|  |
| --- |
|  |
| МИНОБРНАУКИ РОССИИ |
| Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования  **«Российский технологический университет - МИРЭА»** |
| **РТУ МИРЭА** |

Институт информационных технологий (ИИТ)Кафедра вычислительной техники (ВТ)

**ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ №5**

**по дисциплине**

«Инструментальное программное обеспечение разработки и проектирования информационных систем»

Выполнил студент группы ИКМО-01-20 Баранецкий Д.А.

Фамилия И.О.

Принял Куликов А.А.

Фамилия И.О.

Работы выполнены «\_\_»\_\_\_\_\_\_\_2021 г. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись студента)

«Зачтено» «\_\_»\_\_\_\_\_\_\_2021 г. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись руководителя)

Москва, 2021 г.

Оглавление

[Задание 3](#_Toc71570186)

[Код программы 3](#_Toc71570187)

[Скриншоты 12](#_Toc71570188)

# Задание

**Цель практическое работы:**

Создать умный калькулятор с использование языка Python и QT Designer

**Задачи:**

* Реализовать стандартный набор функций калькулятора:
  + Сложение
  + Вычитание
  + Умножение
  + Деление
  + Возведение в степень
* Реализовать дополнительные функции расчёта: работа со скобками и не целыми числами
* Реализовать основные функции действия:
  + Удалить символ
  + Отчистить строку
  + Выйти
  + Рассчитать
* Создать форму в QT Designer

# Код программы

calculator\_interface.py – файл, формирующий QT-форму калькулятора

# -\*- coding: utf-8 -\*-  
  
# Form implementation generated from reading ui file 'calcUIRU.ui'  
#  
# Created by: PyQt5 UI code generator 5.15.0  
#  
# WARNING: Any manual changes made to this file will be lost when pyuic5 is  
# run again. Do not edit this file unless you know what you are doing.  
  
  
from PyQt5 import QtCore, QtGui, QtWidgets  
  
  
class Ui\_Calculator(object):  
 def setupUi(self, Calculator):  
 Calculator.setObjectName("Calculator")  
 Calculator.resize(472, 316)  
 self.centralwidget = QtWidgets.QWidget(Calculator)  
 self.centralwidget.setObjectName("centralwidget")  
  
 self.verticalLayout\_5 = QtWidgets.QVBoxLayout(self.centralwidget)  
 self.verticalLayout\_5.setObjectName("verticalLayout\_5")  
 self.verticalLayout\_4 = QtWidgets.QVBoxLayout()  
 self.verticalLayout\_4.setObjectName("verticalLayout\_4")  
 self.verticalLayout\_3 = QtWidgets.QVBoxLayout()  
 self.verticalLayout\_3.setObjectName("verticalLayout\_3")  
 self.horizontalLayout = QtWidgets.QHBoxLayout()  
 self.horizontalLayout.setObjectName("horizontalLayout")  
  
 self.TitleLabel = QtWidgets.QLabel(self.centralwidget)  
 font = QtGui.QFont()  
 font.setFamily("Arial")  
 font.setPointSize(16)  
 self.TitleLabel.setFont(font)  
 self.TitleLabel.setObjectName("TitleLabel")  
 self.horizontalLayout.addWidget(self.TitleLabel)  
 spacerItem = QtWidgets.QSpacerItem(138, 20, QtWidgets.QSizePolicy.Expanding, QtWidgets.QSizePolicy.Minimum)  
 self.horizontalLayout.addItem(spacerItem)  
 self.verticalLayout\_3.addLayout(self.horizontalLayout)  
 self.line = QtWidgets.QFrame(self.centralwidget)  
 self.line.setFrameShape(QtWidgets.QFrame.HLine)  
 self.line.setFrameShadow(QtWidgets.QFrame.Sunken)  
 self.line.setObjectName("line")  
 self.verticalLayout\_3.addWidget(self.line)  
 self.verticalLayout\_4.addLayout(self.verticalLayout\_3)  
 self.verticalLayout\_2 = QtWidgets.QVBoxLayout()  
 self.verticalLayout\_2.setObjectName("verticalLayout\_2")  
  
 self.ExpressionLineEdit = QtWidgets.QLineEdit(self.centralwidget)  
 font = QtGui.QFont()  
 font.setFamily("Arial")  
 font.setPointSize(16)  
 self.ExpressionLineEdit.setFont(font)  
 self.ExpressionLineEdit.setText("")  
 self.ExpressionLineEdit.setAlignment(QtCore.Qt.AlignRight|QtCore.Qt.AlignTrailing|QtCore.Qt.AlignVCenter)  
 self.ExpressionLineEdit.setReadOnly(True)  
 self.ExpressionLineEdit.setObjectName("ExpressionLineEdit")  
 self.verticalLayout\_2.addWidget(self.ExpressionLineEdit)  
  
 self.ResultLineEdit = QtWidgets.QLineEdit(self.centralwidget)  
 font = QtGui.QFont()  
 font.setFamily("Arial")  
 font.setPointSize(16)  
 self.ResultLineEdit.setFont(font)  
 self.ResultLineEdit.setAlignment(QtCore.Qt.AlignRight|QtCore.Qt.AlignTrailing|QtCore.Qt.AlignVCenter)  
 self.ResultLineEdit.setReadOnly(True)  
 self.ResultLineEdit.setObjectName("ResultLineEdit")  
 self.verticalLayout\_2.addWidget(self.ResultLineEdit)  
 self.ButtonsGroupBox = QtWidgets.QGroupBox(self.centralwidget)  
 self.ButtonsGroupBox.setTitle("")  
 self.ButtonsGroupBox.setObjectName("ButtonsGroupBox")  
  
 self.verticalLayout = QtWidgets.QVBoxLayout(self.ButtonsGroupBox)  
 self.verticalLayout.setObjectName("verticalLayout")  
 self.ButtonsGridLayout = QtWidgets.QGridLayout()  
 self.ButtonsGridLayout.setObjectName("ButtonsGridLayout")  
  
 self.Exp\_Button = QtWidgets.QPushButton(self.ButtonsGroupBox)  
 font = QtGui.QFont()  
 font.setFamily("Arial")  
 font.setPointSize(16)  
 self.Exp\_Button.setFont(font)  
 self.Exp\_Button.setObjectName("Exp\_Button")  
 self.ButtonsGridLayout.addWidget(self.Exp\_Button, 0, 2, 1, 1)  
  
 self.Del\_Button = QtWidgets.QPushButton(self.ButtonsGroupBox)  
 font = QtGui.QFont()  
 font.setFamily("Arial")  
 font.setPointSize(16)  
 self.Del\_Button.setFont(font)  
 self.Del\_Button.setObjectName("Del\_Button")  
 self.ButtonsGridLayout.addWidget(self.Del\_Button, 0, 5, 1, 1)  
  
 self.Nine\_Button = QtWidgets.QPushButton(self.ButtonsGroupBox)  
 font = QtGui.QFont()  
 font.setFamily("Arial")  
 font.setPointSize(16)  
 self.Nine\_Button.setFont(font)  
 self.Nine\_Button.setObjectName("Nine\_Button")  
 self.ButtonsGridLayout.addWidget(self.Nine\_Button, 1, 2, 1, 1)  
  
 self.Mult\_Button = QtWidgets.QPushButton(self.ButtonsGroupBox)  
 font = QtGui.QFont()  
 font.setFamily("Arial")  
 font.setPointSize(16)  
 self.Mult\_Button.setFont(font)  
 self.Mult\_Button.setObjectName("Mult\_Button")  
 self.ButtonsGridLayout.addWidget(self.Mult\_Button, 1, 3, 1, 1)  
  
 self.Zero\_Button = QtWidgets.QPushButton(self.ButtonsGroupBox)  
 font = QtGui.QFont()  
 font.setFamily("Arial")  
 font.setPointSize(16)  
 self.Zero\_Button.setFont(font)  
 self.Zero\_Button.setObjectName("Zero\_Button")  
 self.ButtonsGridLayout.addWidget(self.Zero\_Button, 4, 1, 1, 1)  
  
 self.RightBracket\_Button = QtWidgets.QPushButton(self.ButtonsGroupBox)  
 font = QtGui.QFont()  
 font.setFamily("Arial")  
 font.setPointSize(16)  
 self.RightBracket\_Button.setFont(font)  
 self.RightBracket\_Button.setObjectName("RightBracket\_Button")  
 self.ButtonsGridLayout.addWidget(self.RightBracket\_Button, 0, 1, 1, 1)  
  
 self.Div\_Button = QtWidgets.QPushButton(self.ButtonsGroupBox)  
 font = QtGui.QFont()  
 font.setFamily("Arial")  
 font.setPointSize(16)  
 self.Div\_Button.setFont(font)  
 self.Div\_Button.setObjectName("Div\_Button")  
 self.ButtonsGridLayout.addWidget(self.Div\_Button, 0, 3, 1, 1)  
  
 self.Four\_Button = QtWidgets.QPushButton(self.ButtonsGroupBox)  
 font = QtGui.QFont()  
 font.setFamily("Arial")  
 font.setPointSize(16)  
 self.Four\_Button.setFont(font)  
 self.Four\_Button.setObjectName("Four\_Button")  
 self.ButtonsGridLayout.addWidget(self.Four\_Button, 2, 0, 1, 1)  
  
 self.Three\_Button = QtWidgets.QPushButton(self.ButtonsGroupBox)  
 font = QtGui.QFont()  
 font.setFamily("Arial")  
 font.setPointSize(16)  
 self.Three\_Button.setFont(font)  
 self.Three\_Button.setObjectName("Three\_Button")  
 self.ButtonsGridLayout.addWidget(self.Three\_Button, 3, 2, 1, 1)  
  
 self.Minus\_Button = QtWidgets.QPushButton(self.ButtonsGroupBox)  
 font = QtGui.QFont()  
 font.setFamily("Arial")  
 font.setPointSize(16)  
 self.Minus\_Button.setFont(font)  
 self.Minus\_Button.setObjectName("Minus\_Button")  
 self.ButtonsGridLayout.addWidget(self.Minus\_Button, 2, 3, 1, 1)  
  
 self.One\_Button = QtWidgets.QPushButton(self.ButtonsGroupBox)  
 font = QtGui.QFont()  
 font.setFamily("Arial")  
 font.setPointSize(16)  
 self.One\_Button.setFont(font)  
 self.One\_Button.setObjectName("One\_Button")  
 self.ButtonsGridLayout.addWidget(self.One\_Button, 3, 0, 1, 1)  
  
 self.Two\_Button = QtWidgets.QPushButton(self.ButtonsGroupBox)  
 font = QtGui.QFont()  
 font.setFamily("Arial")  
 font.setPointSize(16)  
 self.Two\_Button.setFont(font)  
 self.Two\_Button.setObjectName("Two\_Button")  
 self.ButtonsGridLayout.addWidget(self.Two\_Button, 3, 1, 1, 1)  
  
 self.Eight\_Button = QtWidgets.QPushButton(self.ButtonsGroupBox)  
 font = QtGui.QFont()  
 font.setFamily("Arial")  
 font.setPointSize(16)  
 self.Eight\_Button.setFont(font)  
 self.Eight\_Button.setObjectName("Eight\_Button")  
 self.ButtonsGridLayout.addWidget(self.Eight\_Button, 1, 1, 1, 1)  
  
 self.Seven\_Button = QtWidgets.QPushButton(self.ButtonsGroupBox)  
 font = QtGui.QFont()  
 font.setFamily("Arial")  
 font.setPointSize(16)  
 self.Seven\_Button.setFont(font)  
 self.Seven\_Button.setObjectName("Seven\_Button")  
 self.ButtonsGridLayout.addWidget(self.Seven\_Button, 1, 0, 1, 1)  
  
 self.Dot\_Button = QtWidgets.QPushButton(self.ButtonsGroupBox)  
 font = QtGui.QFont()  
 font.setFamily("Arial")  
 font.setPointSize(16)  
 self.Dot\_Button.setFont(font)  
 self.Dot\_Button.setObjectName("Dot\_Button")  
 self.ButtonsGridLayout.addWidget(self.Dot\_Button, 4, 2, 1, 1)  
  
 self.Plus\_Button = QtWidgets.QPushButton(self.ButtonsGroupBox)  
 font = QtGui.QFont()  
 font.setFamily("Arial")  
 font.setPointSize(16)  
 self.Plus\_Button.setFont(font)  
 self.Plus\_Button.setObjectName("Plus\_Button")  
 self.ButtonsGridLayout.addWidget(self.Plus\_Button, 3, 3, 1, 1)  
  
 self.Equal\_Button = QtWidgets.QPushButton(self.ButtonsGroupBox)  
 font = QtGui.QFont()  
 font.setFamily("Arial")  
 font.setPointSize(16)  
 self.Equal\_Button.setFont(font)  
 self.Equal\_Button.setObjectName("Equal\_Button")  
 self.ButtonsGridLayout.addWidget(self.Equal\_Button, 4, 3, 1, 1)  
  
 self.Five\_Button = QtWidgets.QPushButton(self.ButtonsGroupBox)  
 font = QtGui.QFont()  
 font.setFamily("Arial")  
 font.setPointSize(16)  
 self.Five\_Button.setFont(font)  
 self.Five\_Button.setObjectName("Five\_Button")  
 self.ButtonsGridLayout.addWidget(self.Five\_Button, 2, 1, 1, 1)  
  
 self.Six\_Button = QtWidgets.QPushButton(self.ButtonsGroupBox)  
 font = QtGui.QFont()  
 font.setFamily("Arial")  
 font.setPointSize(16)  
 self.Six\_Button.setFont(font)  
 self.Six\_Button.setObjectName("Six\_Button")  
 self.ButtonsGridLayout.addWidget(self.Six\_Button, 2, 2, 1, 1)  
  
 self.LeftBracket\_Button = QtWidgets.QPushButton(self.ButtonsGroupBox)  
 font = QtGui.QFont()  
 font.setFamily("Arial")  
 font.setPointSize(16)  
 self.LeftBracket\_Button.setFont(font)  
 self.LeftBracket\_Button.setObjectName("LeftBracket\_Button")  
 self.ButtonsGridLayout.addWidget(self.LeftBracket\_Button, 0, 0, 1, 1)  
  
 self.Exit\_Button = QtWidgets.QPushButton(self.ButtonsGroupBox)  
 font = QtGui.QFont()  
 font.setFamily("Arial")  
 font.setPointSize(16)  
 self.Exit\_Button.setFont(font)  
 self.Exit\_Button.setObjectName("Exit\_Button")  
 self.ButtonsGridLayout.addWidget(self.Exit\_Button, 4, 0, 1, 1)  
  
 self.Clear\_Button = QtWidgets.QPushButton(self.ButtonsGroupBox)  
 font = QtGui.QFont()  
 font.setFamily("Arial")  
 font.setPointSize(16)  
 self.Clear\_Button.setFont(font)  
 self.Clear\_Button.setObjectName("Clear\_Button")  
 self.ButtonsGridLayout.addWidget(self.Clear\_Button, 1, 5, 1, 1)  
  
 self.verticalLayout.addLayout(self.ButtonsGridLayout)  
 self.verticalLayout\_2.addWidget(self.ButtonsGroupBox)  
 self.verticalLayout\_4.addLayout(self.verticalLayout\_2)  
 self.verticalLayout\_5.addLayout(self.verticalLayout\_4)  
 Calculator.setCentralWidget(self.centralwidget)  
  
 self.retranslateUi(Calculator)  
 QtCore.QMetaObject.connectSlotsByName(Calculator)  
  
 def retranslateUi(self, Calculator):  
 \_translate = QtCore.QCoreApplication.translate  
 Calculator.setWindowTitle(\_translate("Calculator", "Calculator"))  
 self.TitleLabel.setText(\_translate("Calculator", "Калькулятор"))  
 self.ExpressionLineEdit.setPlaceholderText(\_translate("Calculator", "Введите выражение"))  
 self.ResultLineEdit.setPlaceholderText(\_translate("Calculator", "Результат"))  
 self.Exp\_Button.setText(\_translate("Calculator", "^"))  
 self.Del\_Button.setText(\_translate("Calculator", "Del"))  
 self.Nine\_Button.setText(\_translate("Calculator", "9"))  
 self.Mult\_Button.setText(\_translate("Calculator", "\*"))  
 self.Zero\_Button.setText(\_translate("Calculator", "0"))  
 self.RightBracket\_Button.setText(\_translate("Calculator", ")"))  
 self.Div\_Button.setText(\_translate("Calculator", "/"))  
 self.Four\_Button.setText(\_translate("Calculator", "4"))  
 self.Three\_Button.setText(\_translate("Calculator", "3"))  
 self.Minus\_Button.setText(\_translate("Calculator", "-"))  
 self.One\_Button.setText(\_translate("Calculator", "1"))  
 self.Two\_Button.setText(\_translate("Calculator", "2"))  
 self.Eight\_Button.setText(\_translate("Calculator", "8"))  
 self.Seven\_Button.setText(\_translate("Calculator", "7"))  
 self.Dot\_Button.setText(\_translate("Calculator", "."))  
 self.Plus\_Button.setText(\_translate("Calculator", "+"))  
 self.Equal\_Button.setText(\_translate("Calculator", "="))  
 self.Five\_Button.setText(\_translate("Calculator", "5"))  
 self.Six\_Button.setText(\_translate("Calculator", "6"))  
 self.LeftBracket\_Button.setText(\_translate("Calculator", "("))  
 self.Exit\_Button.setText(\_translate("Calculator", "Exit"))  
 self.Clear\_Button.setText(\_translate("Calculator", "C"))  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 import sys  
 app = QtWidgets.QApplication(sys.argv)  
 Calculator = QtWidgets.QMainWindow()  
 ui = Ui\_Calculator()  
 ui.setupUi(Calculator)  
 Calculator.show()  
 sys.exit(app.exec\_())

main.py – файл, содержащий логику калькулятора с привязкой к QT-форме

from PyQt5 import QtWidgets  
import sys  
import calcUIRU  
import re  
  
  
class Calc(QtWidgets.QMainWindow, calcUIRU.Ui\_Calculator):  
 def \_\_init\_\_(self):  
 super().\_\_init\_\_()  
 self.setupUi(self)  
 self.setFocus()  
 self.msg\_box = QtWidgets.QMessageBox()  
 self.msg\_box.setWindowTitle(self.windowTitle())  
 self.stack\_of\_numbers = []  
 self.stack\_of\_operations = []  
 self.operations\_priority = {"+": 1, "-": 1, "\*": 2, "/": 2, "^": 3, "(": 0, ")": 0}  
 self.operations = {"+": self.sum, "-": self.sub, "\*": self.mul, "/": self.div, "^": self.exp}  
   
 self.Zero\_Button.clicked.connect(lambda: self.ExpressionLineEdit.insert("0"))  
 self.Dot\_Button.clicked.connect(lambda: self.ExpressionLineEdit.insert("."))  
 self.Equal\_Button.clicked.connect(lambda: self.calculate(self.ExpressionLineEdit.text()))  
 self.Plus\_Button.clicked.connect(lambda: self.ExpressionLineEdit.insert("+"))  
 self.Minus\_Button.clicked.connect(lambda: self.ExpressionLineEdit.insert("-"))  
 self.One\_Button.clicked.connect(lambda: self.ExpressionLineEdit.insert("1"))  
 self.Two\_Button.clicked.connect(lambda: self.ExpressionLineEdit.insert("2"))  
 self.Three\_Button.clicked.connect(lambda: self.ExpressionLineEdit.insert("3"))  
 self.Mult\_Button.clicked.connect(lambda: self.ExpressionLineEdit.insert("\*"))  
 self.Div\_Button.clicked.connect(lambda: self.ExpressionLineEdit.insert("/"))  
 self.Four\_Button.clicked.connect(lambda: self.ExpressionLineEdit.insert("4"))  
 self.Five\_Button.clicked.connect(lambda: self.ExpressionLineEdit.insert("5"))  
 self.Six\_Button.clicked.connect(lambda: self.ExpressionLineEdit.insert("6"))  
 self.LeftBracket\_Button.clicked.connect(lambda: self.ExpressionLineEdit.insert("("))  
 self.RightBracket\_Button.clicked.connect(lambda: self.ExpressionLineEdit.insert(")"))  
 self.Seven\_Button.clicked.connect(lambda: self.ExpressionLineEdit.insert("7"))  
 self.Eight\_Button.clicked.connect(lambda: self.ExpressionLineEdit.insert("8"))  
 self.Nine\_Button.clicked.connect(lambda: self.ExpressionLineEdit.insert("9"))  
 self.Exp\_Button.clicked.connect(lambda: self.ExpressionLineEdit.insert("^"))  
 self.Del\_Button.clicked.connect(lambda: self.ExpressionLineEdit.backspace())  
 self.Clear\_Button.clicked.connect(lambda: self.ExpressionLineEdit.clear())  
 self.Exit\_Button.clicked.connect(self.close)  
   
 self.Zero\_Button.setShortcut("0")  
 self.Dot\_Button.setShortcut(".")  
 self.Equal\_Button.setShortcut("=")  
 self.Plus\_Button.setShortcut("+")  
 self.Minus\_Button.setShortcut("-")  
 self.One\_Button.setShortcut("1")  
 self.Two\_Button.setShortcut("2")  
 self.Three\_Button.setShortcut("3")  
 self.Mult\_Button.setShortcut("\*")  
 self.Div\_Button.setShortcut("/")  
 self.Four\_Button.setShortcut("4")  
 self.Five\_Button.setShortcut("5")  
 self.Six\_Button.setShortcut("6")  
 self.LeftBracket\_Button.setShortcut("(")  
 self.RightBracket\_Button.setShortcut(")")  
 self.Seven\_Button.setShortcut("7")  
 self.Eight\_Button.setShortcut("8")  
 self.Nine\_Button.setShortcut("9")  
 self.Exp\_Button.setShortcut("^")  
 self.Del\_Button.setShortcut("Backspace")  
 self.Clear\_Button.setShortcut("CTRL+C")  
 self.Exit\_Button.setShortcut("ESC")  
  
 def sum(self) -> None:  
 b = self.stack\_of\_numbers.pop()  
 a = self.stack\_of\_numbers.pop()  
 self.stack\_of\_numbers.append(float(a) + float(b))  
 if self.get\_top\_of\_the\_stack(self.stack\_of\_operations) == "+":  
 self.stack\_of\_operations.pop()  
  
 def sub(self) -> None:  
 b = self.stack\_of\_numbers.pop()  
 a = self.stack\_of\_numbers.pop()  
 self.stack\_of\_numbers.append(float(a) - float(b))  
 if self.get\_top\_of\_the\_stack(self.stack\_of\_operations) == "-":  
 self.stack\_of\_operations.pop()  
  
 def mul(self) -> None:  
 b = self.stack\_of\_numbers.pop()  
 a = self.stack\_of\_numbers.pop()  
 self.stack\_of\_numbers.append(float(a) \* float(b))  
 if self.get\_top\_of\_the\_stack(self.stack\_of\_operations) == "\*":  
 self.stack\_of\_operations.pop()  
  
 def div(self) -> None:  
 b = self.stack\_of\_numbers.pop()  
 a = self.stack\_of\_numbers.pop()  
 self.stack\_of\_numbers.append(float(a) / float(b))  
 if self.get\_top\_of\_the\_stack(self.stack\_of\_operations) == "/":  
 self.stack\_of\_operations.pop()  
  
 def exp(self) -> None:  
 b = self.stack\_of\_numbers.pop()  
 a = self.stack\_of\_numbers.pop()  
 self.stack\_of\_numbers.append(float(a) \*\* float(b))  
 if self.get\_top\_of\_the\_stack(self.stack\_of\_operations) == "^":  
 self.stack\_of\_operations.pop()  
  
 @staticmethod  
 def get\_top\_of\_the\_stack(stack: list) -> str:  
 return stack[len(stack) - 1]  
  
 @staticmethod  
 def get\_tokens(expression: str) -> list:  
 raw\_list = re.split(r"(\D)", expression)  
 tokens\_list = []  
 for i in raw\_list:  
 if i != " " and i != "":  
 if i == ".":  
 tokens\_list[len(tokens\_list) - 1] += i  
 tokens\_list.append(i)  
 elif len(tokens\_list) > 0 and tokens\_list[len(tokens\_list) - 1] == ".":  
 tokens\_list.pop()  
 tokens\_list[len(tokens\_list) - 1] += i  
 else:  
 tokens\_list.append(i)  
 return tokens\_list  
  
 def calculate(self, expression: str) -> None:  
 self.stack\_of\_numbers.clear()  
 self.stack\_of\_operations.clear()  
 tokens = self.get\_tokens(expression)  
 try:  
 for token in tokens:  
 if len(self.stack\_of\_numbers) == 0 and token in self.operations\_priority and \  
 (token != '(' and token != ')'):  
 self.stack\_of\_numbers.append(float(0))  
 if tokens.index(token) == len(tokens) - 1 and token in self.operations\_priority and \  
 (token != '(' and token != ')'):  
 self.stack\_of\_numbers.append(float(0))  
 if token not in self.operations\_priority:  
 self.stack\_of\_numbers.append(float(token))  
 elif len(self.stack\_of\_operations) > 0 and (token != '(' and token != ')'):  
 while (len(self.stack\_of\_operations) != 0 and self.operations\_priority[token] <=  
 self.operations\_priority[self.get\_top\_of\_the\_stack(self.stack\_of\_operations)]):  
 self.operations[self.get\_top\_of\_the\_stack(self.stack\_of\_operations)]()  
 self.stack\_of\_operations.append(token)  
 elif token == ')':  
 self.operations[self.get\_top\_of\_the\_stack(self.stack\_of\_operations)]()  
 self.stack\_of\_operations.reverse()  
 for i in self.stack\_of\_operations:  
 if i == '(':  
 self.stack\_of\_operations.remove(i)  
 break  
 self.stack\_of\_operations.reverse()  
 else:  
 self.stack\_of\_operations.append(token)  
 while len(self.stack\_of\_operations) != 0:  
 self.operations[self.get\_top\_of\_the\_stack(self.stack\_of\_operations)]()  
 except (KeyError, IndexError, ValueError, ZeroDivisionError):  
 self.msg\_box.setText("Ошибка выполнения выражения")  
 self.msg\_box.exec()  
 else:  
 if len(self.stack\_of\_numbers) != 0:  
 self.ResultLineEdit.setText(str(self.stack\_of\_numbers[0]))  
  
  
def main() -> None:  
 app = QtWidgets.QApplication(sys.argv)  
 window = Calc()  
 window.show()  
 sys.exit(app.exec\_())  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 main()

# Скриншоты

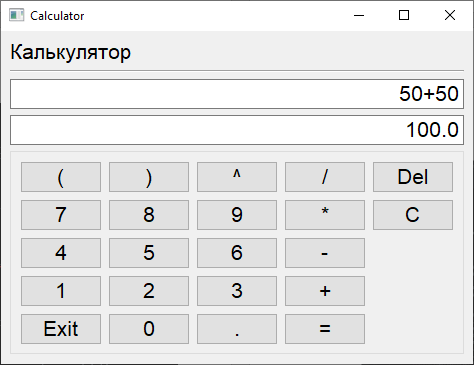


Рис.1. – Интерфейс программы и пример отработки

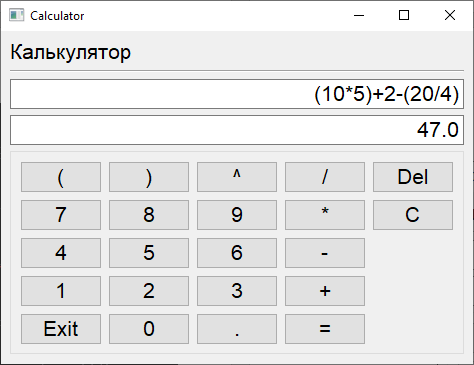


Рис.2. – Пример отработки более сложной расчётной функции