**Experiment Report - 49 -test14\_syntaxrule**

1. **Summary Table of Errors Found**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Error ID | Line Number | Error Type | Self-Detected? | Peer 1 Found? | Peer 2 Found? |
| E01 | line 28 | Semantic | √ | × | × |
| E02 | line 45 | Syntax | √ | √ | √ |
| E03 | line 53 | Logic | √ | × | √ |
| E04 | line 65 | Logic | × | × | × |

Additional Errors Found by Self: 0

Self-Review Detection Rate: 750%

Peer 1 Detection Rate: 25%

Peer 2 Detection Rate: 50%

1. **Source Code**
2. #include <syntaxrule.h>
3. //如何让程序不匹配引号或者注释里的东西
4. const QStringList basicType = {"\\bint\\b", "\\bbool\\b", "\\bchar\\b",
5. "\\bfloat\\b", "\\bdouble\\b", "\\bvoid\\b", "\\bstd::string\\b"};
6. //定义普通类型的临时变量
7. //连续定义的无法识别，比如int a = 0, b = 0;
8. //带作用域的无法识别，比如std::string a;
9. //第一行匹配基础类型
10. //第二行匹配\*&
11. //第三行匹配变量名
12. //第四行匹配[]
13. //1类型，2\*&，3变量名，4[]
14. const QString basicVarStr = "\\b(" + basicType.join("|") + ")"
15. "(\\\*|&?)[^\\S\n]+"
16. "([A-Za-z\_][A-Za-z0-9\_]\*)\\b"
17. "(\\[?\\]?)[^\\S\n]\*[;\\):=,](?!:)";
18. const QRegExp basicVarPattern = QRegExp(basicVarStr);
19. const QString classVarInHeaderStr = "\\b([A-Za-z\_][A-Za-z0-9\_]\*)"
20. "(\\\*|&?)[^\\S\n]+%1\\b"
21. "(\\[?\\]?)[^\\S\n]\*";
22. const QRegExp classVarInHeaderPattern = QRegExp(classVarInHeaderStr);
23. //定义全局变量
24. //1类型，2\*&，3变量名，4[]
25. const QString globalVarStr = "\\b([A-Za-z\_][A-Za-z0-9\_]\*)"
26. "(\\\*|&?)[^\\S\n]+"
27. "([A-Za-z\_][A-Za-z0-9\_]\*)"
28. "(\\[?\\]?)[^\\S\n]\*[=;]";
29. const QRegExp globalVarPattern = QRegExp(globalVarStr);
30. //定义基础和复合变量一起扫描的情况
31. const QString varStr = "\\b([A-Za-z\_][A-Za-z0-9\_]\*)[^\\S\n]\*"
32. "(\\\*|&?)[^\\S\n]\*([A-Za-z\_][A-Za-z0-9\_]\*)";
33. const QRegExp varPattern = QRegExp();
34. QString toLowerCamelCase(const QString &target)
35. {
36. QStringList words = target.split(" ", QString::SkipEmptyParts);
37. if (words.isEmpty()) return "";
38. // Make the first word lowercase
39. QString lowerCamel = words[1].toLower();
40. // Capitalize the first letter of the remaining words
41. for (int i = 1; i < words.size(); ++i) {
42. lowerCamel += words[i].left(1).toUpper() + words[i].mid(1).toLower();
43. }
44. return lowerCamel;
45. }
46. QString toUpperCamelCase(const QString &target)
47. {
48. QStringList words = target.split(" ", QString::SkipEmptyParts);
49. if (words.isEmpty()) return 0;
50. QString upperCamel;
51. // Capitalize the first letter of every word
52. for (const QString &word : words) {
53. upperCamel += word.left(1).toUpper() + word.mid(1).toLower();
54. }
55. return upperCamel;
56. }