OODP- OOA/OOD– Session 3-c

Session times

PT group 1	– Monday	18:00-21:00	room: Malet 403
PT group 2	– Thursday	18:00-21:00	room: Malet 407
FT	- Tuesday	13:30-17:00	room: Malet 404

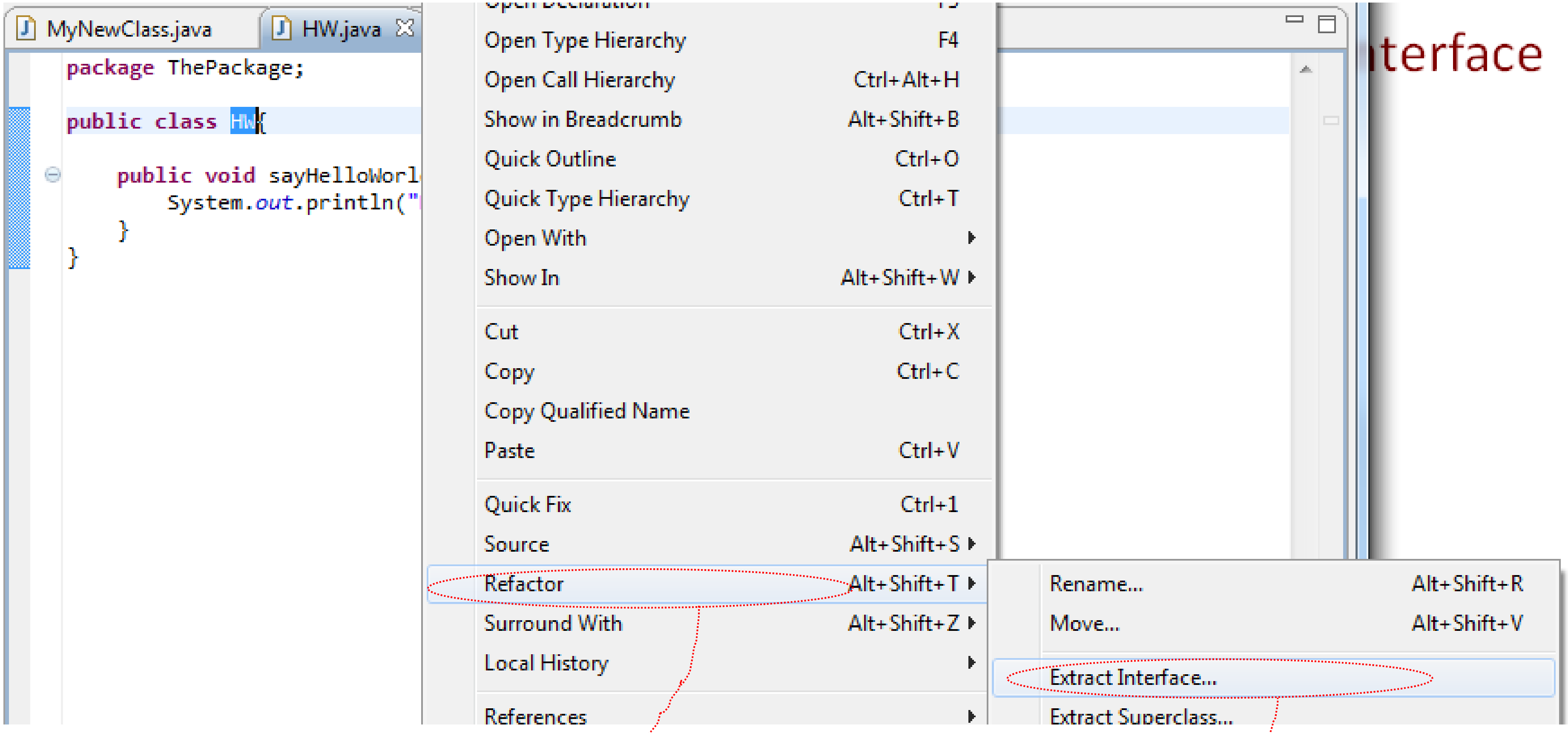
Email: oded@dcs.bbk.ac.uk

Web Page: http://www.dcs.bbk.ac.uk/~oded

Visiting Hours: Tuesday 17:00 to 19:00

Refactor – Extract - Interface

1. **Make sure you are on page**
2. **Mouse right click**

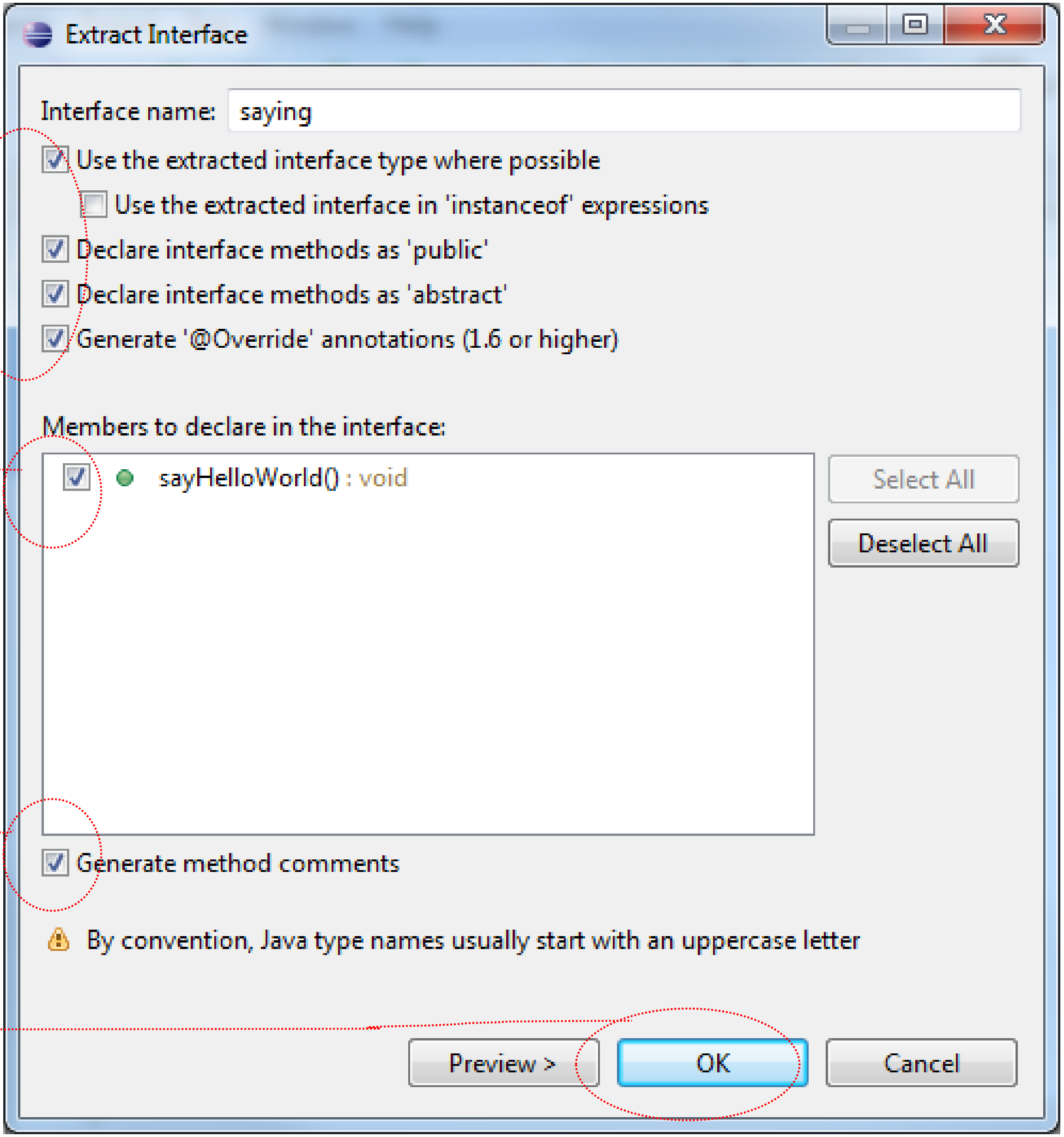


```
package ThePackage;

public class HW{

    public void sayHelloWorl
        System.out.println("
    }
}
```

3. Select

4. Select

Extract Interface



Interface name: saying

☑ Use the extracted interface type where possible

□ Use the extracted interface in 'instanceof' expressions

☑ Declare interface methods as 'public'

☑ Declare interface methods as 'abstract'

☑ Generate '@Override' annotations (1.6 or higher)

Members to declare in the interface:

☑ ● sayHelloWorld() : void

☑ Generate method comments

⚠ By convention, Java type names usually start with an uppercase letter

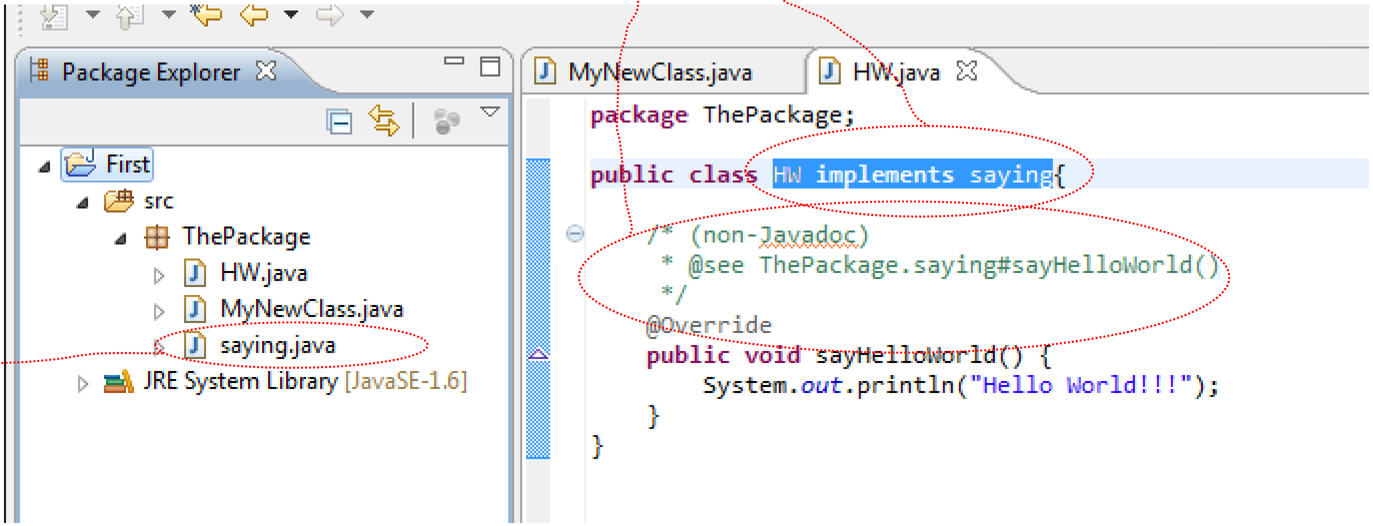
1. Make sure selected

2. select

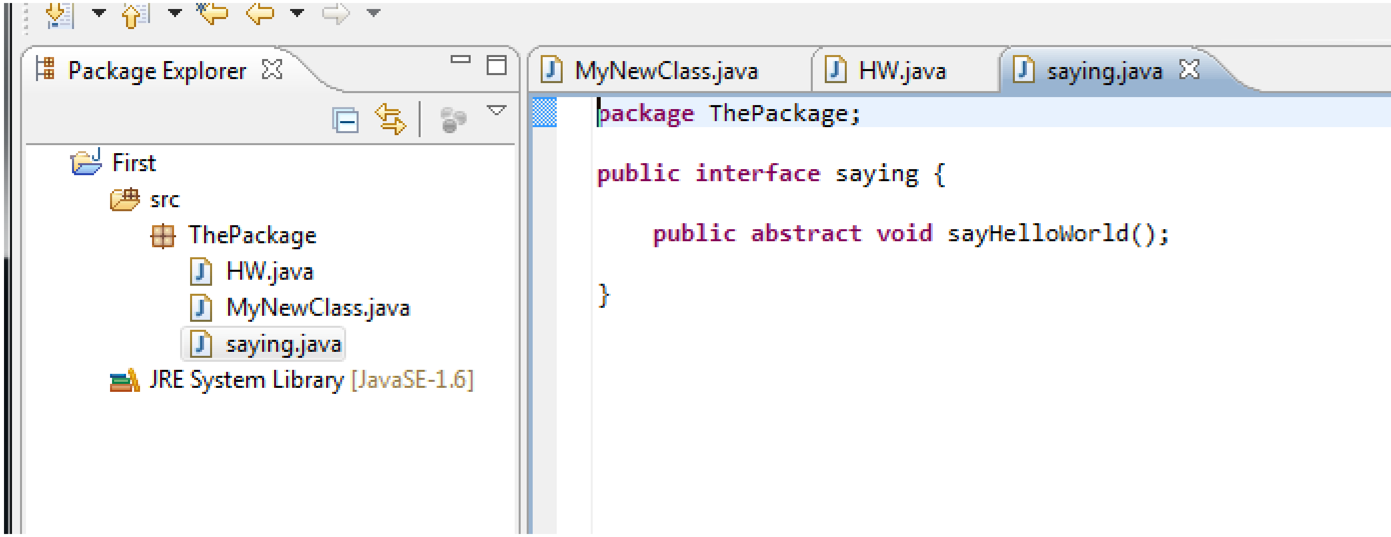
Extract-Interface

1. observe

2. select



```
package ThePackage;

public class HW implements saying{

    /* (non-Javadoc)
     * @see ThePackage.saying#sayHelloWorld()
     */
    @Override
    public void sayHelloWorld() {
        System.out.println("Hello World!!!");
    }
}
```

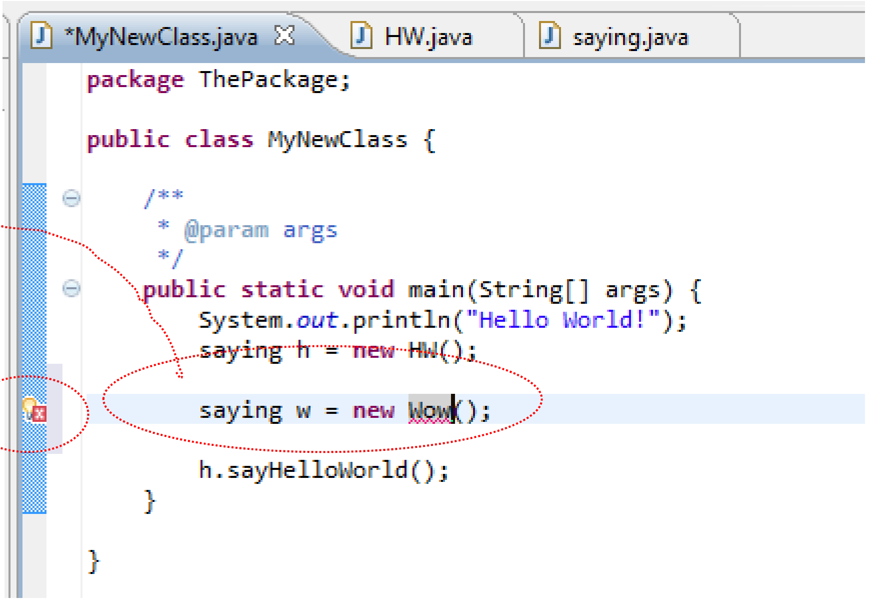


```
package ThePackage;

public interface saying {

    public abstract void sayHelloWorld();

}
```

New line of code

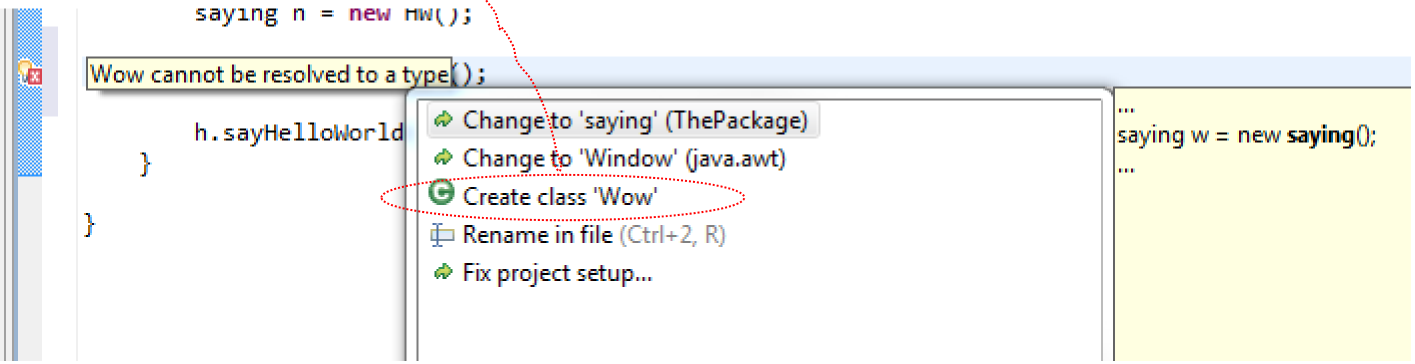
1. Add line of code

2. Select the x



```
package ThePackage;

public class MyNewClass {

    /**
     * @param args
     */
    public static void main(String[] args) {
        System.out.println("Hello World!");
        saying h = new HW();

        saying w = new Wow();

        h.sayHelloWorld();
    }

}
```

3.select



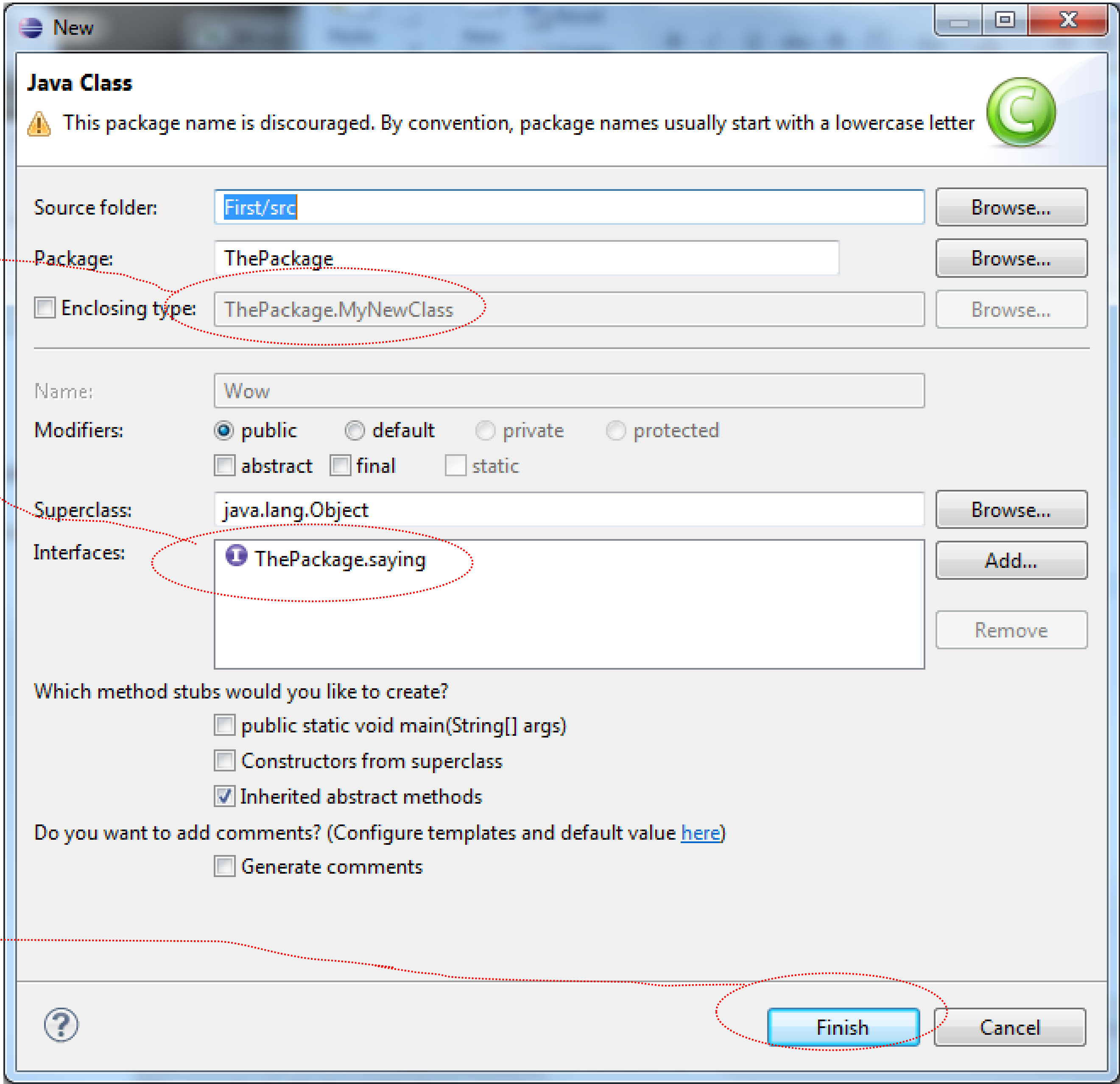
```
        saying h = new HW();
        Wow cannot be resolved to a type();

        h.sayHelloWorld
    }

}
```

Wow cannot be resolved to a type

Change to 'saying' (ThePackage)
Change to 'Window' (java.awt)
Create class 'Wow'
Rename in file (Ctrl+2, R)
Fix project setup...

```
...
saying w = new saying();
...
```

Create new class



New

Java Class

⚠ This package name is discouraged. By convention, package names usually start with a lowercase letter

Source folder: First/src [Browse...]

Package: ThePackage [Browse...]

□ Enclosing type: ThePackage.MyNewClass [Browse...]

Name: Wow

Modifiers: ● public ○ default ○ private ○ protected

□ abstract □ final □ static

Superclass: java.lang.Object [Browse...]

Interfaces: ThePackage.saying [Add...] [Remove]

Which method stubs would you like to create?

□ public static void main(String[] args)

□ Constructors from superclass

☑ Inherited abstract methods

Do you want to add comments? (Configure templates and default value here)

□ Generate comments

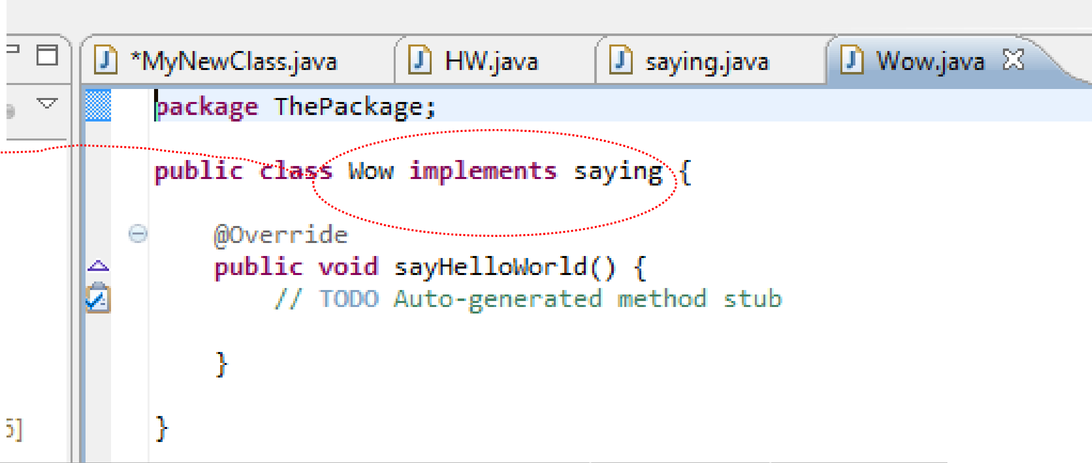
[Finish] [Cancel]

1. Observe

2. Select

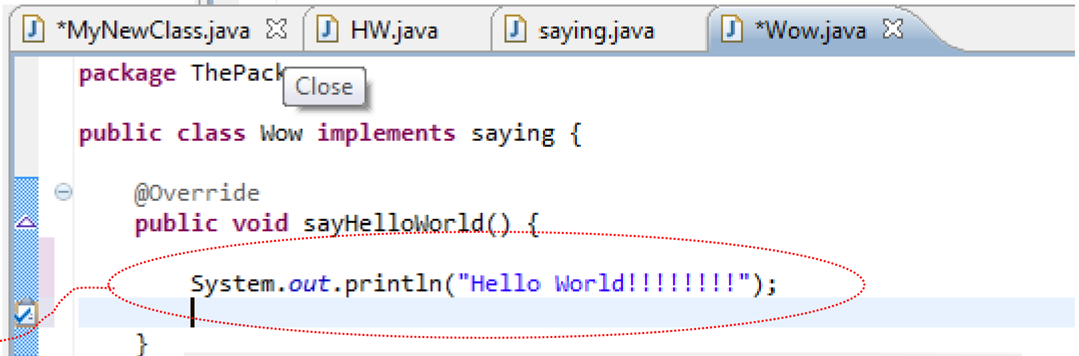
New class

1. Observe

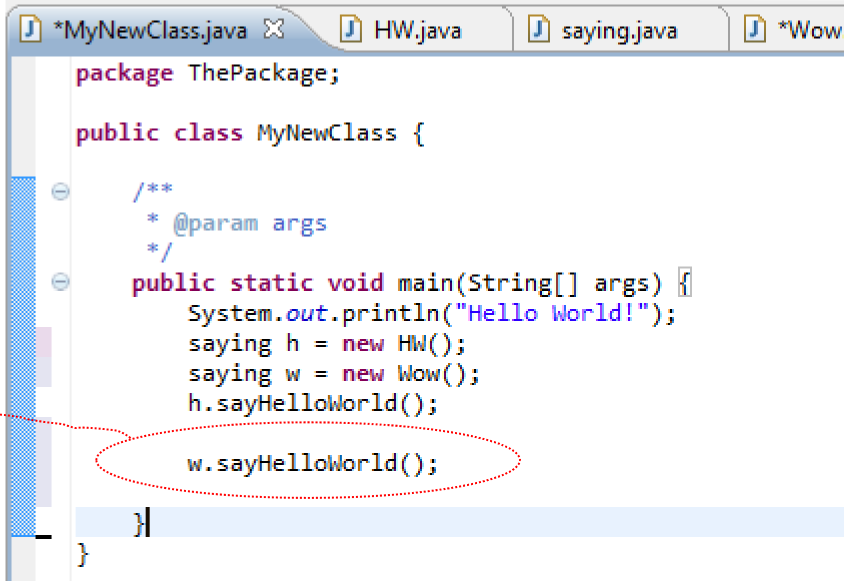


```
package ThePackage;

public class Wow implements saying {

    @Override
    public void sayHelloWorld() {
        // TODO Auto-generated method stub

    }

}
```

2. Add code



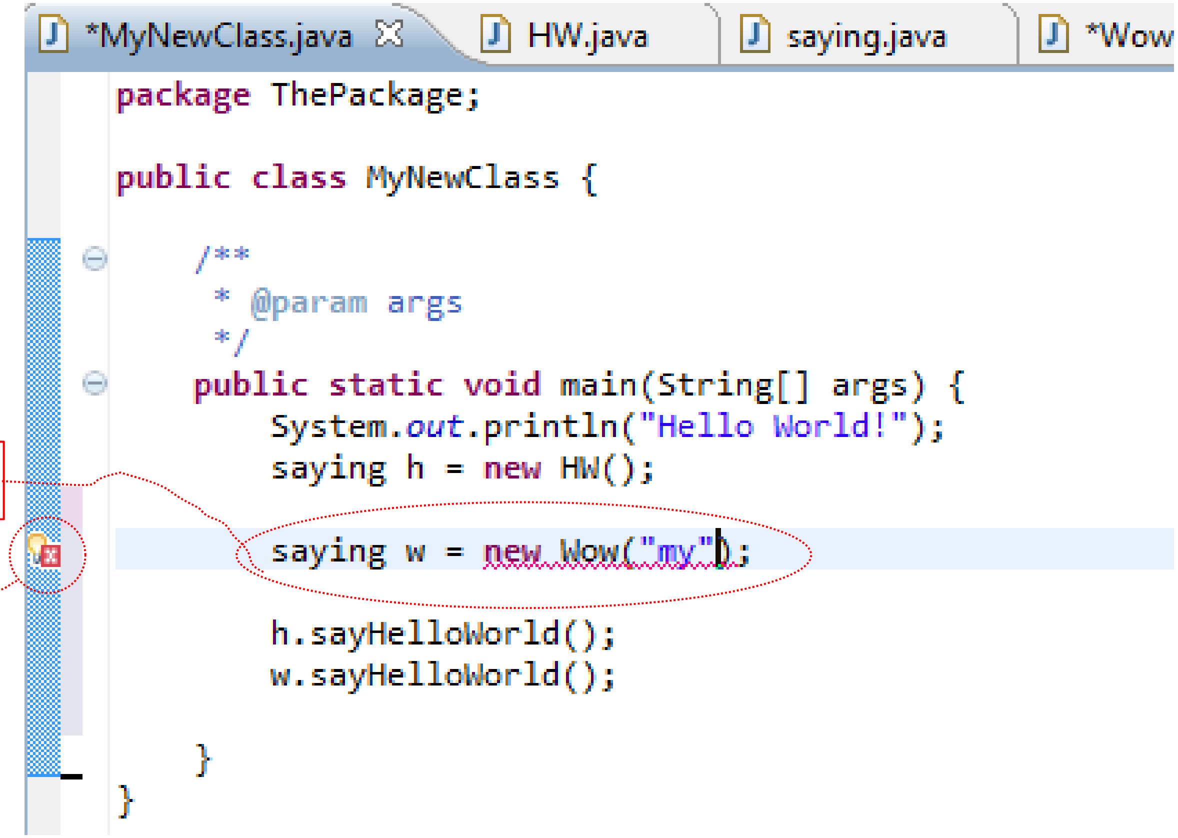
```
package ThePack

public class Wow implements saying {

    @Override
    public void sayHelloWorld() {

        System.out.println("Hello World!!!!!!!!");

    }

}
```



```
package ThePackage;

public class MyNewClass {

    /**
     * @param args
     */
    public static void main(String[] args) {
        System.out.println("Hello World!");
        saying h = new HW();
        saying w = new Wow();
        h.sayHelloWorld();

        w.sayHelloWorld();

    }
}
```

3. Run

Automatically generated constructor



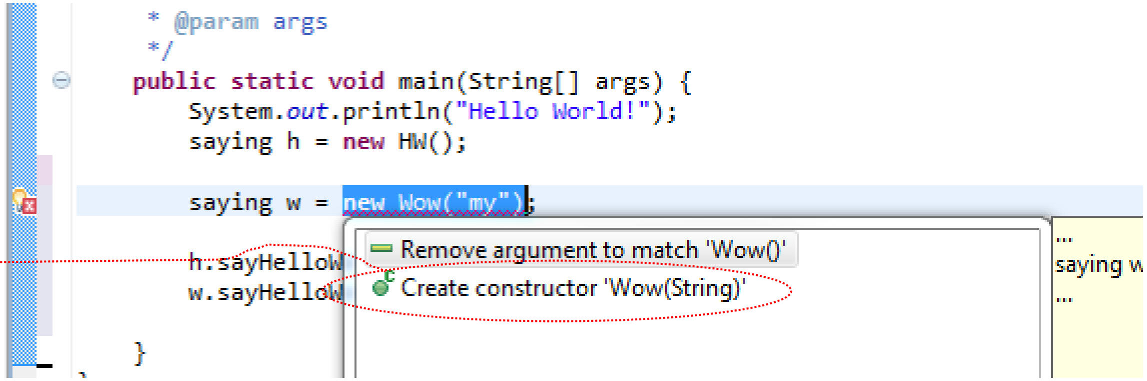
```
package ThePackage;

public class MyNewClass {

    /**
     * @param args
     */
    public static void main(String[] args) {
        System.out.println("Hello World!");
        saying h = new HW();

        saying w = new Wow("my");

        h.sayHelloWorld();
        w.sayHelloWorld();

    }
}
```

2. Add code

2. Select the x

Automatically generated constructor

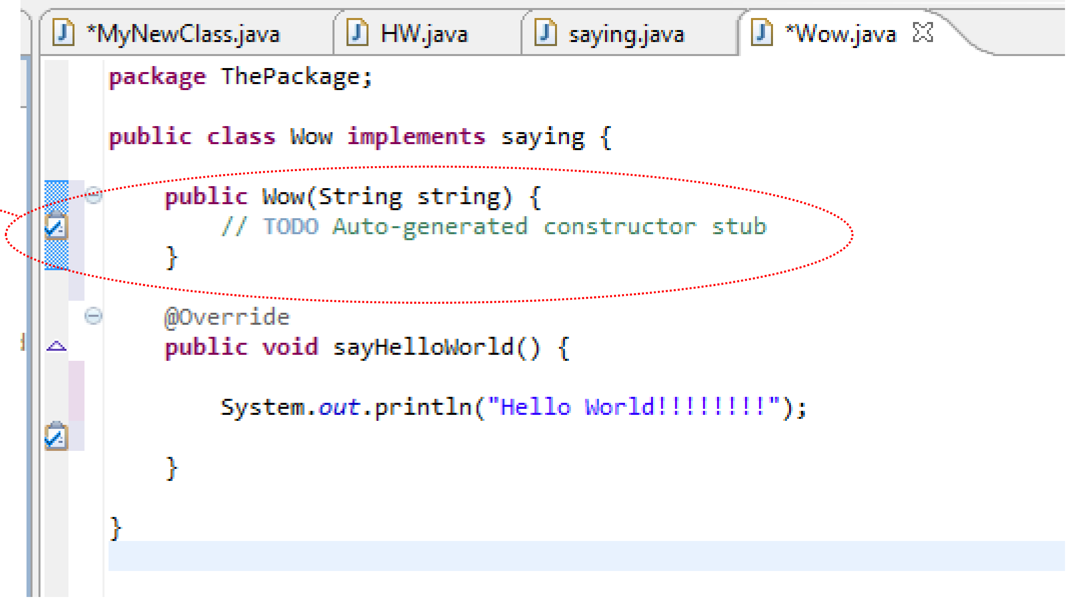
1. select



```
     * @param args
     */
    public static void main(String[] args) {
        System.out.println("Hello World!");
        saying h = new HW();

        saying w = new Wow("my");

        h.sayHelloW
        w.sayHelloW

    }
```

Remove argument to match 'Wow()'
Create constructor 'Wow(String)'
...
saying w
...

2. observe

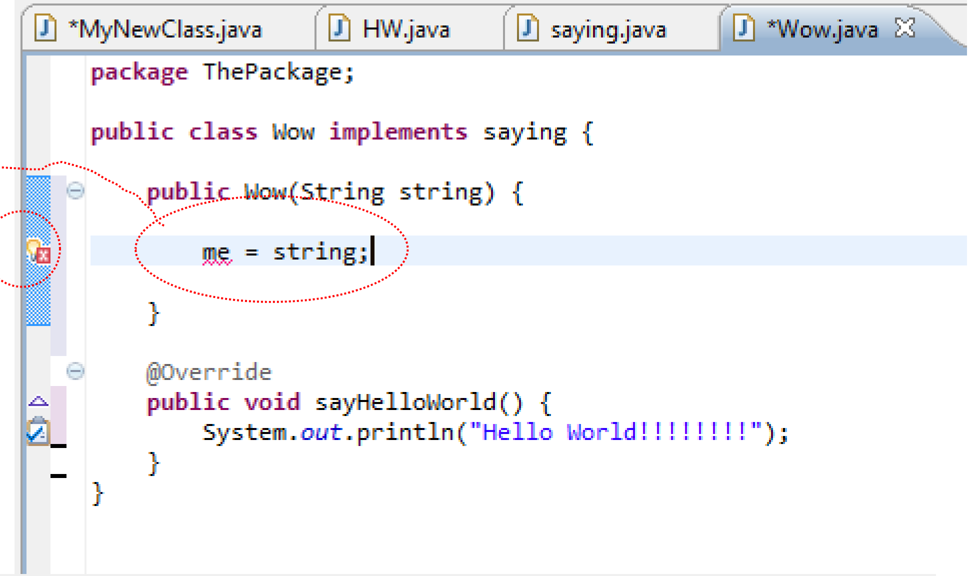


```
package ThePackage;

public class Wow implements saying {

    public Wow(String string) {
        // TODO Auto-generated constructor stub
    }

    @Override
    public void sayHelloWorld() {

        System.out.println("Hello World!!!!!!!!");

    }

}
```

Making constructor interesting

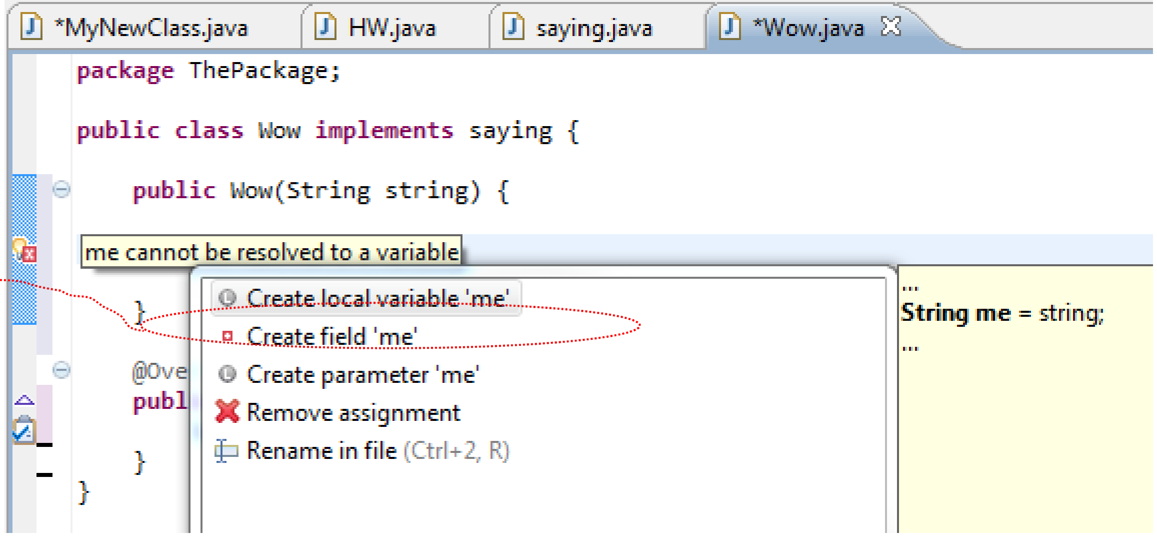
1. Add code

2. Select the x



```
package ThePackage;

public class Wow implements saying {

    public Wow(String string) {

        me = string;

    }

    @Override
    public void sayHelloWorld() {
        System.out.println("Hello World!!!!!!!!");
    }
}
```

3. Select



```
package ThePackage;

public class Wow implements saying {

    public Wow(String string) {
```

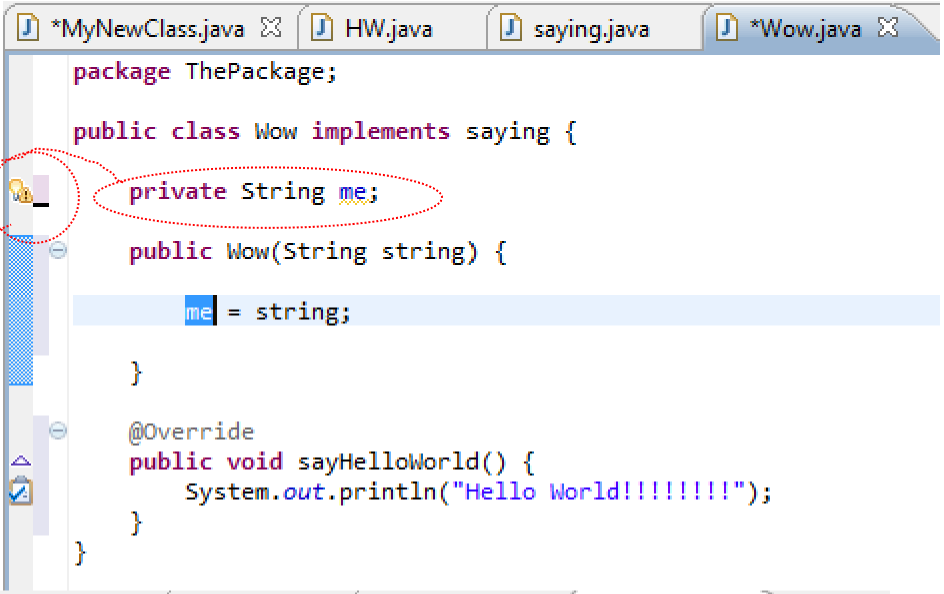
me cannot be resolved to a variable

```
...
String me = string;
...
```

New field

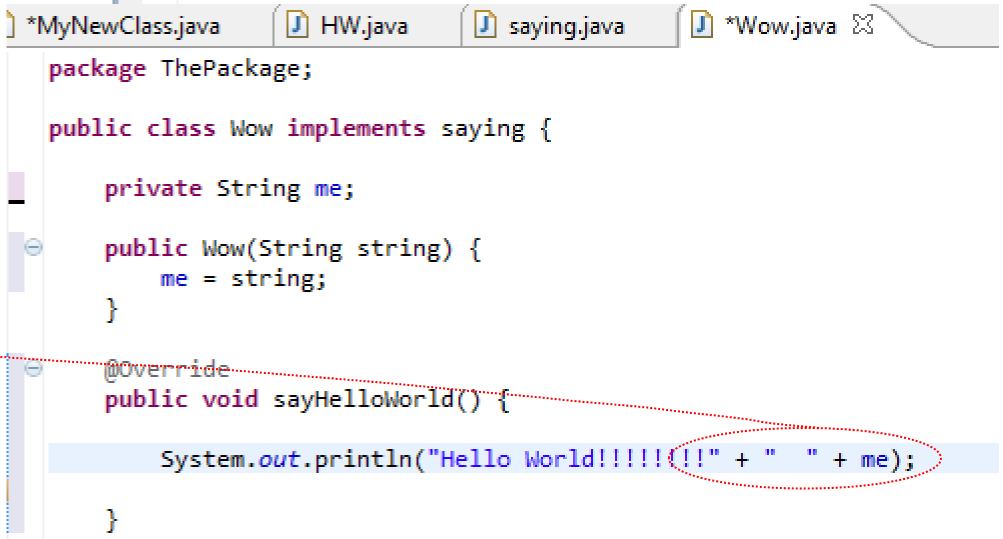
1. observe

Appeared because String “me” is not used



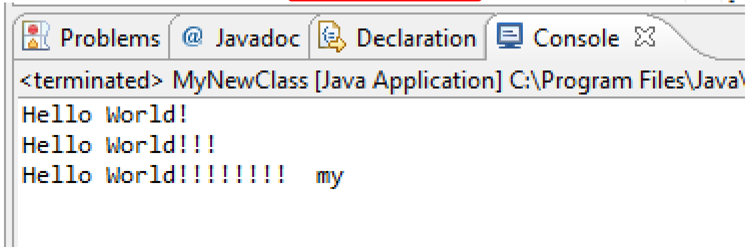
```
package ThePackage;

public class Wow implements saying {

    private String me;

    public Wow(String string) {

        me = string;

    }

    @Override
    public void sayHelloWorld() {
        System.out.println("Hello World!!!!!!!!");
    }
}
```

2. Add code



```
package ThePackage;

public class Wow implements saying {

    private String me;

    public Wow(String string) {
        me = string;
    }

    @Override
    public void sayHelloWorld() {

        System.out.println("Hello World!!!!!!!!" + "  " + me);

    }
```

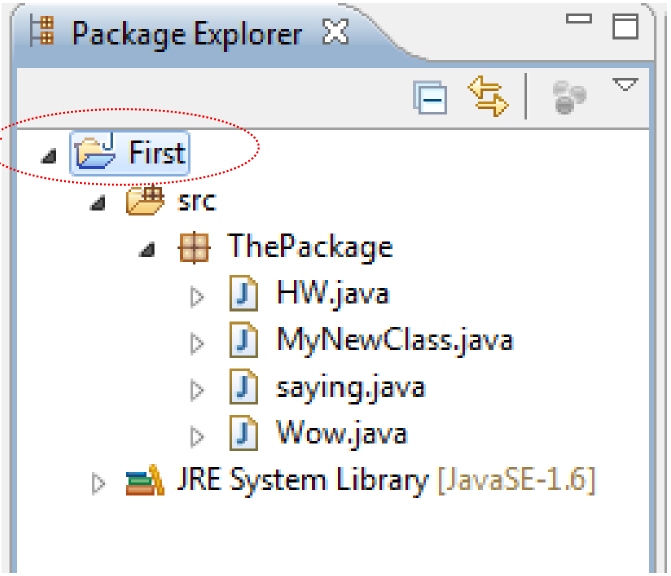
3. Run

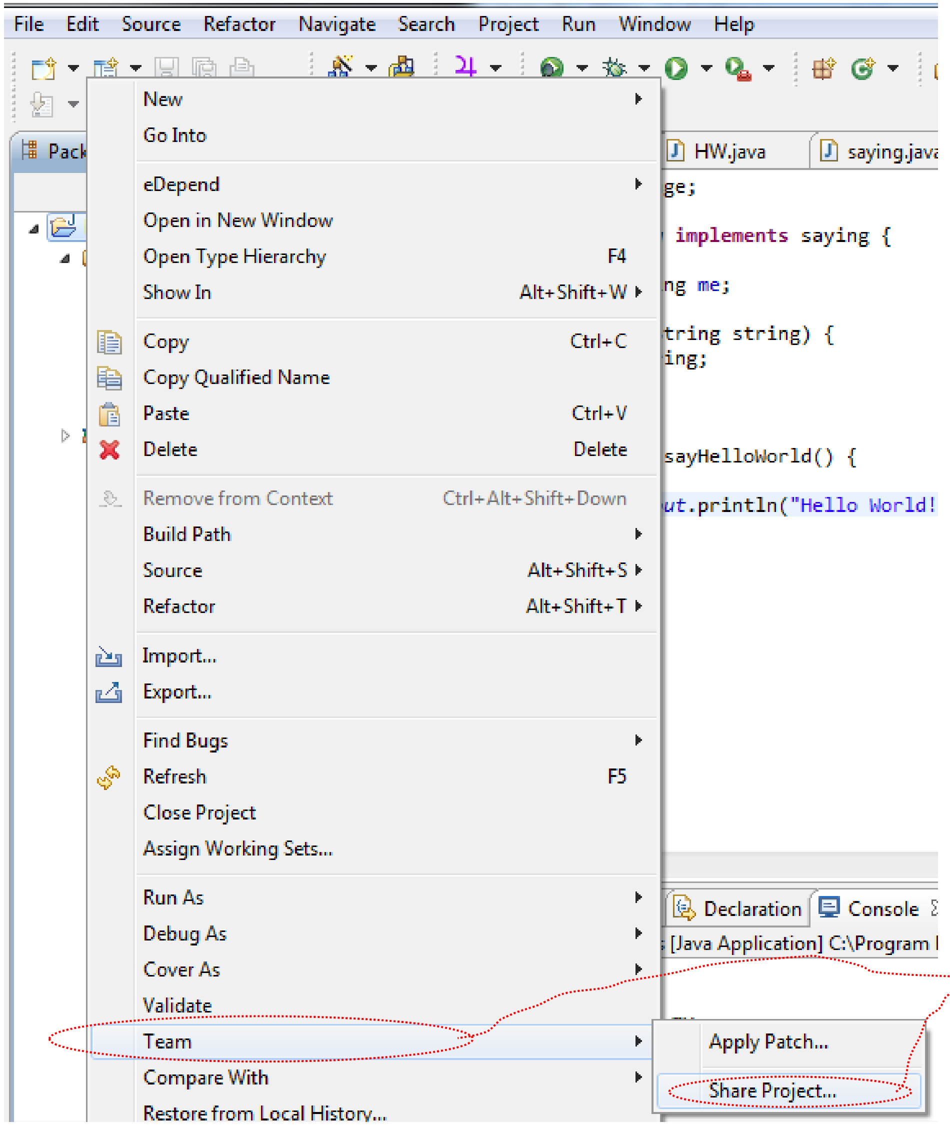


```
<terminated> MyNewClass [Java Application] C:\Program Files\Java\
Hello World!
Hello World!!!
Hello World!!!!!!!!  my
```

Git

1. Right click





```
ge;
 implements saying {
ng me;
tring string) {
ing;
sayHelloWorld() {
ut.println("Hello World!
```

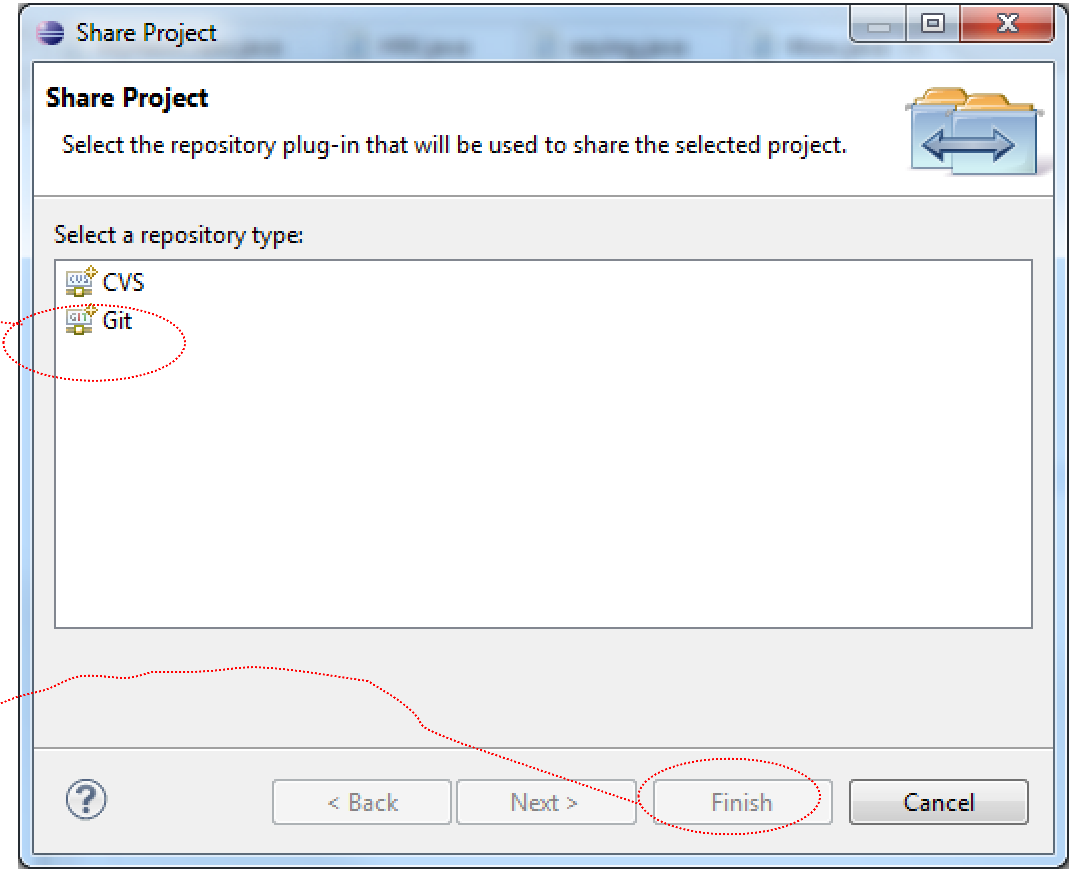
[Java Application] C:\Program

2. Select

Share Project

1. select

3. select



Share Project

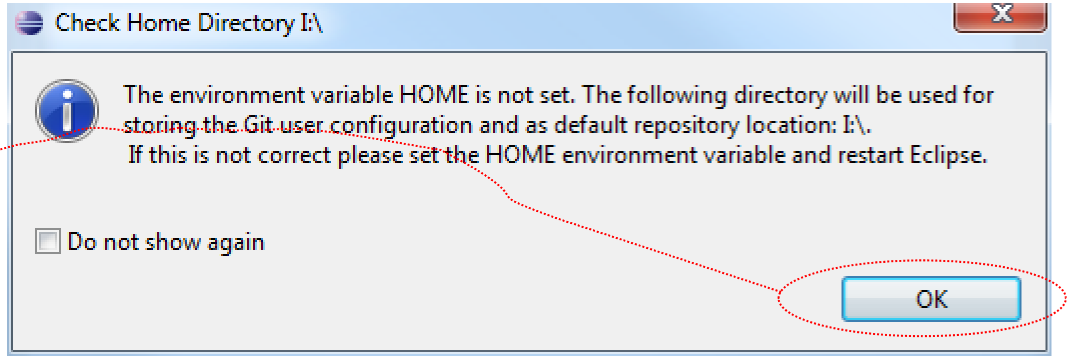
Share Project

Select the repository plug-in that will be used to share the selected project.

Select a repository type:

- CVS
- Git

2. select



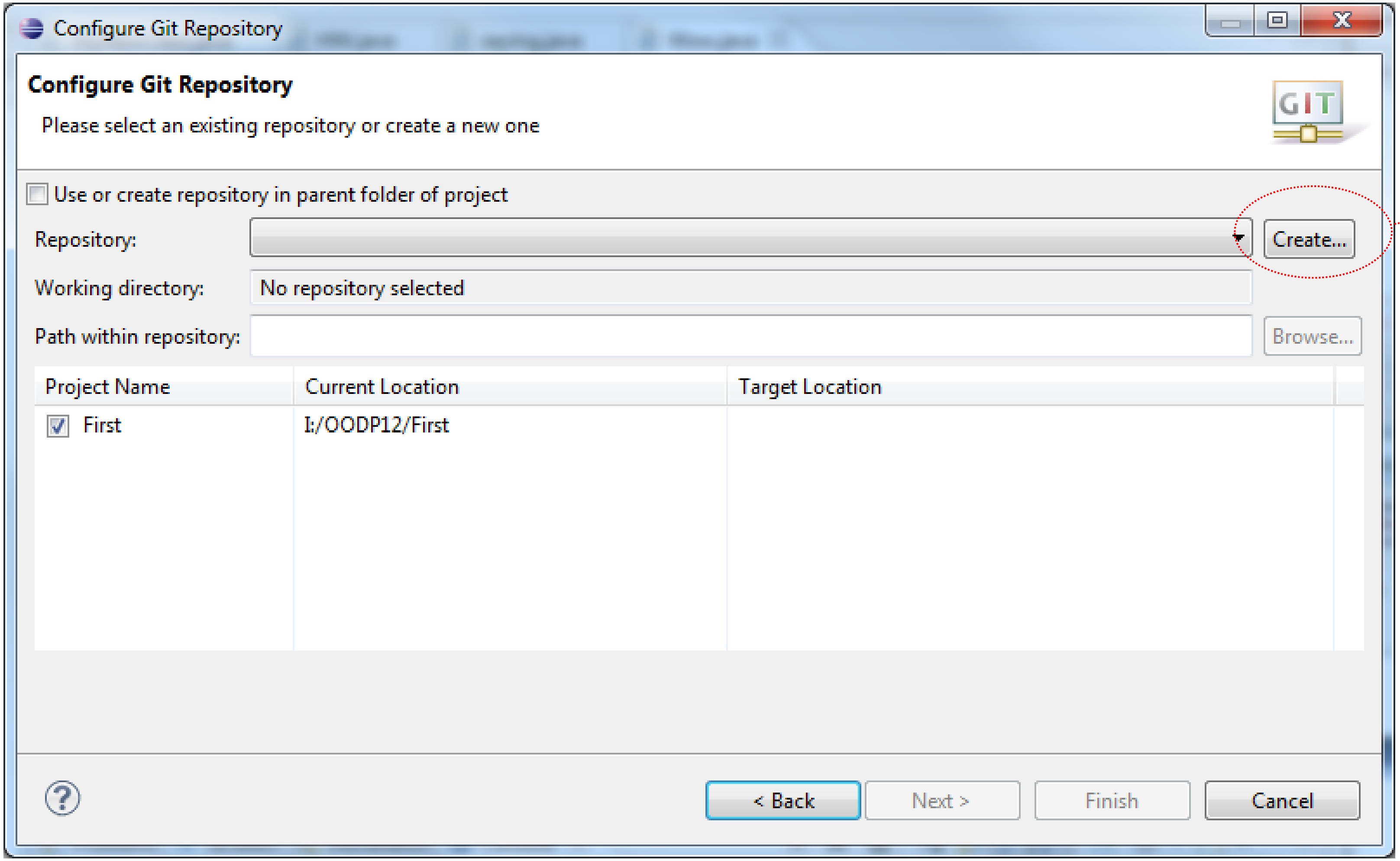
Check Home Directory I:\

The environment variable HOME is not set. The following directory will be used for storing the Git user configuration and as default repository location: I:\.
If this is not correct please set the HOME environment variable and restart Eclipse.

☐ Do not show again

Configure git repository

2. select



Configure Git Repository

Configure Git Repository

Please select an existing repository or create a new one

□ Use or create repository in parent folder of project

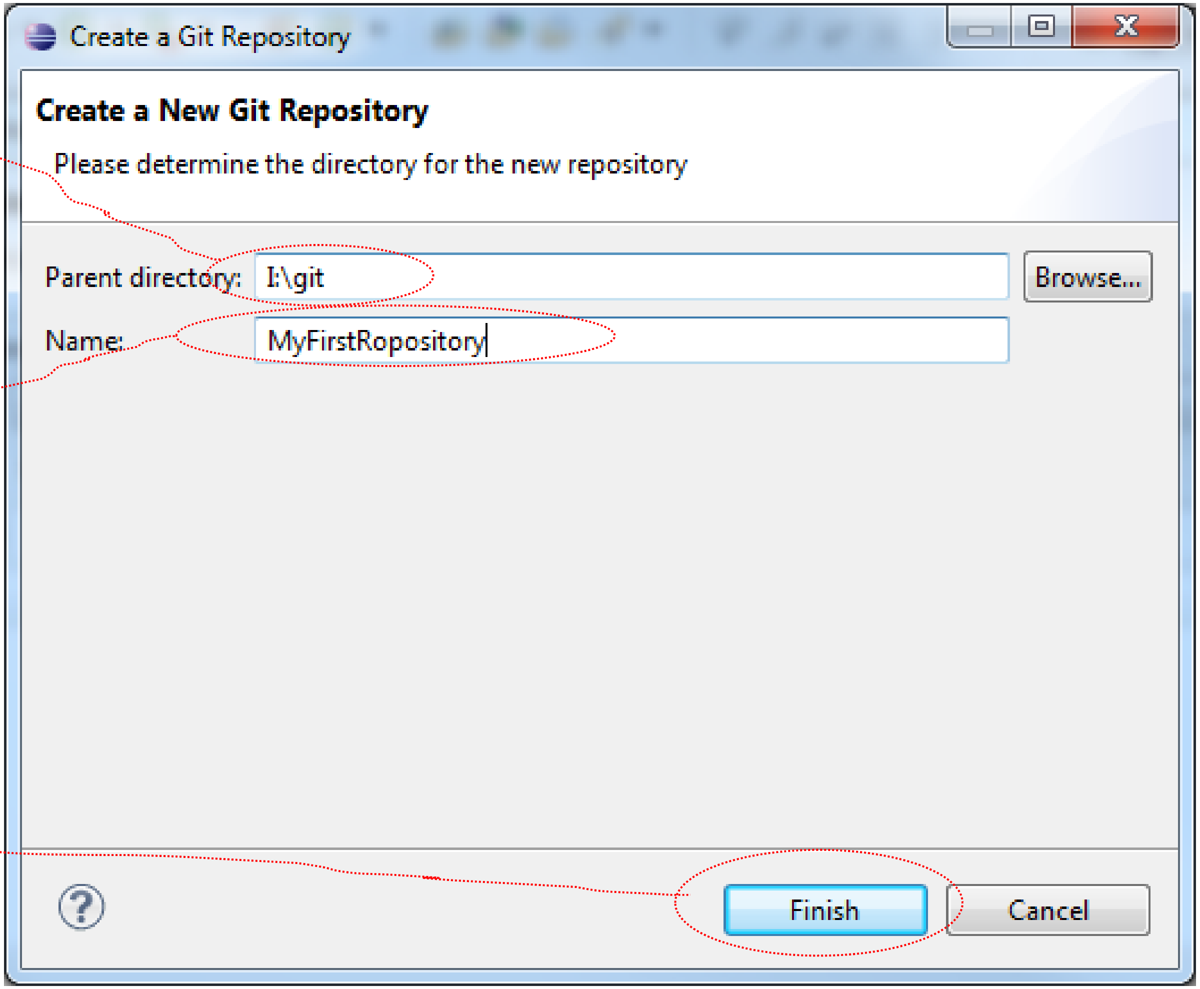
Repository:

Working directory: No repository selected

Path within repository:

Project Name	Current Location	Target Location
☑ First	I:/OODP12/First	

Refactor-Rename



Create a New Git Repository

Please determine the directory for the new repository

Parent directory: I:\git

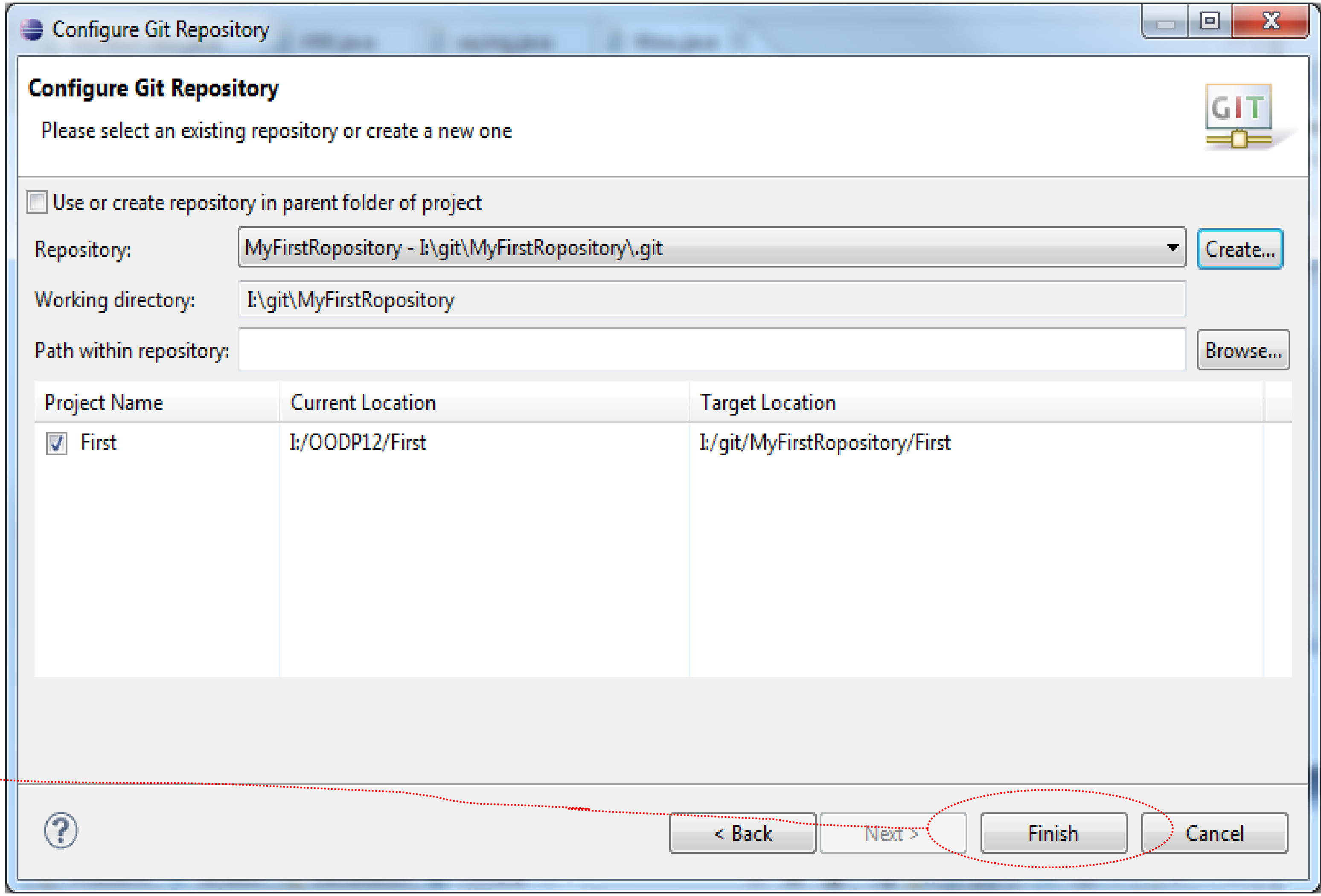
Name: MyFirstRopository

Repository directory

1. Fill in

2. Select

Configure git repository



Configure Git Repository

Configure Git Repository

Please select an existing repository or create a new one

☐ Use or create repository in parent folder of project

Repository: MyFirstRopository - I:\git\MyFirstRopository\.git

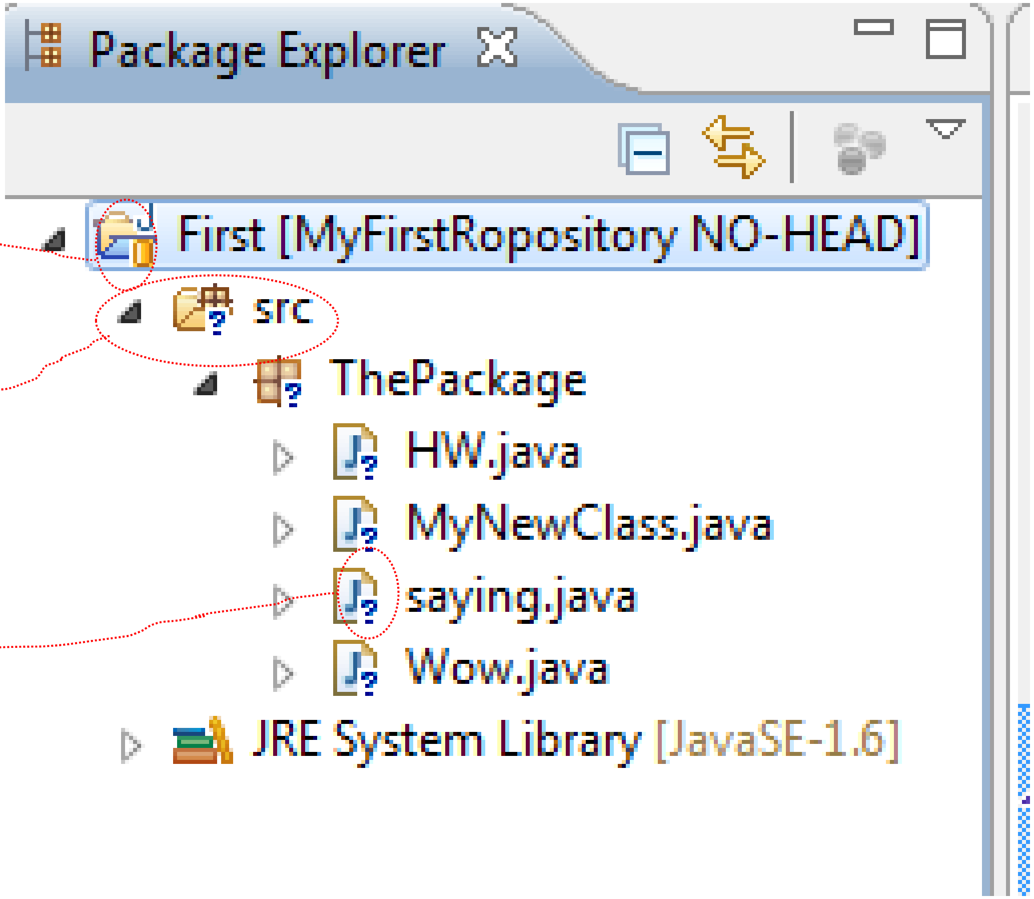
Working directory: I:\git\MyFirstRopository

Path within repository:

Project Name	Current Location	Target Location
☑ First	I:/OODP12/First	I:/git/MyFirstRopository/First

1. Select

Refactor-Rename



First [MyFirstRopository NO-HEAD]

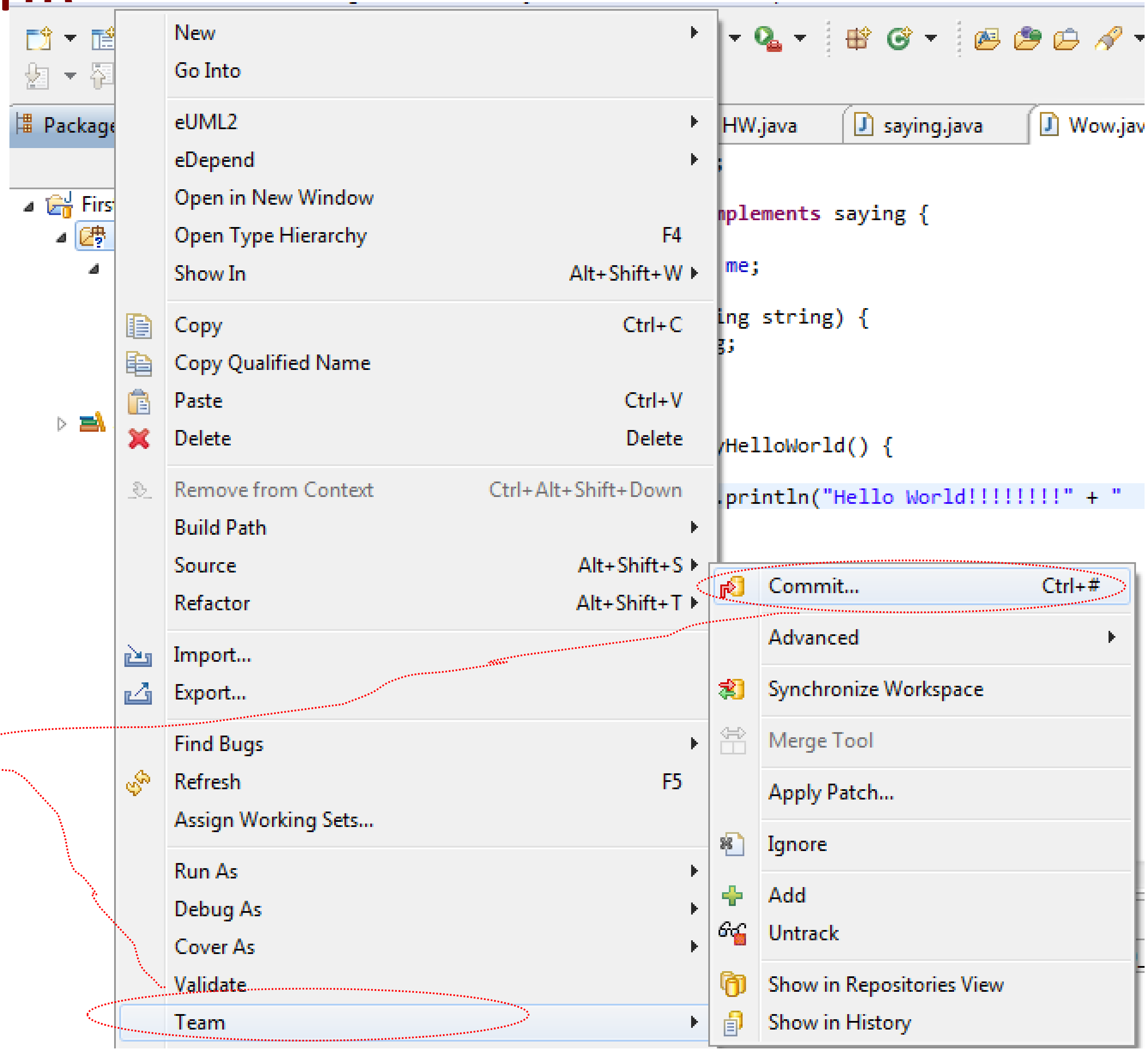
```
src
  ThePackage
    HW.java
    MyNewClass.java
    saying.java
    Wow.java
JRE System Library [JavaSE-1.6]
```

Project has repository

1. Right click

Repository does not know what to do about this

First Commit



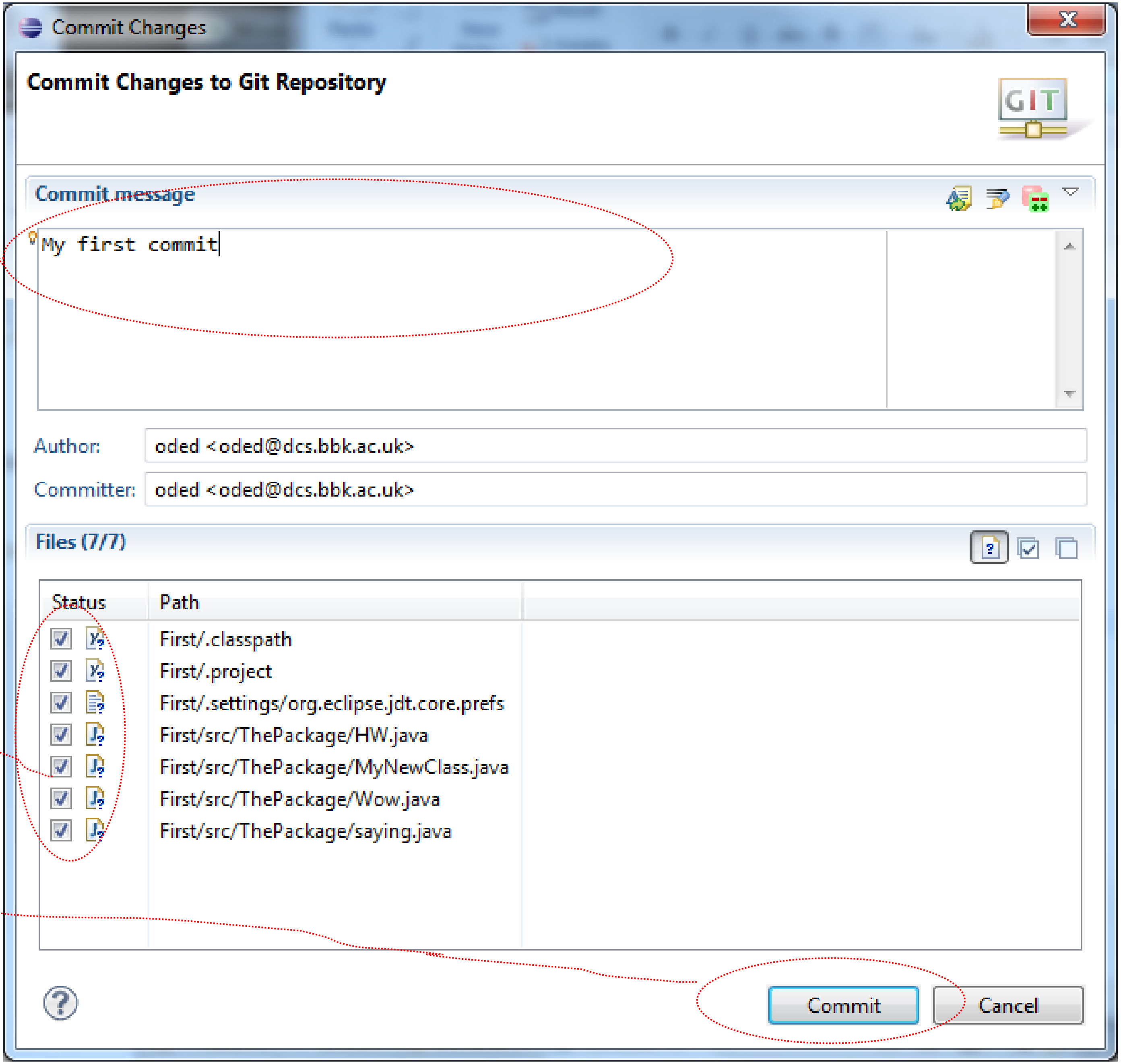
```
mplements saying {
 me;
ing string) {
g;
yHelloWorld() {
.println("Hello World!!!!!!!!" + "
```

1. select

1. write message

2. tick

3. select



Commit Changes to Git Repository

Commit message

```
My first commit
```

Author: oded <oded@dcs.bbk.ac.uk>

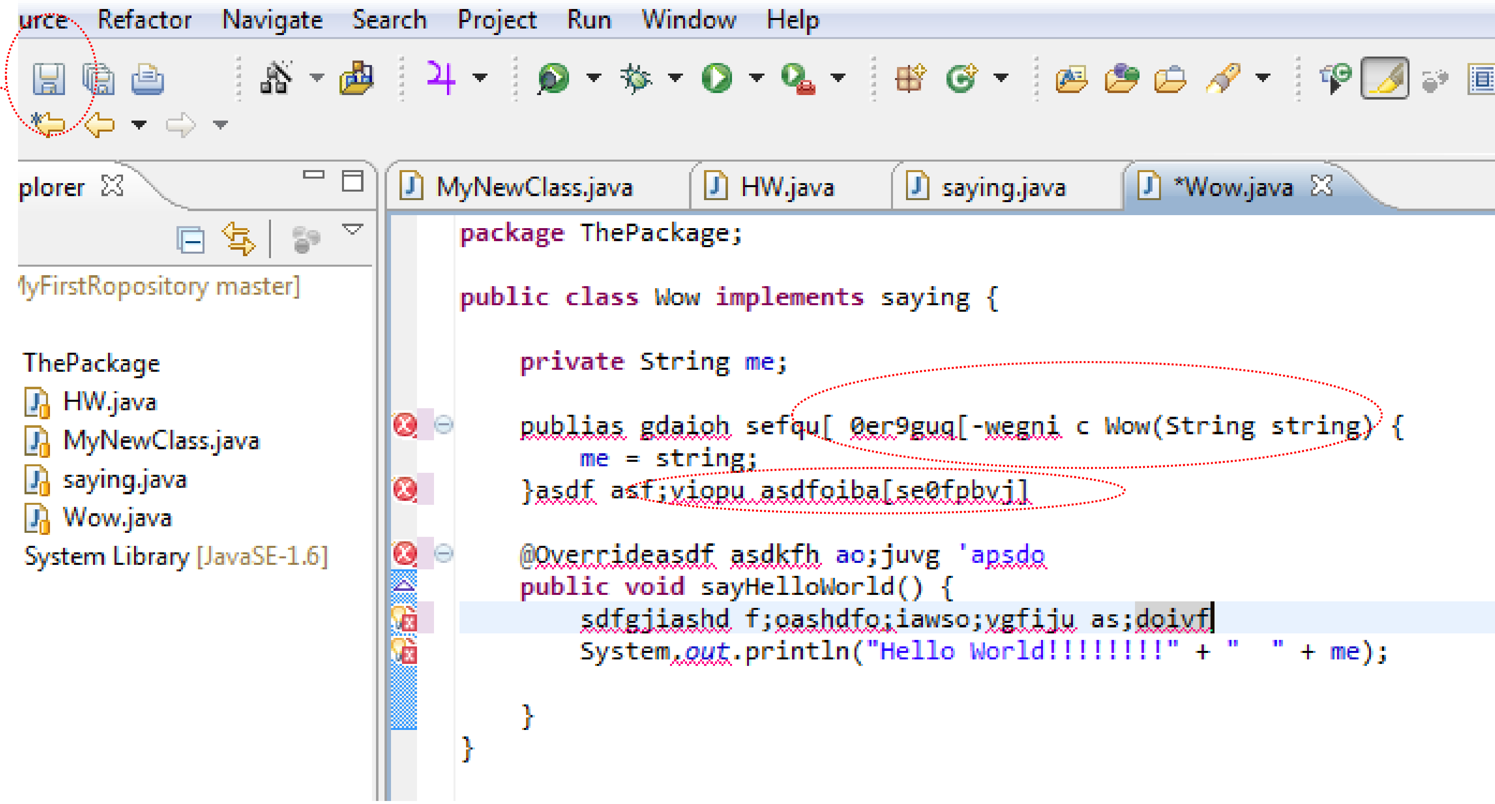
Committer: oded <oded@dcs.bbk.ac.uk>

Files (7/7)

Status	Path
☑	First/.classpath
☑	First/.project
☑	First/.settings/org.eclipse.jdt.core.prefs
☑	First/src/ThePackage/HW.java
☑	First/src/ThePackage/MyNewClass.java
☑	First/src/ThePackage/Wow.java
☑	First/src/ThePackage/saying.java

Trach code

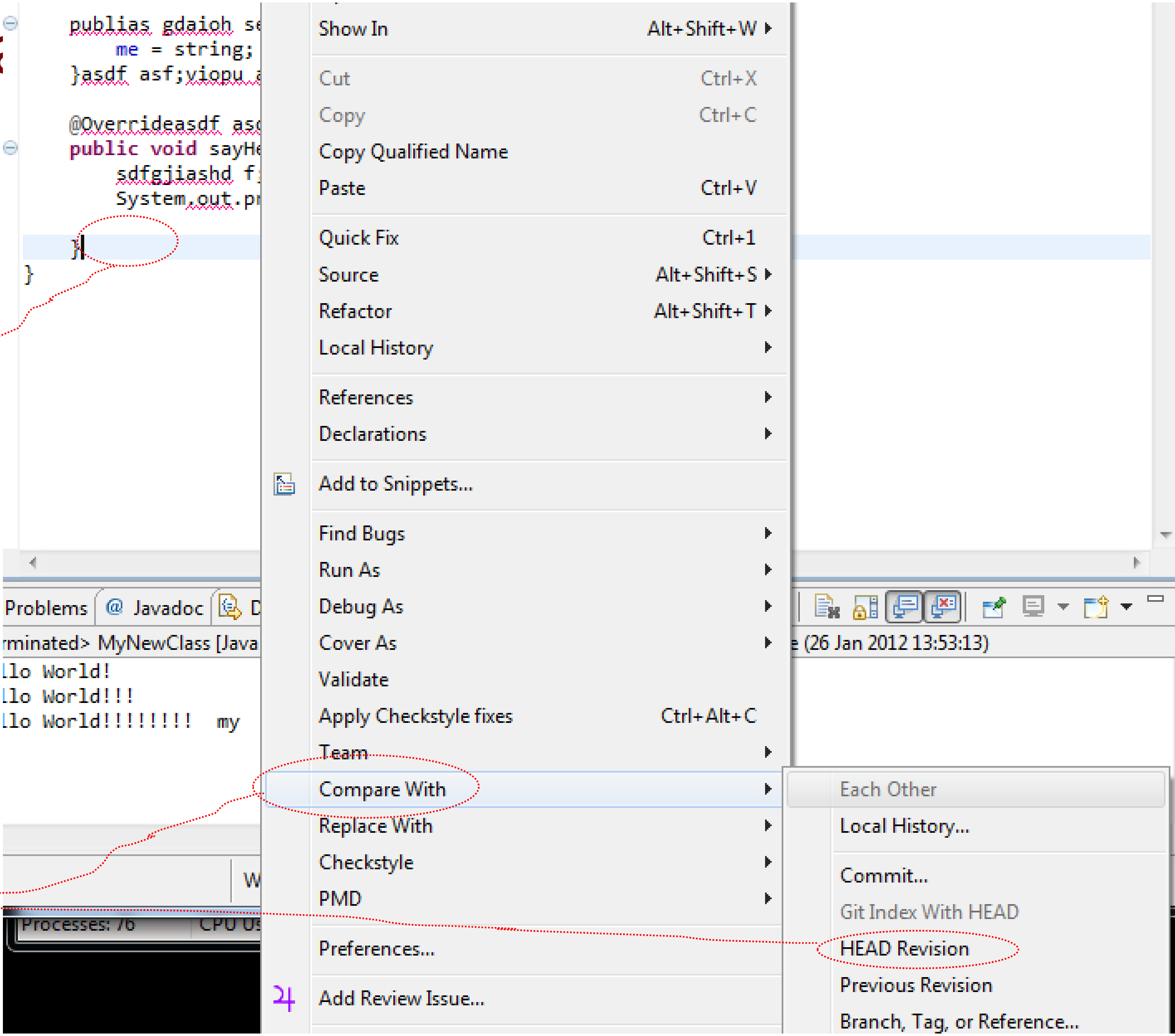
Save



```
package ThePackage;

public class Wow implements saying {

    private String me;

    publias gdaioh sefqu[ 0er9guq[-wegni c Wow(String string) {
        me = string;
    }asdf asf;viopu asdfoiba[se0fpbvj]

    @Overrideasdf asdkfh ao;juvg 'apsdo
    public void sayHelloWorld() {
        sdfgjiashd f;oashdfo;iawso;vgfiju as;doivf
        System.out.println("Hello World!!!!!!!!" + "   " + me);

    }
}
```

2. Press continue on anything that appears afterwards

Git Compa



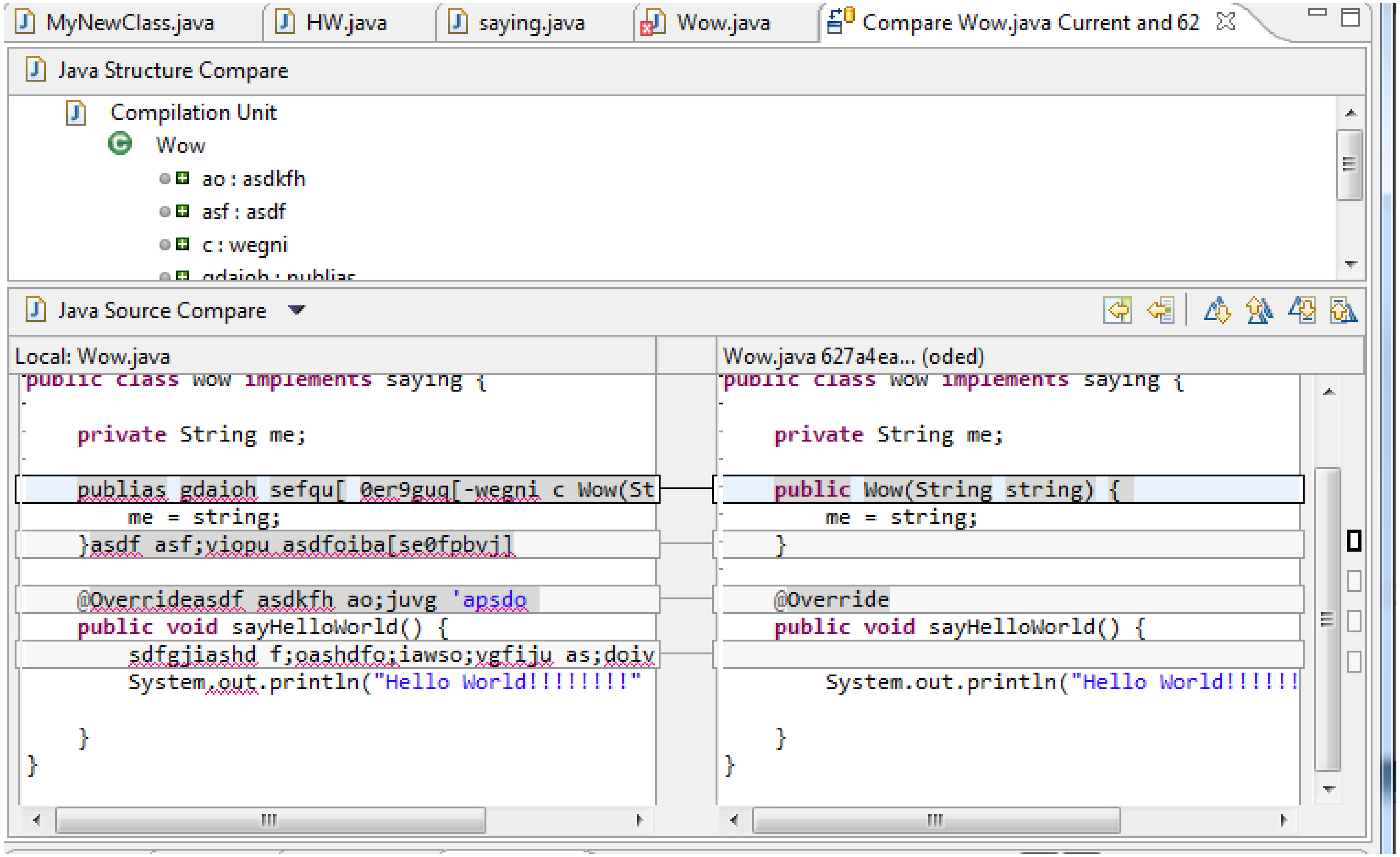
```
publias gdaioh se
    me = string;
}asdf asf;viopu a

@Overrideasdf asc
public void sayHe
    sdfgjiashd f;
    System.out.pr

    }
}
```

```
rminated> MyNewClass [Java                    e (26 Jan 2012 13:53:13)
llo World!
llo World!!!
llo World!!!!!!!!   my
```

1. Right click

2. Select

Look what we got



Compilation Unit
Wow
ao : asdkfh
asf : asdf
c : wegni
gdaioh : publias

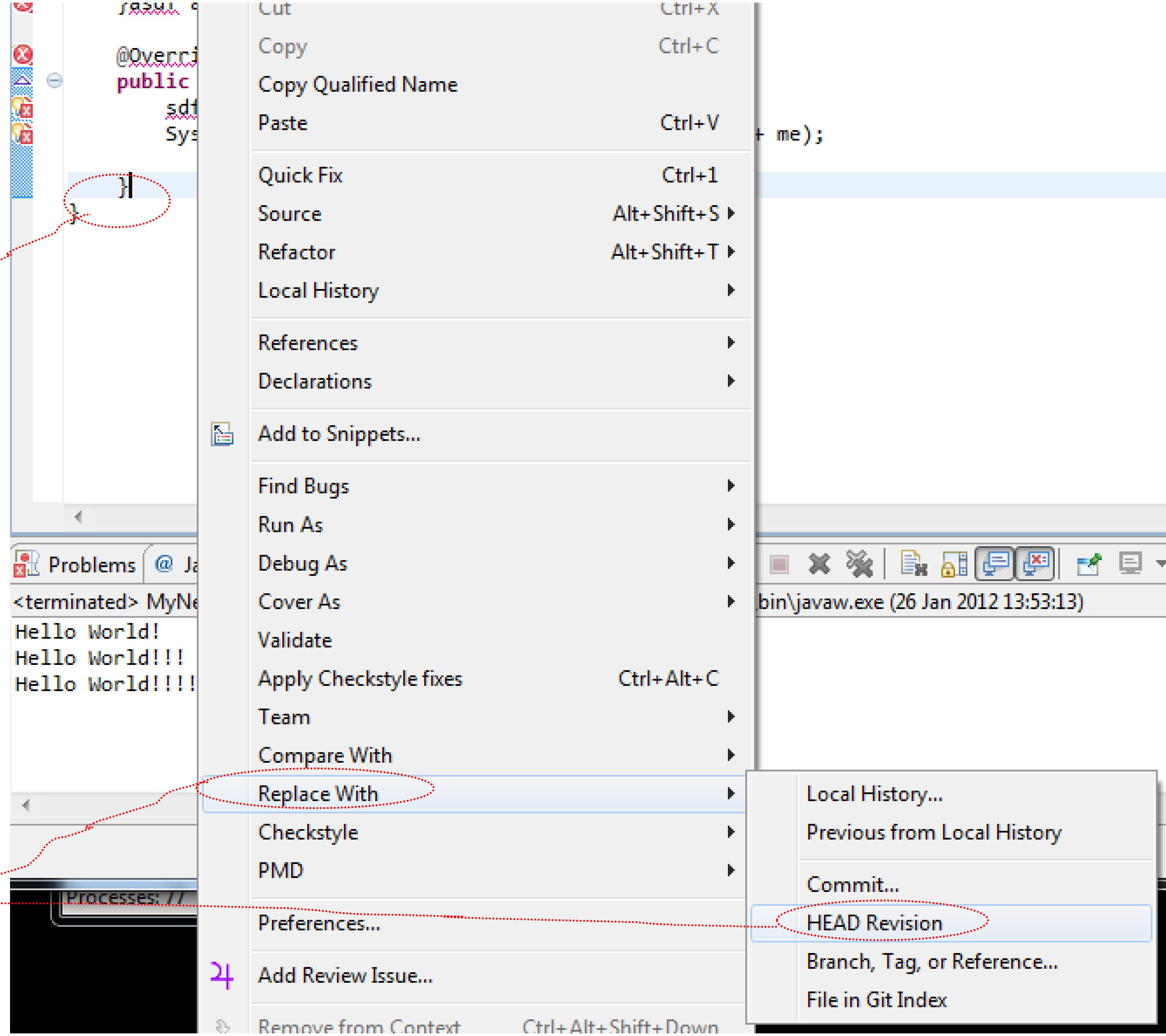
Local: Wow.java

```
public class Wow implements saying {

    private String me;

    publias gdaioh sefqu[ 0er9guq[-wegni c Wow(St
        me = string;
    }asdf asf;viopu asdfoiba[se0fpbvj]

    @Overrideasdf asdkfh ao;juvg 'apsdo
    public void sayHelloWorld() {
        sdfgjiashd f;oashdfo;iawso;vgfiju as;doiv
        System.out.println("Hello World!!!!!!!!"

    }
}
```

Wow.java 627a4ea... (oded)

```
public class Wow implements saying {

    private String me;

    public Wow(String string) {
        me = string;
    }

    @Override
    public void sayHelloWorld() {

        System.out.println("Hello World!!!!!!

    }
}
```

Git Compare



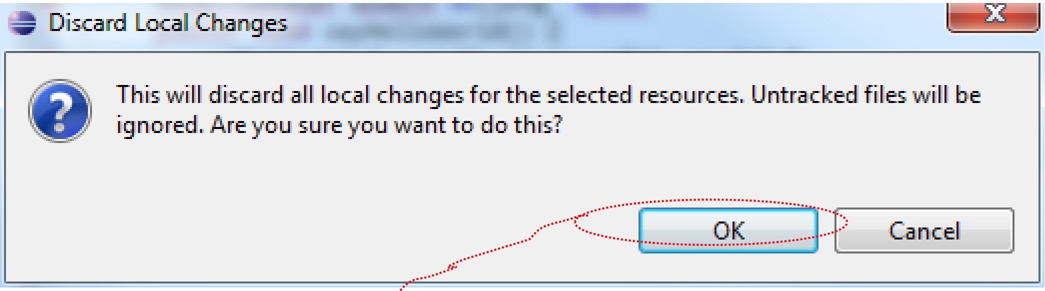
```
@Overri
public
      sdf
      Sys                                         + me);

    }
}
```

```
<terminated> MyNe                                bin\javaw.exe (26 Jan 2012 13:53:13)
Hello World!
Hello World!!!
Hello World!!!!
```

1. Right click

2. Select

Replace with



Discard Local Changes

This will discard all local changes for the selected resources. Untracked files will be ignored. Are you sure you want to do this?

1. Select

What Happened?