



# Основы Qt

## Лекция 2 Виджеты

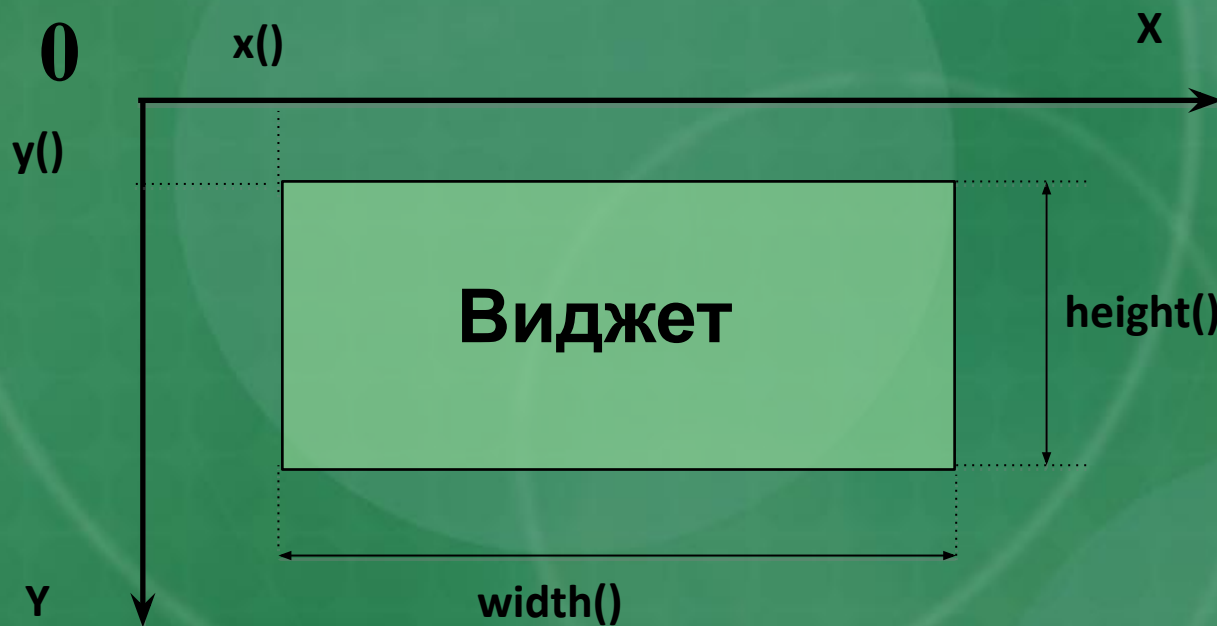


# Виджеты

- Виджеты - это части, из которых создается пользовательский интерфейс.
- Виджеты способны обрабатывать поступающие сигналы и события, отсылать свои сигналы.
- Все базовые виджеты Qt унаследованы от класса QWidget.



# Виджет





# Виджеты

| Метод   | Описание   |
|---|--|
| <code>void setEnabled(bool b)</code>                      | В зависимости от параметра делает виджет доступным или недоступным |
| <code>bool isEnabled()</code>                             | Возвращает текущее состояние виджета                               |
| <code>void setWindowTitle("")</code>                      | Устанавливает надпись в заголовок окна                             |
| <code>int height()</code>                                 | Возвращает высоту виджета  |
| <code>int width()</code>                                  | Возвращает ширину виджета  |
| <code>int x()</code>                                      | Возвращает абсциссу виджета  |
| <code>int y()</code>                                      | Возвращает ординату виджета  |
| <code>void move(int x, int y)</code>                      | Перемещает виджет в заданные координаты                            |
| <code>void resize(int w, int h)</code>                    | Изменяет размеры виджета   |
| <code>void setGeometry(int x, int y, int w, int h)</code> | Изменяет расположение и размеры виджета                            |

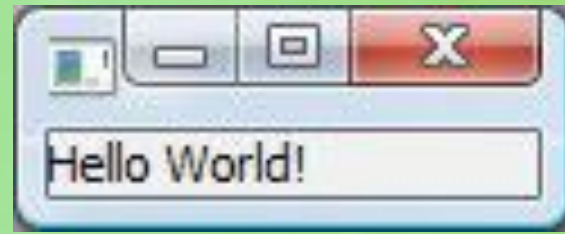


Виджеты

# ЭЛЕМЕНТЫ ОТОБРАЖЕНИЯ



# QLabel



- QLabel – виджет, способный отображать текстовую и графическую информацию

| Метод   | Описание  |
|---|---|
| <code>QLabel* lb = new QLabel("text");</code> | Пример создания   |
| <code>QLabel* lb = new QLabel();</code>       | Пример создания   |
| <code>lb-&gt;setText("text");</code>          | Устанавливает текст в виджет, текст можно форматировать html тэгами |
| <code>lb-&gt;setPixmap(QPixmap);</code>       | Устанавливает изображение в виджет                                  |



# QLCDNumber

- QLCDNumber – виджет, отображающий численные значения в виде семисегментного индикатора

| Метод  | Описание                                 |
|--|--|
| <code>QLCDNumber* num = new QLCDNumber();</code> | Пример создания                          |
| <code>num-&gt;display(int);</code>               | Задаёт целочисленное значение индикатора |
| <code>num-&gt;display(double);</code>            | Задаёт вещественное значение индикатора  |





# Кнопки

| Сигнал        | Описание  |
|---------------|---|
| pressed()     | отправляется при нажатии кнопкой мыши                       |
| clicked(bool) | отправляется при щелчке кнопкой мыши                        |
| released()    | отправляется при отпускании кнопки мыши                     |
| toggled(bool) | отправляется при изменении состояния переключающейся кнопки |

| QPushButton - Кнопка нажатия                                  | Описание        |
|---|-----------------|
| <code>QPushButton* button = new QPushButton("Button");</code> | Пример создания |

| QCheckBox – Кнопка-флажок                                 | Описание        |
|---|-----------------|
| <code>QCheckBox* cb = new QCheckBox("ButtonName");</code> | Пример создания |

| QRadioButton – Кнопки переключатели                          | Описание        |
|--|-----------------|
| <code>QRadioButton* rb = new QRadioButton("ButtonName</code> | Пример создания |



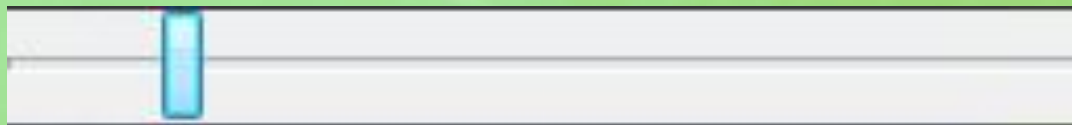


Виджеты

# Элементы ввода настроек



# QSlider



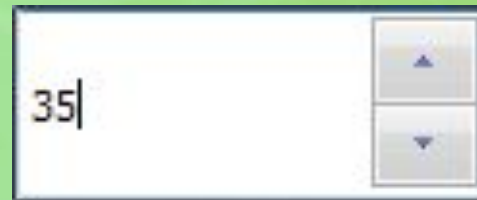
## ■ QSlider - Ползунок

| Метод  | Описание   |
|--|--|
| <code>QSlider* slider = new QSlider(Qt::Orientation);</code> | Пример создания, где "Qt::Orientation" либо "Qt::Horizontal" либо "Qt::Vertical" |
| <code>slider-&gt;setOrientation(Qt::Orientation);</code>     | Изменить ориентацию ползунка   |
| <code>slider-&gt;setMinimum(int min);</code>                 | Задание минимального значения  |
| <code>slider-&gt;setMaximum(int max);</code>                 | Задание максимального значения   |
| <code>slider-&gt;setRange(int min, int max);</code>          | Задание диапазона  |
| <code>slider-&gt;setValue(int v);</code>                     | Слот, устанавливающий текущее значение   |
| <code>int v = slider-&gt;value();</code>                     | Возвращает текущее значение  |

| Сигнал                        | Описание                          |
|-------------------------------|-----------------------------------|
| <code>sliderMoved(int)</code> | Высылается при изменении значения |



# QSpinBox



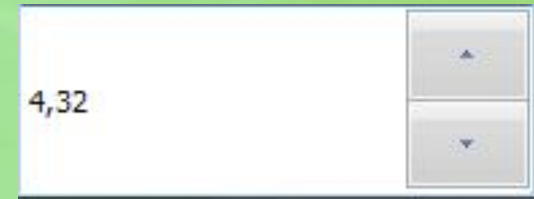
## ■ QSpinBox - Счетчик

| Метод   | Описание                               |
|---|--|
| <code>QSpinBox* spin = new QSpinBox();</code>     | Пример создания                        |
| <code>spin-&gt;setMinimum(int min);</code>        | Задание минимального значения          |
| <code>spin-&gt;setMaximum(int max);</code>        | Задание максимального значения         |
| <code>spin-&gt;setRange(int min, int max);</code> | Задание диапазона                      |
| <code>spin-&gt;setValue(int v);</code>            | Слот, устанавливающий текущее значение |
| <code>spin-&gt;setPrefix("");</code>              | Задание префикса                       |
| <code>spin-&gt;setSuffix("");</code>              | Задание суффикса                       |

| Сигнал                         | Описание                          |
|--------------------------------|-----------------------------------|
| <code>valueChanged(int)</code> | Высылается при изменении значения |



# QDoubleSpinBox



## ■ QDoubleSpinBox – Элемент ввода double

| Метод  | Описание                               |
|--|--|
| <code>QDoubleSpinBox* spin = new QDoubleSpinBox ();</code> | Пример создания                        |
| <code>spin-&gt;setMinimum(double min);</code>              | Задание минимального значения          |
| <code>spin-&gt;setMaximum(double max);</code>              | Задание максимального значения         |
| <code>spin-&gt;setRange(double min, double max);</code>    | Задание диапазона                      |
| <code>spin-&gt;setValue(double v);</code>                  | Слот, устанавливающий текущее значение |
| <code>spin-&gt;setPrefix("");</code>                       | Задание префикса                       |
| <code>spin-&gt;setSuffix("");</code>                       | Задание суффикса                       |

| Сигнал                         | Описание                          |
|--------------------------------|-----------------------------------|
| <code>valueChanged(int)</code> | Высылается при изменении значения |



# QLineEdit

text|

## ■ QLineEdit - Текстовое поле

| Метод   | Описание  |
|---|---|
| <code>QLineEdit *text = new QLineEdit();</code> | Пример создания   |
| <code>text-&gt;setReadOnly(bool);</code>        | Только для чтения   |
| <code>text-&gt;setEchoMode(EchoMode);</code>    | Установить режим, где "EchoMode" принимает значения "Normal", "NoEcho", "Password" или "PasswordEchoEdit" |
| <code>QString s = text-&gt;text();</code>       | Возвращение текста  |

| Сигнал                              | Описание                        |
|-------------------------------------|---------------------------------|
| <code>textChanged(QString) )</code> | Высылается при изменении текста |
| <code>returnPressed()</code>        | Высылается по нажатию Enter     |



Виджеты

Лейауты



# Лейауты

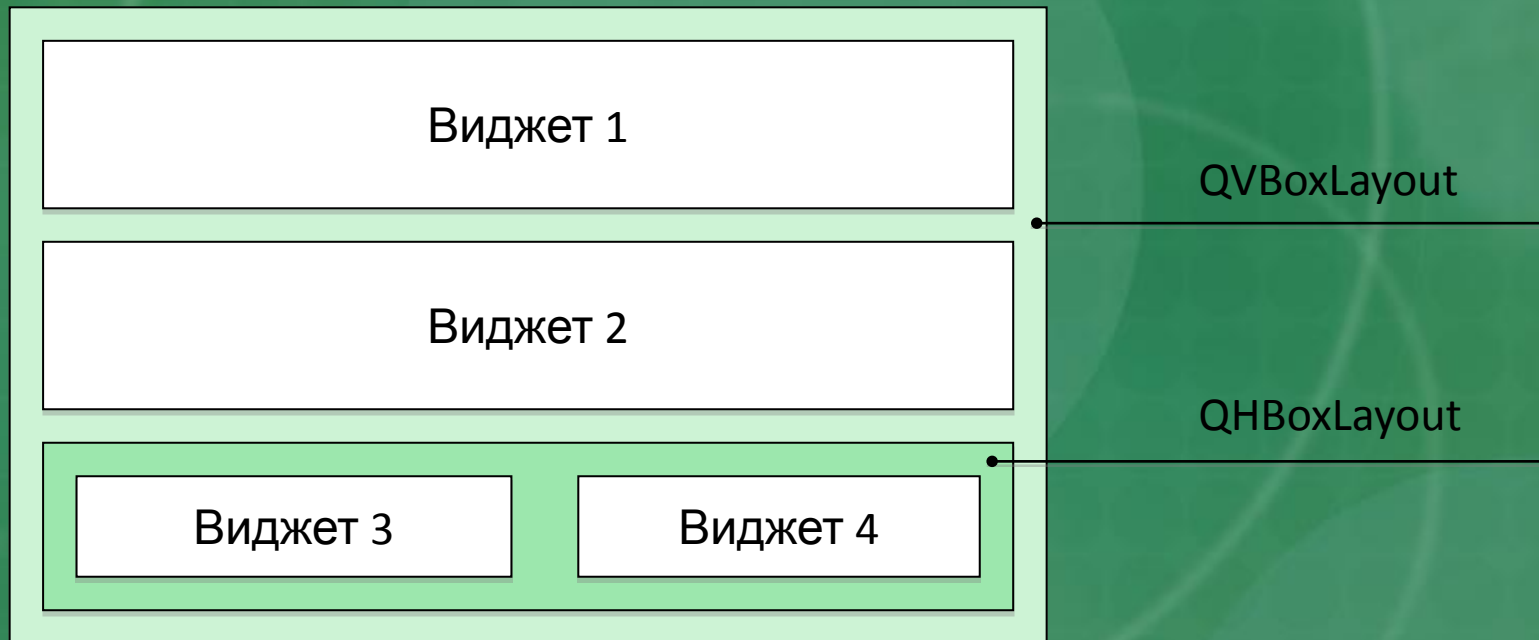
- Лейауты - это элементы для размещения и упорядочивания виджетов и других лейаутов.

| Метод   | Описание  |
|---|---|
| <code>layout-&gt;addWidget(QWidget * w);</code>   | Добавление виджета на лейаут, где "w" - это указатель на добавляемый виджет |
| <code>layout-&gt;removeWidget(QWidget *w);</code> | Удаление виджета  |
| <code>layout-&gt;addLayout(QLayout *);</code>     | Добавление лейаута на лейаут  |





# Лейауты





# Лейауты

## **QGridLayout** - Табличное размещение виджетов

| Метод  | Описание        |
|--|-----------------|
| <code>QGridLayout * layout = new QGridLayout();</code> | Пример создания |

## **QVBoxLayout** - Вертикальное размещение виджетов

| Метод  | Описание        |
|--|-----------------|
| <code>QGridLayout * layout = new QGridLayout();</code> | Пример создания |

## **QHBoxLayout** - Горизонтальное размещение виджетов

| Метод  | Описание        |
|--|-----------------|
| <code>QGridLayout * layout = new QGridLayout();</code> | Пример создания |



# Лейауты

## **QGridLayout** - Табличное размещение виджетов

| Метод  | Описание        |
|--|-----------------|
| <code>QGridLayout * layout = new QGridLayout();</code> | Пример создания |

## **QVBoxLayout** - Вертикальное размещение виджетов

| Метод  | Описание        |
|--|-----------------|
| <code>QGridLayout * layout = new QGridLayout();</code> | Пример создания |

## **QHBoxLayout** - Горизонтальное размещение виджетов

| Метод  | Описание        |
|--|-----------------|
| <code>QGridLayout * layout = new QGridLayout();</code> | Пример создания |



# Лейауты

```
#include "mainwindow.h"
#include <QtGui>

MainWindow::MainWindow(QWidget *parent) : QMainWindow(parent)
{
    QLabel *lb = new QLabel(); // Объявление и вызов конструктора
    lb->setText("text"); // Задание текста
    QLCDNumber *lcd = new QLCDNumber(); // Объявление и вызов конструктора
    lcd->display(20); // Установка значения
    QHBoxLayout *layout1 = new QHBoxLayout(); // Объявление и вызов конструктора
    layout1->addWidget(lb); // Добавление виджета на слой
    layout1->addWidget(lcd); // Добавление виджета на слой
    QLineEdit *line = new QLineEdit(); // Объявление и вызов конструктора
    line->setText("Type text"); // Задание текста
    QVBoxLayout *layout2 = new QVBoxLayout(); // Объявление и вызов конструктора
    layout2->addWidget(line); // Добавление виджета
    layout2->addLayout(layout1); // Добавление лейаута на лейаут
    QWidget*central=new QWidget(this); // Объявление и вызов конструктора виджета "central" - он будет
    // основным виджетом приложения
    central->setLayout(layout2); // Установка лейаута на виджет
    setCentralWidget(central); // Установка виджета "central" в качестве основного виджета приложения
}

MainWindow::~MainWindow() {} // Деструктор
```



# Задание №2: Добавление виджетов вручную

## Цель лабораторной работы:

- Освоить добавление виджетов в проект.
- Написать программу, состоящую из нескольких виджетов.

## Порядок выполнения лабораторной работы:

1. Создать новый проект без формы.
2. Создать любой виджет на свое усмотрение (QLabel, QPushButton, QLCDNumber, QSlider и т.п.).
3. Сделать его главным виджетом приложения с помощью метода `setCentralWidget()`.
4. Скомпилировать и запустить программу.



# Задание №3: Компоновка виджетов с помощью лейаутов

## Цель лабораторной работы:

- Разобраться с размещением виджетов на лейаутах.
- Написать программу со скомпонованными с помощью лейаутов виджетами.

## Порядок выполнения лабораторной работы:

1. Создать новый проект.
2. Создать несколько виджетов и лейаутов
3. Скомпоновать виджеты в соответствии со схемой
4. Скомпилировать и запустить программу.

