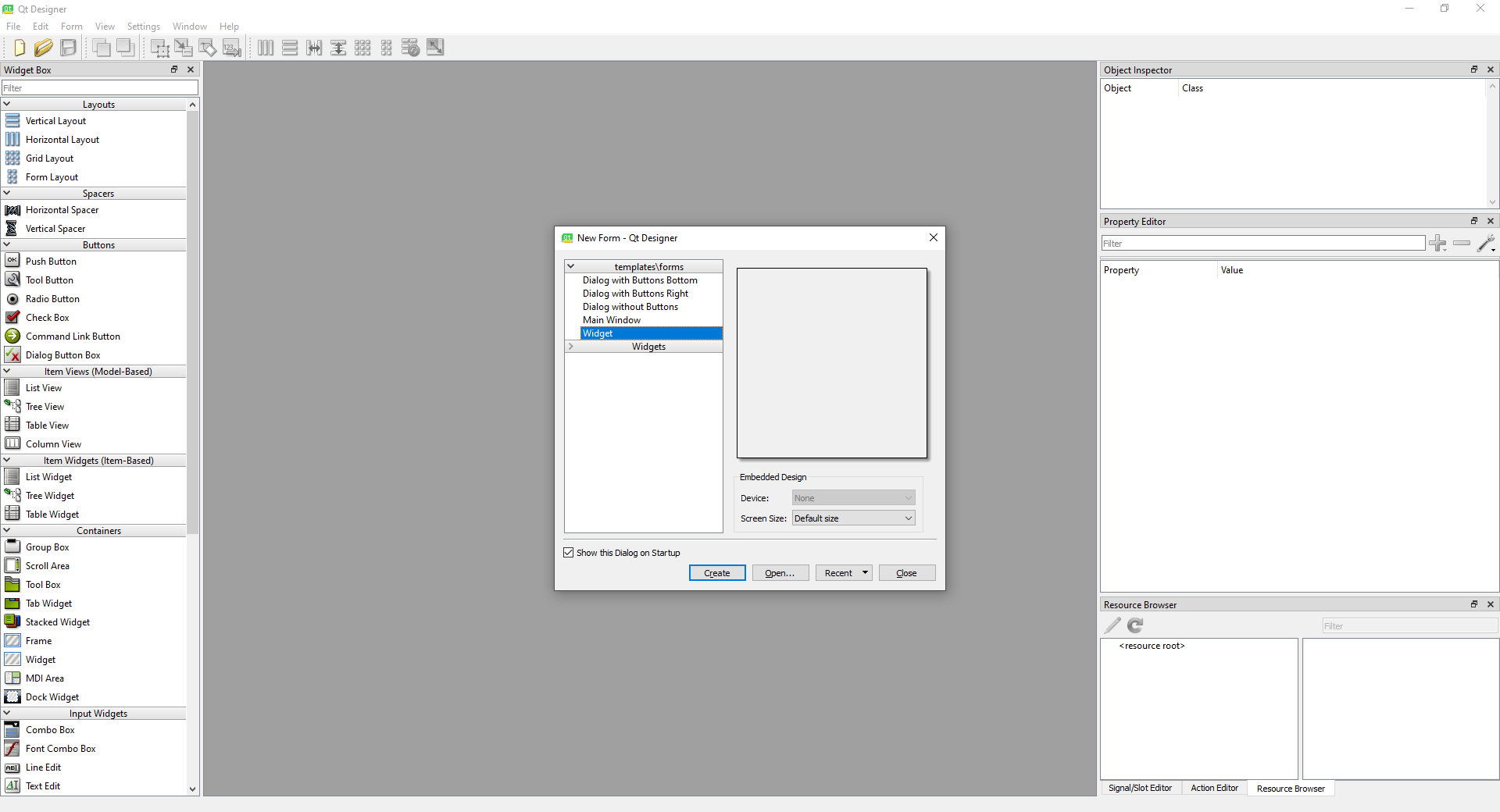
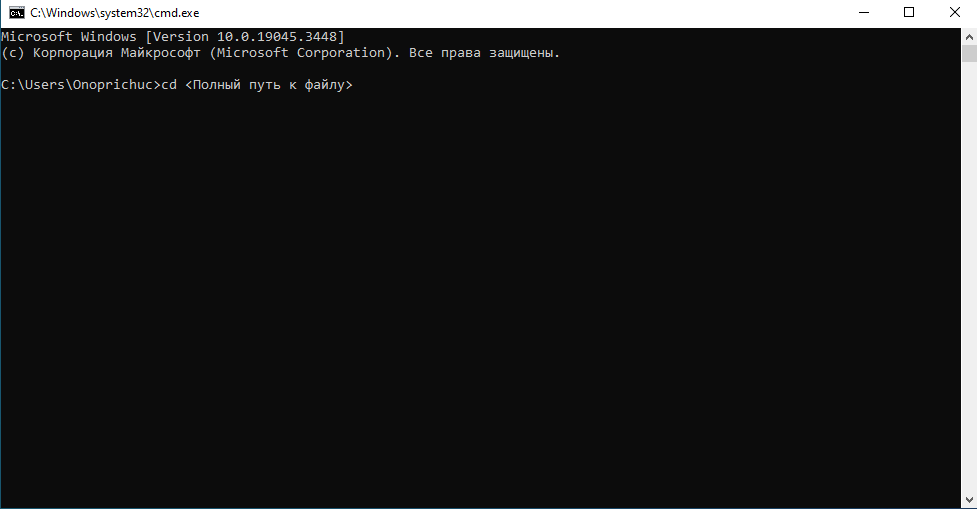
**Начало работы с Qt Designer**

Чтобы начать работать с Qt Designer, запустите саму программу



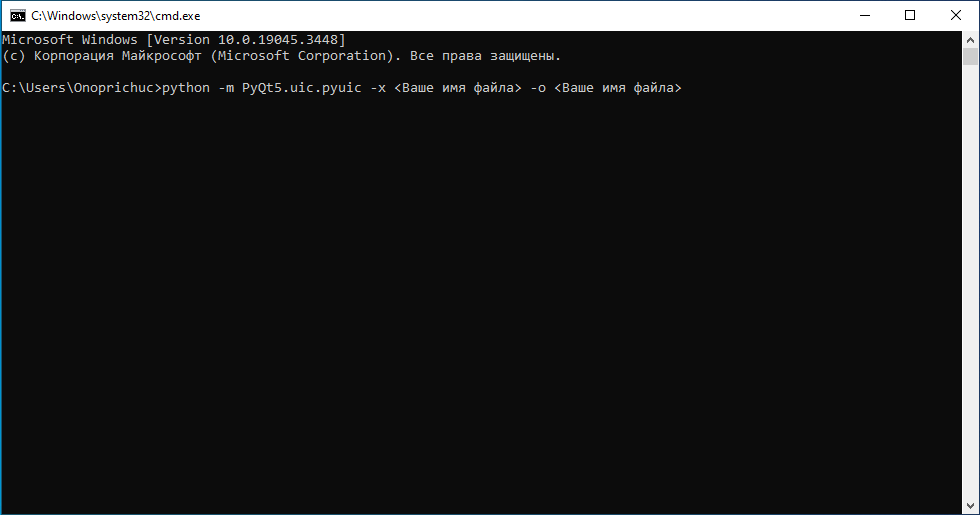
Создайте макет на основе **Widget**, создайте папку и сохраните данный файл формата .ui в неё, для того чтобы начать полноценную работу нам потребуется конвертировать файл из .ui в .py.

Инструкция как это сделать :



Для начала зайдем в командную строку Windows (горячие клавиши Win+R, в появившейся строке вводим cmd)

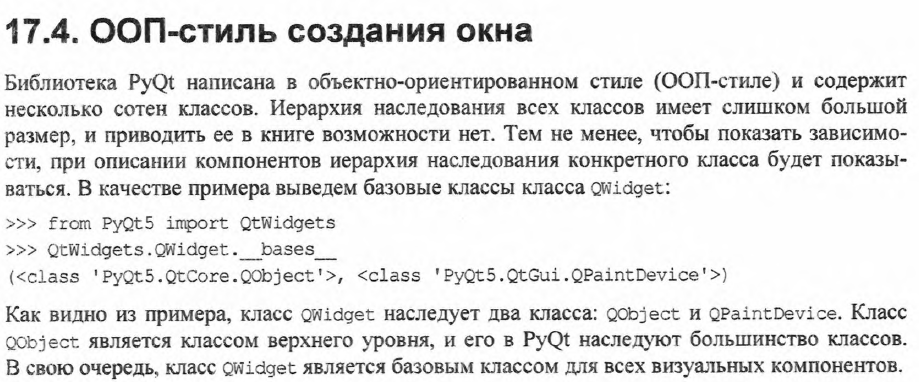
Для того чтобы перейти в директорию папки, где нужно конвертировать определенные файлы нам нужно прописать команду cd <Полный путь к папке>

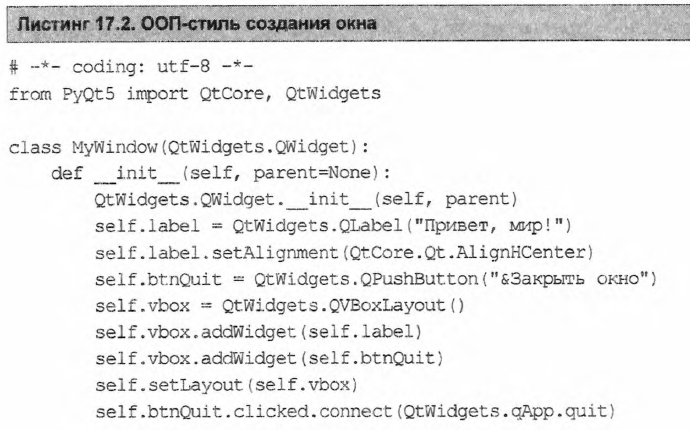


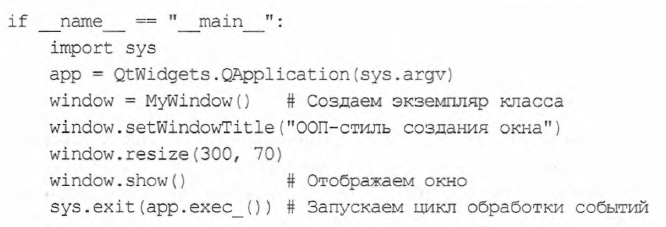
**Как только мы окажемся в директории, пишем следующее :**

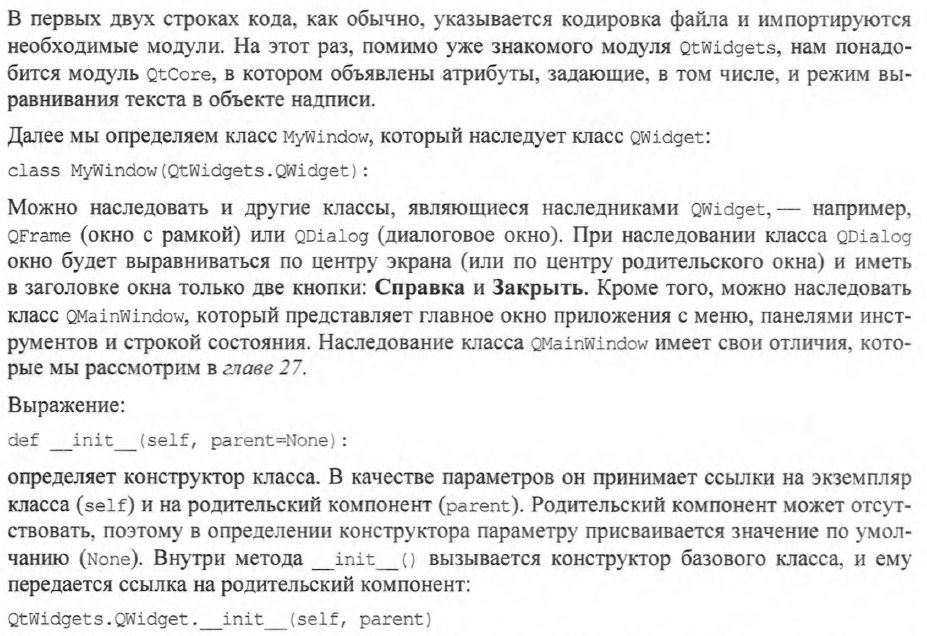
**-m PyQt5.uic.pyuic -x <ИМЯ ФАЙЛА.ui> -o <ИМЯ ФАЙЛА.py> (1 вариант)**

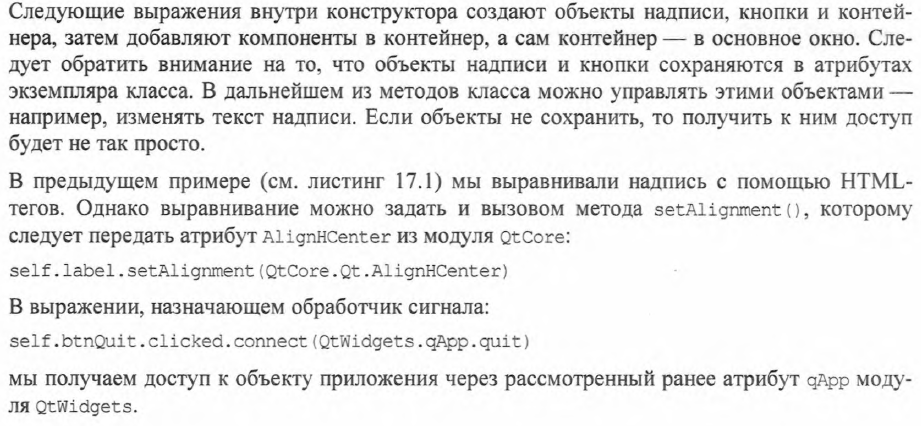
****



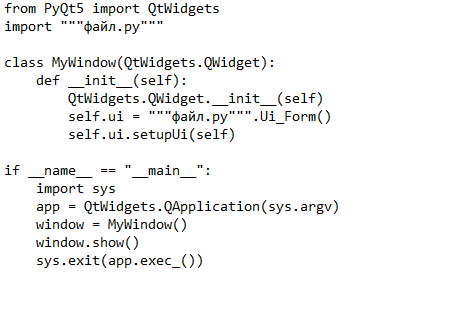




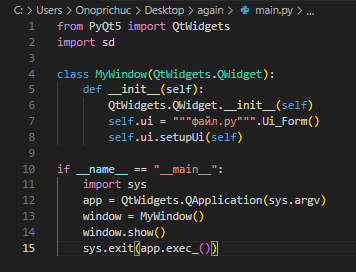




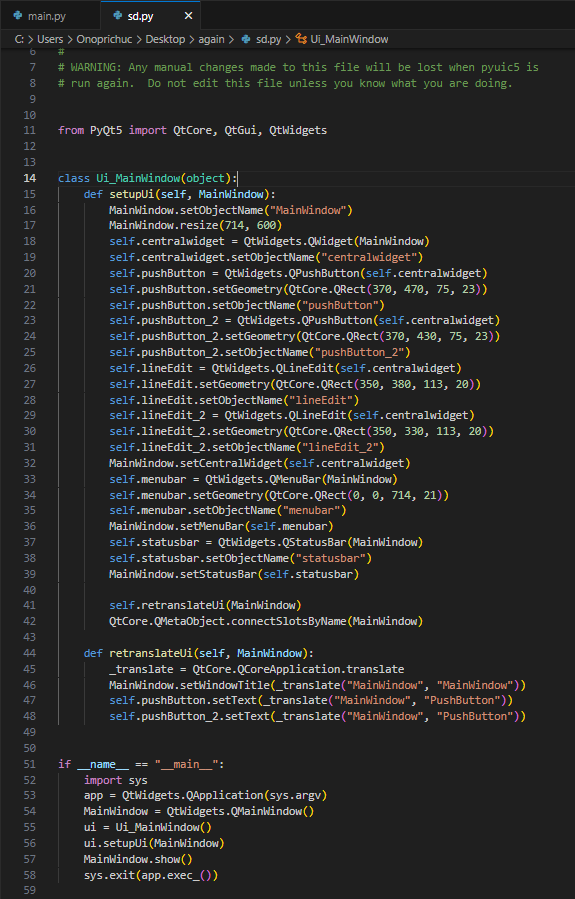
**Далее нам нужно будет создать экземпляр окна в отдельном файле:**

****

**Итого у нас два файла:**

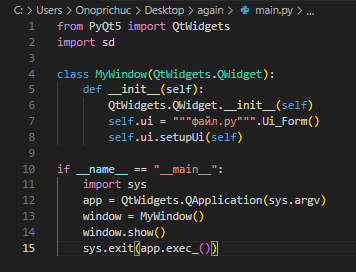
****

**Файл который вы создаете для макета приложения и написания кода. Там где написано «файл.ру» вы должны написать название ui файла который вы сконвертировали (в данном случае sd)**

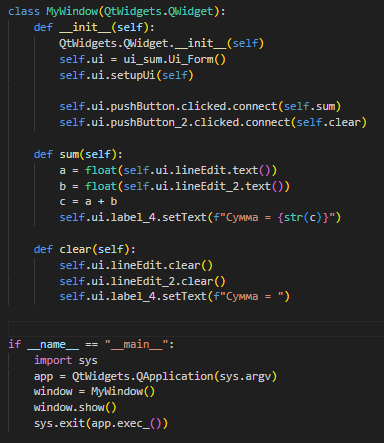
****

**Файл который мы сконвертировали из .ui в .py (автоматически)**

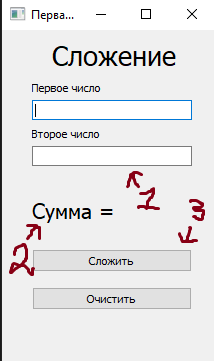
**Весь код и запуск программы мы будем делать при помощи данного файла, он интересует нас больше всего.**

****

**Сегодня нас интересует написание простенького калькулятора в две строчки и 2 кнопки, посмотрим как мы это можем сделать:**

****

**И так что здесь происходит: создаем две функции sum (для математических расчетов) и clear (для очищения строк где будут значения)**

****

**В самом редакторе QtDesigner мы должны добавить следующие вещи:**

1. **lineEdit – позволяет вводить пользователю данные (аналог input)**
2. **label – строка в которую можно что-либо записать или как-либо ее видоизменить**
3. **button – кнопка, позволяющая при нажатии (или напр. Зажатии) исполнить определенный код**

**Self.ui.pushButton.clicked.connect(self.sum) нужен нам для того, чтобы приконнектить нашу кнопку к определенному куску кода, т.е у нас есть условно кнопка посчитать, и чтобы она что-то делала мы должны объявить для нее функцию в которой что-то будет.**

**Def sum(self) содержит в себе 3 переменных для элементарных математических вычислений, при этом переменные а и b забирают в себя те значения, которые введет пользователь в нужном эдите (lineEdit – позволяет вводить пользователю какие-либо сообщения),**

**А после label.setText выводит сообщение о том, чему равняется сумма. Сам же label представляет из себя простую строку, в которой можно объявить текст или видоизменить его как написано здесь.**

**Метод clear() как вы уже догадались просто очищает эдит от всего, что мог написать пользователь.**



**Задание 1:**

**Напишите программу которая выводит текст «Hello world» и содержит в себе кнопку «ОК»**

**Задание 2:**

**Напишите по примеру выше простой калькулятор который может подсчитывать сумму при помощи кнопки, и при помощи другой кнопки очищать заданные параметры в эдите.**