**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»**

**Лабораторная работа №3**

**«Создание калькулятора»**

**По дисциплине**

**«Введение в ИТ»**

Выполнил: студент группы

БИН2002 Дёмин П.Д.

Проверил: Чайка А.Д.

Москва 2021 г.

Оглавление

[**1.** **Цель работы** 3](#_Toc89954318)

[**2.** **Выполнение** 3](#_Toc89954319)

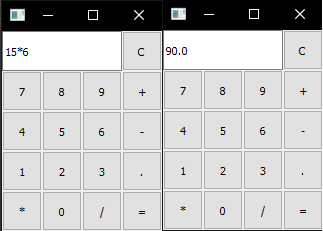
[**3.** **Вывод** 4](#_Toc89954320)

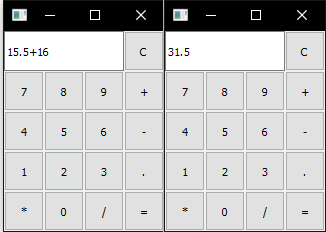
1. **Цель работы**

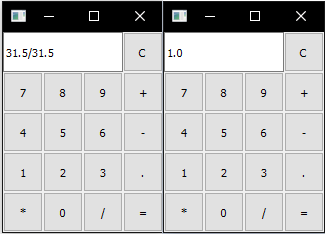
Создание калькулятора с использованием Python и библиотеки PyQt5

1. **Выполнение**

Калькулятор может выполнять все требуемые функции (рисунок 1)

**





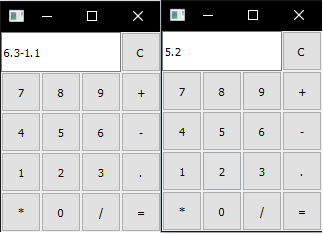


Рисунок 1 - проверка работы калькулятора

При попытке деления на ноль приложение выдает ошибку (рисунок 2)

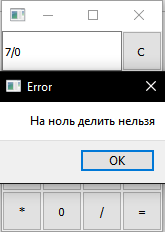


Рисунок 2 - попытка деления на 0

1. **Вывод**

В ходе выполнения работы, я научился созданию оконных приложений с помощью PyQt

**Приложение:**

**Программный код файла main.py**

from PyQt5.QtWidgets import QApplication, QWidget, QLineEdit, QPushButton, QMessageBox  
import sys  
  
calc = 0  
pd = 0  
numb = ''  
  
  
class Window(QWidget):  
 def prov(self):  
 global calc, pd, numb  
 if pd == 0:  
 calc += float(numb)  
 elif pd == 1:  
 calc += float(numb)  
 elif pd == 2:  
 calc -= float(numb)  
 elif pd == 3:  
 calc \*= float(numb)  
 elif pd == 4:  
 try:  
 calc /= float(numb)  
 except:  
 QMessageBox.about(self, "Error", "На ноль делить нельзя")  
 numb = ''  
  
 def \_\_init\_\_(self):  
 super(Window, self).\_\_init\_\_()  
 self.le = QLineEdit(self)  
 self.set\_ui()  
  
 def set\_ui(self):  
 self.setGeometry(200, 200, 160, 200)  
 self.setWindowTitle('Калькулятор')  
 self.le.setGeometry(0, 0, 120, 40)  
  
 for i in range(10):  
 self.btn = QPushButton(str(i), self)  
 if i == 0:  
 self.btn.setGeometry(40, 160, 40, 40)  
 elif i <= 3:  
 self.btn.setGeometry(40 \* (i - 1), 120, 40, 40)  
 elif i <= 6:  
 self.btn.setGeometry(40 \* (i - 4), 80, 40, 40)  
 elif i <= 9:  
 self.btn.setGeometry(40 \* (i - 7), 40, 40, 40)  
 self.btn.clicked.connect(lambda checked=None, j=i: self.click\_event(j))  
  
 self.btne = QPushButton('=', self)  
 self.btne.setGeometry(120, 160, 40, 40)  
 self.btne.clicked.connect(self.click\_event\_e)  
  
 self.btnp = QPushButton('+', self)  
 self.btnp.setGeometry(120, 40, 40, 40)  
 self.btnp.clicked.connect(self.click\_event\_p)  
  
 self.btnm = QPushButton('-', self)  
 self.btnm.setGeometry(120, 80, 40, 40)  
 self.btnm.clicked.connect(self.click\_event\_m)  
  
 self.btny = QPushButton('\*', self)  
 self.btny.setGeometry(0, 160, 40, 40)  
 self.btny.clicked.connect(self.click\_event\_y)  
  
 self.btnd = QPushButton('/', self)  
 self.btnd.setGeometry(80, 160, 40, 40)  
 self.btnd.clicked.connect(self.click\_event\_d)  
  
 self.btnc = QPushButton('C', self)  
 self.btnc.setGeometry(120, 0, 40, 40)  
 self.btnc.clicked.connect(self.click\_event\_c)  
  
 self.btnt = QPushButton('.', self)  
 self.btnt.setGeometry(120, 120, 40, 40)  
 self.btnt.clicked.connect(self.click\_event\_t)  
  
 self.show()  
  
 def click\_event(self, j):  
 global numb  
 self.le.setText(self.le.text() + str(j))  
 numb += str(j)  
  
 def click\_event\_t(self):  
 global numb  
 self.le.setText(self.le.text() + '.')  
 numb += '.'  
  
 def click\_event\_e(self):  
 global pd, numb, calc  
 self.prov()  
 self.le.setText(str(calc))  
 pd = 0  
 numb = str(calc)  
 calc = 0  
  
 def click\_event\_p(self):  
 global pd  
 self.prov()  
 self.le.setText(self.le.text() + '+')  
 pd = 1  
  
 def click\_event\_m(self):  
 global pd  
 self.prov()  
 self.le.setText(self.le.text() + '-')  
 pd = 2  
  
 def click\_event\_y(self):  
 global pd  
 self.prov()  
 self.le.setText(self.le.text() + '\*')  
 pd = 3  
  
 def click\_event\_d(self):  
 global pd  
 self.prov()  
 self.le.setText(self.le.text() + '/')  
 pd = 4  
  
 def click\_event\_c(self):  
 global pd, numb, calc  
 self.le.setText('')  
 calc = 0  
 pd = 0  
 numb = ''  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 app = QApplication(sys.argv)  
  
 window = Window()  
 sys.exit(app.exec\_())