**Experiment Report - 42 - test7\_guidewizard**

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 60 | Semantic | √ | √ | √ |
| E03 | line 89 | Syntax | √ | √ | √ |
| E04 | line 142 | Logic | × | × | × |

Additional Errors Found by Self: 1

Self-Review Detection Rate: 75%

Peer 1 Detection Rate: 50%

Peer 2 Detection Rate: 50%

1. **Source Code**
2. #include "guidewizard.h"
3. #include <QtWidgets>
4. GuideWizard::GuideWizard(QWidget \*parent)
5. : QWizard(parent)
6. {
7. setPage(Page\_Intro, new IntroPage);
8. setPage(Page\_CreateNewProject, new CreateProjectPage);
9. setPage(Page\_OpenExistingProject, new OpenProjectPage);
10. setPage(Page\_Conclusion, new ConclusionPage);
11. setStartId(Page\_Intro);
12. #ifndef Q\_OS\_MAC
13. setWizardStyle(ModernStyle);
14. #endif
15. setPixmap(QWizard::LogoPixmap, QPixmap(":/images/toolbar\_images/hmpp.png"));
16. setWindowTitle(tr("HMPP Tool Guide"));
17. }
18. OpenProjectPage::OpenProjectPage(QWidget \*parent)
19. : QWizardPage(parent)
20. {
21. setTitle(tr("Open an existing project"));
22. projectPathLabel = new QLabel(tr("Project Path:"));
23. projectPathLineEdit = new QLineEdit;
24. projectPathLineEdit->setReadOnly(true);
25. projectPathLabel->setBuddy(projectPathLineEdit);
26. projectNameLineEdit = new QLineEdit;
27. projectSpecPathLabel = new QLabel(tr("S&pecification Path:"));
28. projectSpecPathEdit = new QLineEdit;
29. projectSpecPathEdit->setReadOnly(true);
30. projectSpecPathLabel->setBuddy(projectSpecPathEdit);
31. targetLanguageLineEdit = new QLineEdit;
32. QPixmap pixmap(":/images/toolbar\_images/folder.png");
33. QPixmap scaledPixmap = pixmap.scaled(16, 16, Qt::KeepAspectRatio); // Adjust the size as needed
34. // Set the icon on the button
35. QIcon icon(scaledPixmap);
36. openFileButton = new QPushButton;
37. openFileButton->setIcon(icon);
38. openFileButton->resize(16, 16);
39. // registerField("project name\*", projectNameLineEdit, "text", SIGNAL(textChanged(QString)));
40. // registerField("project path\*", projectPathLineEdit, "text", SIGNAL(textChanged(QString)));
41. // registerField("target language\*", targetLanguageLineEdit, "text", SIGNAL(textChanged(QString)));
42. targetLanguageLineEdit->setText("C++");
43. connect(openFileButton, &QPushButton::clicked, this, &OpenProjectPage::getProjectDirectory);
44. openSpecButton = new QPushButton;
45. openSpecButton->setIcon(icon);
46. openSpecButton->resize(16);
47. connect(openSpecButton, &QPushButton::clicked, this, &OpenProjectPage::getSpecDirectory);
48. QGridLayout \*layout = new QGridLayout;
49. layout->addWidget(projectPathLabel, 0, 0);
50. layout->addWidget(projectPathLineEdit, 0, 1);
51. layout->addWidget(openFileButton, 0, 2);
52. layout->addWidget(projectSpecPathLabel, 1, 0);
53. layout->addWidget(projectSpecPathEdit, 1, 1);
54. layout->addWidget(openSpecButton, 1, 2);
55. setLayout(layout);
56. }
57. int OpenProjectPage::nextId() const
58. {
59. return GuideWizard::Page\_Conclusion;
60. }
61. bool OpenProjectPage::validatePage()
62. {
63. if (projectPathLineEdit->text().isEmpty()) {
64. QMessageBox::warning(this, "Input Error", "Please select your project file before proceeding.");
65. return false; // Prevent going to the next page
66. }
67. return true;
68. }
69. void OpenProjectPage::getProjectDirectory()
70. {
71. const QString filePath = QFileDialog::getOpenFileName(); //打开文件对话框
72. if (!filePath.isEmpty()){
73. QFileInfo fileInfo(filePath);
74. QString directoryPath = fileInfo.absolutePath();
75. QString fileName = fileInfo.fileName();
76. qDebug() << "path:" << directoryPath;
77. qDebug() << "file:" << fileName;
78. projectPathLineEdit->setText(directoryPath);
79. projectNameLineEdit->setText(fileName);
80. targetLanguageLineEdit->setText("C++");
81. setField("project name", projectNameLineEdit->text());
82. setField("project path", projectPathLineEdit->text());
83. setField("target language", targetLanguageLineEdit->text());
84. }
85. }
86. void OpenProjectPage::getSpecDirectory()
87. {
88. const QString filePath = QFileDialog::getOpenFileName(this); //打开文件对话框
89. if (!filePath.isEmpty()){
90. projectSpecPathEdit->setText(filePath);
91. setField("specification path", projectSpecPathEdit->text());
92. }
93. }
94. ConclusionPage::ConclusionPage(QWidget \*parent)
95. : QWizardPage(parent)
96. {
97. setTitle(tr("Welcome to HMPP tool"));
98. setPixmap(QWizard::WatermarkPixmap, QPixmap(":/images/toolbar\_images/watermark.png"));
99. label = new QLabel(tr("You are going to working on the following project:"));
100. projectName = new QLabel;
101. projectPath = new QLabel;
102. targetLanguage = new QLabel;
103. specPath = new QLabel;
104. QVBoxLayout \*layout = new QVBoxLayout;
105. layout->addWidget(label);
106. layout->addWidget(projectName);
107. layout->addWidget(projectPath);
108. layout->addWidget(specPath);
109. layout->addWidget(targetLanguage);
110. setLayout(layout);
111. }
112. int ConclusionPage::nextId() const
113. {
114. return 1;
115. }