

## Первое приложение на PyQT

Откроем Qt Designer.

Здесь нам предлагается выбрать, какой виджет будет главным в создаваемой программе. Выбираем виджет MainWindows. Для этого нажимаем «Создать». На экране появится чистая заготовка для дальнейшей работы.

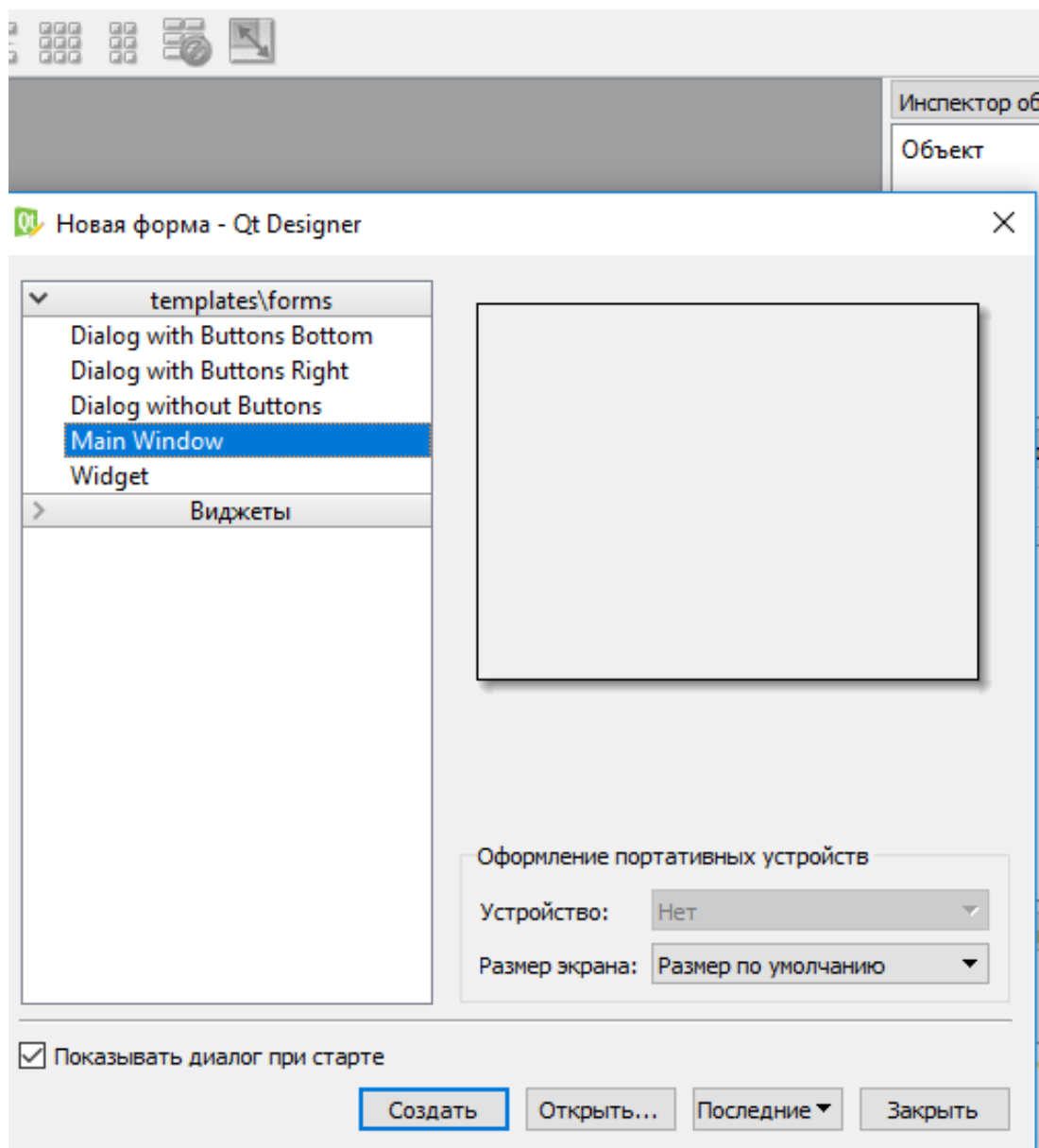
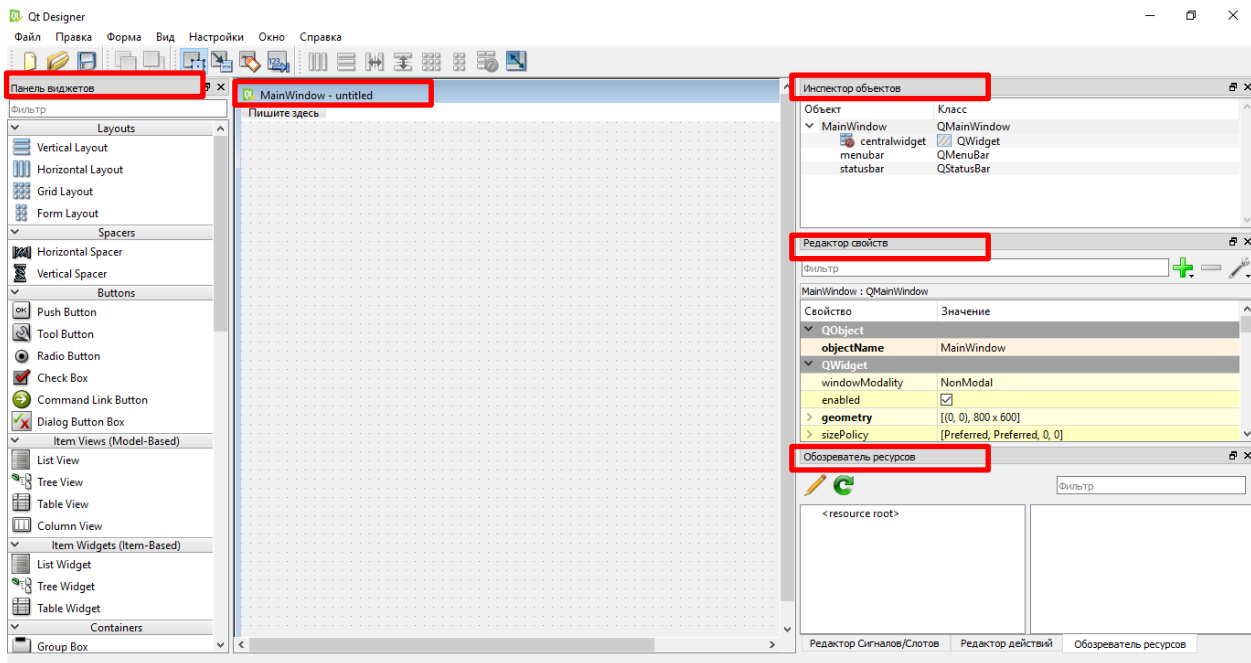


Рисунок 1. Диалоговое окно при запуске Qt Designer

## Ознакомимся с интерфейсом Qt Designer.



Все панель отображаются в меню Вид.

Создадим интерфейс программы, которая будет получать от пользователя текст и выводить на экран длину этого текста в символах.

Добавляем 3 виджета:

- Строка ввода (QLineEdit)
- Кнопку (QPushButton)
- Надпись (QLabel)

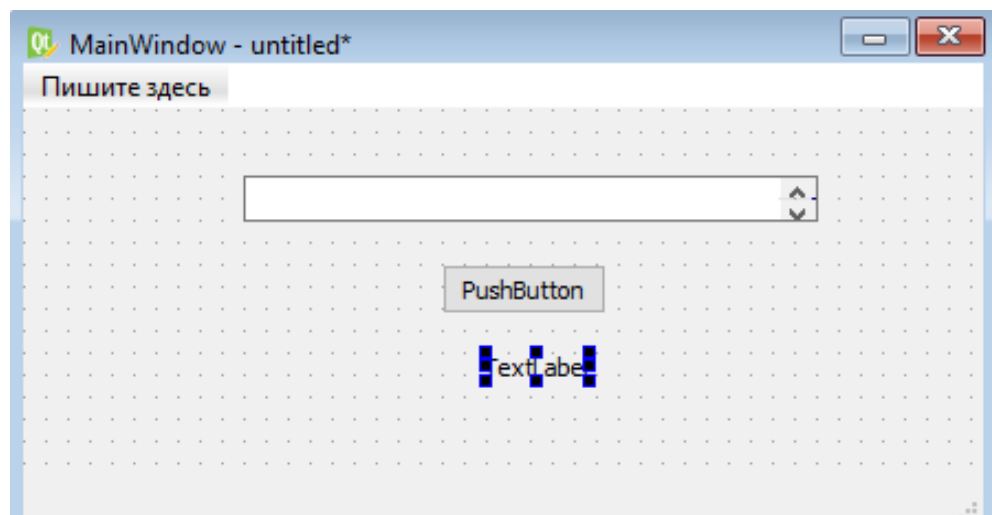


Рисунок 2. Заполнение заготовки

Изменим текст надписи, для этого выделим мышью надпись и в «*Редакторе свойств*» найдем свойство text. Напишем «Длина Вашего текста».

Сохраните интерфейс в файл myinterface.ui.

Сконвертируем файл `ui` в файл `py`. Скопируем файл `myinterface.ui` в директорию (проверьте находится ли в каталоге файл `pyuic5`) `C:\Python36\Lib\site-packages\PyQt5\uic` или `C:\Python*\Scripts`.

Откроем командную строку `Windows` и введем команду «`cmd`» напишем команду `pyuic5 myinterface.ui -o myinterface.py -x` (или создать `bat`-файл). Если всё успешно, то в той же директории появится файл `myinterface.py`. Файл копируем в свой каталог и открываем его в `IDLE`. Запустить `F5`.

Разбираем код:

Первые строки являются комментариями

Производится импорт нужных модулей библиотеки `PyQt5`.

В классе `Ui_MainWindow()` содержатся 2 функции.

Функция `setupUi` строит графический интерфейс, а функция `retranslateUi` создает подписи для виджетов, которые могут иметь подписи (`Button,label,MainWindow`)

Далее инициализируют класс `Ui_MainWindow`, в них строится графический цикл.

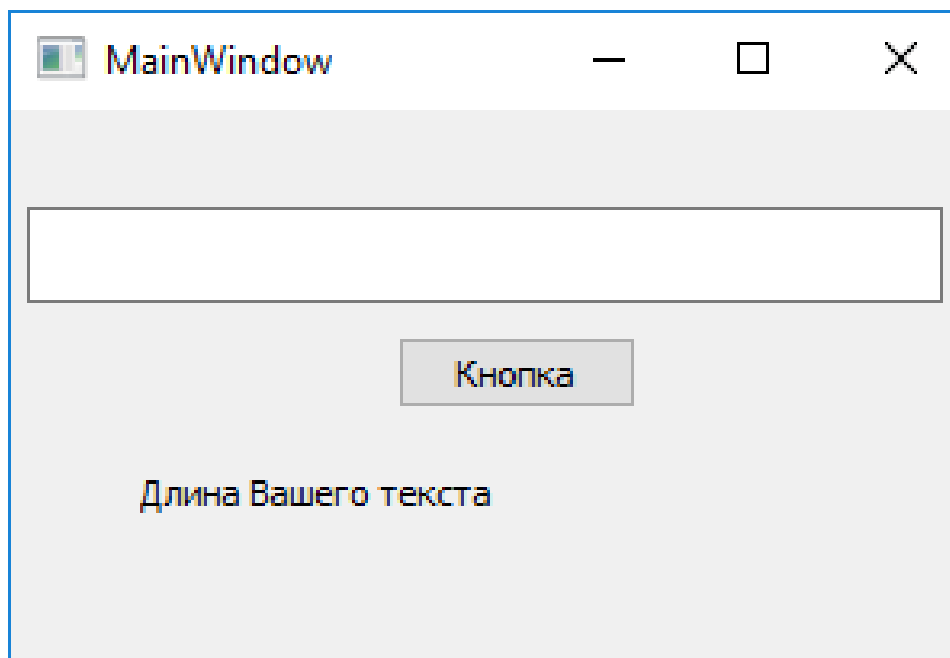


Рисунок 3. Результат работы

#### Самостоятельно:

1. Вместо «`PushButton`» заменим на «`Кнопка`»
2. Поменять размер окна `MainWindow`
3. Строку ввода сделать от одного края окна к другому (`setGeometry (QtCore.QRect (x, y, width, height))`).
4. Изменить заголовок окна на «`My Window`».

## Написание функции

Перед строкой `if __name__ = "__main__"`: добавим функцию

```
def myFunction (self):  
    self.text = self.lineEdit.text() {взять введенный текст  
из строки ввода}  
    self.lenght = len(self.text) {найти длину текста}  
    self.label.setText("Длина    вашего    текста    %d"    %  
self.lenght) {выводить результат в надпись}
```

И вызовем функцию `self.pushButton.clicked.connect(self.myFunction)` перед функцией `def retranslateUi (MainWindow)`. Проверим результат.

### Самостоятельно:

1. Модифицировать программу, чтобы она работала и по нажатию кнопки и по нажатию **Enter** (для его решения, воспользуйтесь литературой и интернетом, в частности на форумах Stack Overflow).
2. Представьте, что пользователь ввел в строку ввода одни пробелы. Модифицируйте программу так, чтобы в этом случае длина текста равнялась нулю. Это не должно касаться случая, когда вместе с пробелами введены и какие-то символы.
3. Усовершенствуйте программу дальше. Сделайте так, чтобы вывод принял форму «a / b», где a - количество всех символов кроме пробелов, b - количество пробелов. Например, при вводе «я - программист» получился бы вывод «13/2».

## Изменение интерфейса.

Секрет в том, чтобы держать в *отдельных файлах* класс интерфейса и функции обработки данных.

Конвертируйте файл интерфейса `myinterface.ui` немного по-другому. Уберите из командной строки параметр `-x`, чтобы получилось следующее: `pyuic5 myinterface.ui -o myinterface.py`. Полученный файл переименуйте в `myinterface2.py`. Этот файл отличается от предыдущего тем, что в нем нет графического цикла и его нельзя запустить самостоятельно. Создадим главный исполняемый файл - назовем его `myintmain.py`. Он должен иметь следующее содержание (см. рис. 3):

```
import sys
from myinterface2 import *
from PyQt5 import QtCore, QtGui, QtWidgets

class MyWin(QtWidgets.QMainWindow):
    def __init__(self, parent=None):
        QtWidgets.QWidget.__init__(self, parent)
        self.ui = Ui_MainWindow()
        self.ui.setupUi(self)

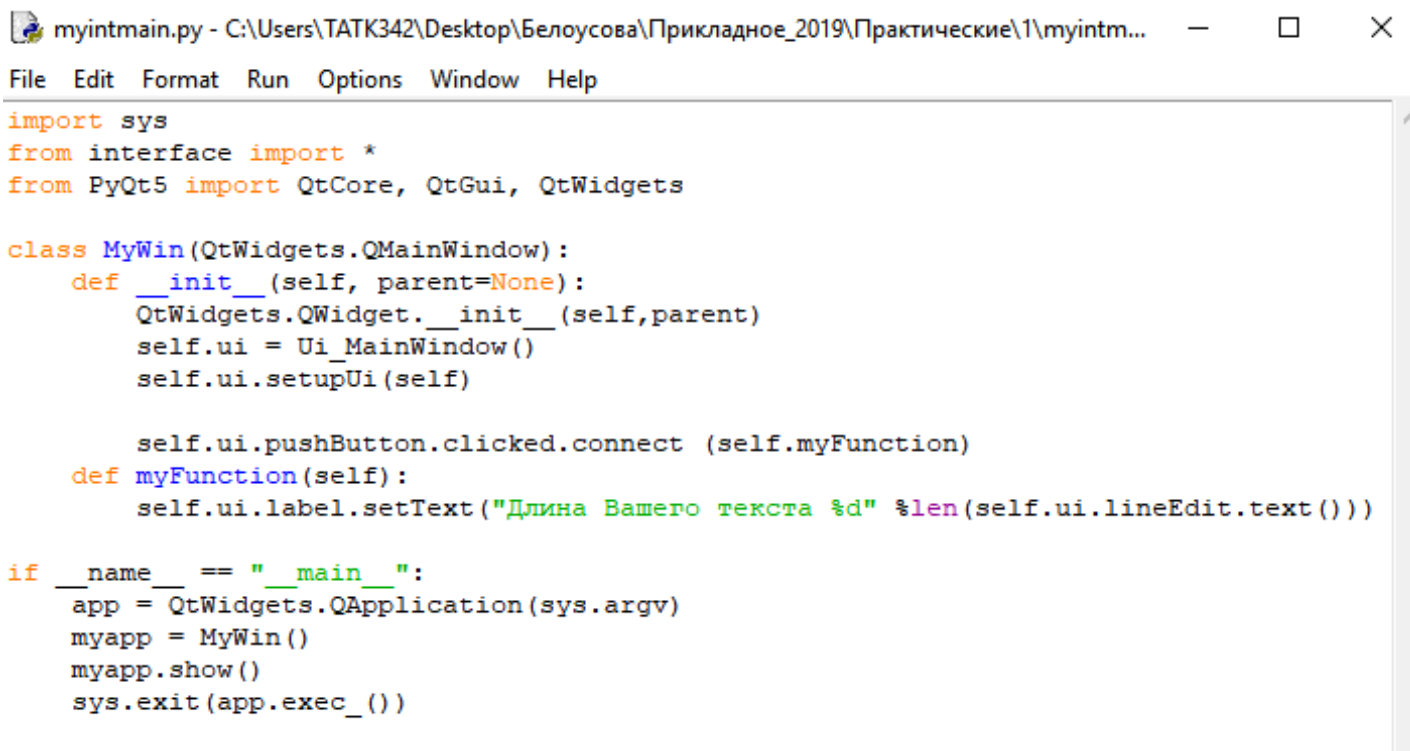
if __name__ == "__main__":
    app = QtWidgets.QApplication(sys.argv)
    myapp = MyWin()
    myapp.show()
    sys.exit(app.exec_())
```

Рисунок 3. Код модуля «Myintmain.py»

Импортируем все классы из файла `myinterface2.py`, в котором находится код интерфейса. Объявляем класс `MyWin()`. Запускаем инициализирующую функцию. Создается бесконечный цикл графического интерфейса.

***! Приведенный выше код можно использовать в качестве универсального, изменяя только строку 2.***

Теперь нужно правильно добавить в программу `myintmain.py` функцию `myFunction()` и привязать ее к кнопке. Внесите нужные изменения и сохраните полученный файл как `myintmain1.py` (см. рис. 4).

The image shows a screenshot of a code editor window titled "myintmain.py". The window has a menu bar with "File", "Edit", "Format", "Run", "Options", "Window", and "Help". The code is written in Python and uses PyQt5 for a Qt-based GUI. It defines a class "MyWin" that inherits from "QtWidgets.QMainWindow". The class has an ".\_\_init\_\_" method that initializes the UI and connects a "pushButton" to a "myFunction". The "myFunction" method updates a label with the length of the text entered in a "lineEdit". A main block of code at the bottom creates a "QApplication", instantiates "MyWin", shows it, and then exits.

```
myintmain.py - C:\Users\TATK342\Desktop\Белоусова\Прикладное_2019\Практические\1\myintm...
File Edit Format Run Options Window Help
import sys
from interface import *
from PyQt5 import QtCore, QtGui, QtWidgets

class MyWin(QtWidgets.QMainWindow):
    def __init__(self, parent=None):
        QtWidgets.QWidget.__init__(self, parent)
        self.ui = Ui_MainWindow()
        self.ui.setupUi(self)

        self.ui.pushButton.clicked.connect (self.myFunction)
    def myFunction(self):
        self.ui.label.setText("Длина Вашего текста %d" %len(self.ui.lineEdit.text()))

if __name__ == "__main__":
    app = QtWidgets.QApplication(sys.argv)
    myapp = MyWin()
    myapp.show()
    sys.exit(app.exec_())
```

Рисунок 4. Код программы myintmain1.py

Теперь можно снова открыть файл myinterface.ui в Qt Designer и изменить с помощью мыши размеры виджетов, надписи, цветовую гамму и многое другое. Измененный файл надо просто конвертировать в файл с расширением .py и все. Изменения сразу вступят в силу при запуске программы главной программы myintmain1.py. Убедитесь в этом сами, поэкспериментируя с интерфейсом. Измените заголовок главного окна и размеры виджетов.