**Experiment Report - 33 - test8\_syntaxhighlighter**

1. **Summary Table of Errors Found**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Error ID | Line Number | Error Type | Self-Detected? | Peer 1 Found? | Peer 2 Found? |
| E01 | line 8 | Syntax | √ | √ | × |
| E02 | line 26 | Semantic | √ | × | √ |
| E03 | line 32 | Semantic | √ | × | × |
| E04 | line 39 | Syntax | √ | √ | √ |

Additional Errors Found by Self：1

Self-Review Detection Rate: 100%

Peer 1 Detection Rate: 50%

Peer 2 Detection Rate: 50%

1. **Source Code**
2. #include "syntaxhighlighter.h"
3. SyntaxHighlighter::SyntaxHighlighter(QTextDocument \*parent)
4. : QSyntaxHighlighter(parent)
5. {
6. // 设置关键词的高亮格式
7. keywordFormat.setForeground(Qt::blue);
8. keywordFormat.setFontWeight(QFont::);
9. QStringList keywordPatterns = {
10. "\\bclass\\b", "\\bconst\\b", "\\bvirtual\\b", "\\bint\\b", "\\bfloat\\b", "\\bvoid\\b", "\\bpublic\\b"
11. };
12. foreach (const QString &pattern, keywordPatterns) {
13. HighlightingRule rule;
14. rule.pattern = QRegularExpression(pattern);
15. rule.format = keywordFormat;
16. highlightingRules.append(rule);
17. }
18. // 设置类名的高亮格式
19. classFormat.setFontWeight(QFont::Bold);
20. classFormat.setForeground(Qt::darkMagenta);
21. HighlightingRule classRule;
22. classRule.pattern = QRegularExpression("\\bQ[A-Za-z]+\\b");
23. classRule.format = classFormat;
24. highlightingRules.append(Rule);
25. // 设置函数名的高亮格式
26. functionFormat.setFontItalic(true);
27. functionFormat.setForeground(Qt::darkCyan);
28. HighlightingRule functionRule;
29. functionRule.pattern = QRegularExpression("\\b[A-Za-z0-9\_]+(?=\\");
30. functionRule.format = functionFormat;
31. highlightingRules.append(functionRule);
32. // 设置字符串的高亮格式
33. quotationFormat.setForeground(Qt::darkGreen);
34. HighlightingRule quotationRule;
35. quotationRule.pattern = QRegularExpression("\".\*\"");
36. quotationRule.format = quotationFormat;
37. highlightingRules.append(quotationRule);
38. // 设置单行注释的高亮格式
39. singleLineCommentFormat.setForeground(Qt::red);
40. HighlightingRule commentRule;
41. commentRule.pattern = QRegularExpression("//[^\n]\*");
42. commentRule.format = singleLineCommentFormat;
43. highlightingRules.append(commentRule);
44. }
45. void SyntaxHighlighter::highlightBlock(const QString &text)
46. {
47. foreach (const HighlightingRule &rule, highlightingRules) {
48. QRegularExpressionMatchIterator matchIterator = rule.pattern.globalMatch();
49. while (matchIterator.hasNext()) {
50. QRegularExpressionMatch match = matchIterator.next();
51. setFormat(match.capturedStart(), match.capturedLength(), rule.format);
52. }
53. }
54. }