

1. My random generator class called RandomTestURLValidator is implemented mainly by using Random class to generate different parts of a url.
2. In this class, there are private functions called generators, including CorrectSiteGenerator(), IncorrectSiteGenerator(), CorrectPortGenerator(), IncorrectPortGenerator(), IncorrectIPGenerator(), CorrectIPGenerator(), are used to generate test cases randomly. Part of the code are as follows,

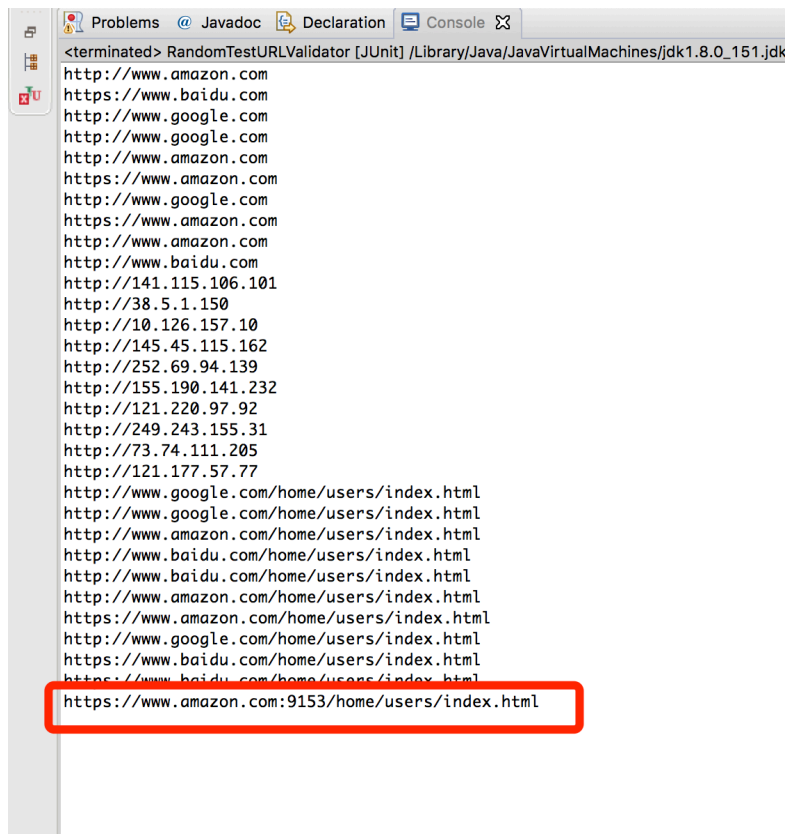
```
25- /**
26  * generates incorrect port number
27  * @return port string like ":xxx".
28  */
29- private String IncorrectPortGenerator() {
30     Random rand = new Random();
31     String ret = "";
32     int rd = rand.nextInt(15000);
33     if (rand.nextBoolean()) {
34         ret = ":-" + rd;
35     } else {
36         rd += 100000;
37         ret = ":" + rd;
38     }
39
40     return ret;
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
66
67
68
69- /**
70  * generate correct ip address
71  * @return
72  */
73- private String CorrectIPGenerator() {
74     Random rand = new Random();
75     String ret = rand.nextInt(255) + "." + rand.nextInt(255) + "." +
76         rand.nextInt(255) + "." + rand.nextInt(255);
77     return ret;
78 }
79
80
81
82
83
84
85
86
87
88
89
90- /**
91  * generate correct ip address
92  * @return
93  */
94- private String IncorrectIPGenerator() {
95     Random rand = new Random();
96     int ip1 = 0;
97     while (ip1 <= 255) {
98         ip1 = rand.nextInt(1000);
99     }
100     String ret = ip1 + "." + rand.nextInt(1000) + "." +
101         rand.nextInt(1000) + "." + rand.nextInt(1000);
102     return ret;
103 }
104
```

3. The calling function, testIsValid(), calls all the generators to generate correct and incorrect test cases respectively, then test isValid() function.

```
114
115 public void testIsValid() {
116     //generate correct test data.
117     UrlValidator urlVal = new UrlValidator(null, null, UrlValidator.ALLOW_ALL_SCHEMES);
118     String correctUrl;
119     String incorrectUrl;
120
121     for (int i = 0; i < 10; i++) {
122         correctUrl = CorrectSiteGenerator();
123         System.out.println(correctUrl);
124         assertTrue(urlVal.isValid(correctUrl));
125     }
126
127     for (int i = 0; i < 10; i++) {
128         correctUrl = CorrectIPGenerator();
129         System.out.println(correctUrl);
130         assertTrue(urlVal.isValid(correctUrl));
131     }
132
133     for (int i = 0; i < 10; i++) {
134         correctUrl = CorrectSiteGenerator() + CorrectPathGenerator();
135         System.out.println(correctUrl);
136         assertTrue(urlVal.isValid(correctUrl));
137     }
138
139     for (int i = 0; i < 10; i++) {
140         correctUrl = CorrectSiteGenerator() + CorrectPortGenerator() + CorrectPathGenerator();
141         System.out.println(correctUrl);
142         assertTrue(urlVal.isValid(correctUrl));
143     }
144 }
```

4. We can see the test case which caused the system fail in the output result.

As the snapshot below, there are output urls passed the test, but the last one failed, then we can go check the last case.



The screenshot shows an IDE interface with a console window. The console title is "<terminated> RandomTestURLValidator [JUnit] /Library/Java/JavaVirtualMachines/jdk1.8.0_151.jdk". The console contains a list of URLs. The last line, "https://www.amazon.com:9153/home/users/index.html", is highlighted with a red rectangle.

```
<terminated> RandomTestURLValidator [JUnit] /Library/Java/JavaVirtualMachines/jdk1.8.0_151.jdk
http://www.amazon.com
https://www.baidu.com
http://www.google.com
http://www.google.com
http://www.amazon.com
https://www.amazon.com
http://www.google.com
https://www.amazon.com
http://www.amazon.com
http://www.baidu.com
http://141.115.106.101
http://38.5.1.150
http://10.126.157.10
http://145.45.115.162
http://252.69.94.139
http://155.190.141.232
http://121.220.97.92
http://249.243.155.31
http://73.74.111.205
http://121.177.57.77
http://www.google.com/home/users/index.html
http://www.google.com/home/users/index.html
http://www.amazon.com/home/users/index.html
http://www.baidu.com/home/users/index.html
http://www.baidu.com/home/users/index.html
http://www.amazon.com/home/users/index.html
https://www.amazon.com/home/users/index.html
http://www.google.com/home/users/index.html
https://www.baidu.com/home/users/index.html
https://www.baidu.com/home/users/index.html
https://www.amazon.com:9153/home/users/index.html
```