Тема 2. Событийно-ориентированное программирование.

Лекция 6. Аппаратно–независимый ввод/вывод

Учебные вопросы

1. Верстка GUI-формы;
2. Подключение GUI-формы в программу;
3. Использование QSettings.

Источники

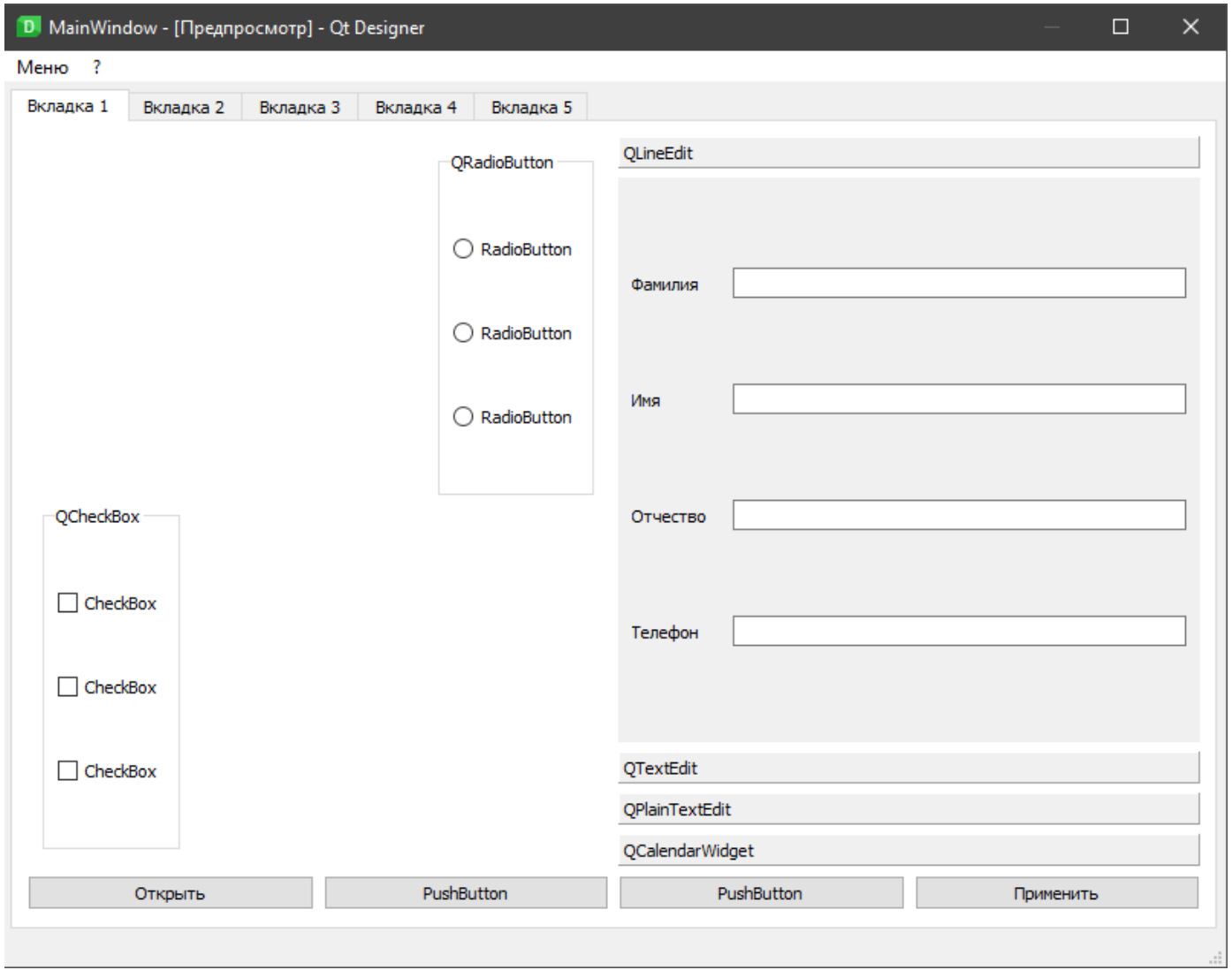
- Официальная документация: https://doc.qt.io/qtforpython
- Прохоренок Н. А., Дронов В. А. Python 3 и PyQt 5. Разработка приложений. 2019 г.

Используемые в курсе инструменты для разработки		
IDE	PyCharm CE	https://www.jetbrains.com/pycharm/download
Окружение	Virtualenv	https://docs.python.org/3/library/venv.html
VSC (рекомендовано)	GIT	https://git-scm.com
Фреймворк	PySide2	https://doc.qt.io/qtforpython/

1. Верстка GUI-формы.

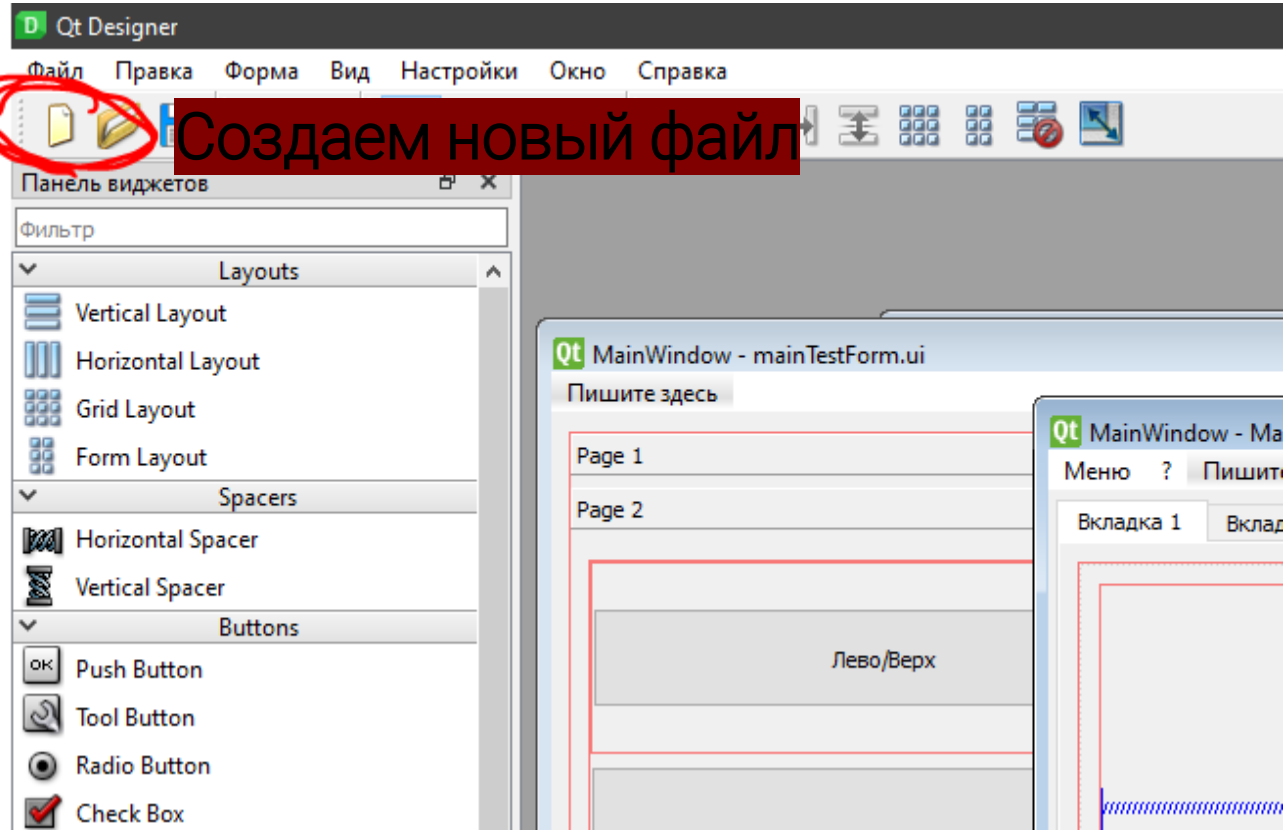
Создание GUI-формы

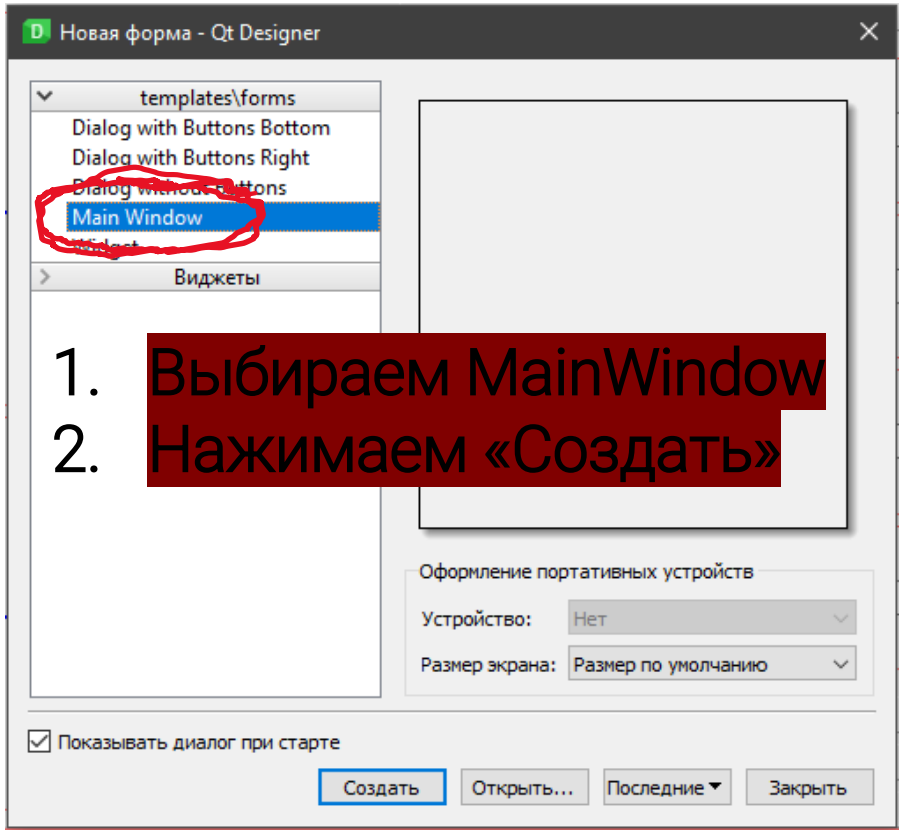
- Задание. Создать GUI-форму следующего вида.



Создание GUI-формы

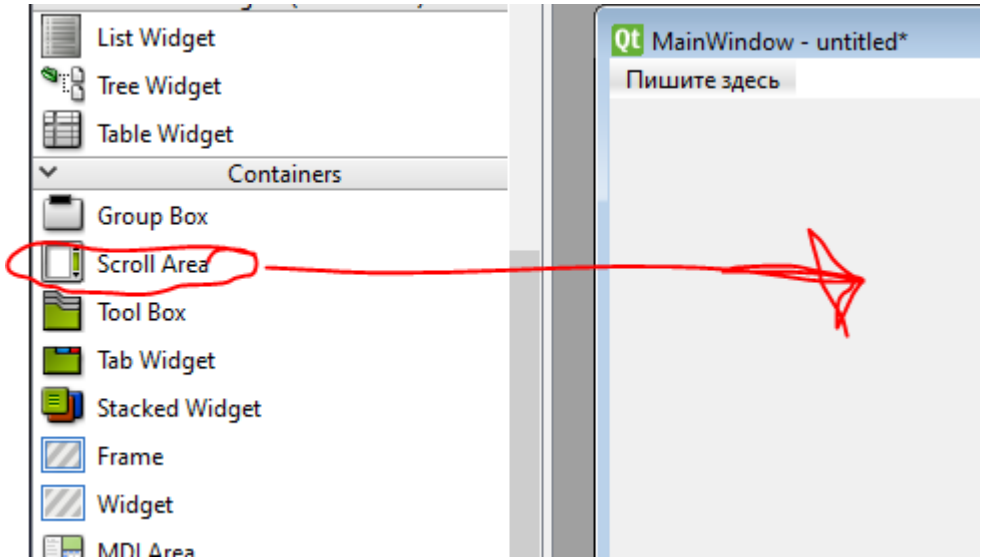
- 1. Т.к. в исходной форме присутствуют классы QMenuBar (верхнее меню), и внизу есть StatusBar, то по этим признакам, можем определить, что окно у нас наследуется от класса QMainWindow.
- 2. Для создания QMainWindow в QtDesigner выбираем:

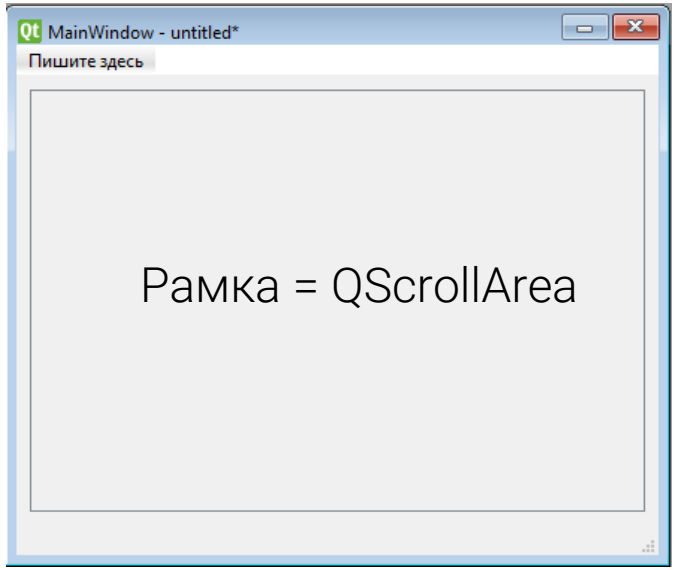




Создание GUI-формы

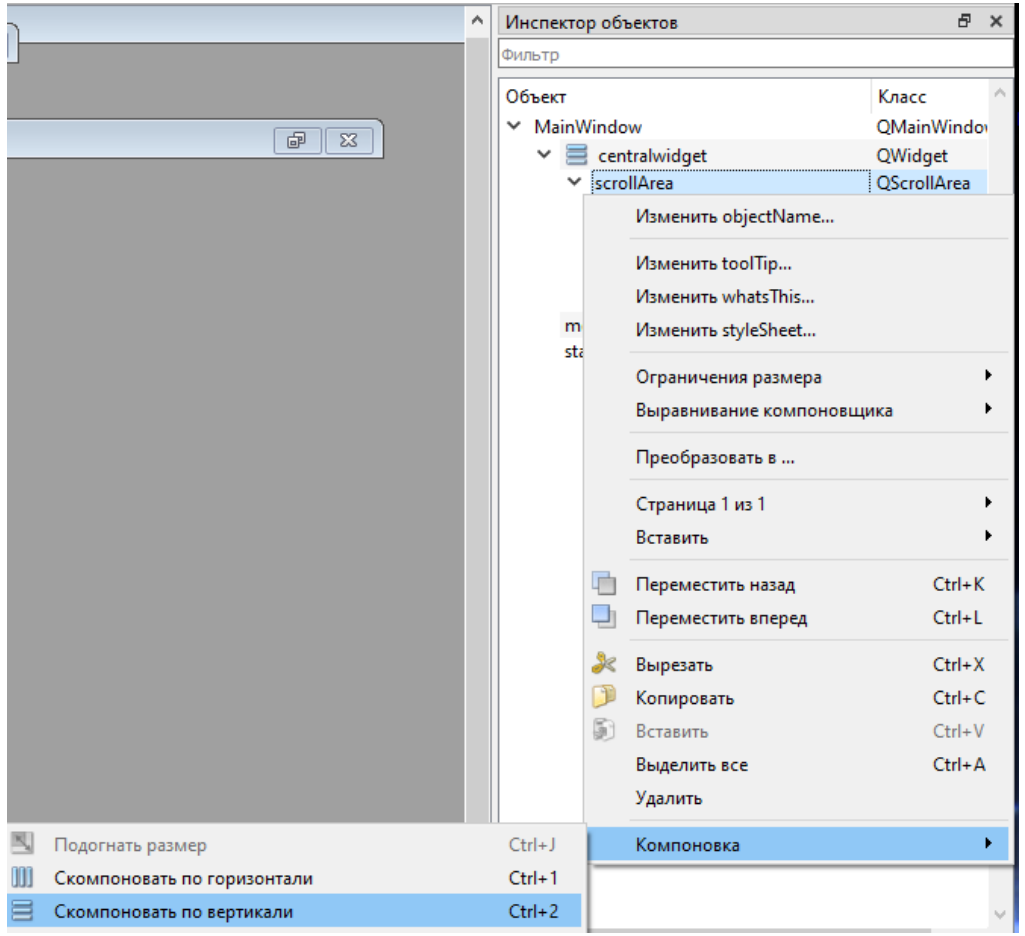
- 1. Для возможности прокрутки окна при его сильном уменьшении, основным виджетом установим QScrollArea. Для этого из левой панели, где расположены виджеты, необходимо на нашу форму перетащить мышкой виджет QScrollArea.
- 2. Далее для того, чтобы QScrollArea, растянулся на всю ширину окна, в окне «Инспектор объектов», выбираем объект MainWindow -> ПКМ -> Компоновка -> «Скомпоновать по горизонтали». Мы должны увидеть, что QScrollArea занял всё пространство MainWindow. И теперь наш виджет изменяет свой размер, вместе с изменением размера окна.





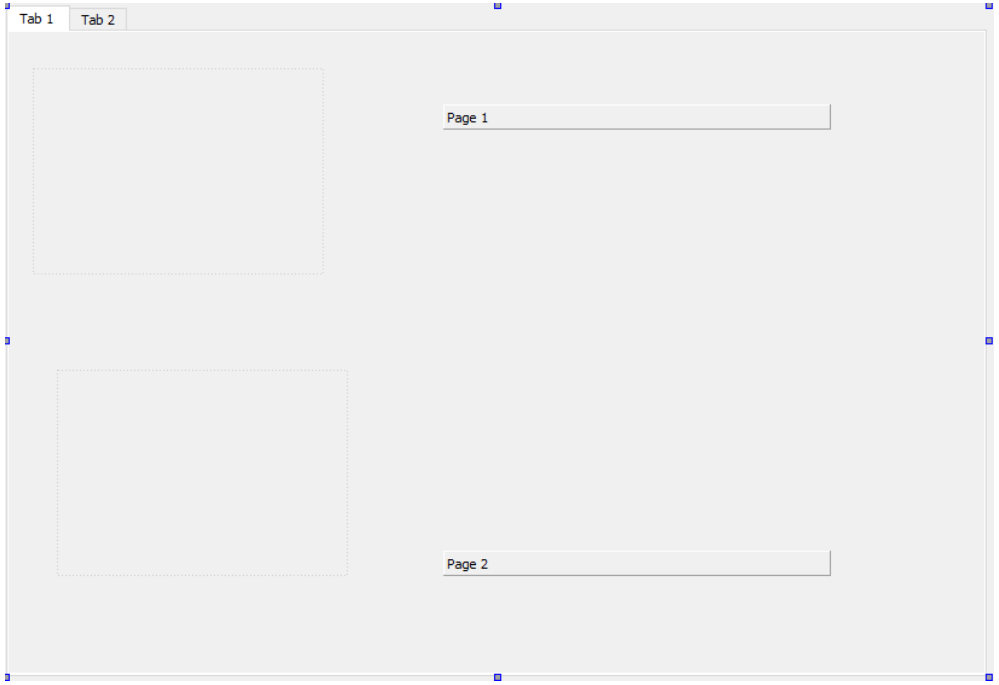
Создание GUI-формы

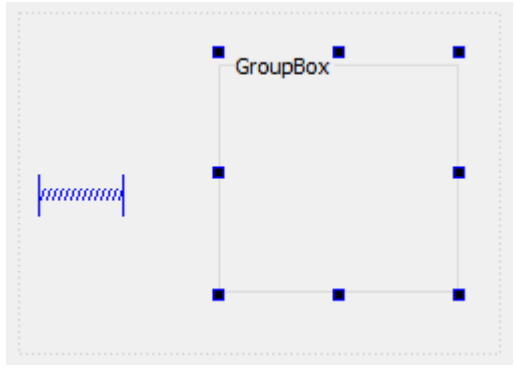
- Следующим виджетом по иерархии слоев, является QTabWidget. Переносим его на QScrollArea и устанавливаем для QScrollArea вертикальную компоновку.
- Добавить вкладки на QTabWidget, можно через контекстное меню, щелчком ПКМ по элементу.
- Переименовать вкладки, можно в окне «Редактор свойств», в категории свойств «QTabWidget» (в самом низу настроек)
- Установим для QTabWidget минимальный размер формы, в том же окне «Редактор свойств», необходимо найти свойство minimumSize и в поле Ширина и Высота установить значения 780 и 550 соответственно



Создание GUI-формы

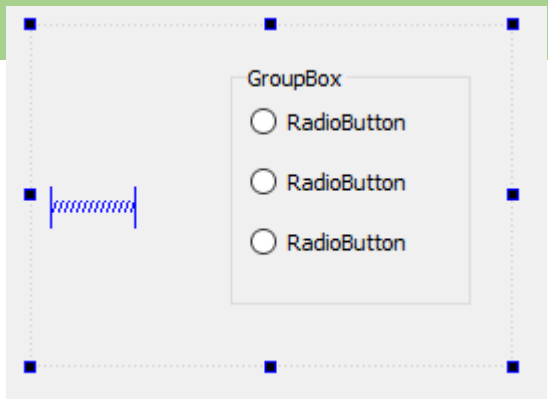
- После установки минимального размера QTabWidget, можем увидеть, что начал отрабатывать виджет QScrollArea, давая возможность просмотра QTabWidget, даже если окно, очень сильно сжато.
- Далее добавляем на форму, два виджета QFrame и один виджет QToolBar.
- Вручную растягиваем виджеты, компоновщик будет определён после их наполнения.
- В первый QFrame добавляем Horizontal Spacer (в левую часть) и QGroupBox (в правую).
- Компоновщик пока не устанавливаем.

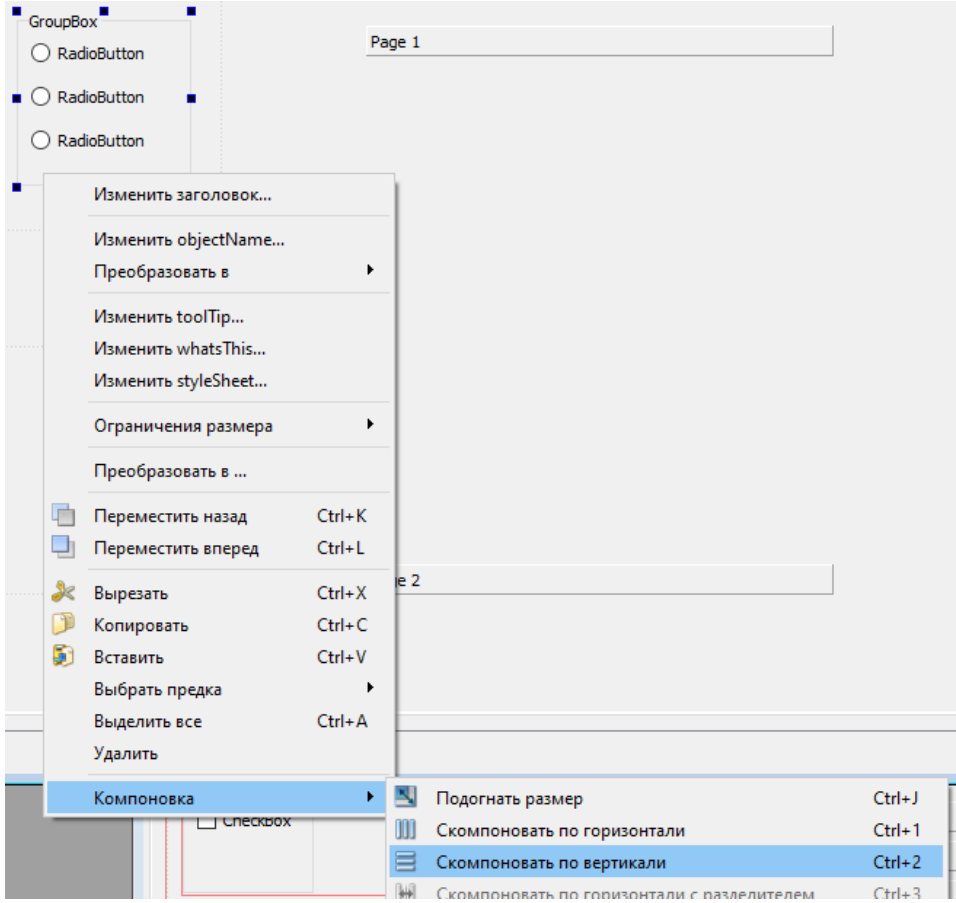


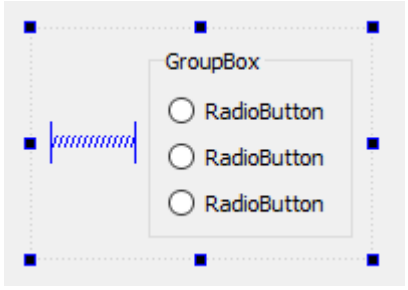


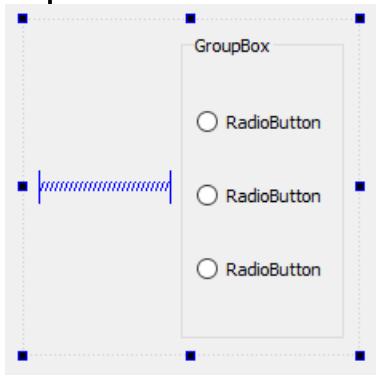
Создание GUI-формы

- Внутрь QGroupBox перетаскиваем три виджета QRadioButton.
- Теперь для QGroupBox, через его контекстное меню, устанавливаем вертикальную компоновку.
- Далее для QFrame, с которым работаем устанавливаем горизонтальную компоновку.
- QFrame примет примерную форму с рисунка и все элементы будут в нем изменять размер или расстояние при попытке его растянуть или сжать.



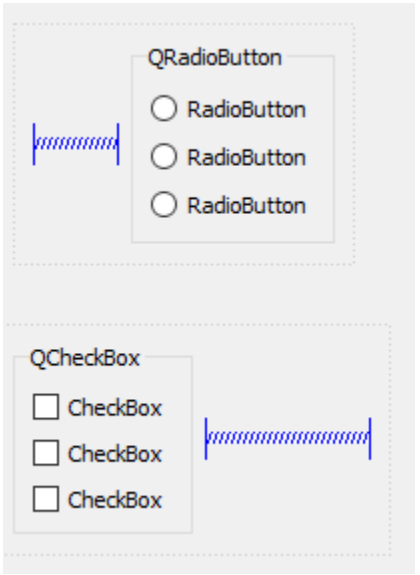


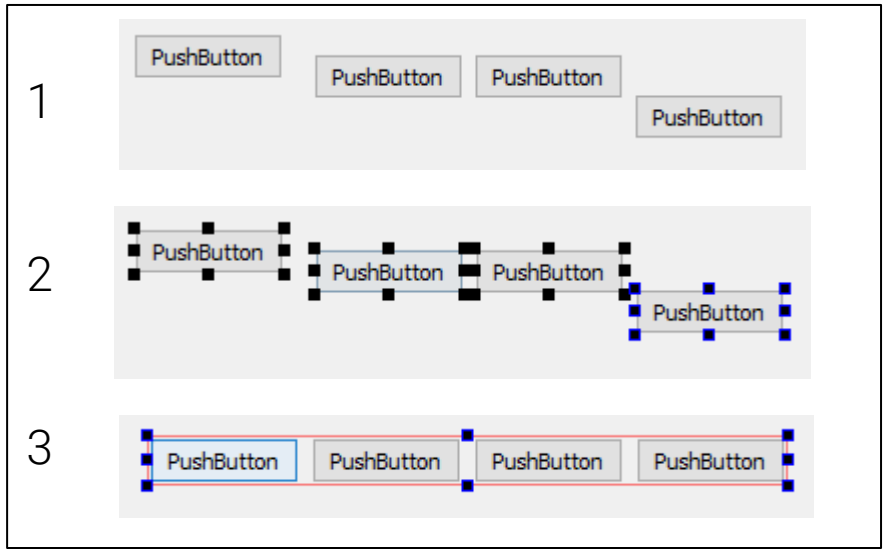




Создание GUI-формы

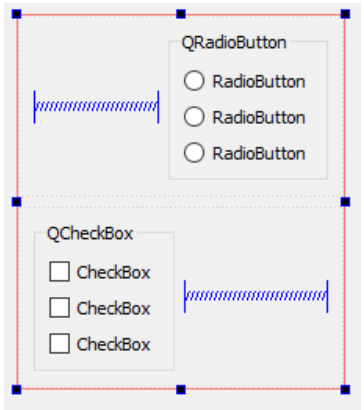
- Повторяем тоже самое для второго QFrame, только вместо QRadioButton, в QGroupBox, устанавливаем QCheckBox и Horizontal Spacer устанавливаем с правой стороны.
- Изменять многие текстовые надписи на виджетах, можно двойным щелчком мыши по ним или в окне «Редактор свойств».
- После наполнения второго QFrame, должно получиться следующее.
- Далее добавим на форму 4 QPushButton. Выделим их через удержание клавиши Ctrl и щелчком по ним. И через контекстное меню установим для них горизонтальную компоновку.

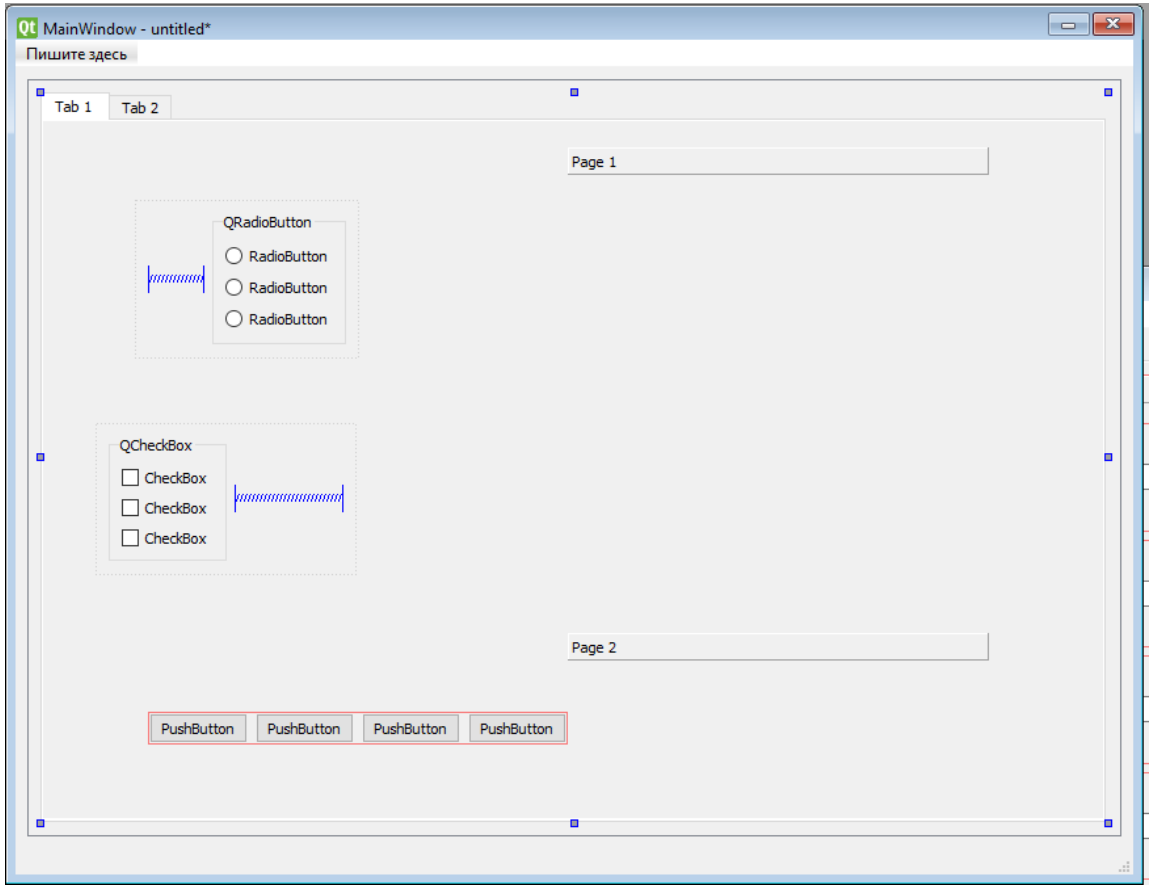




Создание GUI-формы

- На текущий момент должно быть нечто аналогичное

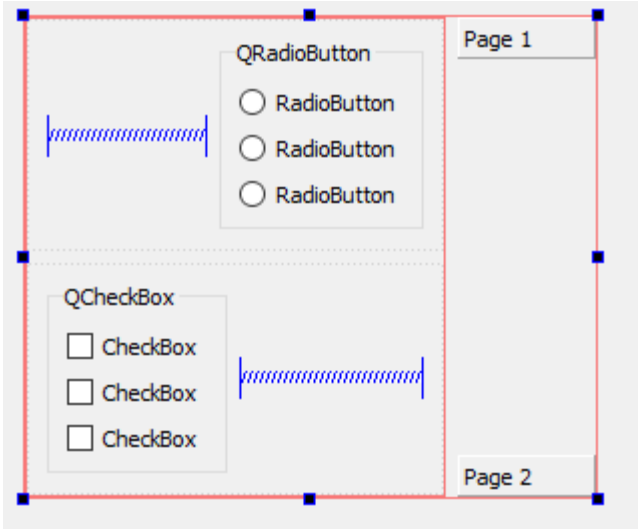


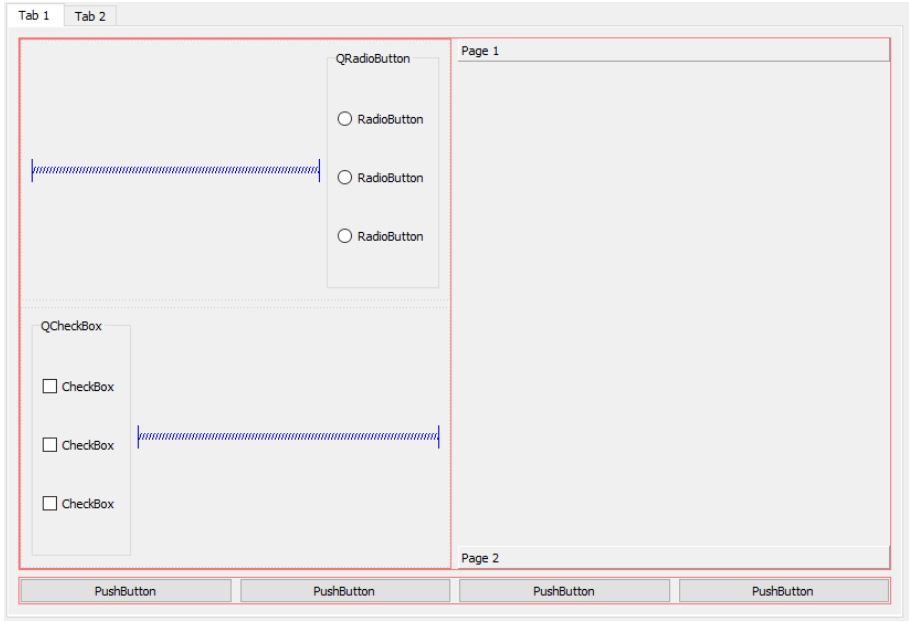


- Далее выделяем два QFrame, с которыми работали ранее и устанавливаем для них двоих вертикальную компоновку.
- НЕ ПУТАТЬ КОМПОНОВКУ МЕЖДУ ЭЛЕМЕНТАМИ И КОМПОНОВКУ ВНУТРИ ВИДЖЕТА

Создание GUI-формы

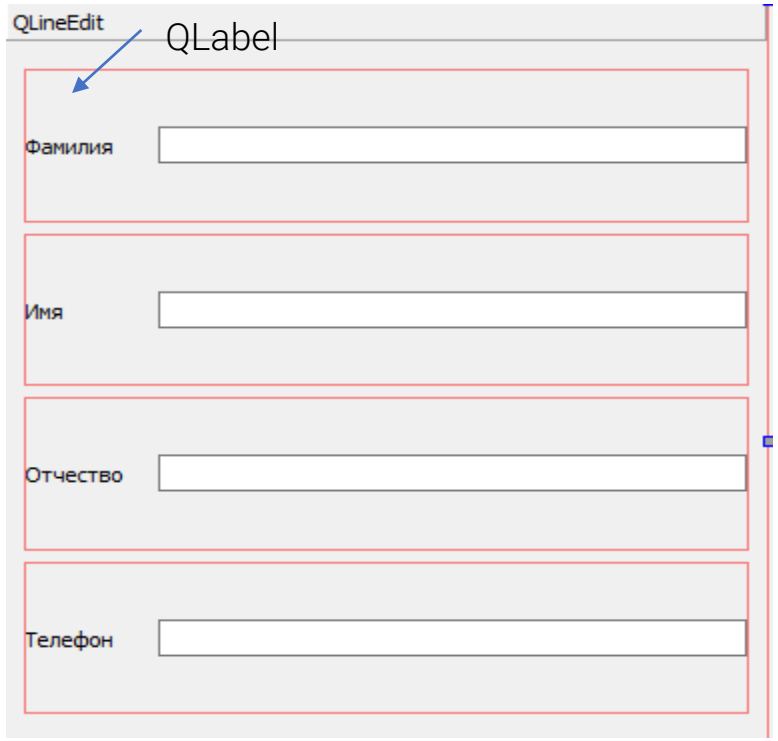
- Выделяем получившуюся компоновку между QFrame и QToolBox и устанавливаем для них горизонтальную компоновку.
- Далее для Tab1 щелчком ПКМ по названию вкладки устанавливаем вертикальную компоновку.
- Наша форма нормально растянулась и приняла вид из задания.



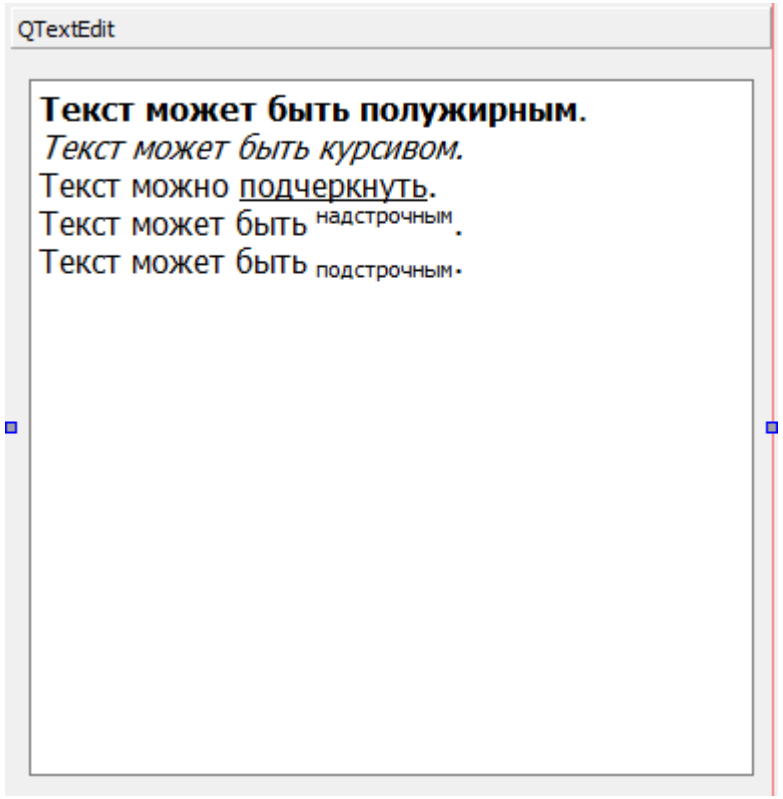


Создание GUI-формы

- САМОСТОЯТЕЛЬНО:
- Заполнить Вкладки QToolBox, используя Горизонтальную или Вертикальную компоновки. В названии вкладок указаны использующиеся внутри виджеты. Текст в окнах редакторов повторять не надо.



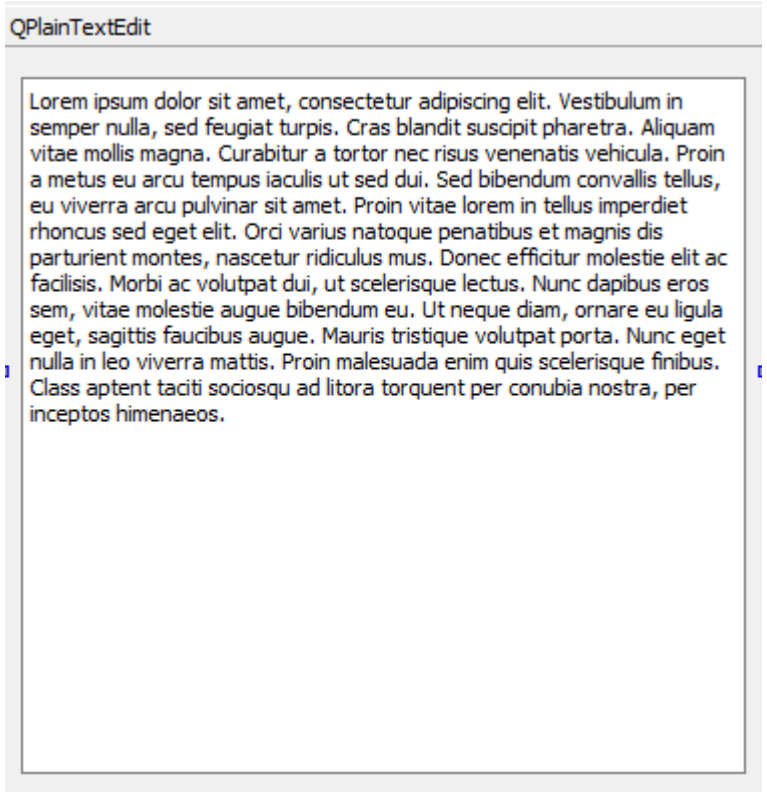
ВКЛАДКА 1



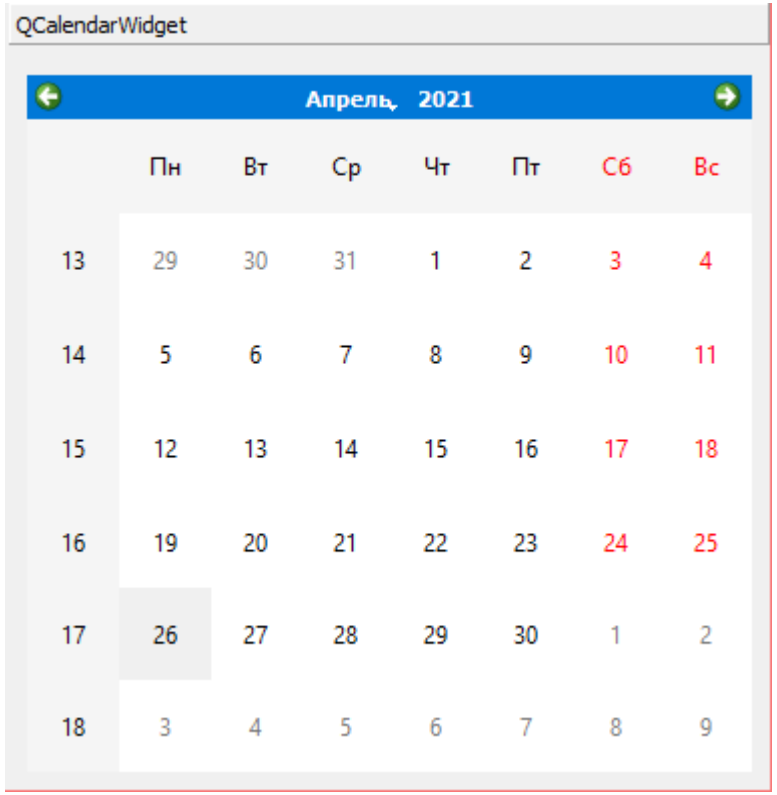
ВКЛАДКА 2

Создание GUI-формы

- САМОСТОЯТЕЛЬНО:
- Заполнить Вкладки QToolBox, используя Горизонтальную или Вертикальную компоновки.



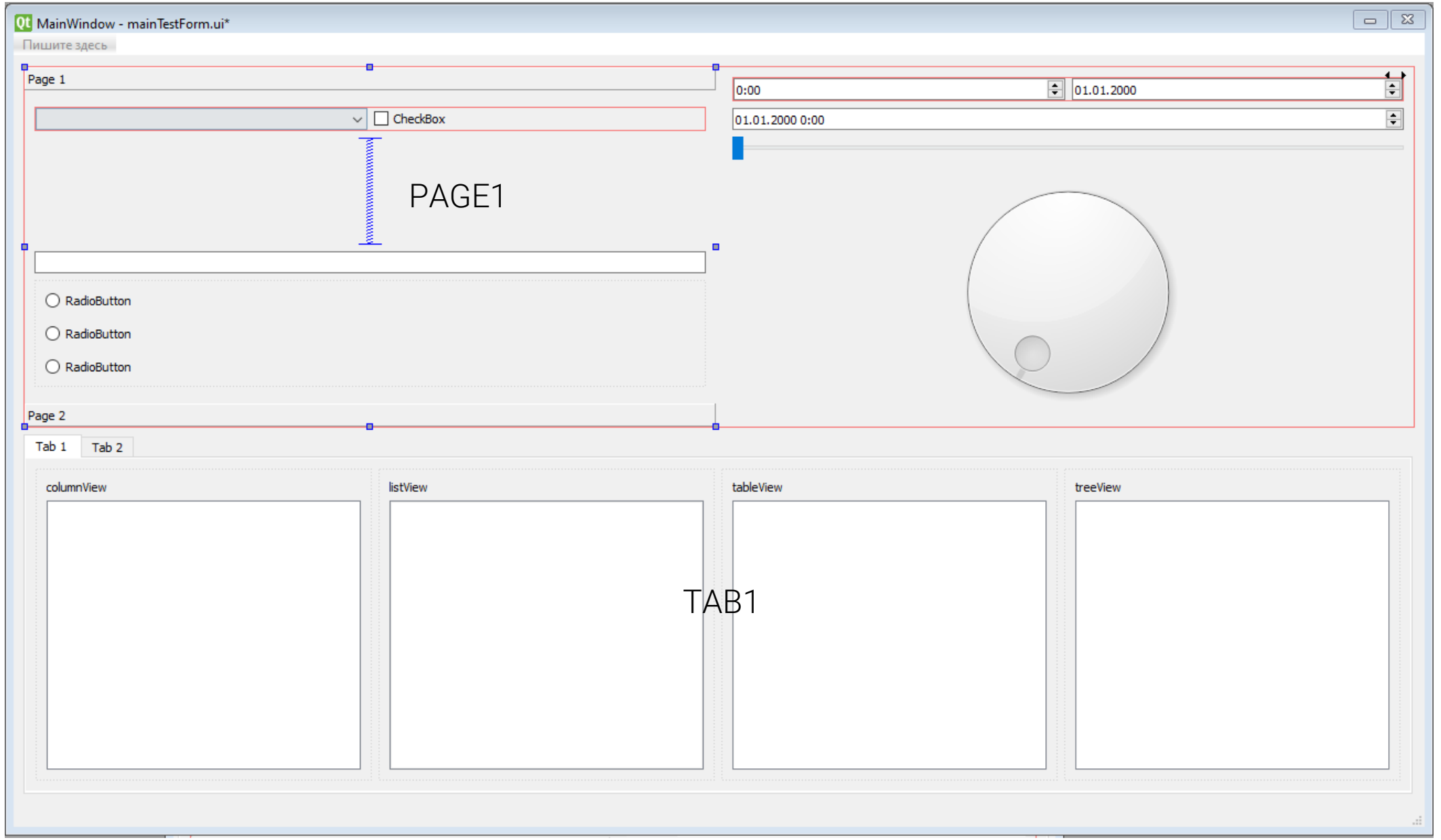
ВКЛАДКА 3



ВКЛАДКА 4

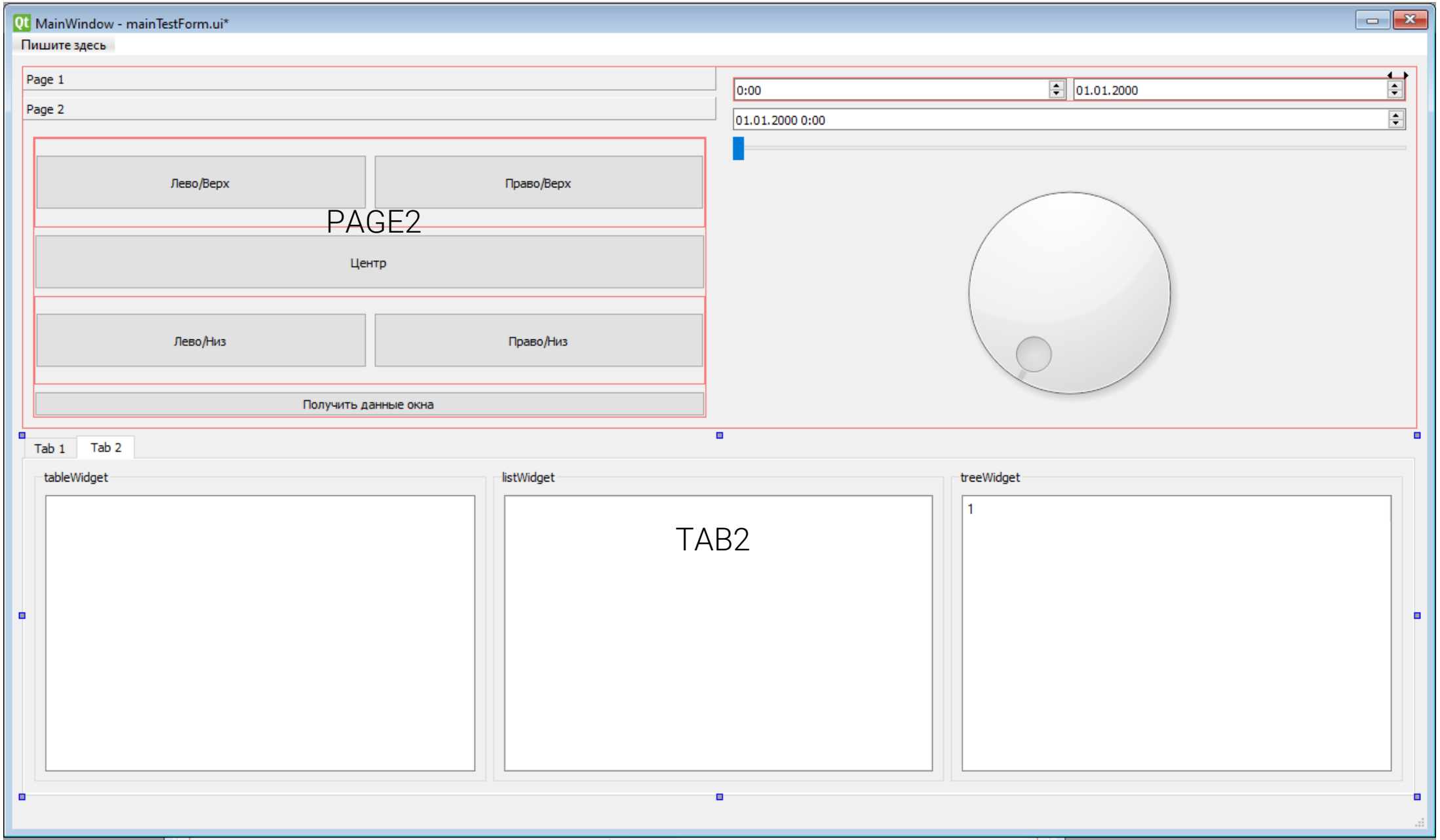
Дополнительно

- САМОСТОЯТЕЛЬНО:
- Повторить форму.



Дополнительно

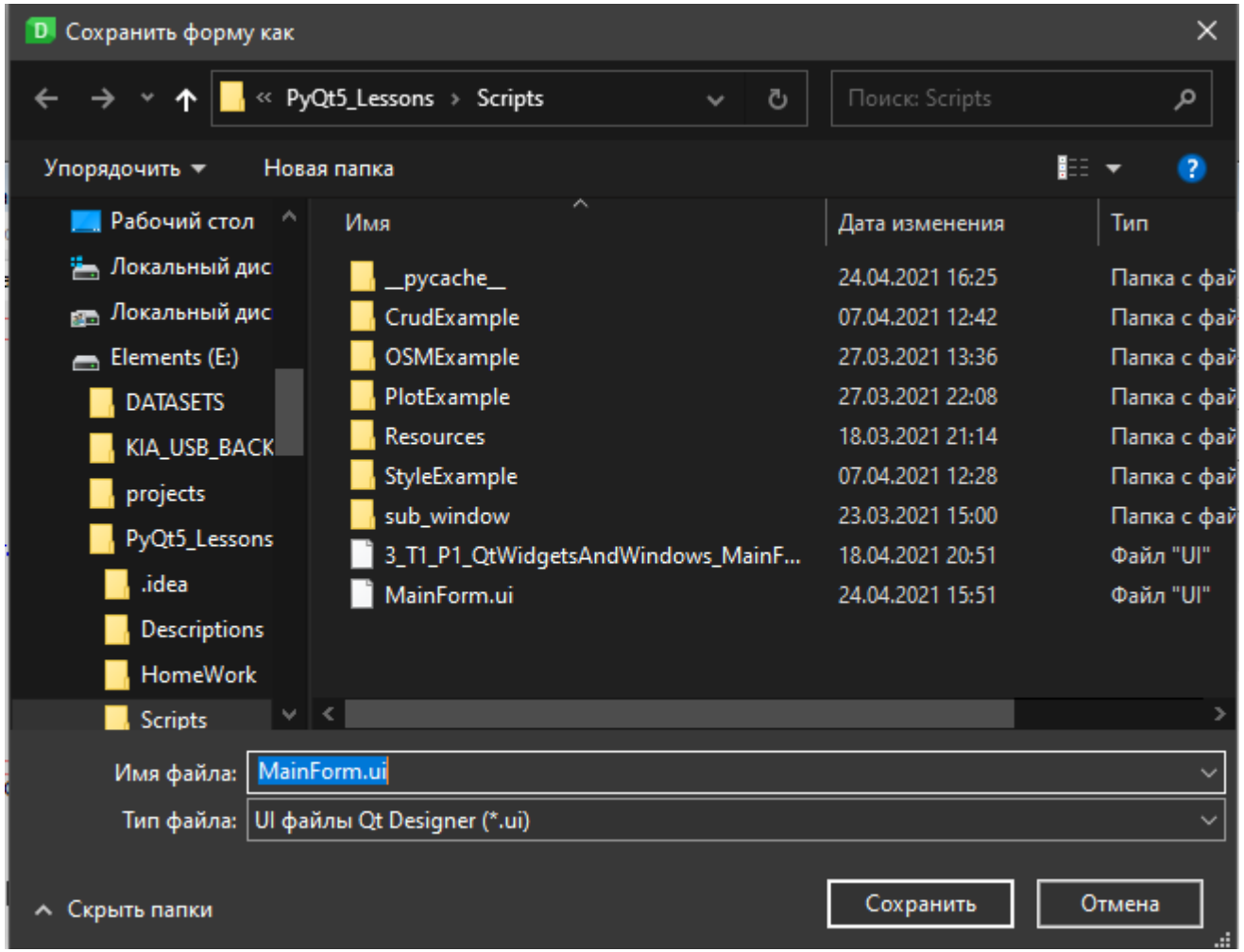
- САМОСТОЯТЕЛЬНО:
- Повторить форму.



2. Подключение GUI-формы в программу.

PySide2-uic

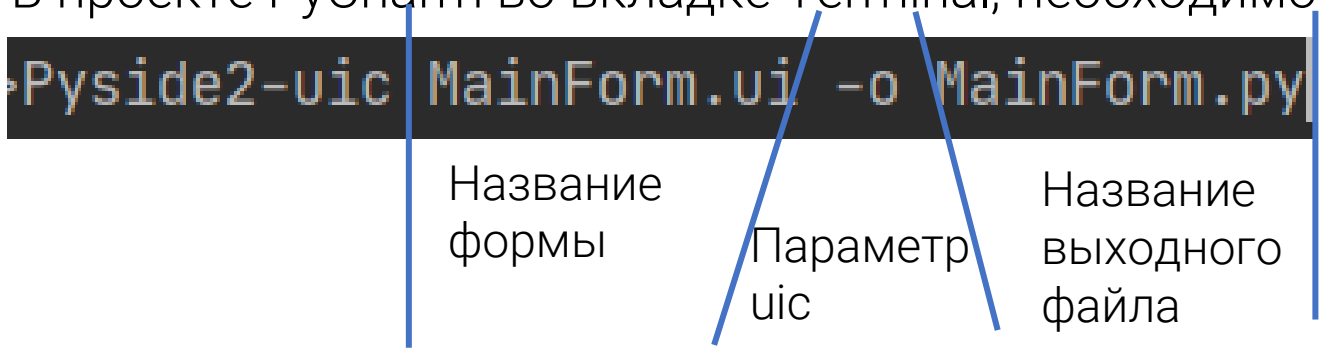
- Для работы с формой в нашей программе необходимо:
- - Сохранить файл формы в папку проекта



PySide2-uic

- Для работы с формой в нашей программе необходимо форму ui сконвертировать в py:

- - В проекте PyCharm во вкладке Terminal, необходимо ввести



- Важно!!! В терминале должна быть установлена директория, где храниться ваша форма .ui
- После выполнения команды, должен появиться файл .py с названием, как у вашей формы.

Подключение в программе

- Далее для вывод элементов нашей формы необходимо импортировать её по имени

```python
from PySide2 import QtWidgets, QtCore, QtGui
import MainForm

class MainTestWindow(QtWidgets.QMainWindow):
    def __init__(self, parent=None):
        super(MainTestWindow, self).__init__(parent)

        self.ui = MainForm.Ui_MainWindow()
        self.ui.setupUi(self)


if __name__ == "__main__":
    app = QtWidgets.QApplication()

    window = MainTestWindow()
    window.show()

    app.exec_()
```

Подключить форму, создав экземпляр класса (как правило называется self.ui) Ui_MainWindow() нашей формы в методе __init__().

Вызвать метод setupUi()

Далее все элементы формы будут доступны по их названиям (objectName), через переменную self.ui
Н-р: self.ui.pushbutton_2

ВНИМАНИЕ!!! ПРИ ИЗМЕНЕНИИ ФОРМЫ В QtDesigner И ПЕРЕСОХРАНЕНИИ ЕЁ, ДЛЯ ОТОБРАЖЕНИЯ ИЗМЕНЕНИЙ В ПРОГРАММЕ, НЕОБХОДИМО ЗАНОВО ПЕРЕКОНВЕРТИРОВАТЬ ФАЙЛ!

3. Использование QSettings.

QSettings

- Для хранения различных настроек в программе, необходимо создать экземляр класса QSettings:

```
self.settings = QtCore.QSettings("MyDataCard")
```

- В конструктор класса передаём имя нашего приложения, с помощью которого потом будем получать сохраненные настройки.

QSettings – сохранение данных

- В данном случае, переопределяем встроенный метод closeEvent (закрытие приложения).
- Для сохранения настроек у экземпляра класса self.settings вызываем метод setValue, в который на первое место передаем имя по которому можно будет найти сохраняемые данные в системе, на второе место передаются сами данные.

```
def closeEvent(self, event: QtGui.QCloseEvent) -> None:
    self.settings.setValue("Name", self.lineEditName.text())
    self.settings.setValue("Surname", self.lineEditSurname.text())
    self.settings.setValue("Telephone", self.lineEditTelephone.text())
    self.settings.setValue("EMail", self.lineEditEMail.text())
    self.settings.setValue("CheckState", self.checkBox.isChecked())
    self.settings.setValue("Size", self.saveGeometry())
```

QSettings – получение данных.

- Рассмотрим следующий метод:

```
def loadData(self):

    self.lineEditName.setText(self.settings.value("Name", "Введите имя"))
    self.lineEditSurname.setText(self.settings.value("Surname", "Введите фамилия"))
    self.lineEditTelephone.setText(self.settings.value("Telephone", "Введите телефон"))
    self.lineEditEMail.setText(self.settings.value("EMail", "Введите e-mail"))

    if self.settings.value("CheckState") == "true":
        self.checkBox.setCheckState(QtCore.Qt.Checked)
    else:
        self.checkBox.setCheckState(QtCore.Qt.Unchecked)
```

- Установка текста в виджет происходит через вызов у экземпляра self.setings метода value которому в параметры на первом месте передаём название переменной под которой у нас хранятся необходимые данные, на втором месте указываем значение по умолчанию (если приложение ещё ничего не сохранило в систему или нужные данные не найдены)