**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 математической кибернетики и информационных технологий

**Отчет по практической работе №4**

на тему «Калькулятор»

по дисциплине «Введение в профессию»

Выполнил: студент группы БВТ2105

Югай Михаил Константинович

Проверил:

Мкртчян Грач Маратович

Москва, 2021

Цель и задачи работы

**Цель**: ознакомиться с библиотекой PyQt5

**Задача:** написать код для калькулятора на языке Python

1. Добавить кнопки с цифрами (0 – 9)
2. Добавить простейшие математические операции
3. Добавить кнопку сброса (C)
4. Добавить исключение на пустой ввод
5. Добавить исключение на деление на 0

Выполнение

Листинг кода файла calculator.py:

import sys  
from PyQt5.QtWidgets import QApplication, QWidget, QLineEdit, QHBoxLayout, QVBoxLayout, QPushButton  
  
class Calculator(QWidget):  
 def \_\_init\_\_(self):  
 super(Calculator, self).\_\_init\_\_()  
  
 self.vbox = QVBoxLayout(self)  
 self.hbox\_input = QHBoxLayout()  
 self.hbox\_first = QHBoxLayout()  
 self.hbox\_second = QHBoxLayout()  
 self.hbox\_result = QHBoxLayout()  
  
 self.vbox.addLayout(self.hbox\_input)  
 self.vbox.addLayout(self.hbox\_first)  
 self.vbox.addLayout(self.hbox\_second)  
 self.vbox.addLayout(self.hbox\_result)  
  
 self.input = QLineEdit(self)  
 self.hbox\_input.addWidget(self.input)  
  
 self.b\_1 = QPushButton("1", self)  
 self.hbox\_first.addWidget(self.b\_1)  
 self.b\_2 = QPushButton("2", self)  
 self.hbox\_first.addWidget(self.b\_2)  
 self.b\_3 = QPushButton("3", self)  
 self.hbox\_first.addWidget(self.b\_3)  
 self.b\_4 = QPushButton("4", self)  
 self.hbox\_first.addWidget(self.b\_4)  
 self.b\_5 = QPushButton("5", self)  
 self.hbox\_first.addWidget(self.b\_5)  
 self.b\_6 = QPushButton("6", self)  
 self.hbox\_first.addWidget(self.b\_6)  
 self.b\_7 = QPushButton("7", self)  
 self.hbox\_first.addWidget(self.b\_7)  
 self.b\_8 = QPushButton("8", self)  
 self.hbox\_first.addWidget(self.b\_8)  
 self.b\_9 = QPushButton("9", self)  
 self.hbox\_first.addWidget(self.b\_9)  
 self.b\_0 = QPushButton("0", self)  
 self.hbox\_first.addWidget(self.b\_0)  
  
 self.b\_plus = QPushButton("+", self)  
 self.hbox\_second.addWidget(self.b\_plus)  
 self.b\_minus = QPushButton("-", self)  
 self.hbox\_second.addWidget(self.b\_minus)  
 self.b\_multiply = QPushButton("\*", self)  
 self.hbox\_second.addWidget(self.b\_multiply)  
 self.b\_divide = QPushButton("/", self)  
 self.hbox\_second.addWidget(self.b\_divide)  
  
 self.b\_result = QPushButton("=", self)  
 self.hbox\_result.addWidget(self.b\_result)  
 self.b\_reset = QPushButton("C", self)  
 self.hbox\_result.addWidget(self.b\_reset)  
  
 self.b\_0.clicked.connect(lambda: self.\_button("0"))  
 self.b\_1.clicked.connect(lambda: self.\_button("1"))  
 self.b\_2.clicked.connect(lambda: self.\_button("2"))  
 self.b\_3.clicked.connect(lambda: self.\_button("3"))  
 self.b\_4.clicked.connect(lambda: self.\_button("4"))  
 self.b\_5.clicked.connect(lambda: self.\_button("5"))  
 self.b\_6.clicked.connect(lambda: self.\_button("6"))  
 self.b\_7.clicked.connect(lambda: self.\_button("7"))  
 self.b\_8.clicked.connect(lambda: self.\_button("8"))  
 self.b\_9.clicked.connect(lambda: self.\_button("9"))  
  
 self.b\_plus.clicked.connect(lambda: self.\_operation("+"))  
 self.b\_minus.clicked.connect(lambda: self.\_operation("-"))  
 self.b\_multiply.clicked.connect(lambda: self.\_operation("\*"))  
 self.b\_divide.clicked.connect(lambda: self.\_operation("/"))  
 self.b\_reset.clicked.connect(self.\_reset)  
 self.b\_result.clicked.connect(self.\_result)  
  
 def \_button(self, param):  
 line = self.input.text()  
 self.input.setText(line + param)  
  
 def \_operation(self, op):  
 self.num\_1 = int(self.input.text())  
 self.op = op  
 self.input.setText("")  
  
 def \_reset(self):  
 self.input.setText("")  
  
 def \_result(self):  
 if self.input.text():  
 self.num\_2 = int(self.input.text())  
 if self.op == "+":  
 self.input.setText(str(self.num\_1 + self.num\_2))  
 elif self.op == "-":  
 self.input.setText(str(self.num\_1 - self.num\_2))  
 elif self.op == "\*":  
 self.input.setText(str(self.num\_1 \* self.num\_2))  
 elif self.op == "/":  
 if self.num\_2 != 0:  
 self.input.setText(str(self.num\_1 / self.num\_2))  
 else:  
 self.input.setText("Can not be divided by 0 :(")  
  
app = QApplication(sys.argv)  
  
win = Calculator()  
win.show()  
  
sys.exit(app.exec\_())

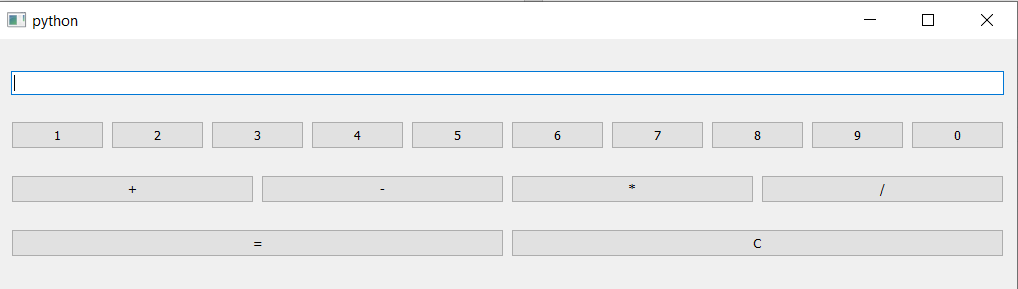
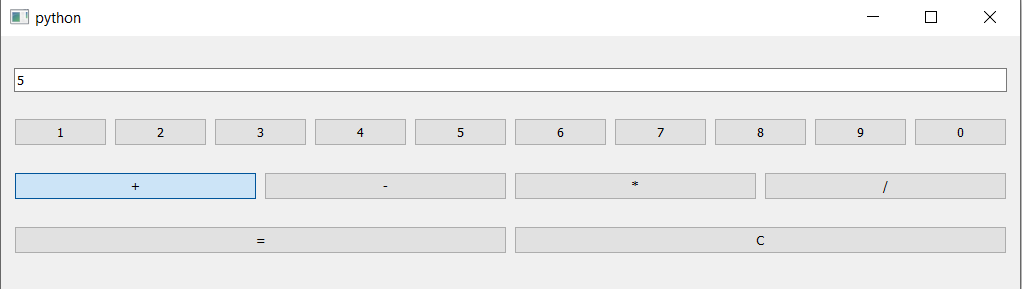
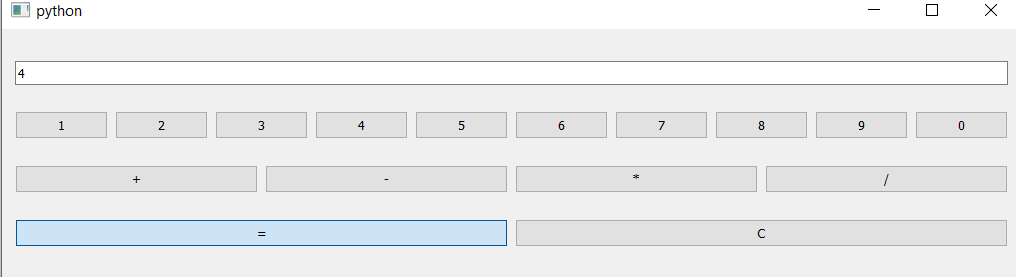
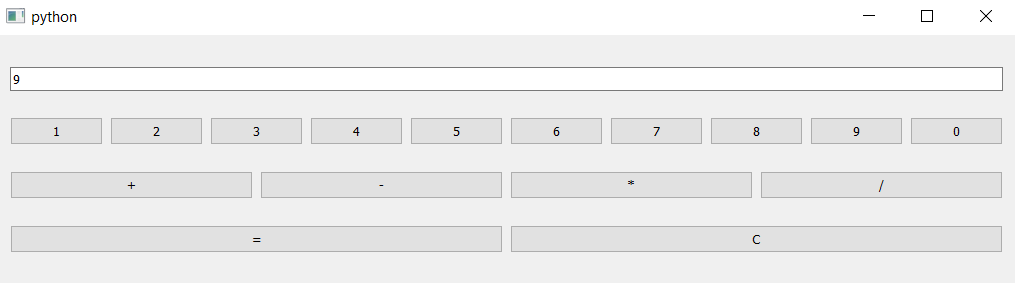
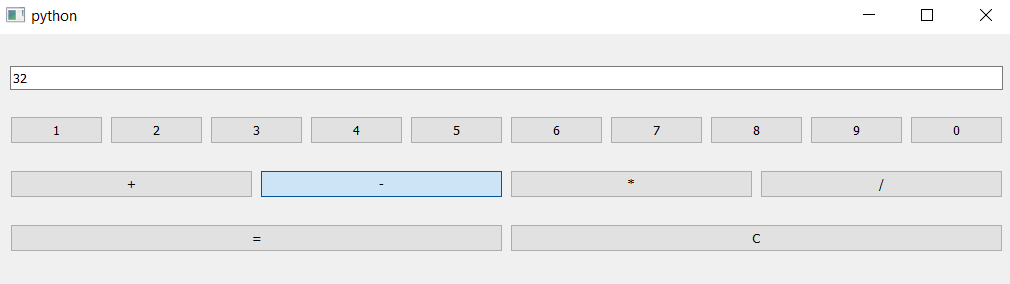
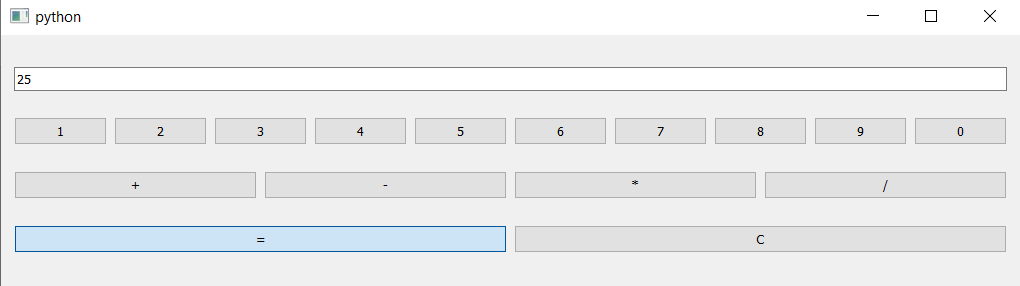
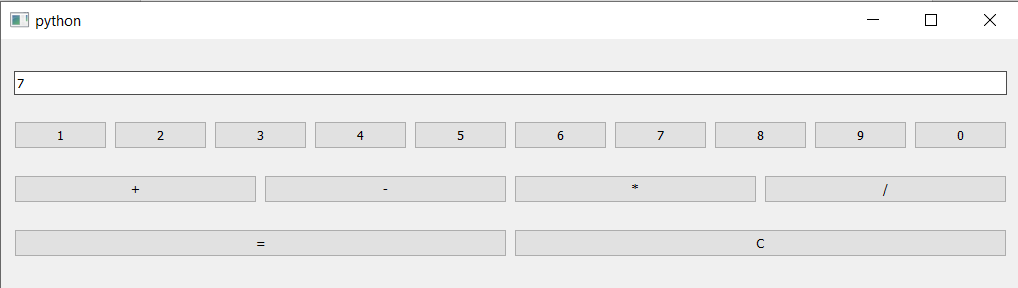


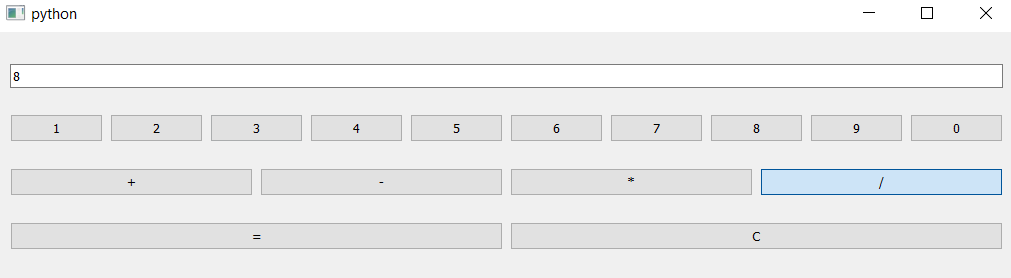
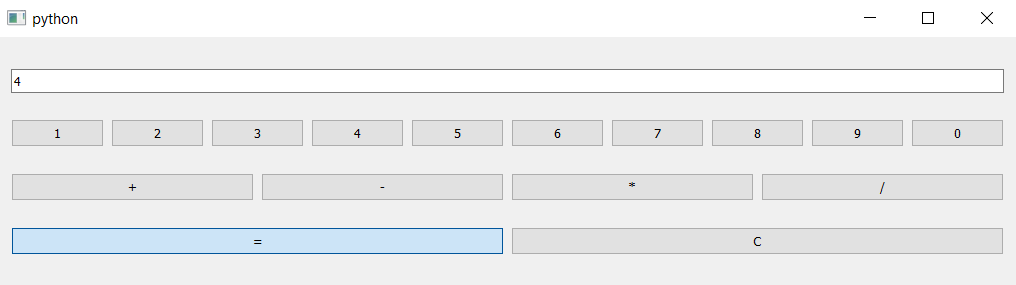
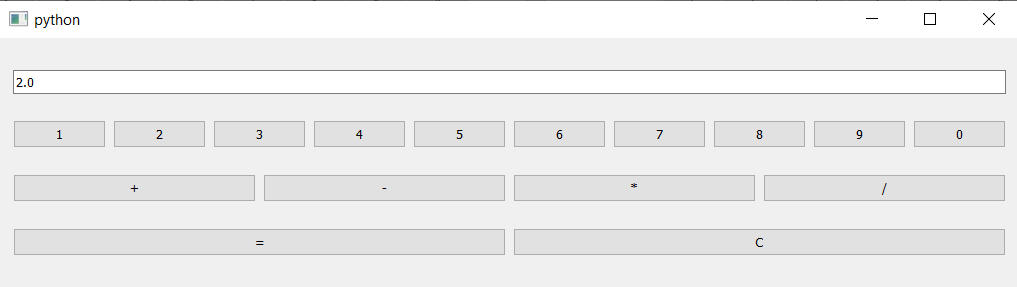
Рисунок 1 – работа программы при запуске компилятора

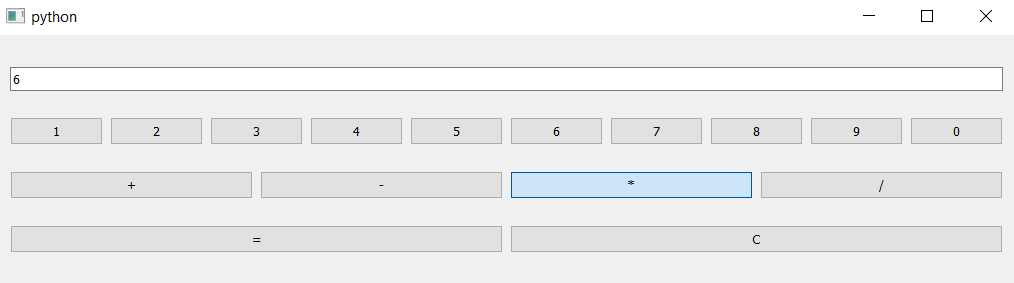
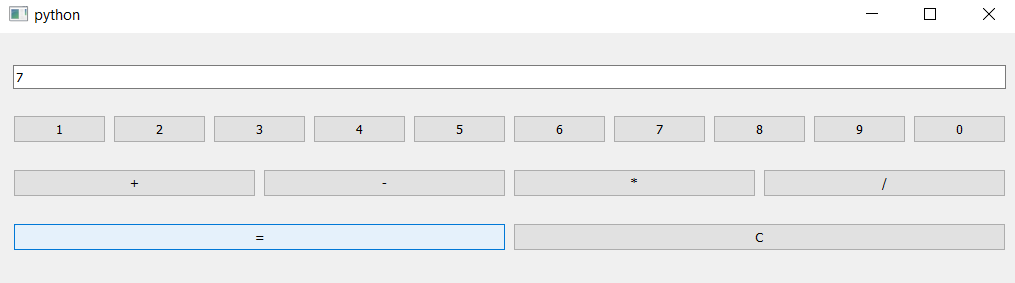
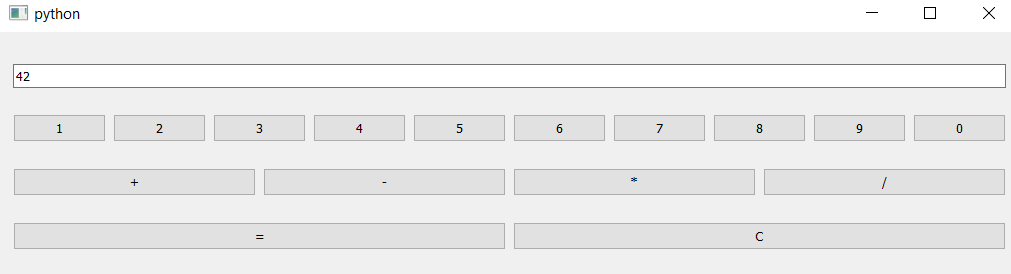
Рисунки 2, 3, 4 – выполнение сложения двух чисел

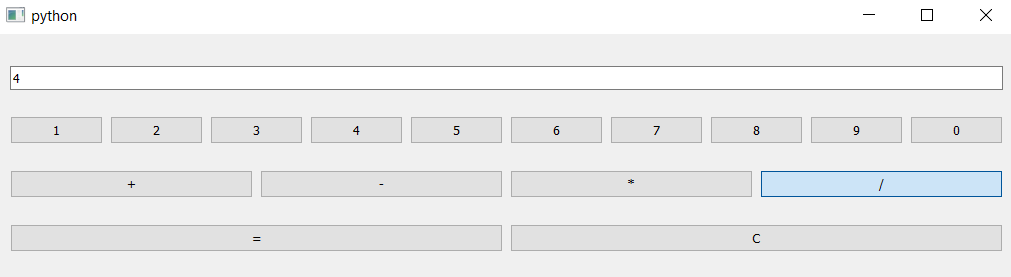
Рисунки 5, 6, 7 – выполнение вычитания двух чисел

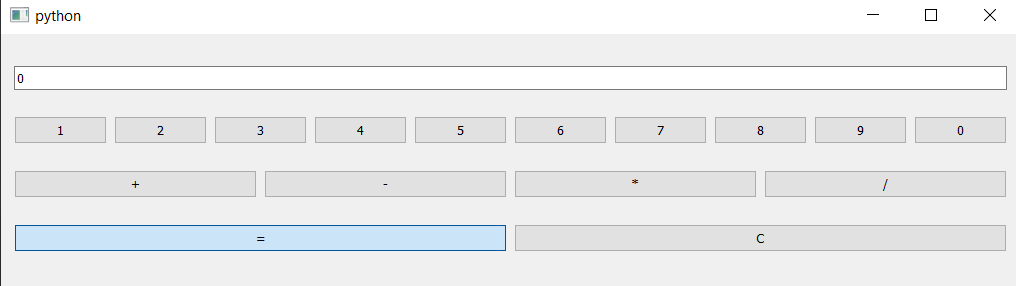
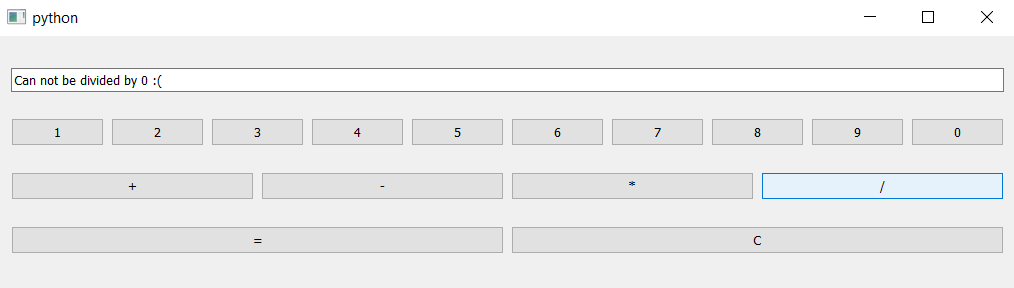
  

Рисунки 8, 9, 10 – выполнение деление двух чисел

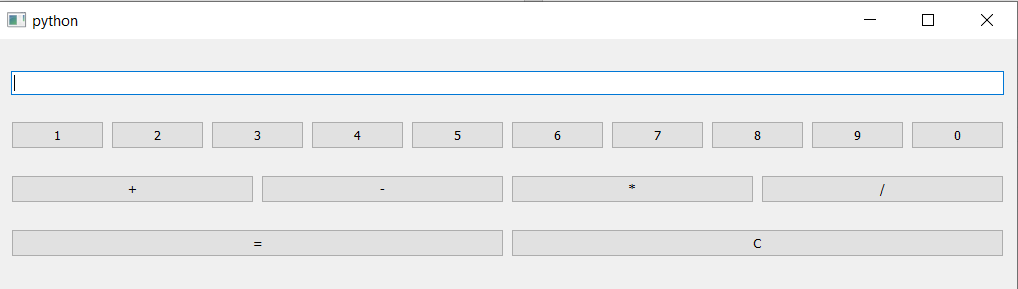
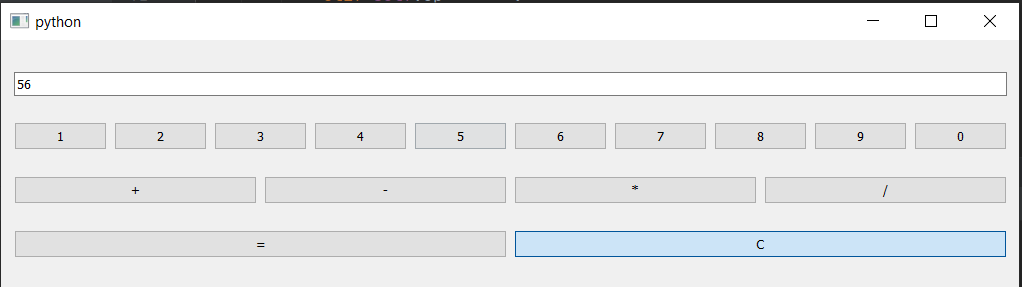
  

Рисунки 11, 12, 13 – выполнения умножения двух чисел

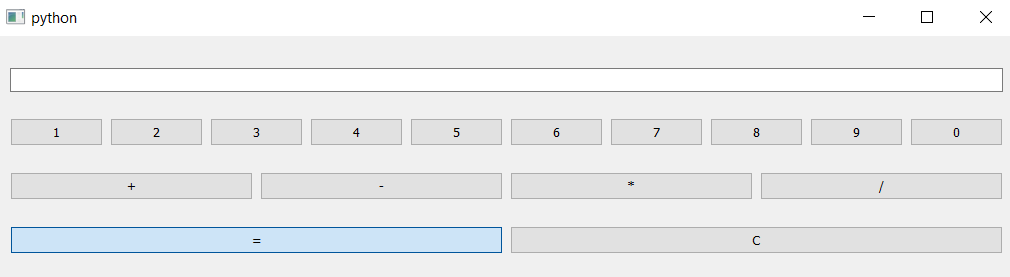


Рисунки 14, 15, 16 – деление на 0



Рисунки 17 - обнуление



Рисунки 17 - обнуление

Выполнение

В результате работы я ознакомился с библиотекой Pyqt5 и благодаря ней написал программу, создающую калькулятор