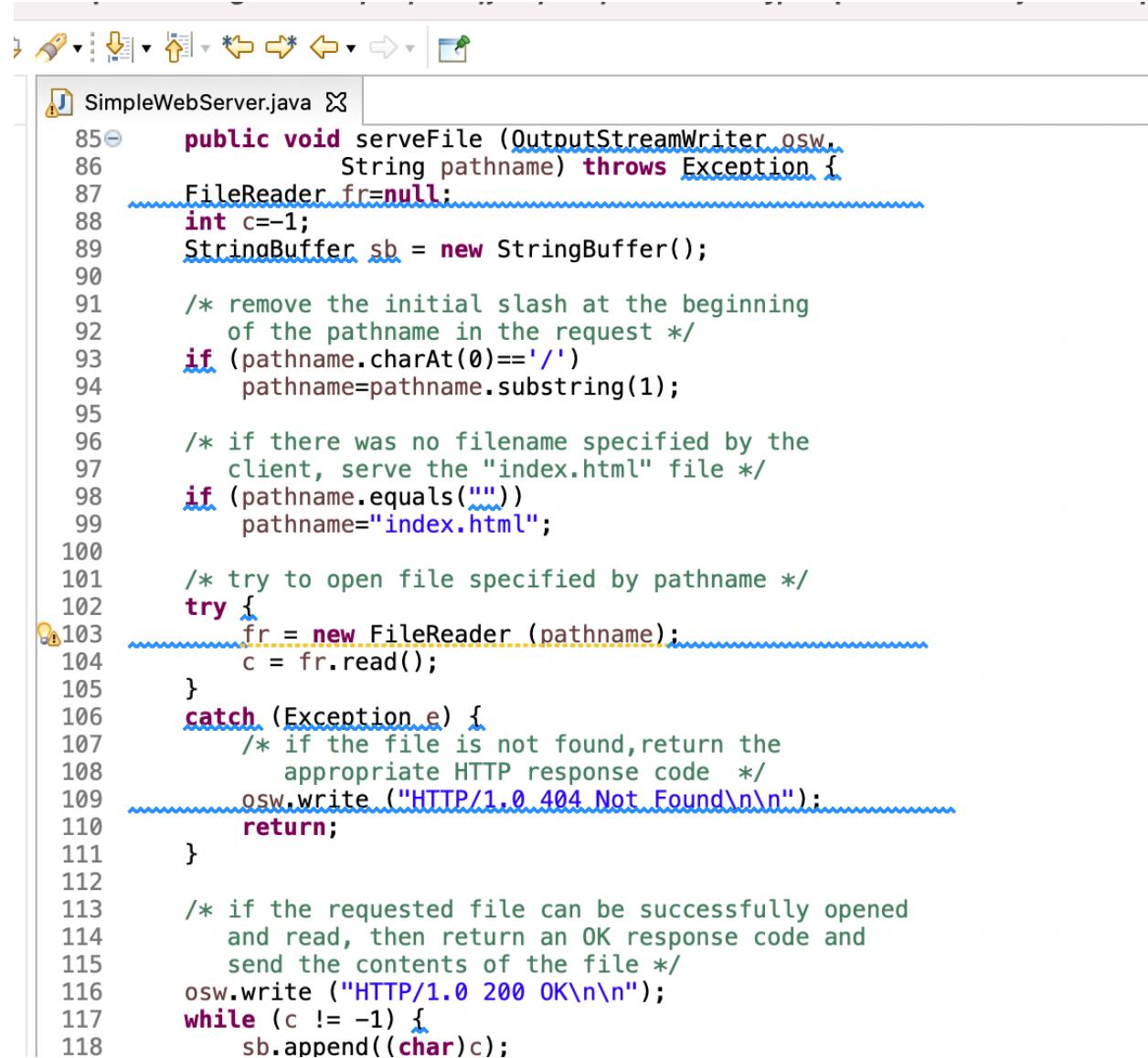


## **Assignment 10 : Understanding and Using Static Code Analysis Tools**

**Submitted By: Begum Fatima Zohra  
ID: 1001880881**

## Manual Analysis:

Line 103 : File Reader fr is not closed



The screenshot shows the Eclipse IDE interface with the Java file "SimpleWebServer.java" open. The code is annotated with several underlines and highlights, indicating potential issues. A yellow circle with a question mark icon is placed over line 103, which is highlighted in orange. The code itself is as follows:

```
85     public void serveFile (OutputStreamWriter osw,
86                           String pathname) throws Exception {
87     FileReader fr=null;
88     int c=-1;
89     StringBuffer sb = new StringBuffer();
90
91     /* remove the initial slash at the beginning
92      of the pathname in the request */
93     if (pathname.charAt(0)=='/')
94         pathname=pathname.substring(1);
95
96     /* if there was no filename specified by the
97      client, serve the "index.html" file */
98     if (pathname.equals(""))
99         pathname="index.html";
100
101    /* try to open file specified by pathname */
102    try {
103        fr = new FileReader (pathname);
104        c = fr.read();
105    }
106    catch (Exception e) {
107        /* if the file is not found, return the
108           appropriate HTTP response code */
109        osw.write ("HTTP/1.0 404 Not Found\n\n");
110        return;
111    }
112
113    /* if the requested file can be successfully opened
114       and read, then return an OK response code and
115       send the contents of the file */
116    osw.write ("HTTP/1.0 200 OK\n\n");
117    while (c != -1) {
118        sb.append((char)c);
```

## Tools Choices:

1. Eclipse IDE

Version: 2021-03 (4.19.0)

The Eclipse IDE is downloaded and used in MacBook.

2. SpotBugs Eclipse Plugin

Version: 3.1.5

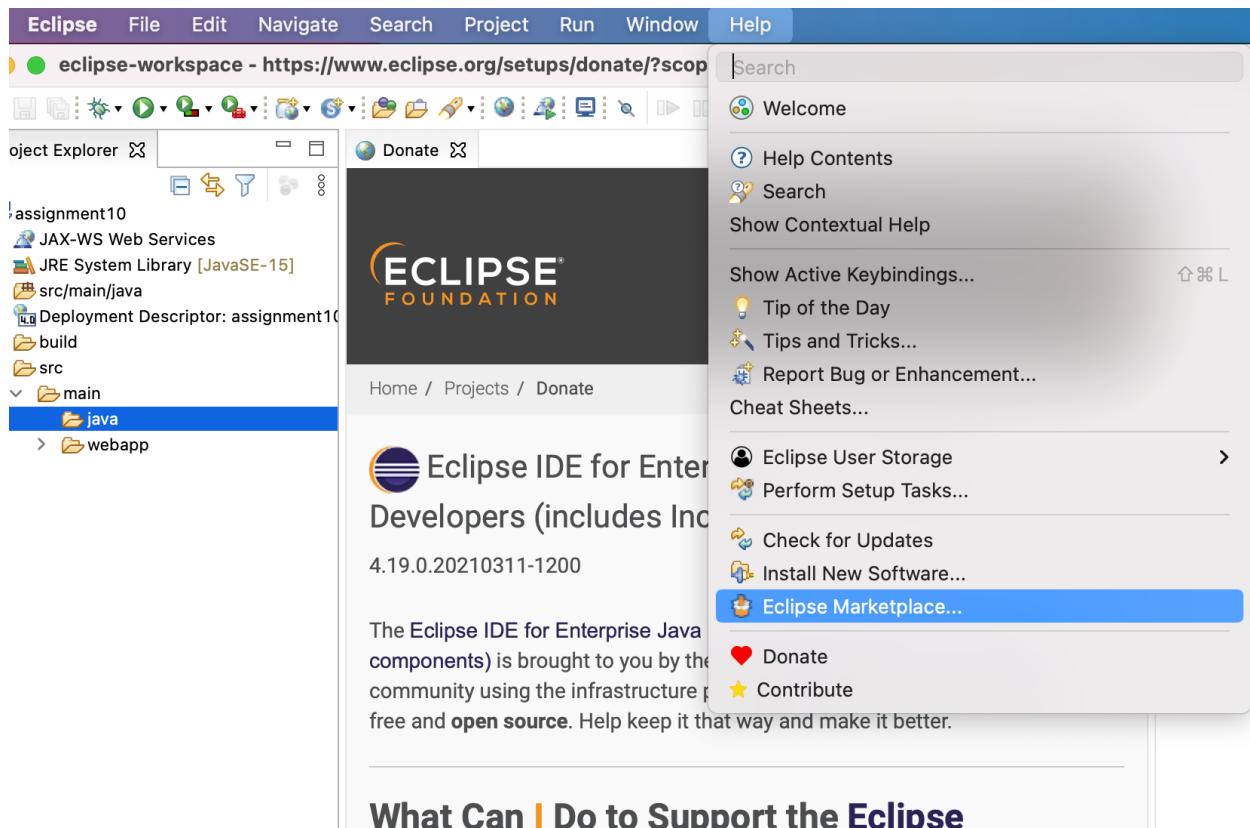
3. SonarLint

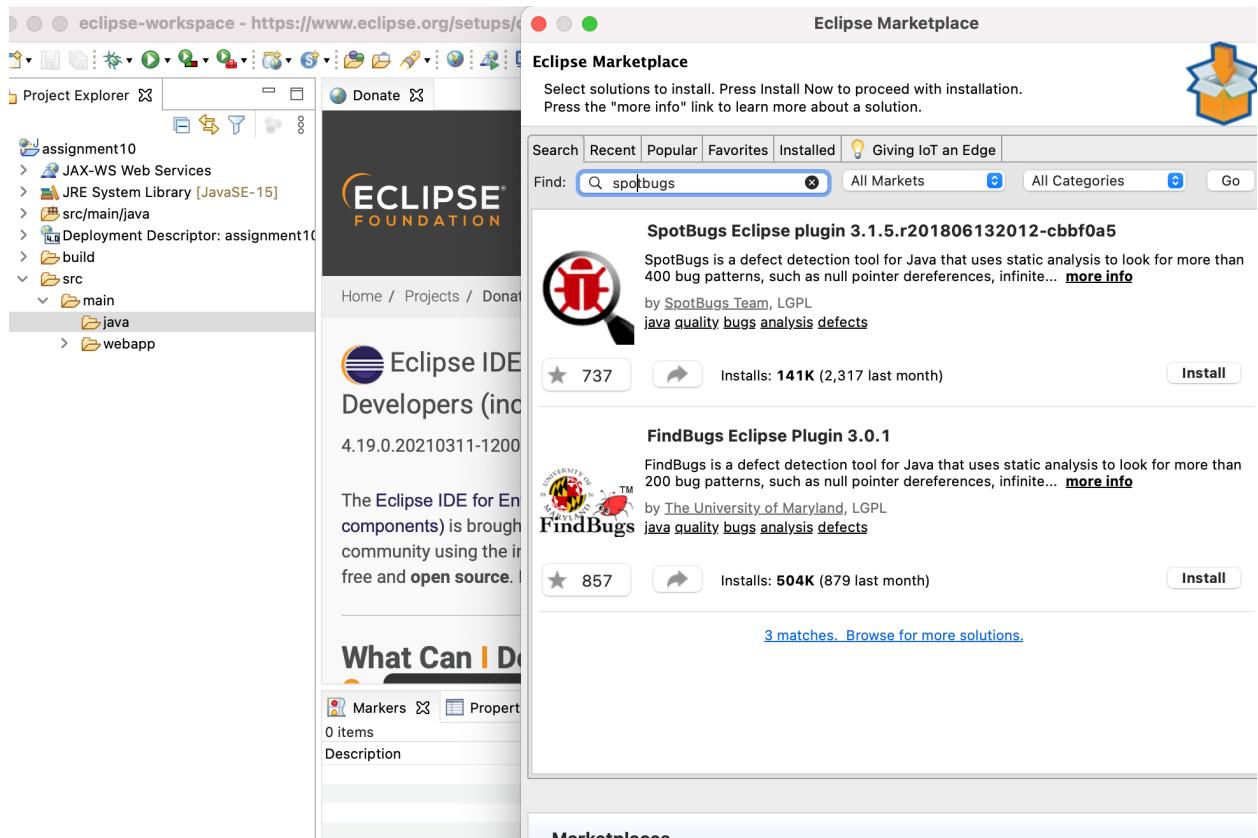
Version: 5.8.1

## Installation of SpotBugs:

Eclipse IDE -> Help -> Eclipse marketplace

In the Find section of the marketplace, type SpotBugs -> Install the recent version of SpotBugs plugin

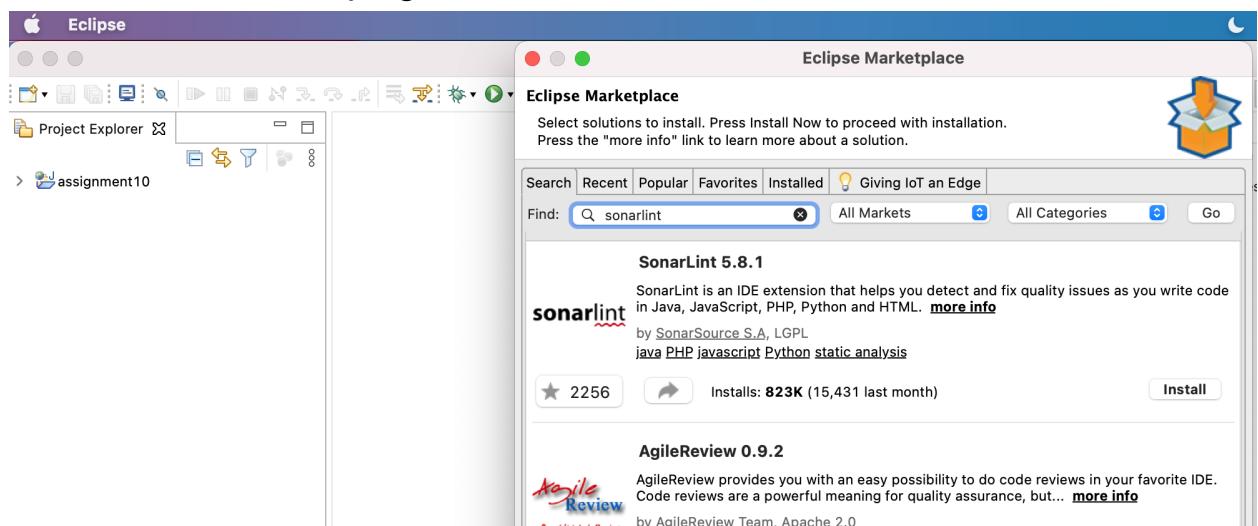




## Installation of SpotBugs:

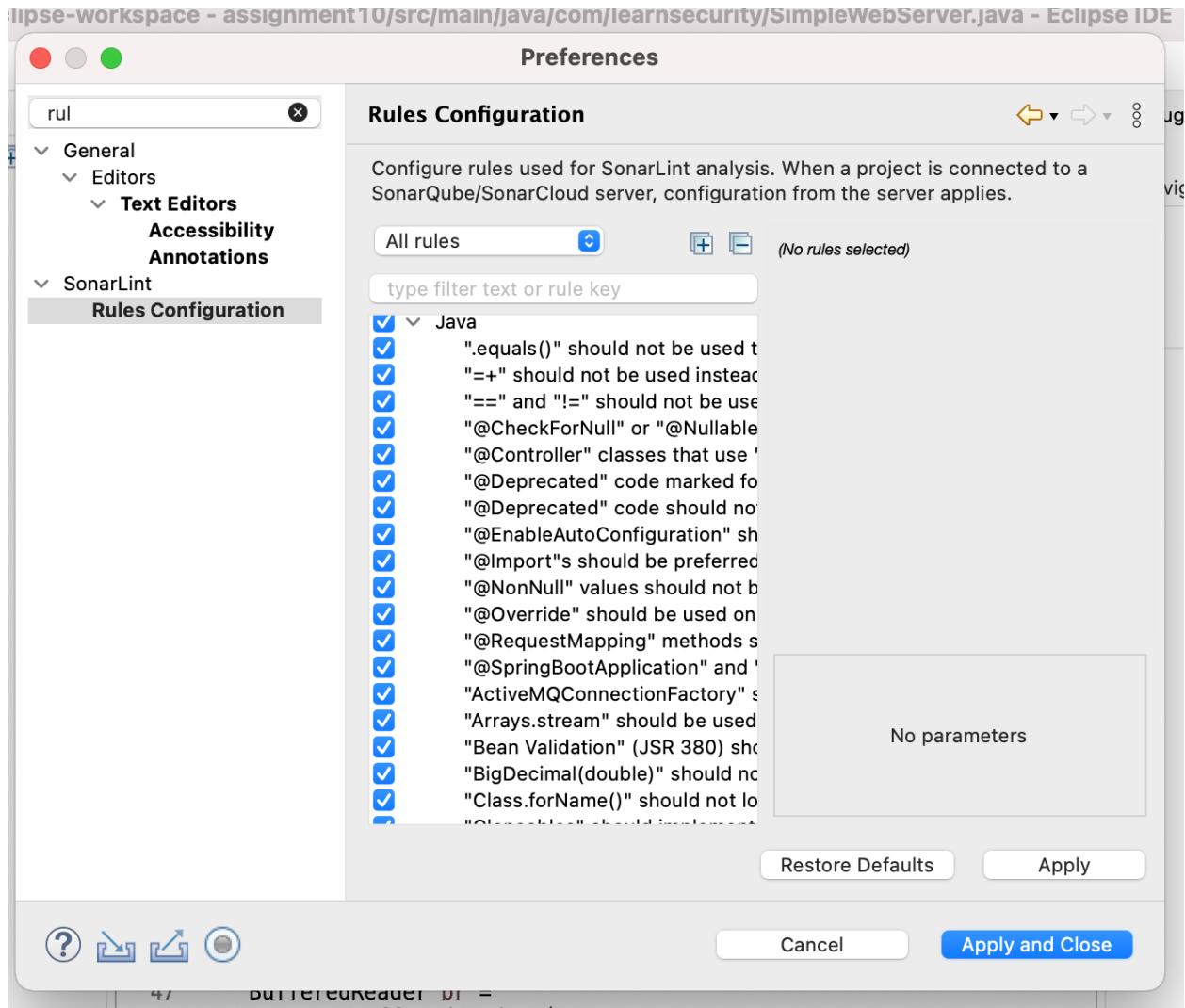
Eclipse IDE -> Help -> Eclipse marketplace

In the Find section of the marketplace, type SonarLint -> Install the recent version of SonarLint plugin



## Enable Security Threats in SonarLint:

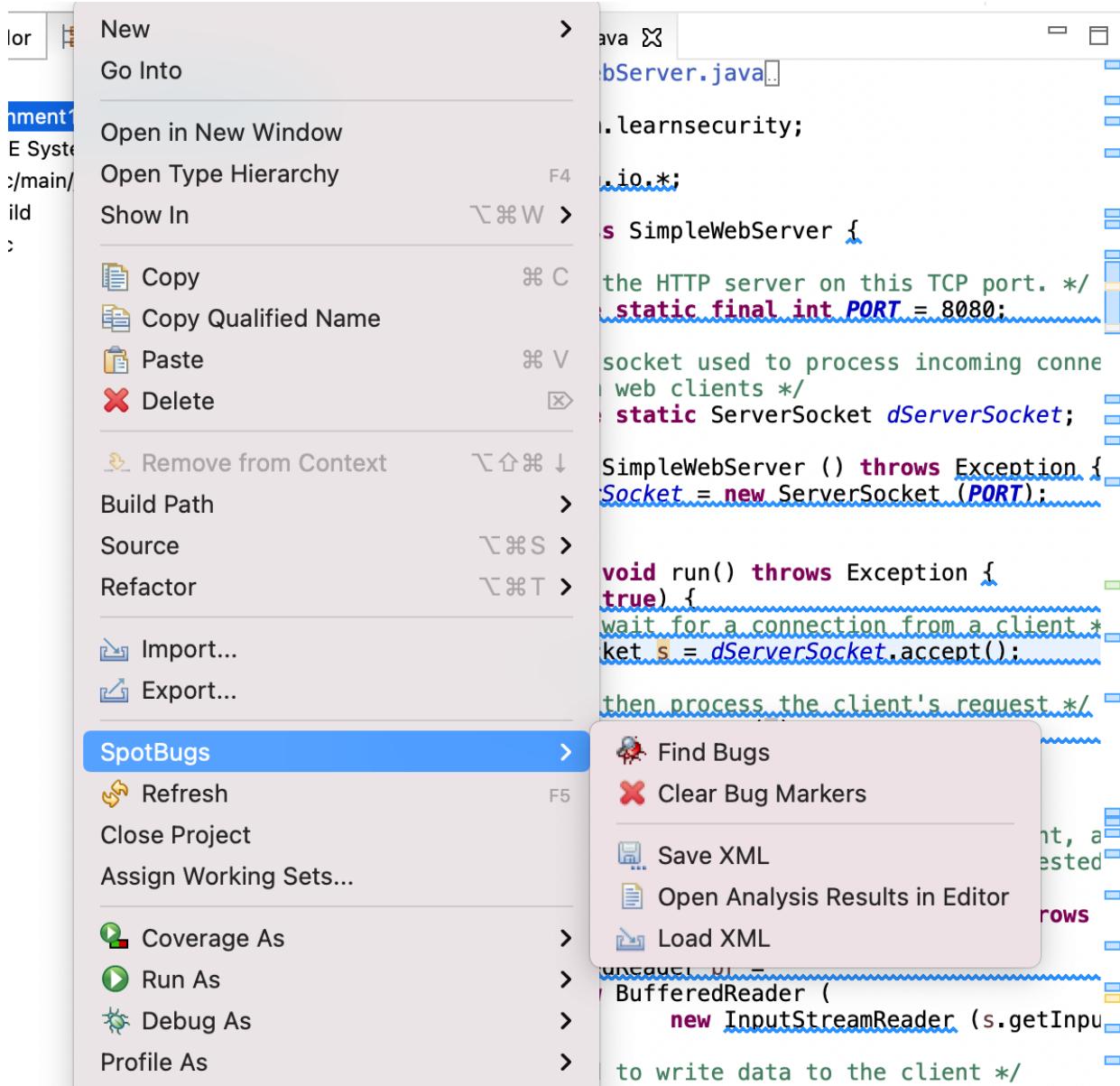
Eclipse -> Preferences -> SonarLint -> Rules Configuration -> Select Java



## XML report using SpotBugs:

Preferences -> SpotBugs -> Check all bug categories

Then, right click on project (assignment10) -> save XML



## Analysis using SpotBugs:

### Case 1 (Default):

Minimum confidence to report: Medium

Minimum rank to report: 15 (Of Concern)

Reported visible bugs categories:

Bad practice

Correctness

Experimental

Internationalization

Malicious code vulnerability

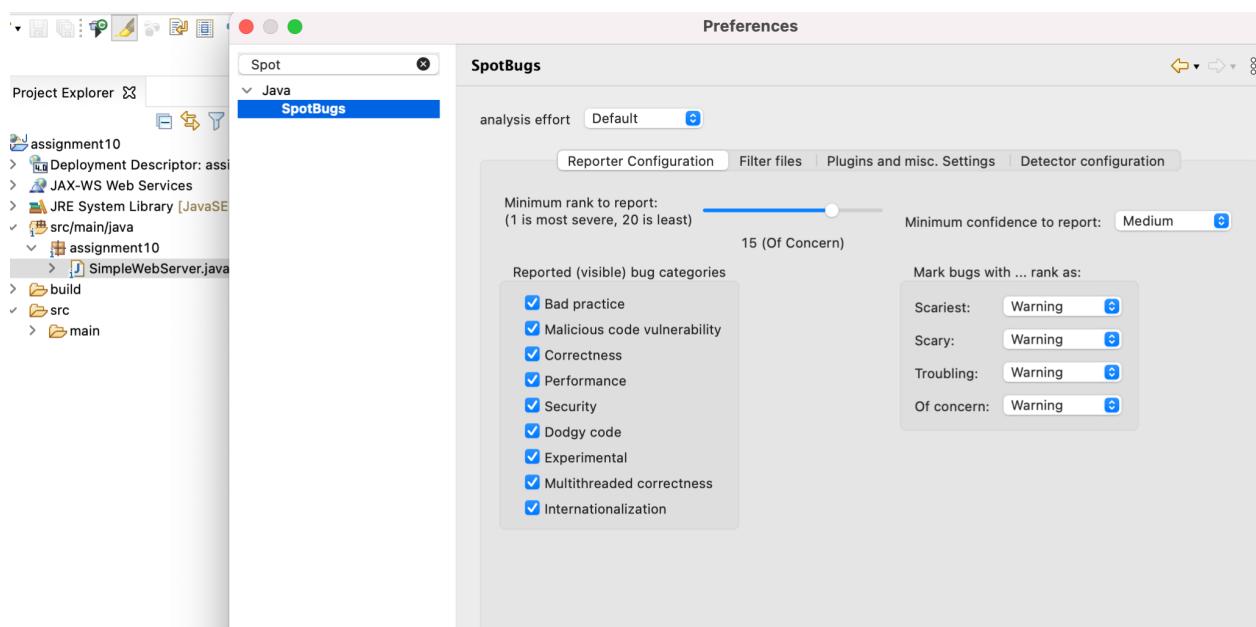
Multithreaded correctness

Performance

Security

Dodgy code

No. Of Bugs: 2



**eclipse-workspace - assignment10/src/main/java/com/learnsecurity/SimpleWebServer.java - Eclipse IDE**

**Bug Explore** | **Package Ex** | **File** | **Edit** | **Search** | **View** | **Project** | **Run** | **Help**

**SimpleWebServer.java** | **SimpleWebServer.java**

30 package com.learnsecurity;

12 import java.io.\*;

13 public class SimpleWebServer {

20 /\* Run the HTTP server on this TCP port \*/

21 private static final int PORT = 8080;

24 /\* The socket used to process incoming from web clients \*/

25 private static ServerSocket dServerSocket;

28 public SimpleWebServer () throws Exception {

29 dServerSocket = new ServerSocket (PORT);

32 public void run() throws Exception {

33 while (true) {

34 /\* wait for a connection from a client \*/

35 Socket s = dServerSocket.accept();

37 /\* then process the client's request \*/

38 processRequest(s);

41 /\* Reads the HTTP request from the client \*/

42 responds with the file the user requested a HTTP error code. \*/

43 public void processRequest(Socket s) {

44 /\* used to read data from the client \*/

45 BufferedReader br =

46 new BufferedReader (

47 new InputStreamReader (s.getInputStream());

50 /\* used to write data to the client \*/

52 OutputStreamWriter osw =

53 new OutputStreamWriter (s.getOutputStream());

55 /\* read the HTTP request from the client \*/

56 String request = br.readLine();

57 String command = null;

58 String pathname = null;

59 }

62 }

65 }

68 }

71 }

74 }

77 }

80 }

83 }

86 }

89 }

92 }

95 }

98 }

101 }

104 }

107 }

110 }

113 }

116 }

119 }

122 }

125 }

128 }

131 }

134 }

137 }

140 }

143 }

146 }

149 }

152 }

155 }

158 }

161 }

164 }

167 }

170 }

173 }

176 }

179 }

182 }

185 }

188 }

191 }

194 }

197 }

200 }

203 }

206 }

209 }

212 }

215 }

218 }

221 }

224 }

227 }

230 }

233 }

236 }

239 }

242 }

245 }

248 }

251 }

254 }

257 }

260 }

263 }

266 }

269 }

272 }

275 }

278 }

281 }

284 }

287 }

290 }

293 }

296 }

299 }

302 }

305 }

308 }

311 }

314 }

317 }

320 }

323 }

326 }

329 }

332 }

335 }

338 }

341 }

344 }

347 }

350 }

353 }

356 }

359 }

362 }

365 }

368 }

371 }

374 }

377 }

380 }

383 }

386 }

389 }

392 }

395 }

398 }

401 }

404 }

407 }

410 }

413 }

416 }

419 }

422 }

425 }

428 }

431 }

434 }

437 }

440 }

443 }

446 }

449 }

452 }

455 }

458 }

461 }

464 }

467 }

470 }

473 }

476 }

479 }

482 }

485 }

488 }

491 }

494 }

497 }

499 }

502 }

505 }

508 }

511 }

514 }

517 }

520 }

523 }

526 }

529 }

532 }

535 }

538 }

541 }

544 }

547 }

550 }

553 }

556 }

559 }

562 }

565 }

568 }

571 }

574 }

577 }

580 }

583 }

586 }

589 }

592 }

595 }

598 }

601 }

604 }

607 }

610 }

613 }

616 }

619 }

622 }

625 }

628 }

631 }

634 }

637 }

640 }

643 }

646 }

649 }

652 }

655 }

658 }

661 }

664 }

667 }

670 }

673 }

676 }

679 }

682 }

685 }

688 }

691 }

694 }

697 }

699 }

702 }

705 }

708 }

711 }

714 }

717 }

720 }

723 }

726 }

729 }

732 }

735 }

738 }

741 }

744 }

747 }

750 }

753 }

756 }

759 }

762 }

765 }

768 }

771 }

774 }

777 }

780 }

783 }

786 }

789 }

792 }

795 }

798 }

801 }

804 }

807 }

810 }

813 }

816 }

819 }

822 }

825 }

828 }

831 }

834 }

837 }

840 }

843 }

846 }

849 }

852 }

855 }

858 }

861 }

864 }

867 }

870 }

873 }

876 }

879 }

882 }

885 }

888 }

891 }

894 }

897 }

899 }

902 }

905 }

908 }

911 }

914 }

917 }

920 }

923 }

926 }

929 }

932 }

935 }

938 }

941 }

944 }

947 }

950 }

953 }

956 }

959 }

962 }

965 }

968 }

971 }

974 }

977 }

980 }

983 }

986 }

989 }

992 }

995 }

998 }

1001 }

1004 }

1007 }

1010 }

1013 }

1016 }

1019 }

1022 }

1025 }

1028 }

1031 }

1034 }

1037 }

1040 }

1043 }

1046 }

1049 }

1052 }

1055 }

1058 }

1061 }

1064 }

1067 }

1070 }

1073 }

1076 }

1079 }

1082 }

1085 }

1088 }

1091 }

1094 }

1097 }

1099 }

1102 }

1105 }

1108 }

1111 }

1114 }

1117 }

1120 }

1123 }

1126 }

1129 }

1132 }

1135 }

1138 }

1141 }

1144 }

1147 }

1150 }

1153 }

1156 }

1159 }

1162 }

1165 }

1168 }

1171 }

1174 }

1177 }

1180 }

1183 }

1186 }

1189 }

1192 }

1195 }

1198 }

1201 }

1204 }

1207 }

1210 }

1213 }

1216 }

1219 }

1222 }

1225 }

1228 }

1231 }

1234 }

1237 }

1240 }

1243 }

1246 }

1249 }

1252 }

1255 }

1258 }

1261 }

1264 }

1267 }

1270 }

1273 }

1276 }

1279 }

1282 }

1285 }

1288 }

1291 }

1294 }

1297 }

1299 }

1302 }

1305 }

1308 }

1311 }

1314 }

1317 }

1320 }

1323 }

1326 }

1329 }

1332 }

1335 }

1338 }

1341 }

1344 }

1347 }

1350 }

1353 }

1356 }

1359 }

1362 }

1365 }

1368 }

1371 }

1374 }

1377 }

1380 }

1383 }

1386 }

1389 }

1392 }

1395 }

1398 }

1401 }

1404 }

1407 }

1410 }

1413 }

1416 }

1419 }

1422 }

1425 }

1428 }

1431 }

1434 }

1437 }

1440 }

1443 }

1446 }

1449 }

1452 }

1455 }

1458 }

1461 }

1464 }

1467 }

1470 }

1473 }

1476 }

1479 }

1482 }

1485 }

1488 }

1491 }

1494 }

1497 }

1499 }

1502 }

1505 }

1508 }

1511 }

1514 }

1517 }

1520 }

1523 }

1526 }

1529 }

1532 }

1535 }

1538 }

1541 }

1544 }

1547 }

1550 }

1553 }

1556 }

1559 }

1562 }

1565 }

1568 }

1571 }

1574 }

1577 }

1580 }

1583 }

1586 }

1589 }

1592 }

1595 }

1598 }

1601 }

1604 }

1607 }

1610 }

1613 }

1616 }

1619 }

1622 }

1625 }

1628 }

1631 }

1634 }

1637 }

1640 }

1643 }

1646 }

1649 }

1652 }

1655 }

1658 }

1661 }

1664 }

1667 }

1670 }

1673 }

1676 }

1679 }

1682 }

1685 }

1688 }

1691 }

1694 }

1697 }

1699 }

1702 }

1705 }

1708 }

1711 }

1714 }

1717 }

1720 }

1723 }

1726 }

1729 }

1732 }

1735 }

1738 }

1741 }

1744 }

1747 }

1750 }

1753 }

1756 }

1759 }

1762 }

1765 }

1768 }

1771 }

1774 }

1777 }

1780 }

1783 }

1786 }

1789 }

1792 }

1795 }

1798 }

1801 }

1804 }

1807 }

1810 }

1813 }

1816 }

1819 }

1822 }

1825 }

1828 }

1831 }

1834 }

1837 }

1840 }

1843 }

1846 }

1849 }

1852 }

1855 }

1858 }

1861 }

1864 }

1867 }

1870 }

1873 }

1876 }

1879 }

1882 }

1885 }

1888 }

1891 }

1894 }

1897 }

1899 }

1902 }

1905 }

1908 }

1911 }

1914 }

1917 }

1920 }

1923 }

1926 }

1929 }

1932 }

1935 }

1938 }

1941 }

1944 }

1947 }

1950 }

1953 }

1956 }

1959 }

1962 }

1965 }

1968 }

1971 }

1974 }

1977 }

1980 }

1983 }

1986 }

1989 }

1992 }

1995 }

1998 }

2001 }

2004 }

2007 }

2010 }

2013 }

2016 }

2019 }

2022 }

2025 }

2028 }

2031 }

2034 }

2037 }

2040 }

2043 }

2046 }

2049 }

2052 }

2055 }

2058 }

2061 }

2064 }

2067 }

2070 }

2073 }

2076 }

2079 }

2082 }

2085 }

2088 }

2091 }

2094 }

2097 }

2099 }

2102 }

2105 }

2108 }

2111 }

2114 }

2117 }

2120 }

2123 }

2126 }

2129 }

2132 }

2135 }

2138 }

2141 }

2144 }

2147 }

2150 }

2153 }

2156 }

2159 }

2162 }

2165 }

2168 }

2171 }

2174 }

2177 }

2180 }

2183 }

2186 }

2189 }

2192 }

2195 }

2198 }

2201 }

2204 }

2207 }

2210 }

2213 }

2216 }

2219 }

2222 }

2225 }

2228 }

2231 }

2234 }

2237 }

2240 }

2243 }

2246 }

2249 }

2252 }

2255 }

2258 }

2261 }

2264 }

2267 }

2270 }

2273 }

2276 }

2279 }

2282 }

2285 }

2288 }

2291 }

2294 }

2297 }

2299 }

2302 }

2305 }

2308 }

2311 }

2314 }

2317 }

2320 }

2323 }

2326 }

2329 }

2332 }

2335 }

2338 }

2341 }

2344 }

2347 }

2350 }

2353 }

2356 }

2359 }

2362 }

2365 }

2368 }

2371 }

2374 }

2377 }

2380 }

2383 }

2386 }

2389 }

2392 }

2395 }

2398 }

2401 }

2404 }

2407 }

2410 }

2413 }

2416 }

2419 }

2422 }

2425 }

2428 }

2431 }

2434 }

2437 }

2440 }

2443 }

2446 }

2449 }

2452 }

2455 }

2458 }

2461 }

2464 }

2467 }

2470 }

2473 }

2476 }

2479 }

2482 }

2485 }

2488 }

2491 }

2494 }

2497 }

2499 }

2502 }

2505 }

2508 }

2511 }

2514 }

2517 }

2520 }

2523 }

2526 }

2529 }

2532 }

2535 }

2538 }

2541 }

2544 }

2547 }

2550 }

2553 }

2556 }

2559 }

2562 }

2565 }

2568 }

2571 }

2574 }

2577 }

2580 }

2583 }

2586 }

2589 }

2592 }

2595 }

2598 }

2601 }

2604 }

2607 }

2610 }

2613 }

2616 }

2619 }

2622 }

2625 }

2628 }

2631 }

2634 }

2637 }

2640 }

2643 }

2646 }

2649 }

2652 }

2655 }

2658 }

2661 }

2664 }

2667 }

2670 }

2673 }

2676 }

2679 }

2682 }

2685 }

2688 }

2691 }

2694 }

2697 }

2699 }

2702 }

2705 }

2708 }

2711 }

2714 }

2717 }

2720 }

2723 }

2726 }

2729 }

2732 }

2735 }

2738 }

2741 }

2744 }

2747 }

2750 }

2753 }

2756 }

2759 }

2762 }

2765 }

2768 }

2771 }

2774 }

2777 }

2780 }

2783 }

2786 }

2789 }

2792 }

2795 }

2798 }

2801 }

2804 }

2807 }

2810 }

2813 }

2816 }

2819 }

2822 }

2825 }

2828 }

2831 }

2834 }

2837 }

2840 }

2843 }

2846 }

2849 }

2852 }

2855 }

2858 }

2861 }

2864 }

2867 }

2870 }

2873 }

2876 }

2879 }

2882 }

2885 }

2888 }

2891 }

2894 }

2897 }

2899 }

2902 }

2905 }

2908 }

2911 }

2914 }

2917 }

2920 }

2923 }

2926 }

2929 }

2932 }

2935 }

2938 }

2941 }

2944 }

2947 }

2950 }

2953 }

2956 }

2959 }

2962 }

2965 }

2968 }

2971 }

2974 }

2977 }

2980 }

2983 }

2986 }

2989 }

2992 }

2995 }

2998 }

3001 }

3004 }

3007 }

3010 }

3013 }

3016 }

3019 }

3022 }

3025 }

3028 }

3031 }

3034 }

3037 }

3040 }

3043 }

3046 }

3049 }

3052 }

3055 }

3058 }

3061 }

3064 }

3067 }

3070 }

3073 }

3076 }

3079 }

3082 }

3085 }

3088 }

3091 }

3094 }

3097 }

3099 }

3102 }

3105 }

3108 }

3111 }

3114 }

3117 }

3120 }

3123 }

3126 }

3129 }

3132 }

3135 }

3138 }

3141 }

3144 }

3147 }

3150 }

3153 }

3156 }

3159 }

3162 }

3165 }

3168 }

3171 }

3174 }

3177 }

3180 }

3183 }

3186 }

3189 }

3192 }

3195 }

3198 }

3201 }

3204 }

3207 }

3210 }

3213 }

3216 }

3219 }

3222 }

3225 }

3228 }

3231 }

3234 }

3237 }

3240 }

3243 }

3246 }

3249 }

3252 }

3255 }

3258 }

3261 }

3264 }

3267 }

3270 }

3273 }

3276 }

3279 }

3282 }

3285 }

3288 }

3291 }

3294 }

3297 }

3299 }

3302 }

3305 }

3308 }

3311 }

3314 }

3317 }

3320 }

3323 }

3326 }

3329 }

3332 }

3335 }

3338 }

3341 }

3344 }

3347 }

3350 }

3353 }

3356 }

3359 }

3362 }

3365 }

3368 }

3371 }

3374 }

3377 }

3380 }

3383 }

3386 }

3389 }

3392 }

3395 }

3398 }

3401 }

3404 }

3407 }

3410 }

3413 }

3416 }

3419 }

3422 }

3425 }

3428 }

3431 }

3434 }

3437 }

3440 }

3443 }

3446 }

3449 }

3452 }

3455 }

3458 }

3461 }

3464 }

3467 }

3470 }

3473 }

3476 }

3479 }

3482 }

3485 }

3488 }

3491 }

3494 }

3497 }

3499 }

3502 }

3505 }

3508 }

3511 }

3514 }

3517 }

3520 }

3523 }

3526 }

3529 }

3532 }

3535 }

3538 }

3541 }

3544 }

3547 }

3550 }

3553 }

3556 }

3559 }

3562 }

3565 }

3568 }

3571 }

3574 }

3577 }

3580 }

3583 }

3586 }

3589 }

3592 }

3595 }

3598 }

3601 }

3604 }

3607 }

3610 }

3613 }

3616 }

3619 }

3622 }

3625 }

3628 }

3631 }

3634 }

3637 }

3640 }

3643 }

3646 }

3649 }

3652 }

3655 }

3658 }

3661 }

3664 }

3667 }

3670 }

3673 }

3676 }

3679 }

3682 }

3685 }

3688 }

3691 }

3694 }

3697 }

3699 }

3702 }

3705 }

3708 }

3711 }

3714 }

3717 }

3720 }

3723 }

3726 }

3729 }

3732 }

3735 }

3738 }

3741 }

3744 }

3747 }

3750 }

3753 }

3756 }

3759 }

3762 }

3765 }

3768 }

3771 }

3774 }

3777 }

3780 }

3783 }

3786 }

3789 }

3792 }

3795 }

3798 }

3801 }

3804 }

3807 }

3810 }

3813 }

3816 }

3819 }

3822 }

3825 }

3828 }

3831 }

3834 }

3837 }

3840 }

3843 }

3846 }

3849 }

3852 }

3855 }

3858 }

3861 }

3864 }

3867 }

3870 }

3873 }

3876 }

3879 }

3882 }

3885 }

3888 }

3891 }

3894 }

3897 }

3899 }

3902 }

3905 }

3908 }

3911 }

3914 }

3917 }

3920 }

3923 }

3926 }

3929 }

3932 }

3935 }

3938 }

3941 }

3944 }

3947 }

3950 }

3953 }

3956 }

3959 }

3962 }

3965 }

3968 }

3971 }

3974 }

3977 }

3980 }

3983 }

3986 }

3989 }

3992 }

3995 }

3998 }

4001 }

4004 }

4007 }

4010 }

4013 }

4016 }

4019 }

4022 }

4025 }

4028 }

4031 }

4034 }

4037 }

4040 }

4043 }

4046 }

4049 }

4052 }

4055 }

4058 }

4061 }

4064 }

4067 }

4070 }

4073 }

4076 }

4079 }

4082 }

4085 }

40

## Case 2: Aggressive

Minimum confidence to report: Low

Minimum rank to report: 20 (Of Concern)

Reported visible bugs categories:

Bad practice

Correctness

Experimental

Internationalization

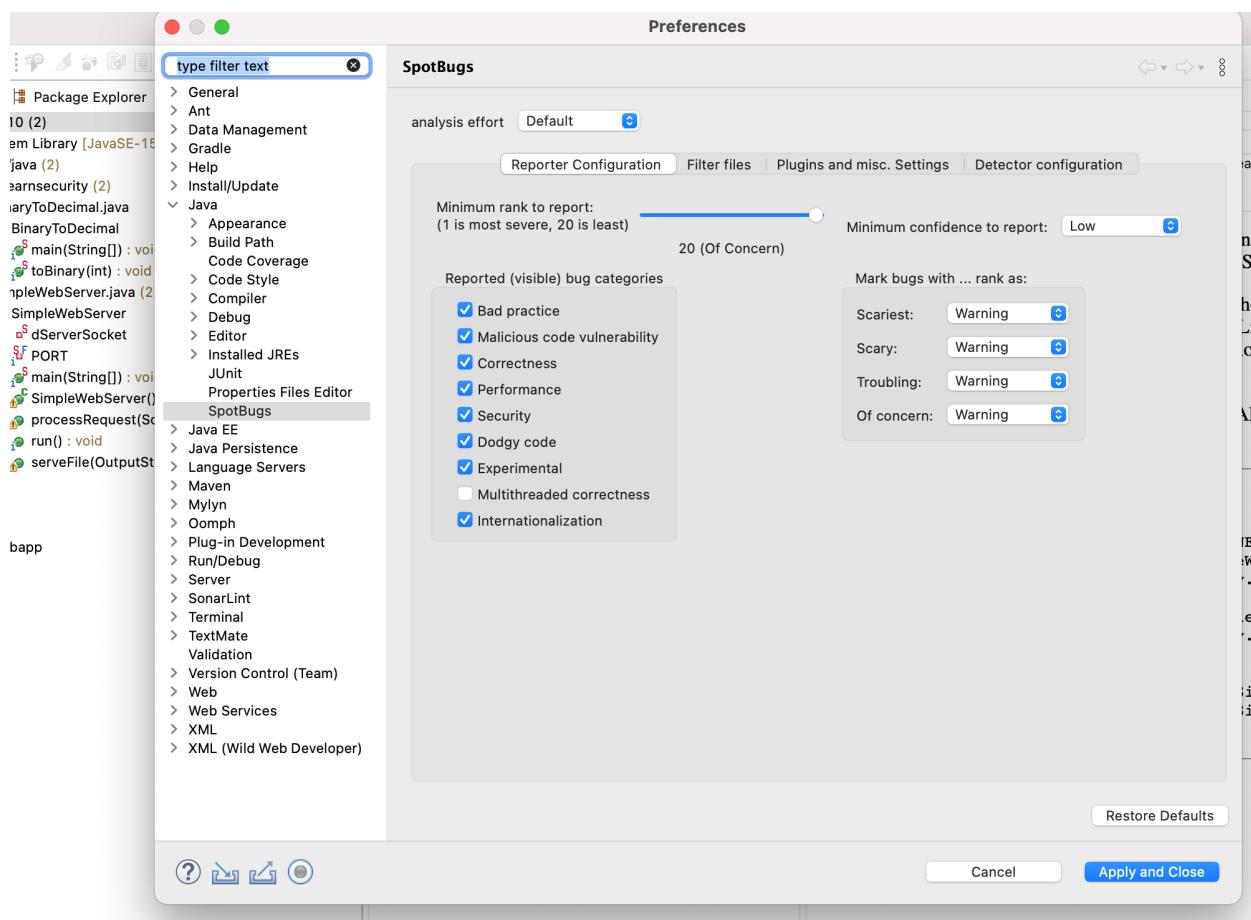
Malicious code vulnerability

Performance

Security

Dodgy code

No. Of Bugs: 8



### Case 3:

## Minimum confidence to report: High

Minimum rank to report: 1 (Of Concern)

## Reported visible bugs categories:

## Bad practice

## Correctness

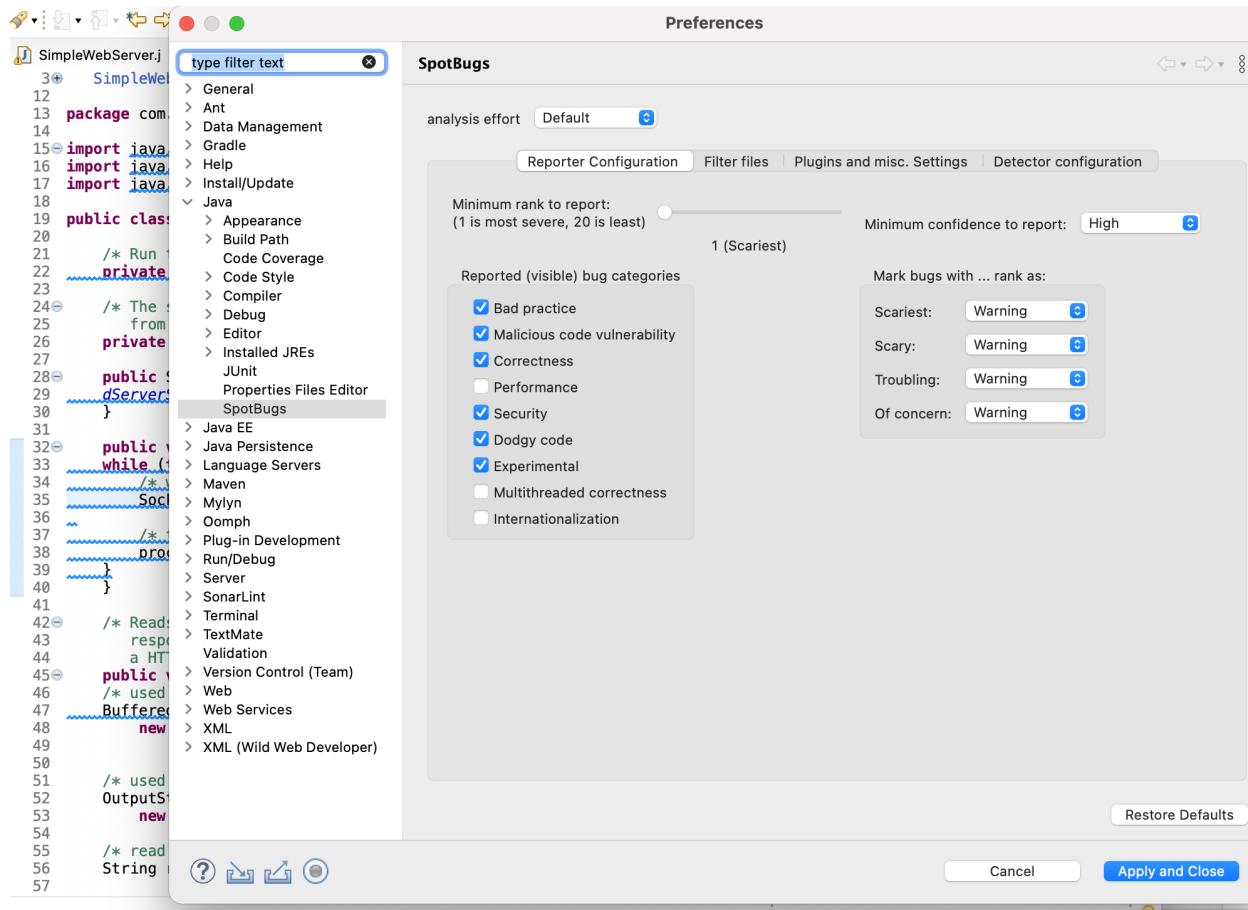
## Experimental

## Malicious code vulnerability

## Security

## Dodgy code

No. Of Bugs: 0



```

3*  SimpleWebServer.java
12
13 package com.learnsecurity;
14
15 import java.io.*;
16 import java.net.*;
17 import java.util.*;
18
19 public class SimpleWebServer {
20
21     /* Run the HTTP server on this TCP port. */
22     private static final int PORT = 8080;
23
24     /* The socket used to process incoming connections
25      from web clients */
26     private static ServerSocket dServerSocket;
27
28     public SimpleWebServer () throws Exception {
29         dServerSocket = new ServerSocket (PORT);
30     }
31
32     public void run() throws Exception {
33         while (true) {
34             /* wait for a connection from a client */
35             Socket s = dServerSocket.accept();
36
37             /* then process the client's request */
38             processRequest(s);
39         }
40     }
41
42     /* Reads the HTTP request from the client, and
43      responds with the file the user requested or
44      a HTTP error code. */
45     public void processRequest(Socket s) throws Exception {
46         /* used to read data from the client */
47         BufferedReader br =
48             new BufferedReader (
49                 new InputStreamReader (s.getInputStream()));
50
51         /* used to write data to the client */
52         OutputStreamWriter osw =
53             new OutputStreamWriter (s.getOutputStream());
54
55         /* read the HTTP request from the client */
56         String request = br.readLine();

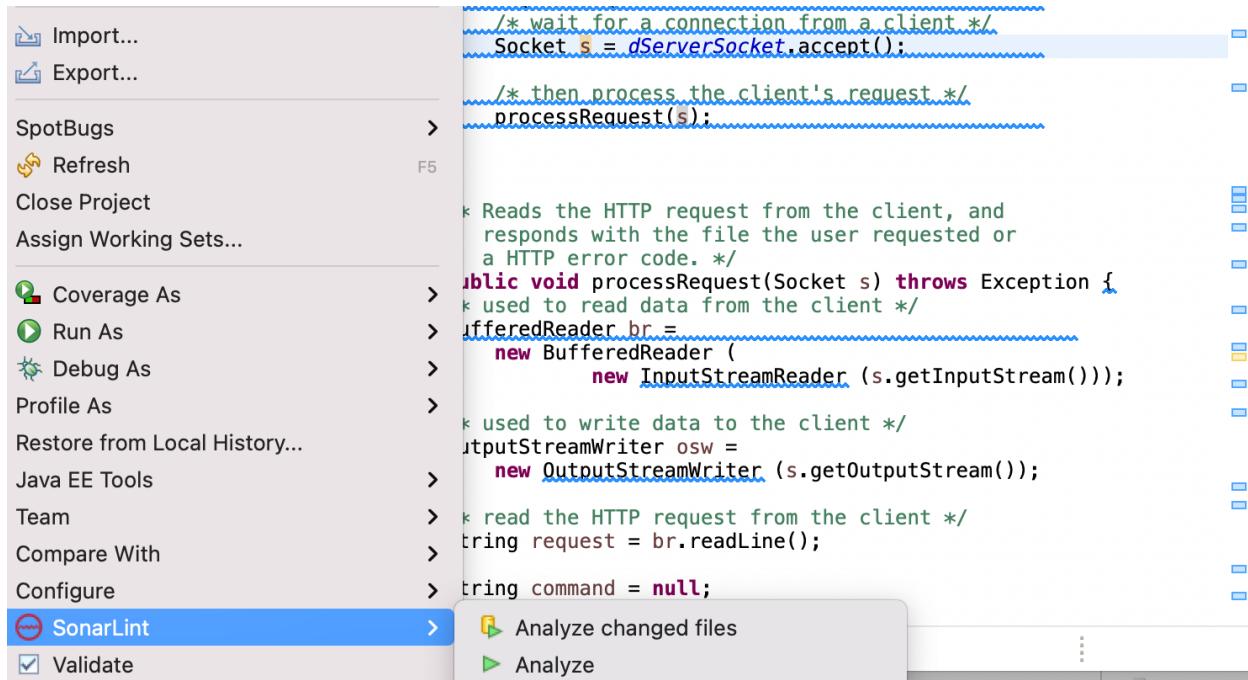
```

Writable

Smart Insert

35 : 17 : 1294

## Analysis using SonarLint



### Case 1: Default

Severity of SonarLint markers: Info

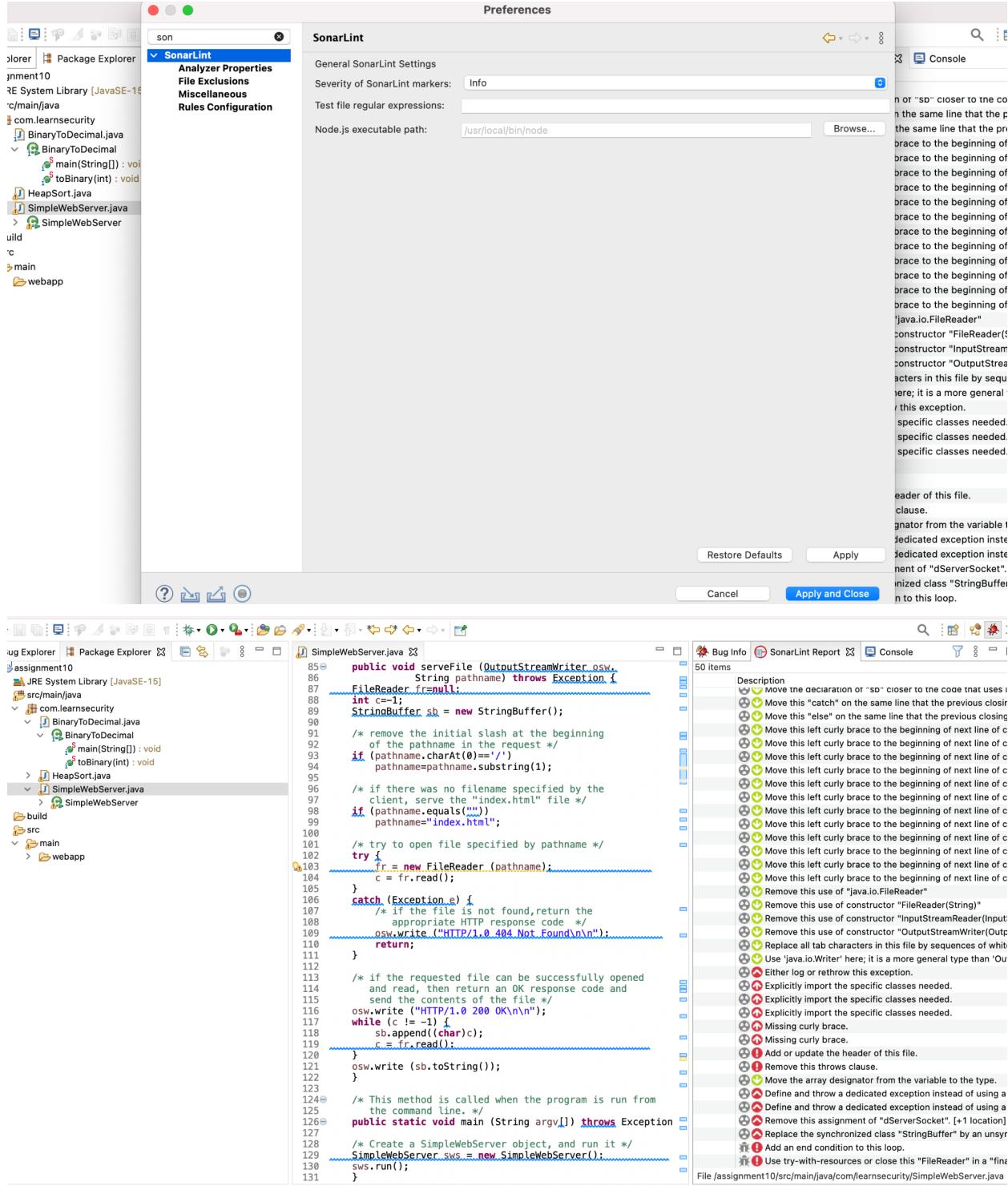
Total code smell: 50

Minor Code Smell: 36

Major Code Smell: 5

Blocker Code Smell: 4

Critical Code Smell: 5



## Case 2:

Severity of SonarLint markers: Warning

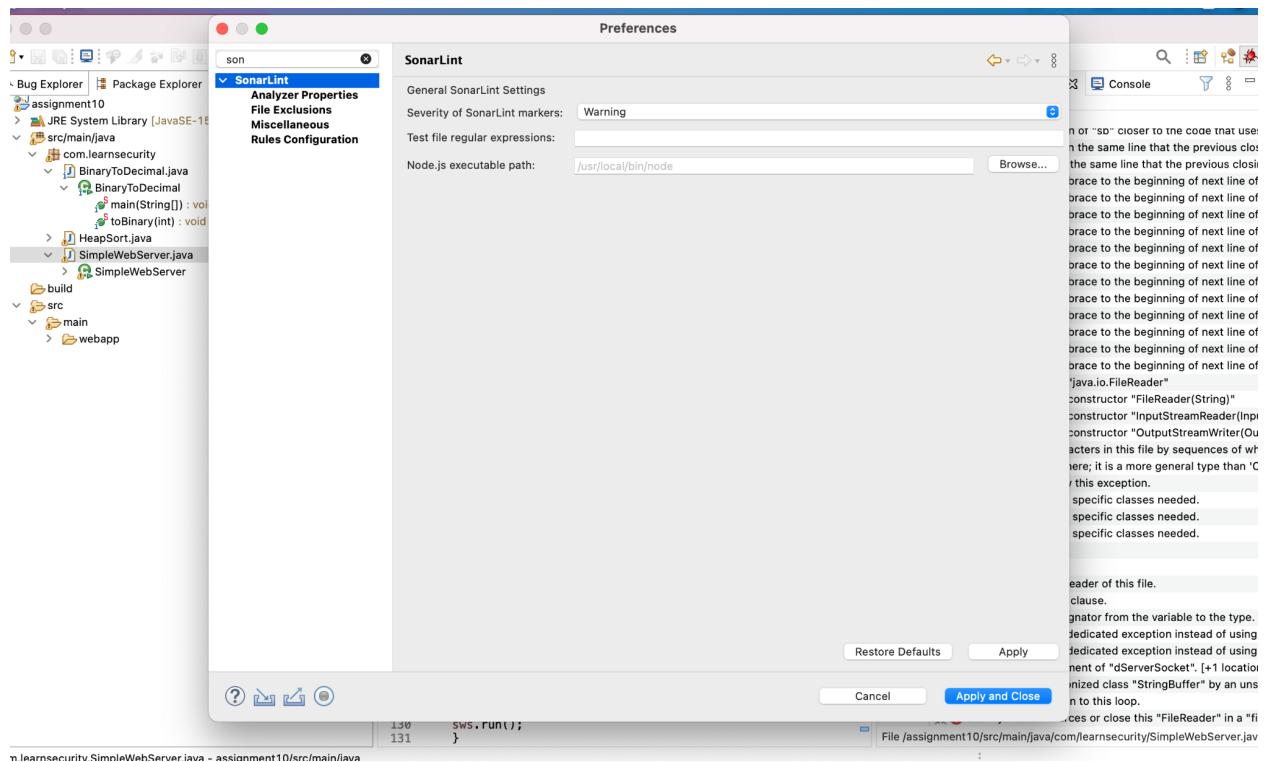
Total code smell: 50

Minor Code Smell: 36

Major Code Smell: 5

Blocker Code Smell: 4

Critical Code Smell: 5



The screenshot shows an IDE interface with two main panes. The left pane displays the Java code for `SimpleWebServer.java`. The right pane shows the SonarLint analysis results for the same file, indicating 0 errors, 50 warnings, and 0 others.

**SimpleWebServer.java**

```
public void serveFile (OutputStreamWriter osw,
                      String pathname) throws Exception {
    FileReader fr=null;
    int c=-1;
    StringBuffer sb = new StringBuffer();
    /* remove the initial slash at the beginning
       of the pathname in the request */
    if (pathname.charAt(0)=='/')
        pathname=pathname.substring(1);
    /* if there was no filename specified by the
       client, serve the "index.html" file */
    if (pathname.equals(""))
        pathname="index.html";
    /* try to open file specified by pathname */
    try {
        fr = new FileReader (pathname);
        c = fr.read();
    } catch (Exception e) {
        /* If the file is not found,return the
           appropriate HTTP response code */
        osw.write ("HTTP/1.0 404 Not Found\n\n");
        return;
    }
    /* if the requested file can be successfully opened
       and read, then return an OK response code and
       send the contents of the file */
    osw.write ("HTTP/1.0 200 OK\n\n");
    while (c != -1) {
        sb.append((char)c);
        c = fr.read();
    }
    osw.write (sb.toString());
}
/* This method is called when the program is run from
   the command line. */
public static void main (String argv[]) throws Exception {
    /* Create a SimpleWebServer object, and run it */
    SimpleWebServer sws = new SimpleWebServer();
    sws.run();
}
```

**Bug Info** | **SonarLint Report** | **Console**  
0 errors, 50 warnings, 0 others

Description

- Move the declaration of "sb" closer to the code that uses it.
- Move this "catch" on the same line that the previous closing curly brace.
- Move this "else" on the same line than the previous closing curly brace.
- Move this left curly brace to the beginning of next line of code.
- Move this left curly brace to the beginning of next line of code.
- Move this left curly brace to the beginning of next line of code.
- Move this left curly brace to the beginning of next line of code.
- Move this left curly brace to the beginning of next line of code.
- Move this left curly brace to the beginning of next line of code.
- Move this left curly brace to the beginning of next line of code.
- Move this left curly brace to the beginning of next line of code.
- Move this left curly brace to the beginning of next line of code.
- Move this left curly brace to the beginning of next line of code.
- Move this left curly brace to the beginning of next line of code.
- Move this left curly brace to the beginning of next line of code.
- Move this left curly brace to the beginning of next line of code.
- Move this left curly brace to the beginning of next line of code.
- Move this left curly brace to the beginning of next line of code.
- Move this left curly brace to the beginning of next line of code.
- Move this left curly brace to the beginning of next line of code.
- Move this left curly brace to the beginning of next line of code.
- Remove this use of "java.io.FileReader"
- Remove this use of constructor "FileReader(String)"
- Remove this use of constructor "InputStreamReader(InputStream)"
- Remove this use of constructor "OutputStreamWriter(OutputStream)"
- Replace all tab characters in this file by sequences of white-spaces.
- Use 'java.io.Writer' here; it is a more general type than 'OutputStreamWriter'.
- Either log or rethrow this exception.
- Explicitly import the specific classes needed.
- Explicitly import the specific classes needed.
- Explicitly import the specific classes needed.
- Missing curly brace.
- Missing curly brace.
- Add or update the header of this file.
- Remove this throws clause.
- Move the array designator from the variable to the type.
- Define and throw a dedicated exception instead of using a generic one.
- Define and throw a dedicated exception instead of using a generic one.
- Remove this assignment of "dServerSocket". [+1 location]
- Replace the synchronized class "StringBuffer" by an unsynchronized one such as
- Add an end condition to this loop.
- Use try-with-resources or close this "FileReader" in a finally clause.

File /assignment10/src/main/java/com/learnsecurity/SimpleWebServer.java (at 25/04/2021 20:38)

### **Case 3:**

## Severity of SonarLint markers: Error

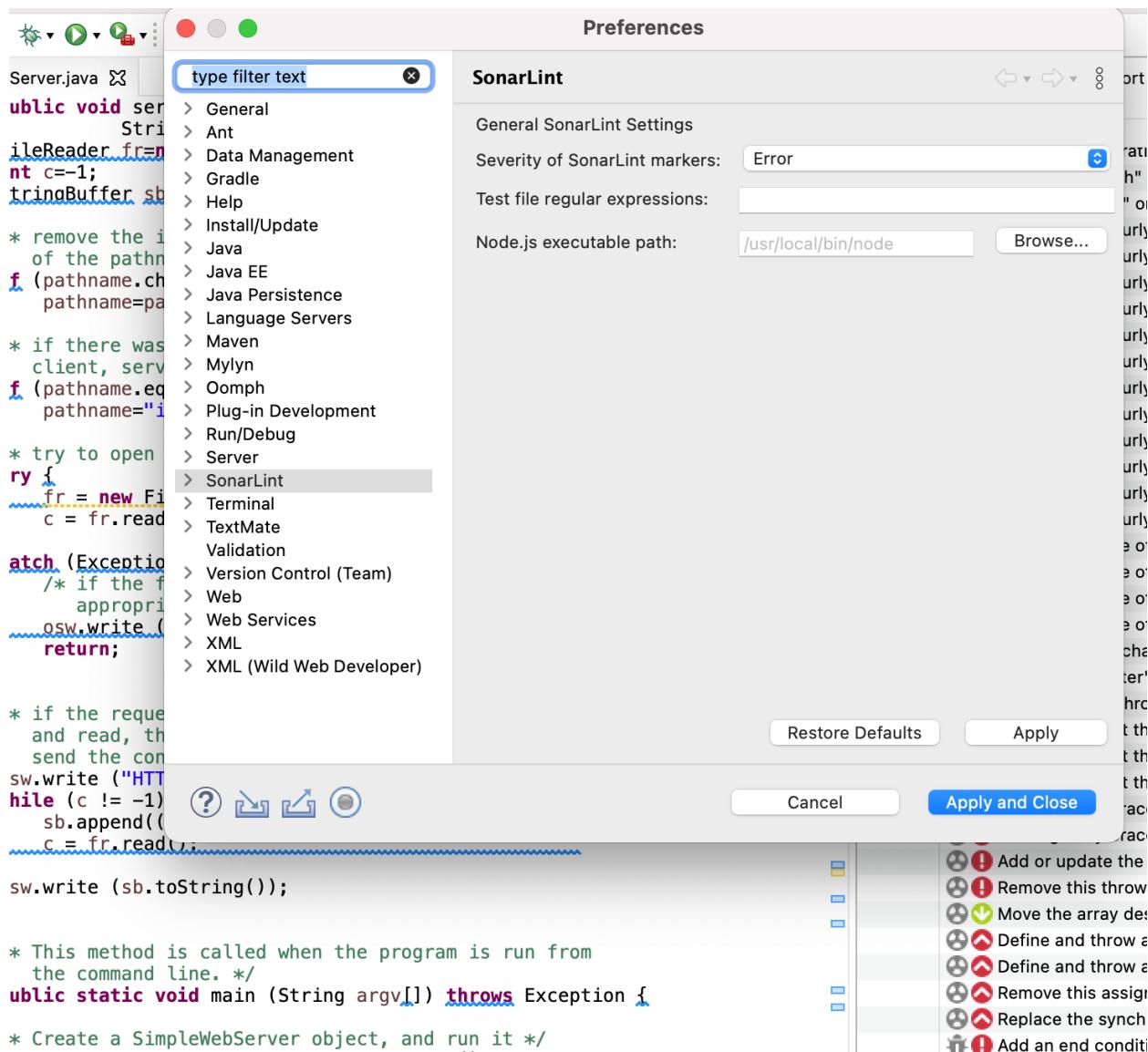
## Total code smell: 50

## Minor Code Smell: 36

## Major Code Smell: 5

## Blocker Code Smell: 4

## Critical Code Smell: 5



```

85    public void serveFile (OutputStreamWriter osw,
86                           String pathname) throws Exception {
87     FileReader fr=null;
88     int c=-1;
89     StringBuffer sb = new StringBuffer();
90
91     /* remove the initial slash at the beginning
92      of the pathname in the request */
93     if (pathname.charAt(0)=='/')
94       pathname=pathname.substring(1);
95
96     /* if there was no filename specified by the
97      client, serve the "index.html" file */
98     if (pathname.equals(""))
99       pathname="index.html";
100
101    /* try to open file specified by pathname */
102    try {
103      fr=new FileReader(pathname);
104      c=fr.read();
105    }
106    catch (Exception e) {
107      /* if the file is not found, return the
108       appropriate HTTP response code */
109      osw.write ("HTTP/1.0 404 Not Found\n\n");
110      return;
111    }
112
113    /* if the requested file can be successfully opened
114     and read, then return an OK response code and
115     send the contents of the file */
116    osw.write ("HTTP/1.0 200 OK\n\n");
117    while (c != -1) {
118      sb.append((char)c);
119      c=fr.read();
120    }
121    osw.write (sb.toString());
122  }
123
124  /* This method is called when the program is run from
125   the command line. */
126  public static void main (String argv[]) throws Exception {
127
128    /* Create a SimpleWebServer object, and run it */
129    SimpleWebServer sws = new SimpleWebServer();
130    sws.run();
131  }
}

```

Bug Info   SonarLint Report   Console

50 errors, 0 warnings, 0 others

**Description**

- Move the declaration or "so" closer to the code that uses it.
- Move this "catch" on the same line that the previous closing curly brace.
- Move this "else" on the same line that the previous closing curly brace.
- Move this left curly brace to the beginning of next line of code.
- Move this left curly brace to the beginning of next line of code.
- Move this left curly brace to the beginning of next line of code.
- Move this left curly brace to the beginning of next line of code.
- Move this left curly brace to the beginning of next line of code.
- Move this left curly brace to the beginning of next line of code.
- Move this left curly brace to the beginning of next line of code.
- Move this left curly brace to the beginning of next line of code.
- Move this left curly brace to the beginning of next line of code.
- Move this left curly brace to the beginning of next line of code.
- Move this left curly brace to the beginning of next line of code.
- Move this left curly brace to the beginning of next line of code.
- Move this left curly brace to the beginning of next line of code.
- Move this left curly brace to the beginning of next line of code.
- Move this left curly brace to the beginning of next line of code.
- Move this left curly brace to the beginning of next line of code.
- Move this left curly brace to the beginning of next line of code.
- Move this left curly brace to the beginning of next line of code.
- Move this left curly brace to the beginning of next line of code.
- Remove this use of "java.io.FileReader"
- Remove this use of constructor "FileReader(String)"
- Remove this use of constructor "InputStreamReader(InputStream)"
- Remove this use of constructor "OutputStreamWriter(OutputStream)"
- Replace all tab characters in this file by sequences of white-spaces.
- Use 'java.io.Writer' here; it is a more general type than 'OutputStreamWriter'.
- Either log or rethrow this exception.
- Explicitly import the specific classes needed.
- Explicitly import the specific classes needed.
- Missing curly brace.
- Missing curly brace.
- Add or update the header of this file.
- Remove this throws clause.
- Move the array designator from the variable to the type.
- Define and throw a dedicated exception instead of using a generic one.
- Define and throw a dedicated exception instead of using a generic one.
- Remove this assignment of "iServerSocket". [+1 location]
- Replace the synchronized class "StringBuffer" by an unsynchronized one such as
- Add an end condition to this loop.
- Use try-with-resources or close this "FileReader" in a "finally" clause.

File /assignment10/src/main/java/com/learnsecurity/SimpleWebServer.java (at 25/04/2021 20:11)

## Comparison of the two tools:

**SpotBugs:** Type checking, Bug finding, Security review.

**SonarLint:** Program understanding, Bug finding, Type Checking, Security Review.

SpotBugs analyzes **binary** as input.

SonarLint analyzes **source** as input.

SpotBugs and SonarLint, both are **Bug Finding Tools**.

## Bug reported by both the tools:

Line number: 103

**SpotBugs:** Found reliance on default encoding in  
com.learnsecurity.SimpleWebServer.serveFile(OutputStreamWriter, String):  
new java.io.FileReader(String)

**SonarLint:** Use try-with-resources or close this "FileReader" in a "finally" clause.

**Bug present in one tool only:**  
**Not Found in SpotBugs but by SonarLint.**  
**SonarLint: Add an end condition to this loop.**

```

1 Jse try-with-resources or close this "FileReader" in a "finally" clause.
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

```

**Add an end condition to this loop.**

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65

```

## Part 2:

### CODE:

```
package com.learnsecurity;

public class BinaryToDecimal {

    public static void toBinary(int decimal){
        int binary[] = new int[40];
        int index = 0;
        while(decimal > 0){
            binary[index++] = decimal%2;
            decimal = decimal/2;
        }
        for(int i = index-1;i >= 0;i--){
            System.out.print(binary[i]);
        }
        System.out.println();//new line
    }

    public static void main(String args[]){
        System.out.println("Decimal of 10 is: ");
        toBinary(10);
        System.out.println("Decimal of 21 is: ");
        toBinary(21);
        System.out.println("Decimal of 31 is: ");
        toBinary(31);
    }
}
```

The screenshot shows the Eclipse IDE interface. On the left, there are two tabs: "SimpleWebServer.java" and "BinaryToDecimal.java". The "BinaryToDecimal.java" tab is active, displaying the following Java code:

```

1 package com.learnsecurity;
2
3 public class BinaryToDecimal {
4
5     public static void toBinary(int decimal){
6         int binary[] = new int[40];
7         int index = 0;
8         while(decimal > 0){
9             binary[index++] = decimal%2;
10            decimal = decimal/2;
11        }
12        for(int i = index-1;i >= 0;i--){
13            System.out.print(binary[i]);
14        }
15        System.out.println();//new line
16    }
17    public static void main(String args[]){
18        System.out.println("Decimal of 10 is: ");
19        toBinary(10);
20        System.out.println("Decimal of 21 is: ");
21        toBinary(21);
22        System.out.println("Decimal of 31 is: ");
23        toBinary(31);
24    }
25

```

On the right side of the interface, there are several tabs: "Bug Info", "SonarLint Report", "Console", and "Output". The "Console" tab is active, showing the output of the program's execution:

```

<terminated> BinaryToDecimal [Java Application] /Applica
Decimal of 10 is:
1010
Decimal of 21 is:
10101
Decimal of 31 is:
11111

```

## Manual Analysis:

The program needs indentation/formatting.

This screenshot shows the same Eclipse IDE setup as above, but with SonarLint annotations applied to the code. The "BinaryToDecimal.java" tab is active, and the "SonarLint" tab is selected in the top bar. The code now includes numerous SonarLint annotations, each with a small icon and a description:

- Line 5: Move the array designator from the variable to the type.
- Line 6: Assign this magic number 40 to a well-named constant, and use the const.
- Line 7: Make this line start after 4 spaces to indent the code consistently.
- Line 8: Make this line start after 2 spaces to indent the code consistently.
- Line 9: Make this line start after 4 spaces to indent the code consistently.
- Line 10: Make this line start after 6 spaces to indent the code consistently.
- Line 11: Make this line start after 6 spaces to indent the code consistently.
- Line 12: Move the array designator from the variable to the type.
- Line 13: Move this closing curly brace to the next line.
- Line 14: Move this left curly brace to the beginning of next line of code.
- Line 15: Move this left curly brace to the beginning of next line of code.
- Line 16: Move this left curly brace to the beginning of next line of code.
- Line 17: Assign this magic number 10 to a well-named constant, and use the const.
- Line 18: Assign this magic number 2 to a well-named constant, and use the const.
- Line 19: Assign this magic number 21 to a well-named constant, and use the const.
- Line 20: Assign this magic number 31 to a well-named constant, and use the const.
- Line 21: Extract this increment or decrement operator into a dedicated statement!
- Line 22: Replace this use of System.out or System.err by a logger.
- Line 23: Replace this use of System.out or System.err by a logger.
- Line 24: Replace this use of System.out or System.err by a logger.
- Line 25: Move this left curly brace to the beginning of next line of code.
- Line 26: Add or update the header of this file.

## Results:

SpotBugs: 0

SolarLint: 27 code smells

## Code Fix:

After formatting the entire set of code the errors reduced to 21 from 27.

The screenshot shows the Eclipse IDE interface. On the left, the Package Explorer displays several Java files under the 'com.learnsecurity' package, including SimpleWebServer.java, BinaryToDecimal.java, and HeapSort.java. In the center, the code editor shows the Java code for BinaryToDecimal.java. On the right, the SolarLint view shows a list of 21 code smells with their descriptions. The code in the editor is as follows:

```
1 package com.learnsecurity;
2
3 public class BinaryToDecimal {
4
5     public static void toBinary(int decimal) {
6         int a = 40;
7         int binary[] = new int[a];
8         int index = 0;
9         while (decimal > 0) {
10             binary[index++] = decimal % 2;
11             decimal = decimal / 2;
12         }
13         for (int i = index - 1; i >= 0; i--) {
14             System.out.print(binary[i]);
15         }
16         System.out.println();
17     }
18
19     public static void main(String args[]) {
20         System.out.println("Decimal of 10 is: ");
21         toBinary(10);
22         System.out.println("Decimal of 21 is: ");
23         toBinary(21);
24         System.out.println("Decimal of 31 is: ");
25         toBinary(31);
26     }
27
28 }
```

The SolarLint view lists 21 items with descriptions:

- Move this left curly brace to the beginning of next line of code.
- Replace all tab characters in this file by sequences of white-spaces.
- Move the array designator from the variable to the type.
- Assign this magic number 40 to a well-named constant, and use the const keyword.
- Move the array designator from the variable to the type.
- Move this left curly brace to the beginning of next line of code.
- Move this left curly brace to the beginning of next line of code.
- Assign this magic number 10 to a well-named constant, and use the const keyword.
- Assign this magic number 2 to a well-named constant, and use the const keyword.
- Assign this magic number 2 to a well-named constant, and use the const keyword.
- Assign this magic number 21 to a well-named constant, and use the const keyword.
- Assign this magic number 31 to a well-named constant, and use the const keyword.
- Extract this increment or decrement operator into a dedicated statement.
- Replace this use of System.out or System.err by a logger.
- Replace this use of System.out or System.err by a logger.
- Replace this use of System.out or System.err by a logger.
- Replace this use of System.out or System.err by a logger.
- Move this left curly brace to the beginning of next line of code.
- Add or update the header of this file.