**Министерство цифрового развития связи и массовых коммуникаций Российской Федерации.**

**Ордена Трудового Красного Знамени федеральное государственное бюджетное образовательное учреждение высшего образования**

**"Московский технический университет связи и информатики"**

**Кафедра математическая кибернетика и информационные технологии**

**Отчет по лабораторной работе № 5**

**“Калькулятор”**

**по дисциплине**

**"Введение в ИТ"**

Выполнил: студент гр. БИН2001 Казаков А.А.

Проверил: Аршинов Е.А.

Москва, 2021 г.

1. #Импортируем необходимые библиотеки
2. import sys
3. import sympy
4. from PyQt5.QtWidgets import QApplication, QWidget, QGridLayout, QPushButton, QLineEdit
6. #Создаём главное окно калькулятора, со всеми необходимыми кнопками
7. class Window(QWidget):
8. def \_\_init\_\_(self):
9. super(Window, self).\_\_init\_\_()
10. grid = QGridLayout()
11. self.calcText = QLineEdit()
12. self.setLayout(grid)
13. grid.addWidget(self.calcText, 0, 0, 1, 4)
14. names = ['Clear', 'Back', '(', ')',
15. '7', '8', '9', '/',
16. '4', '5', '6', '\*',
17. '1', '2', '3', '-',
18. '0', '.', '=', '+']
19. positions = [(i + 1, j) for i in range(6) for j in range(4)]
20. buttons = []
21. for position, name in zip(positions, names):
22. button = QPushButton(name)
23. buttons.append(button)
24. grid.addWidget(button, \*position)
25. self.move(300, 150)
26. self.setWindowTitle('Calculator')
27. self.show()
29. #Упрощение введённого выражения с помощью библиотеки simplify и вычисление с помощью
30. # встроенной функции eval
31. def make\_calculate(self):
32. try:
33. return eval(str(sympy.sympify(self.calcText.text(), evaluate=True)))
34. except Exception as e:
35. print(e)
36. return 'Error'
38. #Функция, которая при нажатии вычисляет пример, либо редактирует текст в текстовом
39. # поле
40. def butonact(self, param):
41. nowLine = self.calcText.text()
42. if param in ['7', '8', '9', '/',
43. '4', '5', '6', '\*',
44. '1', '2', '3', '-',
45. '0', '.', '+', '(', ')']:
46. if len(nowLine) > 0:
47. if nowLine[-1] in ['/', '\*', '-', '.', '+'] and param in ['/', '\*', '-', '.', '+', '=']:
48. pass
49. else:
50. self.calcText.setText(nowLine + str(param))
51. else:
52. self.calcText.setText(nowLine + str(param))
53. elif param in ['Clear', 'Back', '=']:
54. if param == 'Clear':
55. self.calcText.setText('')
56. elif param == 'Back':
57. if len(nowLine) > 0:
58. self.calcText.setText(nowLine[:-1])
59. elif param == '=':
60. res = str(self.make\_calculate())
61. self.calcText.setText(res)
63. #Вызов и запуск приложения
64. app = QApplication(sys.argv)
65. win = Window()
66. win.show()
67. sys.exit(app.exec())

На рисунке 1 представлены результаты вычислений, на рисунке 2 результаты при вводе ошибочных данных

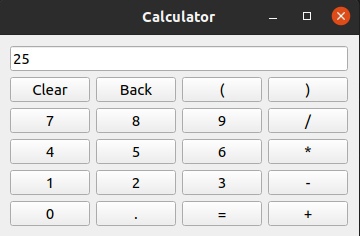
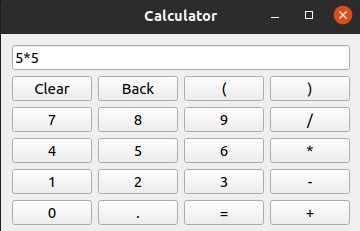


Рисунок 1

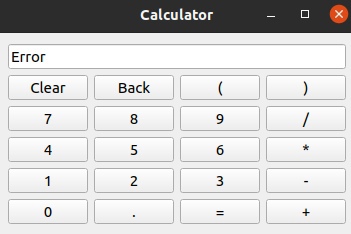
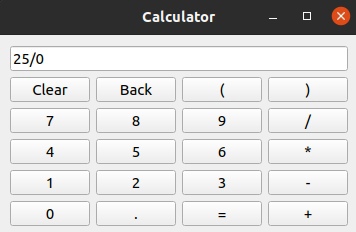


Рисунок 2