

# Виджеты

Основные виды

Дмитрий Фёдоров  
Руководитель отдела разработки ПО



# Проверка связи





## Если у вас нет звука:

- убедитесь, что на вашем устройстве и на колонках включён звук
- обновите страницу вебинара или закройте страницу и заново присоединитесь к вебинару
- откройте вебинар в другом браузере
- перезагрузите компьютер (ноутбук) и зайдите снова



## Поставьте в чат:

-  если меня видно и слышно
-  если нет

# Дмитрий Фёдоров

О спикере:

- более 10 лет в разработке авиационных систем
- возглавляет отдел разработки ПО в НИЦ «ИРТ»



# Вспоминаем прошрое занятие

**Вопрос:** как справились с домашней работой?



# Вспоминаем прошрое занятие

**Вопрос:** что такое Qt Designer?



# Вспоминаем прошрое занятие

**Вопрос:** что такое Qt Designer?

**Ответ:** ПО для разработки пользовательских интерфейсов



# Вспоминаем прошлое занятие

**Вопрос:** при помощи какого инструмента можно достичь пропорционального отображения виджетов при растягивании окна?



# Вспоминаем прошлые занятия

**Вопрос:** при помощи какого инструмента можно достичь пропорционального отображения виджетов при растягивании окна?

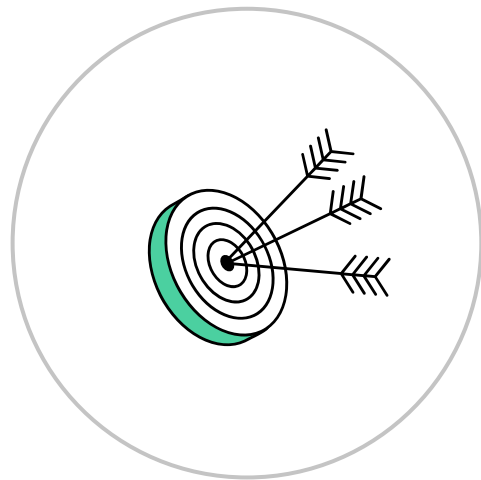
**Ответ:** компоновка





# Цели занятия

- Поговорим о классе QWidget
- Изучим наиболее часто используемые виджеты



# План занятия

1

[Класс QWidget](#)

2

[Элементы ввода](#)

3

[Элементы отображения](#)

4

[Кнопки](#)

5

[Контейнеры](#)

6

[Другие элементы](#)

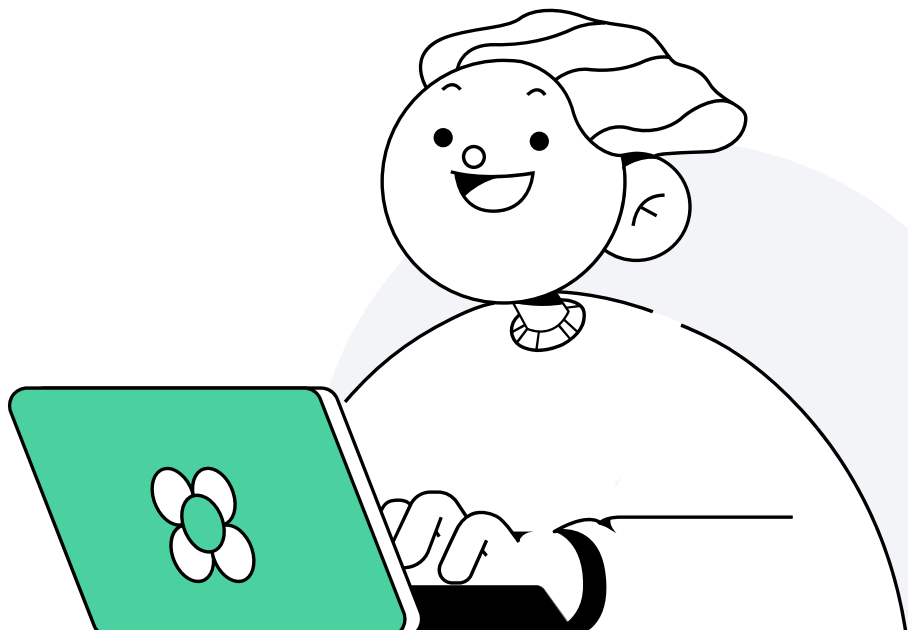
7

[Итоги](#)

8

[Домашнее задание](#)

\* Нажмите на нужный раздел для перехода



# Класс QWidget



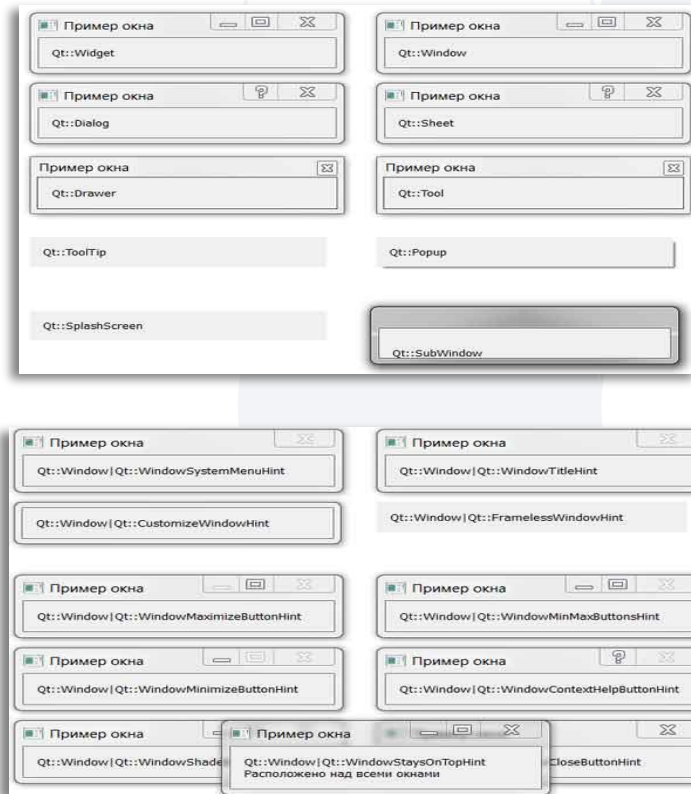
1

# Класс QWidget

- Класс QWidget — базовый для всех объектов пользовательского интерфейса
- QWidget унаследован от класса QObject, а значит — может использовать механизм сигналов/слотов и механизм объектной иерархии

# Конструктор QWidget

- Класс QWidget и большинство унаследованных от него классов имеют конструктор с двумя параметрами
- Первый параметр — указатель на родительский виджет. Если он равен нулю, то этот виджет верхний в иерархии
- Второй параметр служит для задания свойств окна. С его помощью можно управлять внешним видом окна и режимом отображения: чтобы окно не перекрывалось другими окнами и т. д.



# Методы QWidget

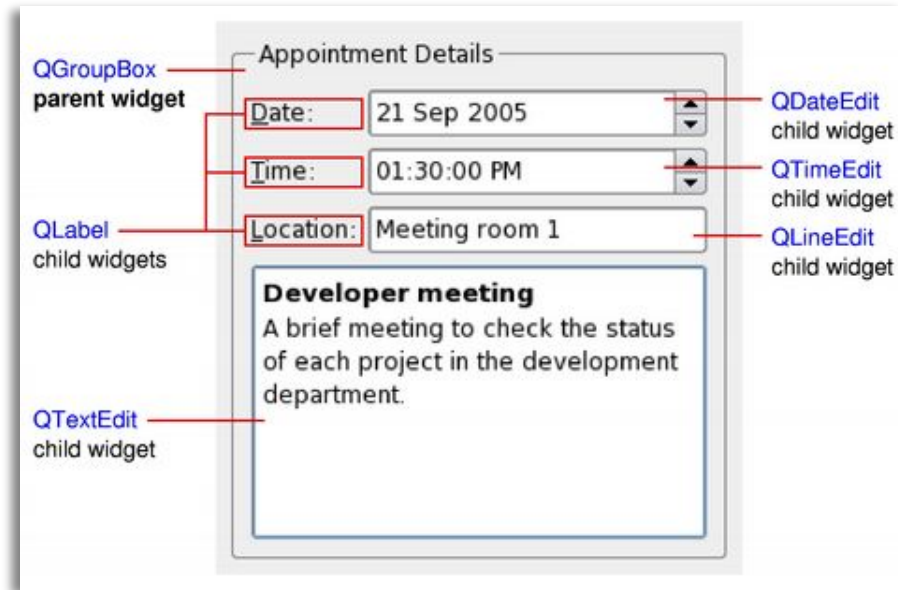
[`setEnabled\(\)`](#) — устанавливает виджет в доступное (enabled) или недоступное (disabled) состояние

[`mousePressEvent\(\)`](#), `mouseMoveEvent()`, `mouseReleaseEvent()` или `mouseDoubleClickEvent()` — реагирование на события мыши

[`setGeometry\(\)`](#) — позволяет изменить размеры и геометрию виджета. `move()` и `resize()` изменяют их по отдельности

[`setCursor\(\)`](#) — позволяет изменить курсор мыши на виджете

[`setMinimumSize\(\)`](#), [`setMaximumSize\(\)`](#) — устанавливают минимальные и максимальные размеры виджета








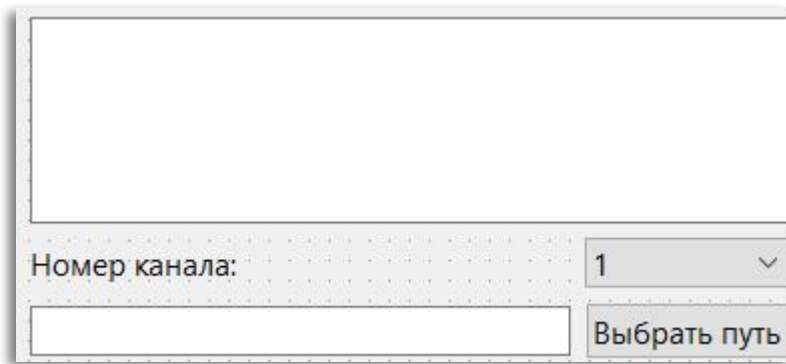
# Элементы ввода



2

# Элементы ввода

-  Line Edit
-  Text Edit
-  Plain Text Edit
-  Spin Box
-  Double Spin Box



Номер канала: 1

Выбрать путь

Большая часть элементов ввода может работать с буфером обмена и поддерживать технологию перетаскивания (drag & drop), что избавляет разработчика от дополнительной реализации. Текст можно выделять с помощью мыши, клавиатуры и контекстного меню



# QlineEdit

QlineEdit — однострочное текстовое поле.



## Основные методы:

- метод `text()` — возвращает текст, находящийся в виджете
- сигнал `textChanged()` — испускается при изменении текста
- сигнал `textEdited()` — испускается, если пользователь изменил текст в GUI
- метод `setReadOnly(bool)` — устанавливает режим только для чтения
- метод `setText()` — служит для программной установки текста
- метод `setEchoMode()` с флагом `Password` — позволяет скрыть вводимые символы

# QTextEdit

[QTextEdit](#) — редактор текста. Позволяет осуществлять просмотр и редактирование как простого текста, так и текста в формате HTML.



## Основные методы:

- методы аналогичны [QLineEdit](#)
- метод [append\(\)](#) позволяет добавлять текст в конец текущего
- [find\(\)](#) позволяет осуществлять поиск строки в тексте

# QSpinBox

[QSpinBox](#) — счётчик. Позволяет пользователю получить доступ к ограниченному диапазону чисел, который настраивается в свойствах этого виджета.



## Основные методы:

- [setRange\(\)](#) — задаёт диапазон значений
- [value\(\)](#) — возвращает текущее значение
- [setValue\(\)](#) — задаёт значение
- сигнал [valueChanged\(\)](#) испускается, как только значение изменится — на самом деле испускаются два сигнала: один с параметром `int`, другой с параметром [QString](#)

# Другие элементы ввода

- [QDateTimeEdit](#) — элемент ввода даты
- [QDoubleSpinBox](#) — счётчик для ввода чисел double
- [QValidator](#) — мощный класс, позволяющий проверять ввод данных в виджеты

Мы не будем рассматривать QValidator на курсе, но пару слов сказать стоит. QValidator основывается на регулярных выражениях и является одной из основ безопасного программирования. Ведь он ограничивает действия пользователей, а они, как мы с вами знаем, всегда хотят что-нибудь сломать )

# Элементы отображения



3

# QListWidget

[QListWidget](#) — это виджет списка, предоставляющий пользователю возможность выбора одного или нескольких элементов.




## Основные методы:

- [addItem\(\)](#) добавляет элемент списка. Можно добавить как строку, так и объект класса [QListWidgetItem](#)
- [clear\(\)](#) — удаляет все элементы списка
- [insertItems\(\)](#) или [insertItem\(\)](#) вставляют элементы в список на определённую позицию
- [currentItem\(\)](#) — возвращает элемент, выбранный пользователем
- [setSelectionMode\(\)](#) — позволяет разрешить множественное выделение элементов
- [sortItems\(\)](#) — используется для сортировки списка

# QLabel

QLabel — виджет надписи служит для показа состояния приложения или поясняющего текста и представляет собой текстовое поле, текст которого не подлежит изменению со стороны пользователя.



Номер канала:

## Основные методы:

- `setText()` — устанавливает текст лейбла
- `setAlignment()` — определяет выравнивание лейбла

# QProgressBar

QProgressBar — индикатор выполнения, демонстрирующий ход процесса выполнения операции и заполняющийся слева направо.



Основные методы:

- setOrientation() — устанавливает ориентацию индикатора (да, они бывают вертикальными)
- setMinimum() и setMaximum() — устанавливают минимальные и максимальные значения
- setValue() — устанавливает текущее значение
- format() — устанавливает формат отображения. %p — в процентах, %v — в абсолютных числах, %m — отображает количество шагов



# Кнопки

4

# Виды кнопок

- **Normal Button** (обычная кнопка) — соответствует той самой кнопке, которую мы привыкли видеть в большинстве случаев. После отпускания кнопка всегда возвращается в своё исходное положение
- **Toggle Button** (выключатель) — может пребывать в двух состояниях: нажатом или не нажатом, которые соответствуют положениям «включено» или «выключено»
- **Flat Button** (плоская кнопка) — по своим функциональным особенностям идентична обычной кнопке. Разница лишь во внешнем виде
- **Pixmap Button** (кнопка с изображением) — представляет собой кнопку, содержащую растровое изображение

# Сигналы кнопок

- `clicked()` — отправляется при щелчке кнопкой мыши
- `pressed()` — отправляется при нажатии на кнопку мыши
- `released()` — отправляется при отпускании кнопки мыши
- `toggled()` — отправляется при изменении состояния кнопки, имеющей статус выключателя

# Опросы состояния кнопок

- `isDown()` — возвращает значение `true`, если кнопка находится в нажатом состоянии. Изменить текущее состояние может или пользователь, нажав на кнопку, или вызов метода `setDown()`
- `isChecked()` — возвращает значение `true`, когда кнопка находится во включённом состоянии. Изменить текущее состояние может или пользователь, нажав на кнопку, или вызов метода `setChecked()`

# Контейнеры



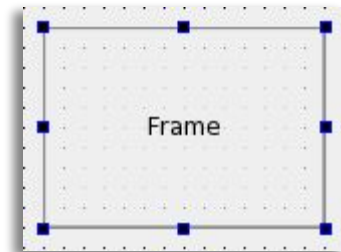
5

# Frames

Рамки используются для вложения и группировки виджетов, а также для декорирования.

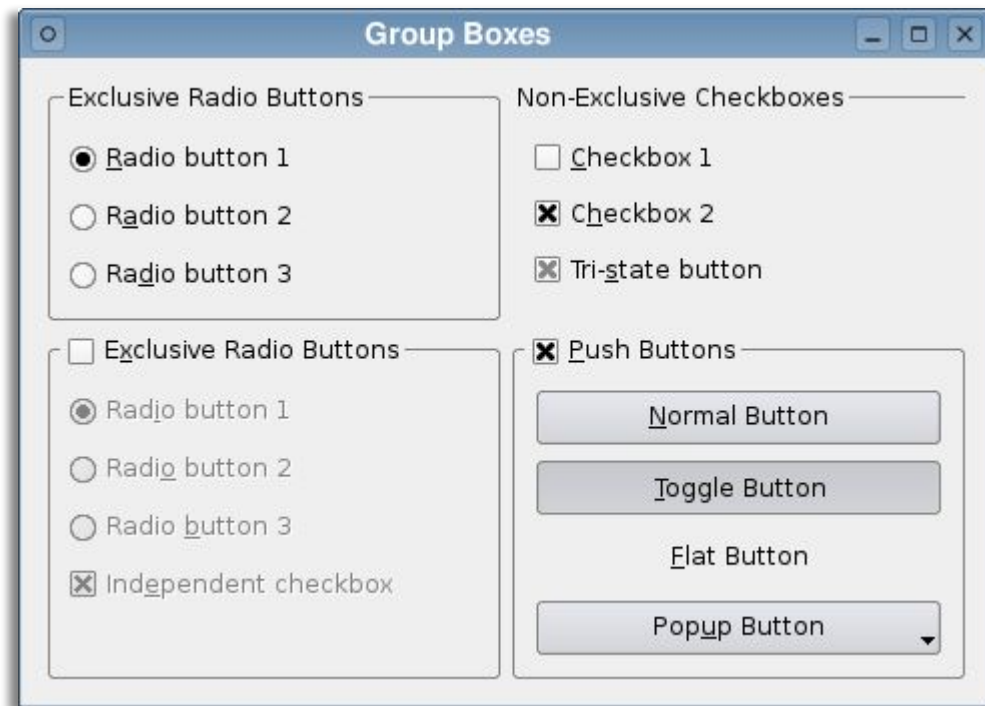
Наиболее важные свойства фреймов:

- `frameShape`
- `frameShadow`
- `lineWidth`
- `midLineWidth`



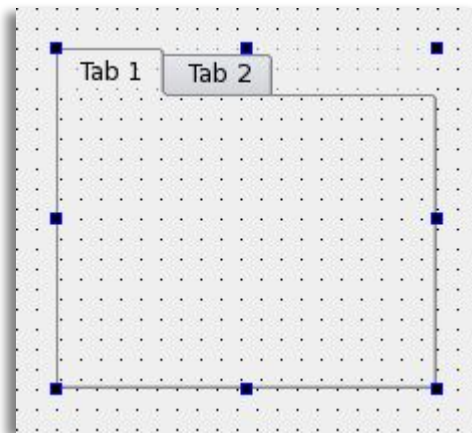
# Group Box

Групповые поля обычно используются для группировки коллекций флажков и радиокнопок с аналогичными целями



# Tab Widgets

Виджеты с вкладками позволяют разработчику разбивать содержимое виджета на различные помеченные секции, только одна из которых отображается в любой момент времени





# Другие элементы



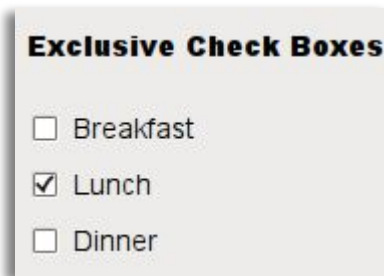
4

# QCheckBox

QCheckBox — это флажок выбора, который может быть включён (выбран) или выключен (снят).

Основные методы:

- `setCheckState()` — устанавливает состояние флажка
- `setTristate()` — определяет, является ли флажок с тремя состояниями
- `stateChanged()` — сигнал испускается, когда флажок меняет состояние



# QRadioButton

QRadioButton — это флажок выбора, который может быть включён (выбран) или выключен (снят). Флажки предоставляют пользователю выбор «один из многих» и объединяются в группы

isChecked() — показывает, выбран текущий флажок или нет

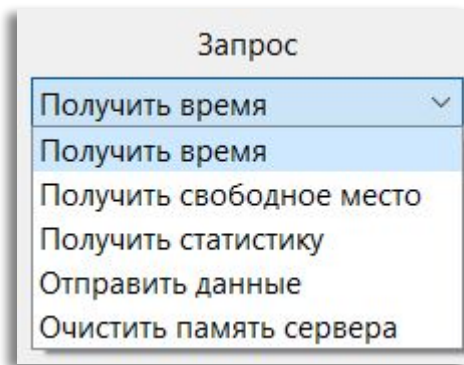


# QComboBox

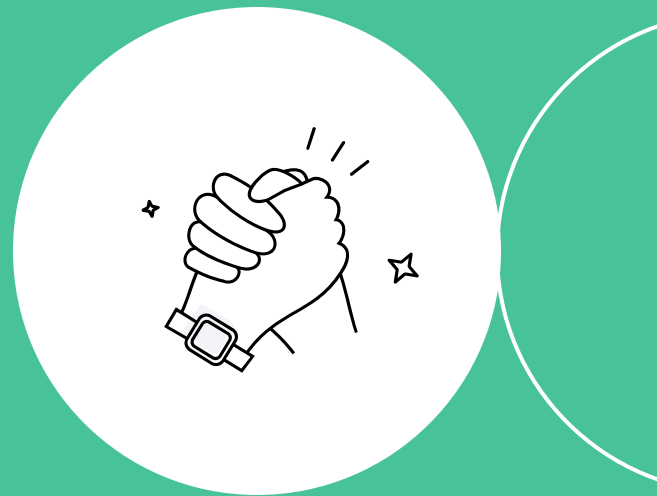
QComboBox — выпадающий список.

Основные методы:

- `currentIndex()` — возвращает текущий выбранный индекс
- `currentText()` — возвращает текущий выбранный текст
- `addItem()` — добавляет в конец списка новое значение
- `currentIndexChanged()` — сигнал испускается, когда текущий индекс изменился



# Итоги



# Итоги занятия

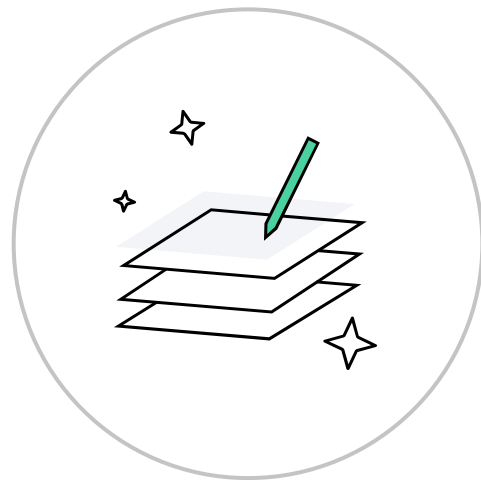
- 1 Узнали, что такое класс QWidget
- 2 Поговорили об основных виджетах
- 3 Узнали основные методы виджетов



# Домашнее задание

Давайте посмотрим ваше [домашнее задание](#).

- 1 Вопросы по домашней работе задавайте в чате группы
- 2 Задачи можно сдавать по частям
- 3 Зачёт по домашней работе ставят после того, как приняты все задачи



# Дополнительные материалы

- Описание модуля [Qt Widgets](#)





**Задавайте вопросы  
и пишите отзыв о лекции**

