МИНИСТЕРСТВО **ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ МАССОВЫХ**

**КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Ордена Трудового Красного Знамени федеральное государственное бюджетное образовательное учреждение высшего образования**

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

Индивидуальный проект

по дисциплине: «Введение в информационные технологии»

«Создание оконного приложения-калькулятора».

Выполнил студент группы БИН2308

Герасимов.И.Д

Проверил преподаватель:

Москва

2023

Цель работы : создать оконное приложение-калькулятор, который по итогу должен будет уметь выполнять все те функции, что делает самый обыкновенный калькулятор.

Ход работы :

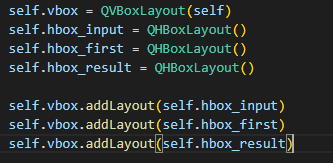
1 ШАГ. Импортируем необходимые библиотеки для создания приложения.



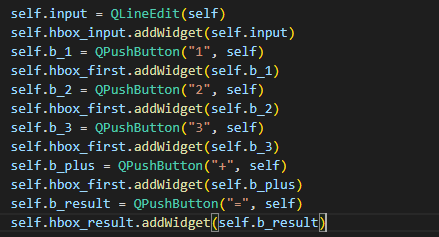
2 ШАГ. Создаём класс Calculator и наследуем его от класса Qwidgets



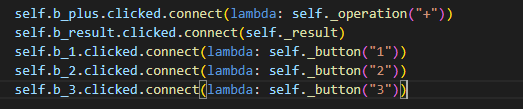
3 ШАГ. Внутри конструктора создаём ось выравнивания.



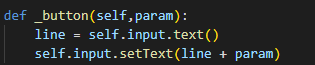
4 ШАГ. В конструкторе создаём виджеты и привязываем их к соответствующим осям выравнивания.



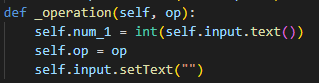
5 ШАГ. Создаём события, отвечающие за реакции на нажатия по кнопкам.



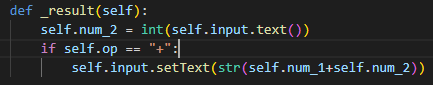
6 ШАГ. Создадим метод-обработчик для наших кнопок, отвечающих за ввод цифр в линию ввода текста.



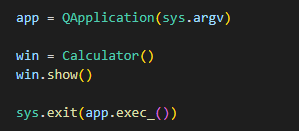
7 ШАГ. Создадим метод-обработчик для наших кнопок, отвечающих за математические операции.



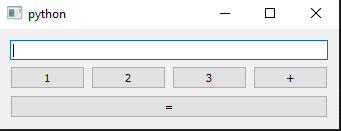
8 ШАГ. Cоздадим очередной метод-обработчик, но уже для результата математического вычисления.



9 ШАГ. Запускаем приложение.



10 ШАГ. Получаем результат.



Домашняя работа :

1. Обработать все возможные исключения
2. Добавить кнопку для добавления плавающей точки
3. Добавить кнопки для математических операций вычитания, умножения, деления
4. Создать для этих кнопок методы-обработчики

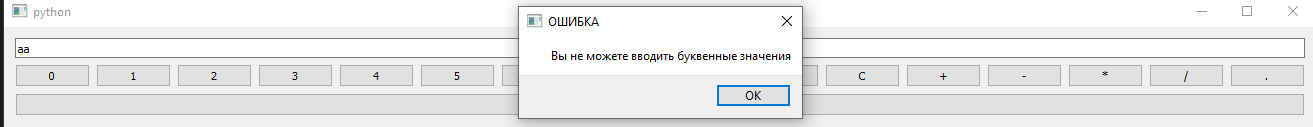
1 пункт :

В калькуляторе может возникнуть 2 исключения, 1, когда пользователь вводит с клавиатуры не цифры, а буквы, а 2, когда пользователь делит свой результат на ноль.

Чтобы обработать 1 исключение, дополним код :



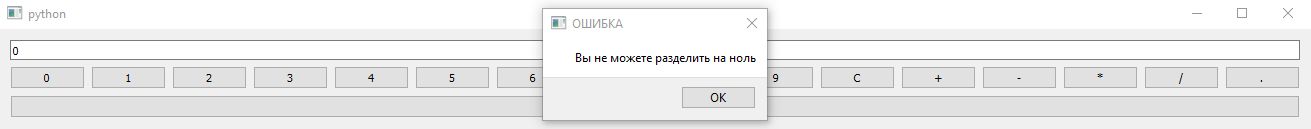
Метод isnumeric() проверяет, все ли символы в строке являются цифрами. Если пользователь вводит буквенное значение, он получает ошибку.



Чтобы обработать 2 исключение, также дополним код :



Программа проверяет, если выбранная операция – это деление и элемент, на который мы делим, равняется нулю, программа выдаёт ошибку.



2 пункт :

Через addWidget добавим плавающую точку.



Не забываем подключить её к самому приложению

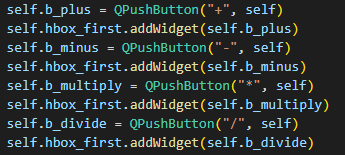


Получаем результат :

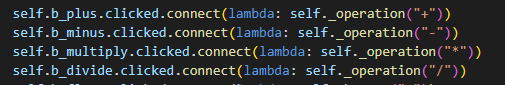


3 и 4 пункт.

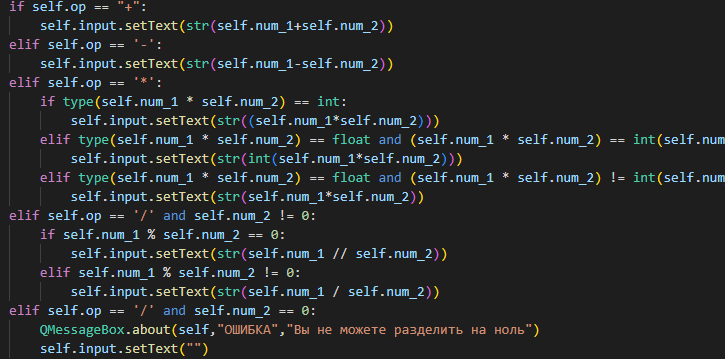
Первым делом через addWidget добавим кнопки для + ; - ; \*; /.



Подключаем к приложению.



Теперь, сделаем так, чтобы наши математические операции работали. Через переменную self.op проверяем вид операции и отправляем результат на экран. Для того чтобы избежать последующих некорректных выводов результата, я к операции умножения и деления добавил несколько условий-обработчиков входных данных.



Доработаем наш калькулятор. Сделаем так, чтобы при вводе буквенных значений, знаков препинания, нескольких плавающих точек пользователь получал ошибку.

 if self.input.text().isnumeric() == False and ('?' or '!' or ';' or ':' or ',' or '@' or '{}' or '[]' or '[}' or '{]' or '|') not in self.input.text() and ".." not in self.input.text():

                QMessageBox.about(self,"ОШИБКА","Вы не можете вводить буквенные значения")

                self.input.setText("")

            elif self.input.text().isnumeric() == False and ('?' or '!' or ';' or ':' or ',' or '@' or '{}' or '[]' or '[}' or '{]' or '|') in self.input.text():

                QMessageBox.about(self,"ОШИБКА","Вы не можете вводить знаки пунктуации")

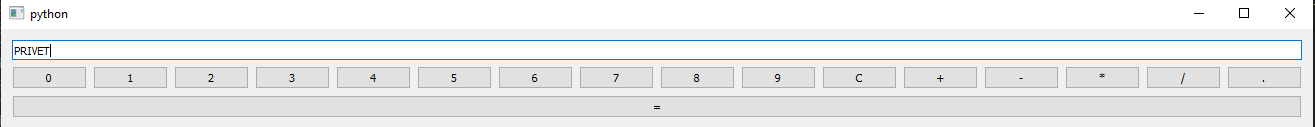
                self.input.setText("")

            elif self.input.text().isnumeric() == False and ('?' or '!' or ';' or ':' or ',' or '@' or '{}' or '[]' or '[}' or '{]' or '|') not in self.input.text() and ".." in self.input.text():

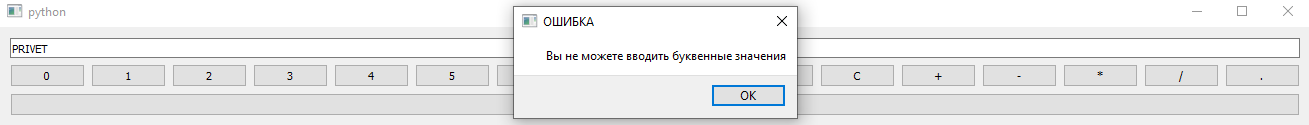
                QMessageBox.about(self,"ОШИБКА","Вы не можете ввести больше 1 плавающей точки в одном числе")

                self.input.setText("")

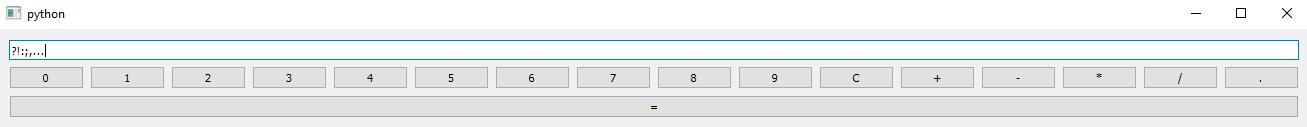
Попробуем ввести буквенное значение:



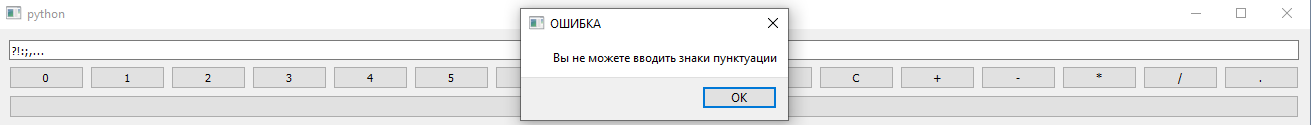
Получаем ошибку :



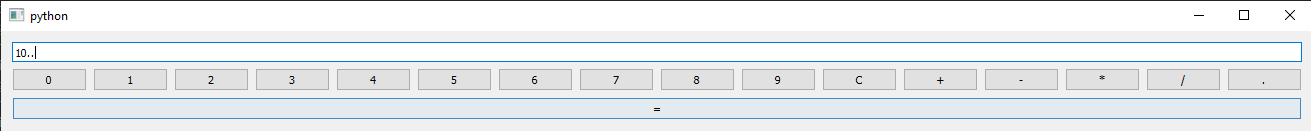
Попробуем ввести знаки препинания:



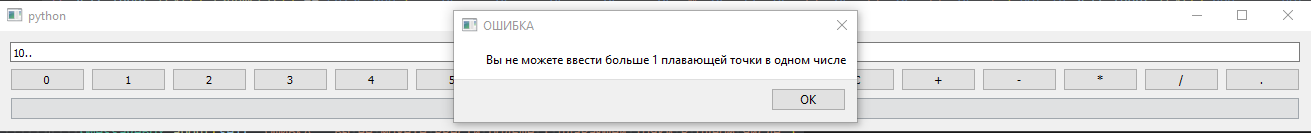
Получаем ошибку :



Попробуем ввести две плавающие точки в числе:



Получаем ошибку :



Вывод : По итогу я смог создать калькулятор, который умеет выполнять все самые базовые математические операции. В этой работе мне понадобилось не только применить навыки Python, но и немного логически поразмышлять о том, какие исключения могут появиться при работе пользователя с нашим калькулятором.