lab 3.1 & 3.2

lab 3.1

overview

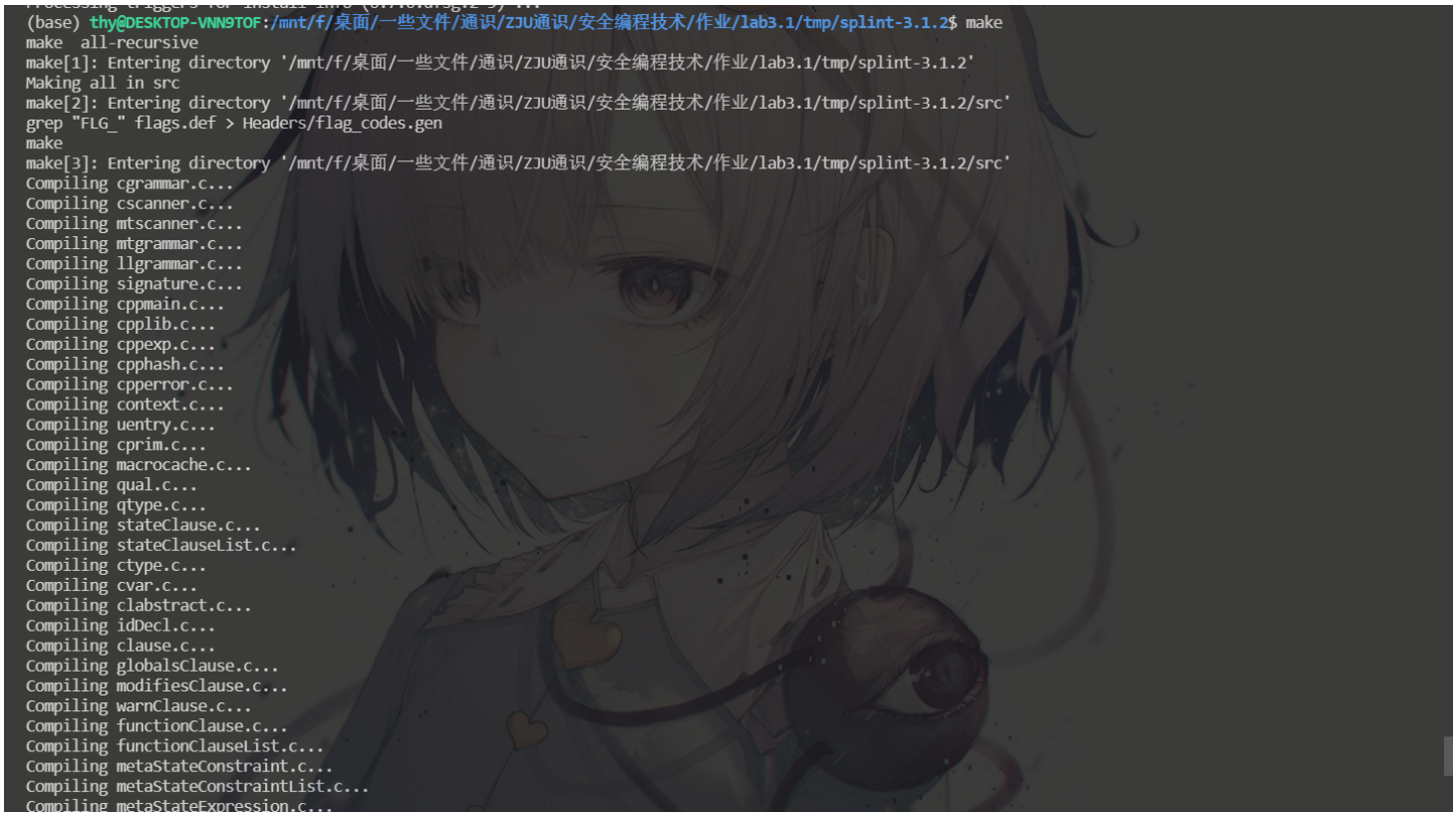
The learning objective of this lab is for students to gain the first-hand experience on using static code analysis tools to check c program for security vulnerabilities and coding mistakes.

your goal is to achieve the followings:

- Install splint;
- Finish code samples with 2 different kinds of problems which can be detected by Splint. You can choose any 2 of 11 problems as above.
- Use splint to detect the 2 kinds of problems. Descibe your observations in your report.

install splint

```
(base) thy@DESKTOP-VNN9TOF:/mnt/f/桌面/一些文件/通识/ZJU通识/安全编程技术/作业/lab3.1/tmp$ sudo mkdir /usr/local/splint
[sudo] password for thy:
(base) thy@DESKTOP-VNN9TOF:/mnt/f/桌面/一些文件/通识/ZJU通识/安全编程技术/作业/lab3.1/tmp$ cd splint-3.1.2
(base) thy@DESKTOP-VNN9TOF:/mnt/f/桌面/一些文件/通识/ZJU通识/安全编程技术/作业/lab3.1/tmp/splint-3.1.2$ ls
Makefile.am         Makefile.binary.in  README        aclocal.m4  Changlog      configure     configure.binary     doc                 import        lib  test
Makefile.binary.am  Makefile.in         acinclude.m4  bin         config.hin    configure.ac  configure.binary.ac  fixBinaryDist.sh  install.html  src
(base) thy@DESKTOP-VNN9TOF:/mnt/f/桌面/一些文件/通识/ZJU通识/安全编程技术/作业/lab3.1/tmp/splint-3.1.2$ ./configure --prefix=/usr/local/splint
checking build system type... x86_64-unknown-linux-gnu
checking host system type... x86_64-unknown-linux-gnu
checking target system type... x86_64-unknown-linux-gnu
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for gawk... gawk
checking whether make sets $(MAKE)... yes
checking for gcc... gcc
checking for C compiler default output file name... a.out
checking whether the C compiler works... yes
checking whether we are cross compiling... no
checking for suffix of executables...
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
checking whether gcc accepts -g... yes
checking for gcc option to accept ANSI C... none needed
checking for style of include used by make... GNU
checking dependency style of gcc... gcc3
checking how to run the C preprocessor... gcc -E
checking for flex... no
checking for lex... no
checking for yywrap in -lfl... no
checking for yywrap in -ll... no
checking for a BSD-compatible install... /usr/bin/install -c
checking whether make sets $(MAKE)... (cached) yes
checking whether ln -s works... yes
checking for bison... no
checking for grep... grep
checking for diff... diff
checking for cat... cat
checking for rm... rm
checking for mv... mv
checking for cp... cp
checking for sed... sed
checking whether we need _ALL_SOURCE to expose mode_t... no
checking whether to include support for LCL files... yes
```



Finish code samples with 2 different kinds of problems

Problem code 1

使用如下代码进行漏洞分析

```
#include<stdio.h>
#include<string.h>

int main() {
        char test[100];
        scanf("%s", test);
        printf(test);
        return 1;
}
```

Problem code 2

使用如下代码进行漏洞分析

```c
#include<stdio.h>

int main() {
    int a = 0;
    int b = 1;
    if(a = b) {
        printf("wrong");
    }
}
```

Use splint to detect the 2 kinds of problems.

对于test1.c,分析得到如下结果,可以看到,splint检测到了两个错误,一个是程序没有注意scanf函数的返回值,另一个是典型的格式化字符串漏洞

```
(base) thy@DESKTOP-VNN9TOF:/mnt/f/桌面/一些文件/通识/ZJU通识/安全编程技术/作业/lab3.1/code$ splint test1.c
Splint 3.1.2 --- 20 Feb 2018

/mnt/f/\377\377\377\377\377\377/\377\377\377\377\377\377\377\377\377\377\377\377/\377\377\377\377\377\377/ZJU\377\377\377\377\377\377/\377\377\377\377\377\
377\377\377\377\377\377\377\377\377\377\377\377\377/\377\377\377\377\377\377/lab3.1/code/test1.c: (in function main)
/mnt/f/\377\377\377\377\377\377/\377\377\377\377\377\377\377\377\377\377\377\377
    /\377\377\377\377\377\377/ZJU\377\377\377\377\377\377/\377\377\377\377\377\377\3
    77\377\377\377\377\377\377\377\377\377\377\377\377/\377\377\377\377\377\377/
    lab3.1/code/test1.c:6:2: Return value (type int) ignored: scanf("%s", test)
  Result returned by function call is not used. If this is intended, can cast
  result to (void) to eliminate message. (Use -retvalint to inhibit warning)
/mnt/f/\377\377\377\377\377\377/\377\377\377\377\377\377\377\377\377\377\377\377
    /\377\377\377\377\377\377/ZJU\377\377\377\377\377\377/\377\377\377\377\377\377\3
    77\377\377\377\377\377\377\377\377\377\377\377\377/\377\377\377\377\377\377/
    lab3.1/code/test1.c:7:2: Format string parameter to printf is not a
    compile-time constant: test
  Format parameter is not known at compile-time.  This can lead to security
  vulnerabilities because the arguments cannot be type checked. (Use
  -formatconst to inhibit warning)

Finished checking --- 2 code warnings
(base) thy@DESKTOP-VNN9TOF:/mnt/f/桌面/一些文件/通识/ZJU通识/安全编程技术/作业/lab3.1/code$
```

对于test2.c,分析得到如下结果,splint检测到了三个错误,一个是使用了if(a = b),一个是if()内的表达式既不是Boolean也不是int,还有一个是main函数没有写返回值

```
(base) thy@DESKTOP-VNN9TOF:/mnt/f/桌面/一些文件/通识/ZJU通识/安全编程技术/作业/lab3.1/code$ splint test2.c
Splint 3.1.2 --- 20 Feb 2018

/mnt/f/\377\377\377\377\377\377/\377\377\377\377\377\377\377\377\377\377\377\377/\377\377\377\377\377\377/ZJU\377\377\377\377\377\377/\377\377\377\377\377\
377\377\377\377\377\377\377\377\377\377\377\377\377/\377\377\377\377\377\377/lab3.1/code/test2.c: (in function main)
/mnt/f/\377\377\377\377\377\377/\377\377\377\377\377\377\377\377\377\377\377\377
    /\377\377\377\377\377\377/ZJU\377\377\377\377\377\377/\377\377\377\377\377\377\3
    77\377\377\377\377\377\377\377\377\377\377\377\377/\377\377\377\377\377\377/
    lab3.1/code/test2.c:6:8: Test expression for if is assignment expression:
    a = b
  The condition test is an assignment expression. Probably, you mean to use ==
  instead of =. If an assignment is intended, add an extra parentheses nesting
  (e.g., if ((a = b)) ...) to suppress this message. (Use -predassign to
  inhibit warning)
/mnt/f/\377\377\377\377\377\377/\377\377\377\377\377\377\377\377\377\377\377\377
    /\377\377\377\377\377\377/ZJU\377\377\377\377\377\377/\377\377\377\377\377\377\3
    77\377\377\377\377\377\377\377\377\377\377\377\377/\377\377\377\377\377\377/
    lab3.1/code/test2.c:6:8: Test expression for if not boolean, type int:
    a = b
  Test expression type is not boolean or int. (Use -predboolint to inhibit
  warning)
/mnt/f/\377\377\377\377\377\377/\377\377\377\377\377\377\377\377\377\377\377\377
    /\377\377\377\377\377\377/ZJU\377\377\377\377\377\377/\377\377\377\377\377\377\3
    77\377\377\377\377\377\377\377\377\377\377\377\377/\377\377\377\377\377\377/
    lab3.1/code/test2.c:9:2: Path with no return in function declared to return
    int
  There is a path through a function declared to return a value on which there
  is no return statement. This means the execution may fall through without
  returning a meaningful result to the caller. (Use -noret to inhibit warning)

Finished checking --- 3 code warnings
(base) thy@DESKTOP-VNN9TOF:/mnt/f/桌面/一些文件/通识/ZJU通识/安全编程技术/作业/lab3.1/code$
```

lab 3.2

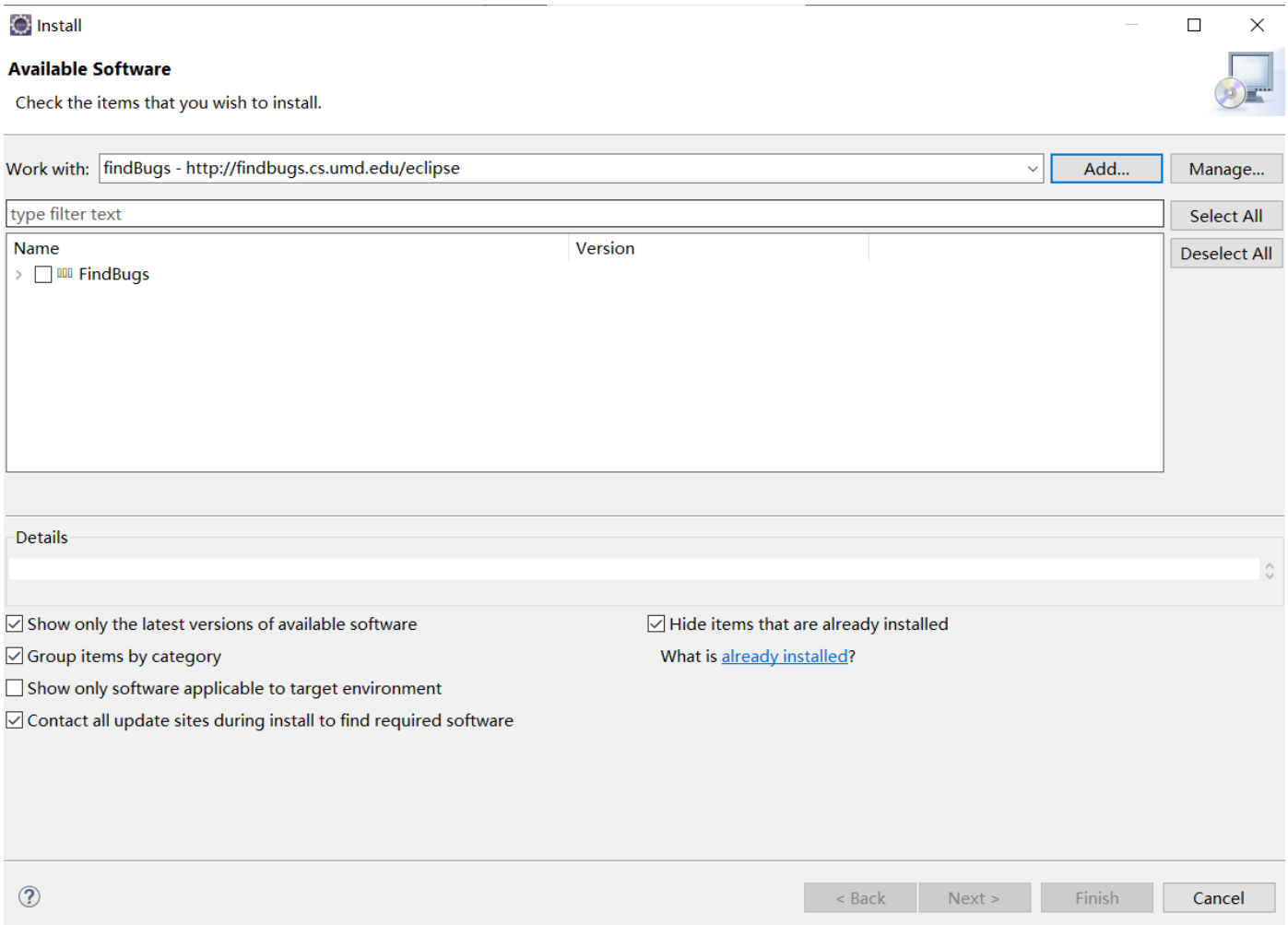
overview

In this Lab, your goal is to achieve the followings:

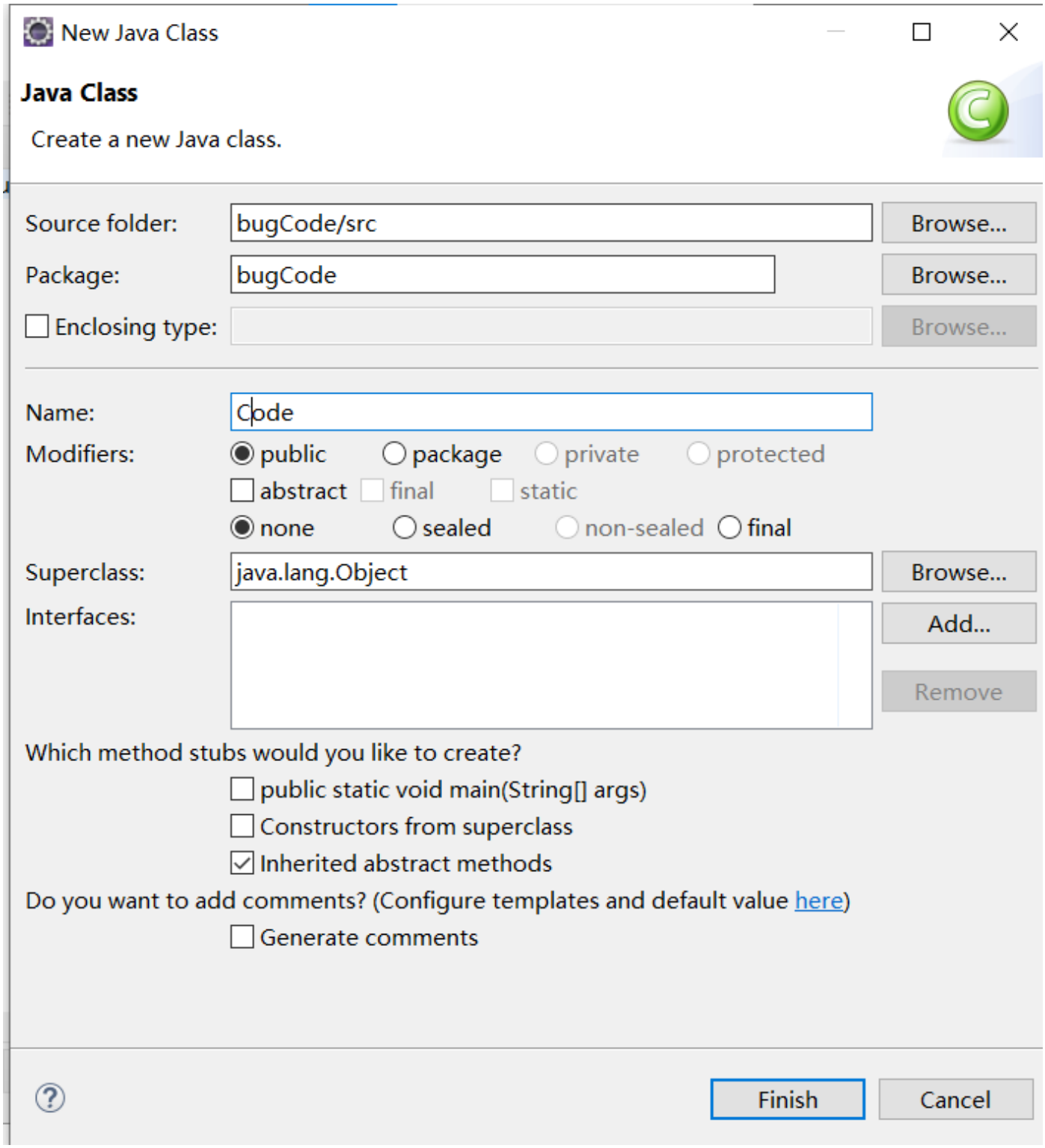
- Install plugins in Java;
- Learn to check Java code by using static code analyzers in - - Eclipse. Descibe your observations in your report.

steps

1. 安装findBugs 插件



2. 创建Java Project

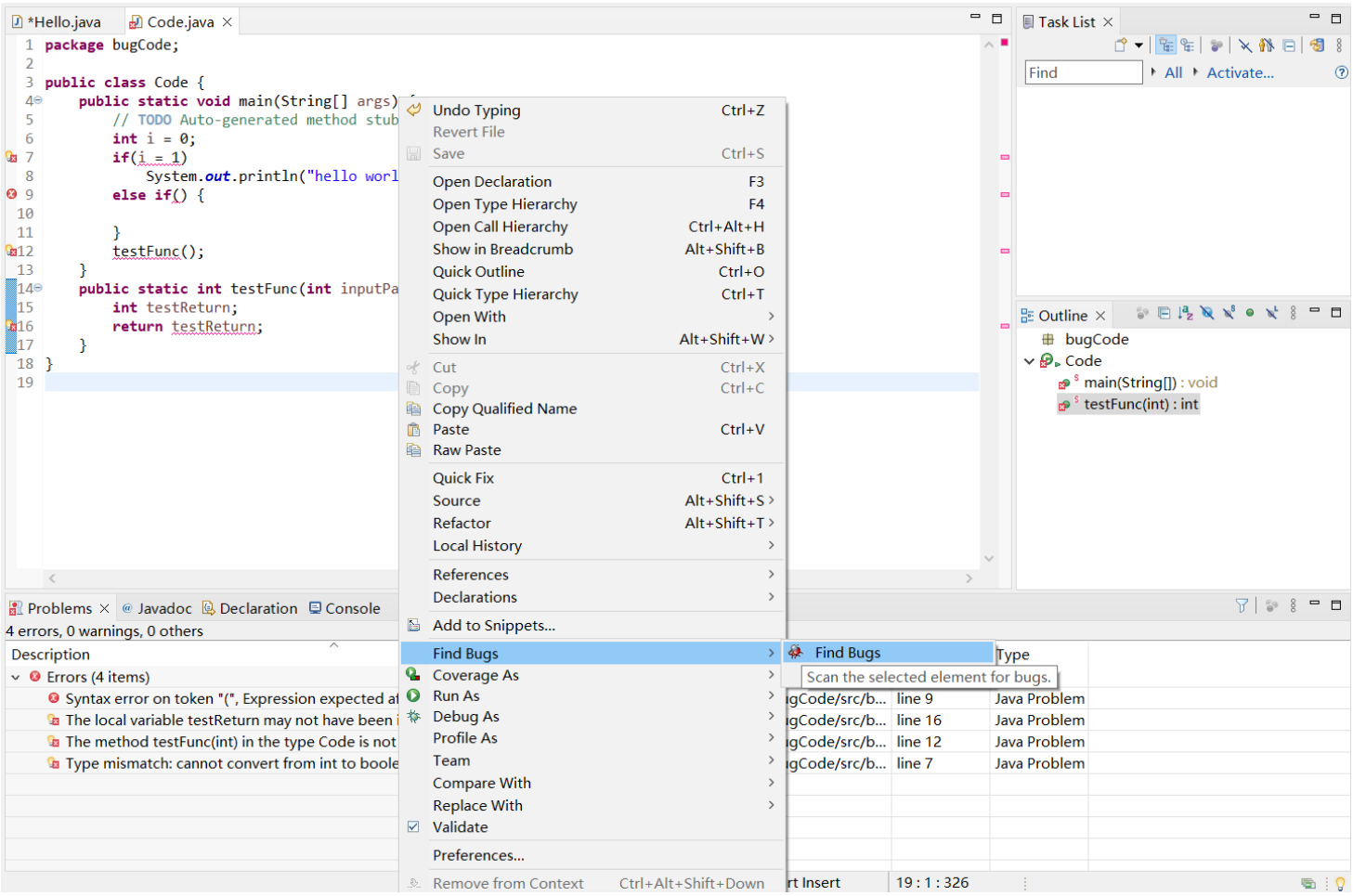


3. 输入以下代码，右键Find Bugs寻找漏洞

```java
package bugCode;

public class Code {
	public static void main(String[] args) {
		// TODO Auto-generated method stub
		int i = 0;
		if(i = 1)
			System.out.println("hello world");
		else if() {

		}
		testFunc();
	}
	public static int testFunc(int inputPara) {
		int testReturn;
		return testReturn;
	}
}
```



可以看到,源代码中的漏洞:

- `if(i = 1)` ,

- 使用函数时未传入参数,
- 函数返回一个未定义的变量,
- if块中没有语句等

  都被找了出来

```java
1  package bugCode;
2
3  public class Code {
4      public static void main(String[] args) {
5          // TODO Auto-generated method stub
6          int i = 0;
7          if(i = 1)
8              System.out.println("hello world");
9          else if() {
10
11         }
12         testFunc();
13     }
14     public static int testFunc(int inputPara) {
15         int testReturn;
16         return testReturn;
17     }
18 }
19
```

Outline

- bugCode
- Code
  - main(String[]) : void
  - testFunc(int) : int

Problems | Javadoc | Declaration | Console

4 errors, 0 warnings, 0 others

Description	Resource	Path	Location	Type
Errors (4 items)				
Syntax error on token "(", Expression expected after this token	Code.java	/bugCode/src/b...	line 9	Java Problem
The local variable testReturn may not have been initialized	Code.java	/bugCode/src/b...	line 16	Java Problem
The method testFunc(int) in the type Code is not applicable for the arguments ()	Code.java	/bugCode/src/b...	line 12	Java Problem
Type mismatch: cannot convert from int to boolean	Code.java	/bugCode/src/b...	line 7	Java Problem