**МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ**

**Ордена Трудового Красного Знамени**

**Федеральное государственное бюджетное образовательное учреждение высшего образования**

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

**Отчет по практической работе №4**

по дисциплине «Введение в информационные технологии» на тему:

Создание калькулятора с использованием Python и библиотеки PyQt5

Выполнил: студент группы БВТ2104

Кравченко Николай Алексеевич

Проверил:

Мкртчян Грач Маратович

Москва

2021

**1 Постановка задачи**

Создать простой калькулятор на языке программирования Python, с использованием библиотеки PyQt5.

**Задание**

* Обработать все возможные исключения
* Добавить кнопку для добавления плавающей точки
* Добавить кнопки для математических операций вычитания, умножения, деления
* Создать для этих кнопок методы-обработчики

2 Результат выполненной работы

Код main.py:

import sys

from PyQt5.QtWidgets import QApplication, QWidget, QLineEdit, QHBoxLayout, QVBoxLayout, QPushButton

class Digit(QPushButton):

def \_\_init\_\_(*self*, text, parent):

super(Digit, *self*).\_\_init\_\_(text, parent)

*self*.clicked.connect(lambda: parent.\_button(text))

class Operation(QPushButton):

def \_\_init\_\_(*self*, text, parent):

super(Operation, *self*).\_\_init\_\_(text, parent)

*self*.clicked.connect(lambda: parent.\_operation(text))

class Calculator(QWidget):

def \_\_init\_\_(*self*):

super(Calculator, *self*).\_\_init\_\_()

*self*.op = None

*self*.hbox = QHBoxLayout(*self*)

*self*.vbox = QVBoxLayout()

*self*.hbox\_input = QHBoxLayout()

*self*.vbox\_operate = QVBoxLayout()

*self*.hbox.addLayout(*self*.vbox)

*self*.vbox.addLayout(*self*.hbox\_input)

*self*.hbox\_digits = []

*self*.b = [Digit(str(number), *self*) for number in range(10)]

for i in range(2, -1, -1):

layout = QHBoxLayout()

*self*.hbox\_digits.append(layout)

*self*.vbox.addLayout(layout)

for j in range(1, 4):

number = i \* 3 + j

*self*.hbox\_digits[-1].addWidget(*self*.b[number])

layout = QHBoxLayout()

*self*.hbox\_digits.append(layout)

*self*.vbox.addLayout(layout)

*self*.hbox\_digits[-1].addWidget(*self*.b[0])

*self*.point = Digit('.', *self*)

*self*.hbox\_digits[-1].addWidget(*self*.point)

*self*.input = QLineEdit(*self*)

*self*.hbox\_input.addWidget(*self*.input)

*self*.hbox.addLayout(*self*.vbox\_operate)

*self*.oper = [Operation(s, *self*) for s in ('-', '+', '/', '\*')]

for i in *self*.oper:

*self*.vbox\_operate.addWidget(i)

*self*.b\_result = QPushButton("=", *self*)

*self*.vbox\_operate.addWidget(*self*.b\_result)

*self*.b\_result.clicked.connect(*self*.\_result)

def \_button(*self*, param):

line = *self*.input.text()

if param == '.' and '.' in line:

line = line[:line.index('.')]

param = ''

*self*.input.setText(line + param)

def \_operation(*self*, op):

text = *self*.input.text().strip()

if text and text.replace('.', '', 1).isdigit():

*self*.num\_1 = float(text)

*self*.op = op

*self*.input.setText("")

def \_result(*self*):

text = *self*.input.text().strip()

if text and text.replace('.', '', 1).isdigit():

*self*.num\_2 = float(text)

if *self*.op:

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 == '/':

if *self*.num\_2 != 0:

*self*.input.setText(str(*self*.num\_1 / *self*.num\_2))

elif *self*.op == '\*':

*self*.input.setText(str(*self*.num\_1 \* *self*.num\_2))

app = QApplication(sys.argv)

win = Calculator()

win.show()

sys.exit(app.exec\_())

**3 Вывод**

В результате проделанной работы мы изучили принципы графических интерфейсов на Qt5.