**«Московский технический университет связи и информатики»**

**Отчёт для проекта**

**«Калькулятор»**

**По дисциплине**

**«Введение в ИТ»**

Выполнил: студент Безяев Даниил Сергеевич БИН 2002

Москва, 2021 год

**Содержание**

[**1. Цель работы** 3](#_Toc91105105)

[**2. Задачи функций** 3](#_Toc91105106)

[**3. Пример работы** 3](#_Toc91105107)

[**4. Программный код** 5](#_Toc91105108)

# **1. Цель работы**

Создать оконное приложение «Калькулятор». Реализовать с использованием библиотеки PyQt5.

# **2. Задачи функций**

* \_\_init\_\_ – отвечает за переопределение методов родительского класса;
* set\_ui – создание и размещение элементов UI;
* without\_part – обрабатывает входящие числа на наличие незначащей дробной части;
* click\_event... – методы обрабатывающие нажатие на кнопки.

**3. Пример работы**

Окно ошибки (рис. 2) выводится, если при подсчётах происходит ошибка, например, деление на ноль. Реализовано с помощью структуры try except.

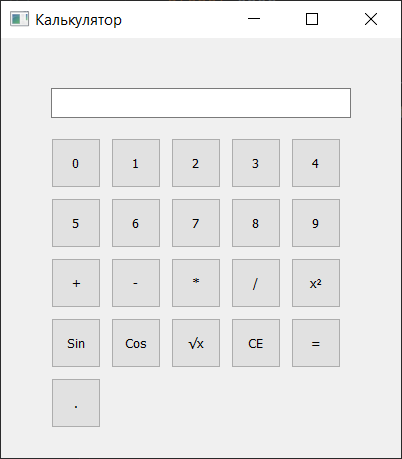


Рисунок 1 – UI главного окна

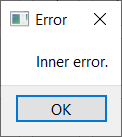


Рисунок 2 – Окно ошибки

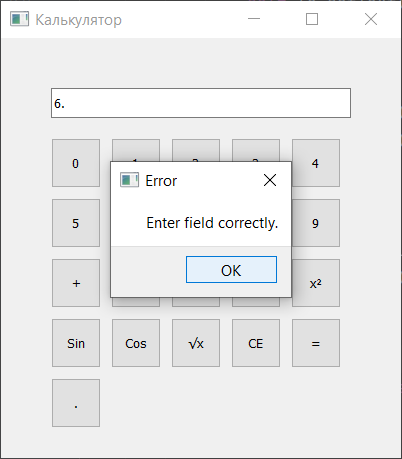


Рисунок 3 – Ошибка при вводе больше чем одной точки дробной части

# **4. Программный код**

from PyQt5.QtWidgets import QApplication, QWidget, QLineEdit, QLabel, QPushButton, QMessageBox  
import sys  
import math  
  
  
class Window(QWidget):  
  
 fst = 0  
 oper = ''  
  
 def \_\_init\_\_(self):  
 super(Window, self).\_\_init\_\_() # переопределение методов родительского класса  
 self.set\_ui()  
  
 def set\_ui(self):  
 self.setGeometry(200, 200, 400, 420)  
 self.setWindowTitle('Калькулятор')  
  
 self.le = QLineEdit(self)  
 self.le.setGeometry(50, 50, 300, 30)  
 self.le.setReadOnly(True)  
  
 self.btn0 = QPushButton(self)  
 self.btn0.setGeometry(50, 100, 50, 50)  
 self.btn0.setText('0')  
 self.btn0.clicked.connect(self.click\_event0)  
  
 self.btn1 = QPushButton(self)  
 self.btn1.setGeometry(110, 100, 50, 50)  
 self.btn1.setText('1')  
 self.btn1.clicked.connect(self.click\_event1)  
  
 self.btn2 = QPushButton(self)  
 self.btn2.setGeometry(170, 100, 50, 50)  
 self.btn2.setText('2')  
 self.btn2.clicked.connect(self.click\_event2)  
  
 self.btn3 = QPushButton(self)  
 self.btn3.setGeometry(230, 100, 50, 50)  
 self.btn3.setText('3')  
 self.btn3.clicked.connect(self.click\_event3)  
  
 self.btn4 = QPushButton(self)  
 self.btn4.setGeometry(290, 100, 50, 50)  
 self.btn4.setText('4')  
 self.btn4.clicked.connect(self.click\_event4)  
  
 self.btn5 = QPushButton(self)  
 self.btn5.setGeometry(50, 160, 50, 50)  
 self.btn5.setText('5')  
 self.btn5.clicked.connect(self.click\_event5)  
  
 self.btn6 = QPushButton(self)  
 self.btn6.setGeometry(110, 160, 50, 50)  
 self.btn6.setText('6')  
 self.btn6.clicked.connect(self.click\_event6)  
  
 self.btn7 = QPushButton(self)  
 self.btn7.setGeometry(170, 160, 50, 50)  
 self.btn7.setText('7')  
 self.btn7.clicked.connect(self.click\_event7)  
  
 self.btn8 = QPushButton(self)  
 self.btn8.setGeometry(230, 160, 50, 50)  
 self.btn8.setText('8')  
 self.btn8.clicked.connect(self.click\_event8)  
  
 self.btn9 = QPushButton(self)  
 self.btn9.setGeometry(290, 160, 50, 50)  
 self.btn9.setText('9')  
 self.btn9.clicked.connect(self.click\_event9)  
  
 self.btnp = QPushButton(self)  
 self.btnp.setGeometry(50, 220, 50, 50)  
 self.btnp.setText('+')  
 self.btnp.clicked.connect(self.click\_eventp)  
  
 self.btnmi = QPushButton(self)  
 self.btnmi.setGeometry(110, 220, 50, 50)  
 self.btnmi.setText('-')  
 self.btnmi.clicked.connect(self.click\_eventmi)  
  
 self.btnmu = QPushButton(self)  
 self.btnmu.setGeometry(170, 220, 50, 50)  
 self.btnmu.setText('\*')  
 self.btnmu.clicked.connect(self.click\_eventmu)  
  
 self.btnd = QPushButton(self)  
 self.btnd.setGeometry(230, 220, 50, 50)  
 self.btnd.setText('/')  
 self.btnd.clicked.connect(self.click\_eventd)  
  
 self.btnsq = QPushButton(self)  
 self.btnsq.setGeometry(290, 220, 50, 50)  
 self.btnsq.setText('x'+ chr(178))  
 self.btnsq.clicked.connect(self.click\_eventsq)  
  
 self.btnsin = QPushButton(self)  
 self.btnsin.setGeometry(50, 280, 50, 50)  
 self.btnsin.setText('Sin')  
 self.btnsin.clicked.connect(self.click\_eventsin)  
  
 self.btncos = QPushButton(self)  
 self.btncos.setGeometry(110, 280, 50, 50)  
 self.btncos.setText('Cos')  
 self.btncos.clicked.connect(self.click\_eventcos)  
  
 self.btnsqrt = QPushButton(self)  
 self.btnsqrt.setGeometry(170, 280, 50, 50)  
 self.btnsqrt.setText(chr(8730) + 'x')  
 self.btnsqrt.clicked.connect(self.click\_eventsqrt)  
  
 self.btnclr = QPushButton(self)  
 self.btnclr.setGeometry(230, 280, 50, 50)  
 self.btnclr.setText('CE')  
 self.btnclr.clicked.connect(self.click\_eventclr)  
  
 self.btneq = QPushButton(self)  
 self.btneq.setGeometry(290, 280, 50, 50)  
 self.btneq.setText('=')  
 self.btneq.clicked.connect(self.click\_eventeq)  
  
 self.btndot = QPushButton(self)  
 self.btndot.setGeometry(50, 340, 50, 50)  
 self.btndot.setText('.')  
 self.btndot.clicked.connect(self.click\_eventdot)  
  
 self.show()  
  
 def without\_part(self, nom):  
 if nom == '' or nom[len(nom)-1] == '.':  
 return nom  
 else:  
 if math.modf(float(nom))[0] == 0.0:  
 return str(math.trunc(float(nom)))  
 else:  
 return nom  
  
 def click\_event0(self):  
 self.le.setText(self.without\_part(self.le.text()) + '0')  
  
 def click\_event1(self):  
 self.le.setText(self.without\_part(self.le.text()) + '1')  
  
 def click\_event2(self):  
 self.le.setText(self.without\_part(self.le.text()) + '2')  
  
 def click\_event3(self):  
 self.le.setText(self.without\_part(self.le.text()) + '3')  
  
 def click\_event4(self):  
 self.le.setText(self.without\_part(self.le.text()) + '4')  
  
 def click\_event5(self):  
 self.le.setText(self.without\_part(self.le.text()) + '5')  
  
 def click\_event6(self):  
 self.le.setText(self.without\_part(self.le.text()) + '6')  
  
 def click\_event7(self):  
 self.le.setText(self.without\_part(self.le.text()) + '7')  
  
 def click\_event8(self):  
 self.le.setText(self.without\_part(self.le.text()) + '8')  
  
 def click\_event9(self):  
 self.le.setText(self.without\_part(self.le.text()) + '9')  
  
 def click\_eventdot(self):  
 if self.le.text().find('.') != -1:  
 QMessageBox.about(self, "Error", "Enter field correctly.")  
 else:  
 self.le.setText(self.le.text() + '.')  
  
 def click\_eventp(self):  
 global fst  
 global oper  
 fst = float(self.le.text())  
 oper = 'p'  
 self.le.setText('')  
  
 def click\_eventmi(self):  
 global fst  
 global oper  
 fst = float(self.le.text())  
 oper = 'mi'  
 self.le.setText('')  
  
 def click\_eventmu(self):  
 global fst  
 global oper  
 fst = float(self.le.text())  
 oper = 'mu'  
 self.le.setText('')  
  
 def click\_eventd(self):  
 global fst  
 global oper  
 fst = float(self.le.text())  
 oper = 'd'  
 self.le.setText('')  
  
 def click\_eventsq(self):  
 global fst  
 global oper  
 fst = 0.0  
 oper = ''  
 self.le.setText(str(float(self.le.text())\*\*2))  
 self.le.setText(self.without\_part(self.le.text()))  
  
 def click\_eventsin(self):  
 global fst  
 global oper  
 fst = 0  
 oper = ''  
 self.le.setText(str(math.sin(int(self.le.text()))))  
 self.le.setText(self.without\_part(self.le.text()))  
  
 def click\_eventcos(self):  
 global fst  
 global oper  
 fst = 0  
 oper = ''  
 self.le.setText(str(math.cos(int(self.le.text()))))  
 self.le.setText(self.without\_part(self.le.text()))  
  
 def click\_eventd(self):  
 global fst  
 global oper  
 fst = float(self.le.text())  
 oper = 'd'  
 self.le.setText('')  
 self.le.setText(self.without\_part(self.le.text()))  
  
 def click\_eventsqrt(self):  
 global fst  
 global oper  
 fst = 0  
 oper = ''  
 self.le.setText(str(math.sqrt(float(self.le.text()))))  
 self.le.setText(self.without\_part(self.le.text()))  
  
 def click\_eventeq(self):  
 global fst  
 global oper  
 second = float(self.le.text())  
 try:  
 if oper == 'p':  
 self.le.setText(str(fst + second))  
 self.le.setText(self.without\_part(self.le.text()))  
 fst = 0  
 oper = ''  
 elif oper == 'mi':  
 self.le.setText(str(fst - second))  
 self.le.setText(self.without\_part(self.le.text()))  
 fst = 0  
 oper = ''  
 elif oper == 'mu':  
 self.le.setText(str(fst \* second))  
 self.le.setText(self.without\_part(self.le.text()))  
 fst = 0  
 oper = ''  
 elif oper == 'd':  
 self.le.setText(str(fst / second))  
 self.le.setText(self.without\_part(self.le.text()))  
 fst = 0  
 oper = ''  
 else:  
 QMessageBox.about(self, "Error", "Inner error.")  
 self.le.setText('')  
 fst = 0  
 oper = ''  
 except:  
 QMessageBox.about(self, "Error", "Inner error.")  
 self.le.setText('')  
 fst = 0  
 oper = ''  
  
 def click\_eventclr(self):  
 global oper  
 global fst  
 oper = ''  
 fst = 0  
 self.le.setText('')  
  
  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
  
 app = QApplication(sys.argv)  
  
 window = Window()  
 sys.exit(app.exec\_())