

Тема 1.

Модули Qt для создания приложений с графическим интерфейсом.

**Лекция 2. Виджеты и окна.
Иерархия классов и объектов.**



Обо мне

• ФОТО

- Харченко Владислав Алексеевич, 26 лет.
- 2012 – 2017 гг. ВГТУ. Факультет Радиотехники и Электроники.
- 2014 – 2018 гг. Фриланс.
- 2017 – 2018 гг. АО «НИИ СВТ». Программист.
- 2018 – 2021 гг. АО «НИИ СВТ». Старший программист.

Учебные вопросы

1. Размещение виджетов в приложении.
2. Основные виджеты Qt.
3. Концепция «модель - представление».
4. Иерархия классов и объектов.

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. Размещение виджетов в приложении.

Создание окна через QMainWindow:

Код для
примера,
запустить
окно можно
и без него

```
1  import sys
2  from PySide2 import QtWidgets
3
4
5  class MyWidgets(QtWidgets.QMainWindow):
6
7      def __init__(self, parent=None):
8          QtWidgets.QMainWindow.__init__(self, parent)
9          # menuBar отсутствует у QWidgets
10         self.fileMenu = self.menuBar().addMenu('File')
11         self.fileMenu.addAction("Open")
12         # toolBar отсутствует у QWidgets
13         self.toolBar = self.addToolBar("Edit")
14         self.toolBar.addAction("Edit")
15         # statusBar отсутствует у QWidgets
16         self.appStatusBar = self.statusBar()
17         self.appStatusBar.showMessage("Status: Ok!")
18
19
20  if __name__ == "__main__":
21      app = QtWidgets.QApplication(sys.argv)
22
23      myWindow = MyWidgets()
24      myWindow.show()
25
26      app.exec_()
```

Создание окна через QWidget:

```
1  import sys
2  from PySide2 import QtWidgets
3
4
5  class MyWidgets(QtWidgets.QWidget):
6
7      def __init__(self, parent=None):
8          QtWidgets.QWidget.__init__(self, parent)
9
10
11  if __name__ == "__main__":
12      app = QtWidgets.QApplication(sys.argv)
13
14      myWindow = MyWidgets()
15      myWindow.show()
16
17      app.exec_()
```

Варианты добавления виджетов

- Напрямую:

```
self.pushButton = QtWidgets.QPushButton("Кнопка", self)
```

- Через компоновку:

QMainWindow

```
centralWidget = QtWidgets.QWidget()
self.setCentralWidget(centralWidget)
layout = QtWidgets.QHBoxLayout()
self.abc = QtWidgets.QPushButton("abc")
layout.addWidget(self.abc)



centralWidget.setLayout(layout)
```

QWidget

```
layout = QtWidgets.QHBoxLayout()
self.abc = QtWidgets.QPushButton("abc")
layout.addWidget(self.abc)
self.setLayout(layout)
```

- Через QtDesigner.

2. Основные виджеты Qt.

Внешний вид	Название	Примечание	Наследование
	QPushButton	Командная кнопка.	QWidget -> QAbstractButton -> QPushButton
	QToolButton	QTollButton. (Создается в QToolBar)	QWidget -> QAbstractButton -> QToolButton
	QRadioButton	Переключатель.	QWidget -> QAbstractButton -> QRadioButton
	QCheckBox	«Флажок»	QWidget -> QAbstractButton -> QCheckBox
	QCommandLinkButton	Ссылка на команду	QWidget -> QAbstractButton -> QPushButton -> QCommandLinkButton
	QDialogButtonBox	Макет группы кнопок	QWidget -> QDialogButtonBox
	QListView QListWidget QUndoView	Список элементов	QWidget -> QFrame -> QAbstractScrollArea -> QAbstractItemView -> QListView -> QListWidget/QUndoView
	QTreeView QTreeWidget	Древовидный список элементов	QWidget -> QFrame -> QAbstractScrollArea -> QAbstractItemView -> QTreeView -> QTreeWidget

Внешний вид	Название	Примечание	Наследование
	QTableView QTableWidget	Таблица	QWidget -> QFrame -> QAbstractScrollArea -> QAbstractItemView -> QTableView -> QTableWidget
	QColumnView	Таблица столбцов	QWidget -> QFrame -> QAbstractScrollArea -> QAbstractItemView -> QColumnView
	QGroupBox	Группировка виджетов с заголовком	QWidget -> QGroupBox
	QScrollArea	Зона с прокруткой	QWidget -> QFrame -> QAbstractScrollArea -> QScrollArea
	QToolBox	«Аккордеон»	QWidget -> QFrame -> QToolBox
	QTabWidget	Зона с вкладками	QWidget -> QTabWidget
	QStackedWidget	Стек окон	QWidget -> QFrame -> QStackedWidget
	QFrame	Группировка виджетов	QWidget -> QFrame

Внешний вид	Название	Примечание	Наследование
	QWidget	Базовый класс	QObject & QPaintDevice -> QWidget
	QMdiArea	Окна MDI (многодокументный интерфейс)	QWidget -> QFrame -> QAbstractScrollArea -> QMdiArea
	QDockWidget	«Плавающее» окно	QWidget -> QDockWidget
	QComboBoxWidget QFontComboBox	Выпадающий список	QWidget -> QComboBox -> QFontComboBox
	QLineEdit QTextEdit QPlainTextEdit QTextBrowser	Зона редактирования текста	QWidget -> QLineEdit QWidget -> QFrame -> QAbstractScrollArea -> QTextEdit QWidget -> QFrame -> QAbstractScrollArea -> QPlainTextEdit QWidget -> QFrame -> QAbstractScrollArea -> QTextEdit -> QTextBrowser
	QSpinBox QDoubleSpinBox	Счётчик с увеличением	QWidget -> QAbstractSpinBox -> QSpinBox QWidget -> QAbstractSpinBox -> QDoubleSpinBox
	QTimeEdit QDateEdit QDateTimeEdit	Работа с временем и датой	QWidget -> QAbstractSpinBox -> QDateTimeEdit -> QTimeEdit QWidget -> QAbstractSpinBox -> QDateTimeEdit -> QDateEdit QWidget -> QAbstractSpinBox -> QDateTimeEdit ->

Внешний вид	Название	Примечание	Наследование
	QDial	Контроль диапазона	QWidget -> QAbstractSlider -> QDial
	QScrollBar	Полоса прокрутки	QWidget -> QAbstractSlider -> QScrollBar
	QSlider	Слайдер	QWidget -> QAbstractSlider -> QSlider
	QKeySequenceEdit	Проверка нажатия	QWidget -> QKeySequenceEdit
	QLabel	Подпись	QWidget -> QFrame -> QLabel
	QGraphicsView	Отображение графики (объектов QGraphicsScene)	QWidget -> QFrame -> QAbstractScrollArea -> QGraphicsView
	QCalendarWidget	Календарь	QWidget -> QCalendarWidget
	QLCDNumber	«Цифровое табло»	QWidget -> QFrame -> QLCDNumber
	QProgressBar	Шкала выполнения	QWidget -> QProgressBar
	QOpenGLWidget	Отображение 2D/3D	QWidget -> QOpenGLWidget

3. Концепция «модель - представление».

Смысл концепции

- **Модель** – «обёртка» над данными, которая позволяет управлять и взаимодействовать с ними.
- **Представление** – служит для отображения элементов на экране. Одну модель можно установить сразу в несколько представлений.
- **Модель выделения** – управляет выделением данных в модели.
- **Промежуточная модель** – «прослойка» между основной моделью и представлением. Служит для сортировки и фильтрации данных без изменения порядка следования элементов в базовой модели.
- **Делегат** – обеспечивает компонент для вывода и редактирования данных.

Виды моделей

- QStringListModel – список строк. Отображение через QListView, QComboBox;
- QStringListModel – двумерная таблица. Отображение через QTableView, QTreeView.
- QStandardItem – создание элементов и вложенных структур.

Виды представлений

- QListView – простой список.
- QTableView – таблица.
- QTreeView – иерархический список.

4. Иерархия классов и объектов.



* изображение актуально для версии Qt=4

Классы основных модулей Qt

- **QtCore:** <https://doc.qt.io/qt-5/qtcore-module.html>
- **QtGui:** <https://doc.qt.io/qt-5/qtgui-module.html>
- **QtWidgets:** <https://doc.qt.io/qt-5/qtwidgets-module.html>
- **QtNetwork:** <https://doc.qt.io/qt-5/qtnetwork-module.html>
- **QtSQL:** <https://doc.qt.io/qt-5/qtsql-module.html>

Спасибо за внимание!