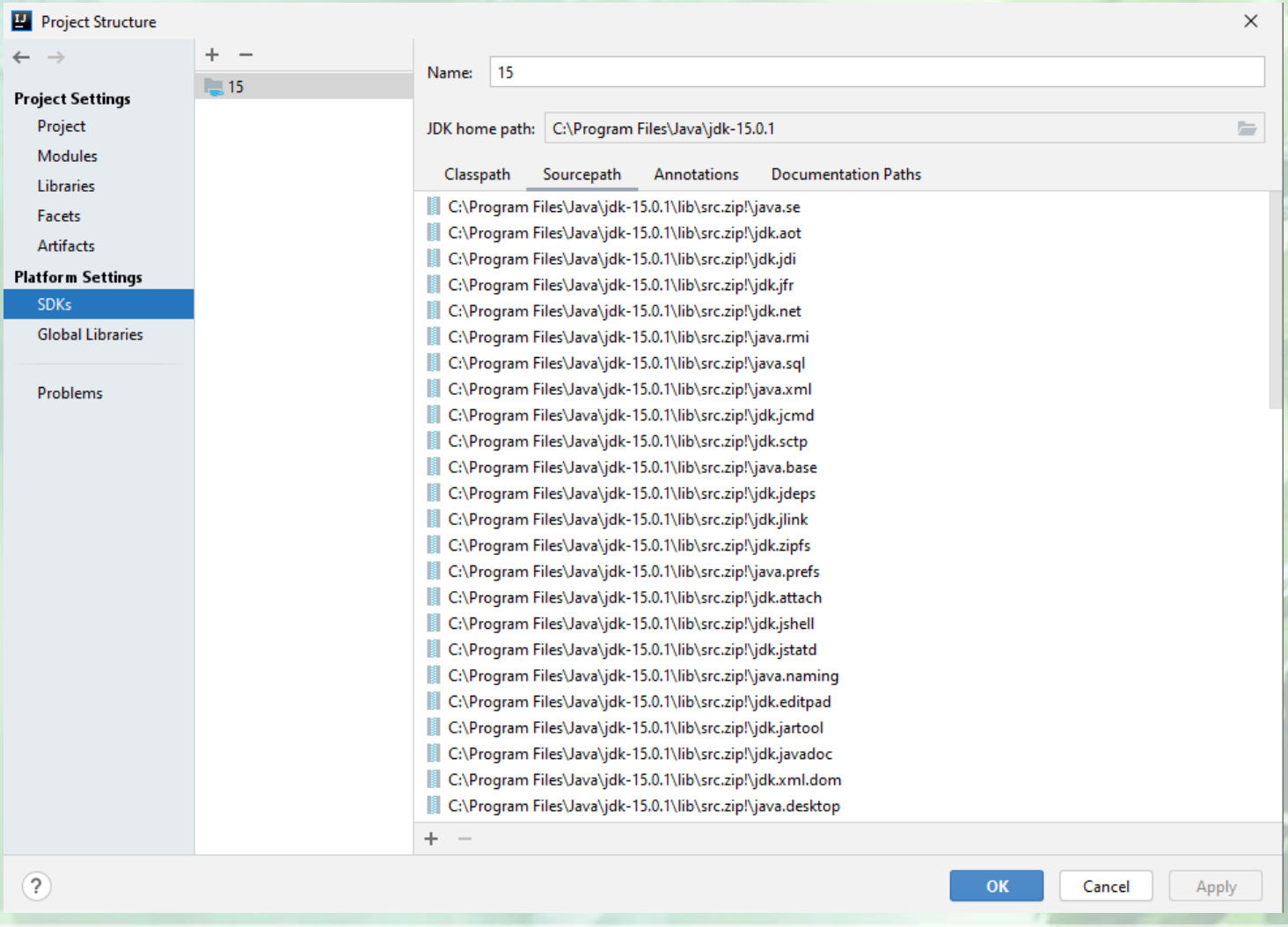


4. You can now see the source code for your selected type



Intellij attach Java src.zip

5. File -> Project Structure -> SDK



MAIN POINT 2

- Eclipse is a leading, open-source, 100% Java, integrated development environment, which provides excellent support for editing, compiling, running, and debugging Java applications. By analogy, to create a good life, we need to handle inner life and at the same time, structure a life-supporting environment – the goal is to live 200% of life.

UNITY CHART

CONNECTING THE PARTS OF KNOWLEDGE

WITH THE WHOLENESS OF KNOWLEDGE

The Usefulness of Java and Eclipse IDE

- Using Java, highly functional applications can be built more quickly and easily with fewer mistakes than is typically possible using C or C++.
- To optimize the use of Java's features, IDE's such as Eclipse ease the work of the developer by handling in the background many routine tasks.

- ***Transcendental Consciousness*** is the source from the field of pure intelligence which is the basis for successful action.
- ***Impulses within the Transcendental Field :*** *At this level of experience, consciousness acting within itself to produce the desired outcome.*
- ***Wholeness moving within Itself****: In Unity Consciousness, the pure intelligence located in TC is found pervading all of creation, from gross to subtle.*