

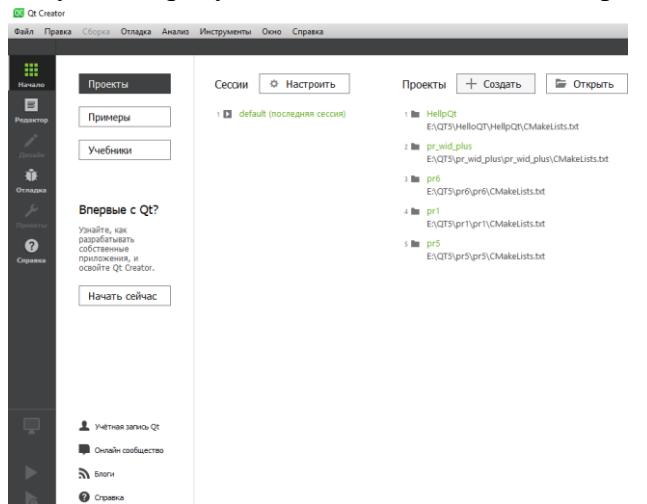
## Лабораторная работа 1.

### Ознакомление с кросс-платформенной средой разработки Qt Creator

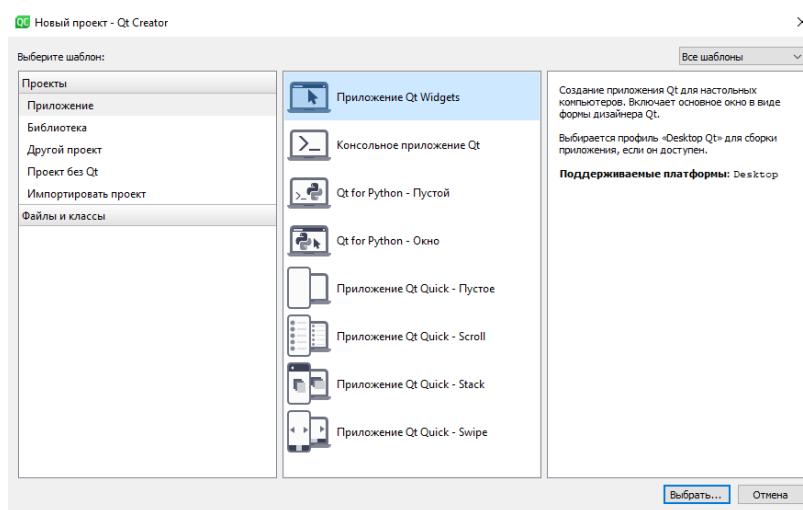
**Цель работы:** научиться создавать приложения в среде Qt Creator

**Порядок выполнения работы:**

1. Запустите среду Qt-Creator. Для этого выберите **Пуск\Qt\Qt Creator**.

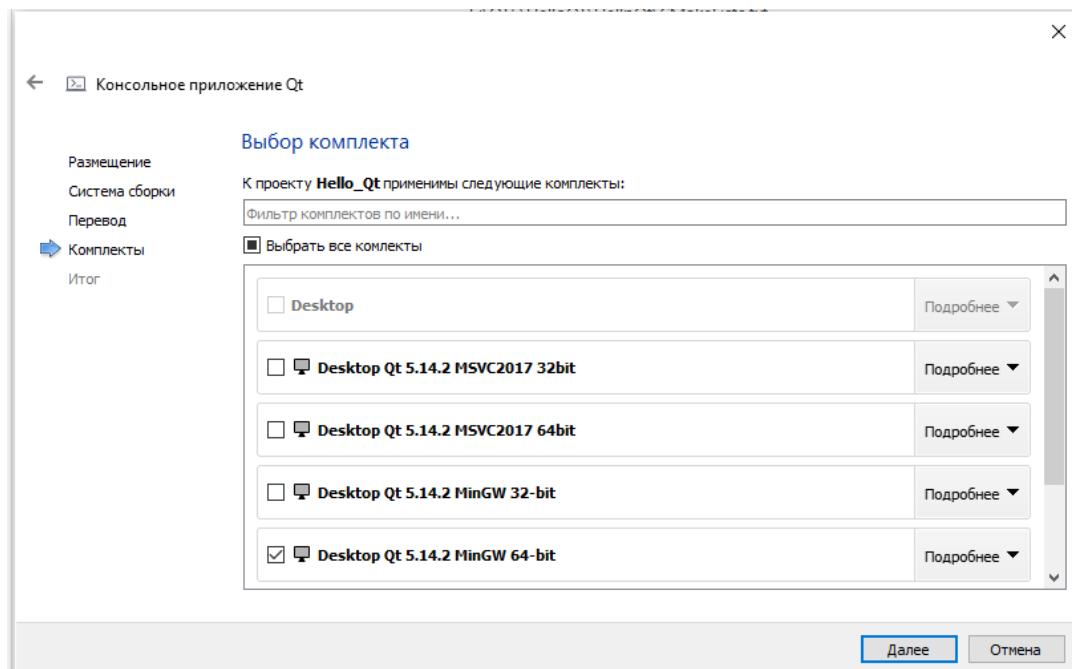


2. Создайте проект. Для этого можно, например, нажать на кнопочку «Создать».
3. Выберите шаблон для проекта «Приложение QT Widgets»

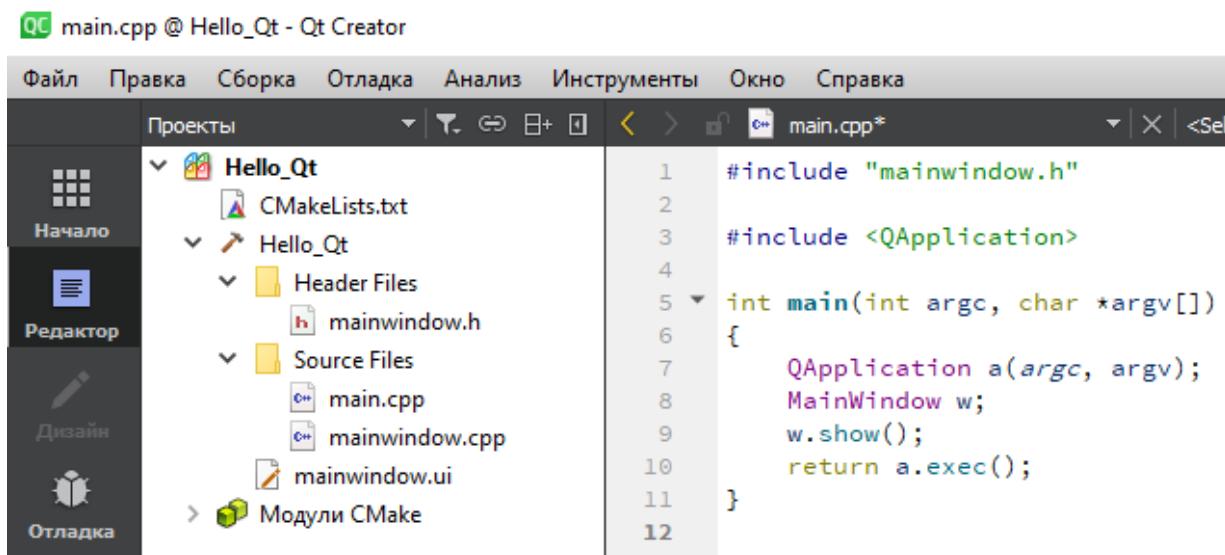


4. Задайте название проекта и его размещение. Рекомендуется для каждого проекта Qt создавать отдельную папку (т.к. для каждого проекта Qt в дальнейшем будет создано несколько подкаталогов). Имя проекта и путь размещения проекта НЕ ДОЛЖНЫ содержать русских букв.

5. Выберите систему сборки (qmake или cmake).
6. Далее следует выбрать подходящий комплект. «Комплектом» (Kit) Qt Creator называет набор из конфигурации Qt и компилятора. Находятся комплексты в меню Tools->Options, раздел Build&Run, вкладка Kits. Обычно, при установке QtCreator скачивается и ставится как минимум один комплект.



7. В результате на экране получим:



8. Данный проект уже можно запустить. В результате получим пустую форму.

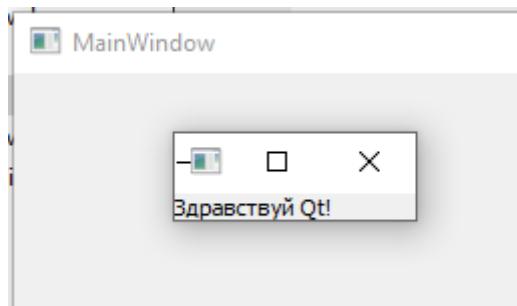
9. Попробуем вывести на форме приветственное сообщение. Для этого внесем в файл некоторые изменения:

```
#include "mainwindow.h"

#include <QApplication>
#include <QLabel> // 1

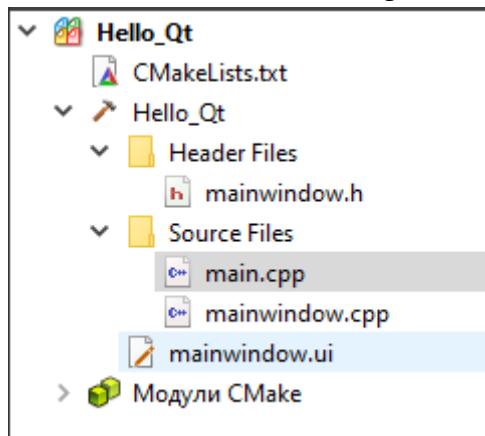
int main(int argc, char *argv[])
{
    QApplication a(argc, argv);
    MainWindow w;
    QLabel lb1("Здравствуй Qt!"); // 2
    w.show();
    lb1.show(); // 3
    return a.exec();
}
```

10. Запустим проект. В результате получим:

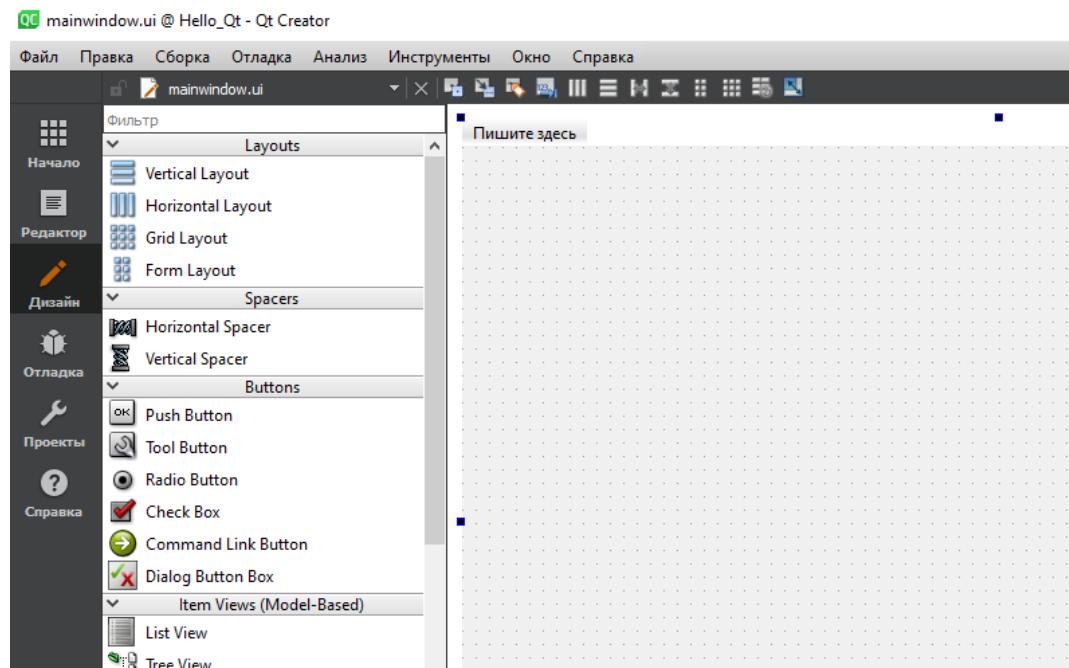


На самом деле, форма нам здесь не нужна и можно было ее не создавать.

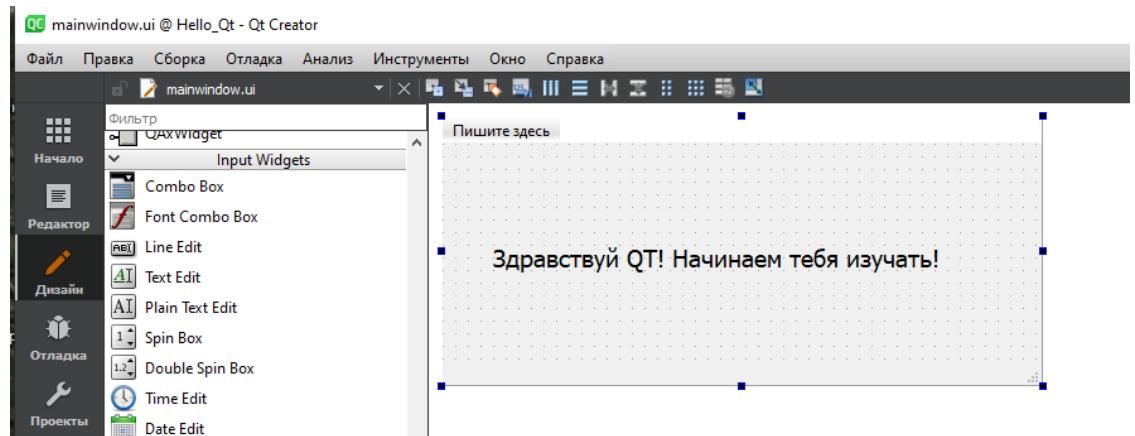
11. Приветственное сообщение можно разместить на форме в конструкторе. Для этого щелкаем мышкой на файле



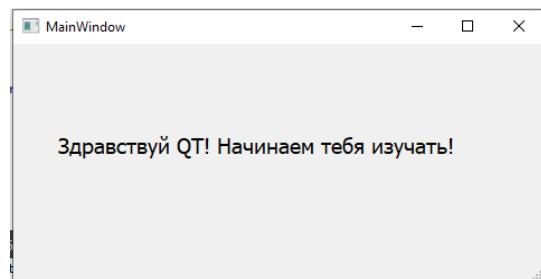
12. В результате перед нами откроется форма в режиме конструктора, на которую можно перетаскивать требуемые виджеты:



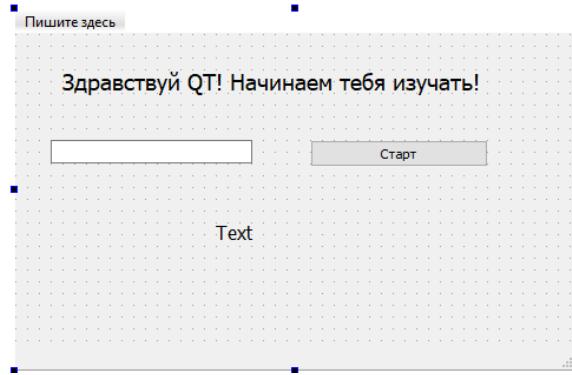
13. «Перетащим» на форму виджет Label, в котором напишем приветствие:



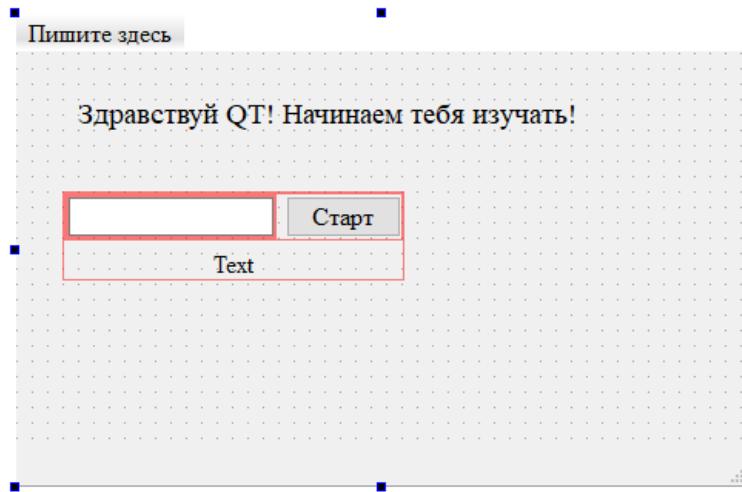
14. Запустим приложение и получим:



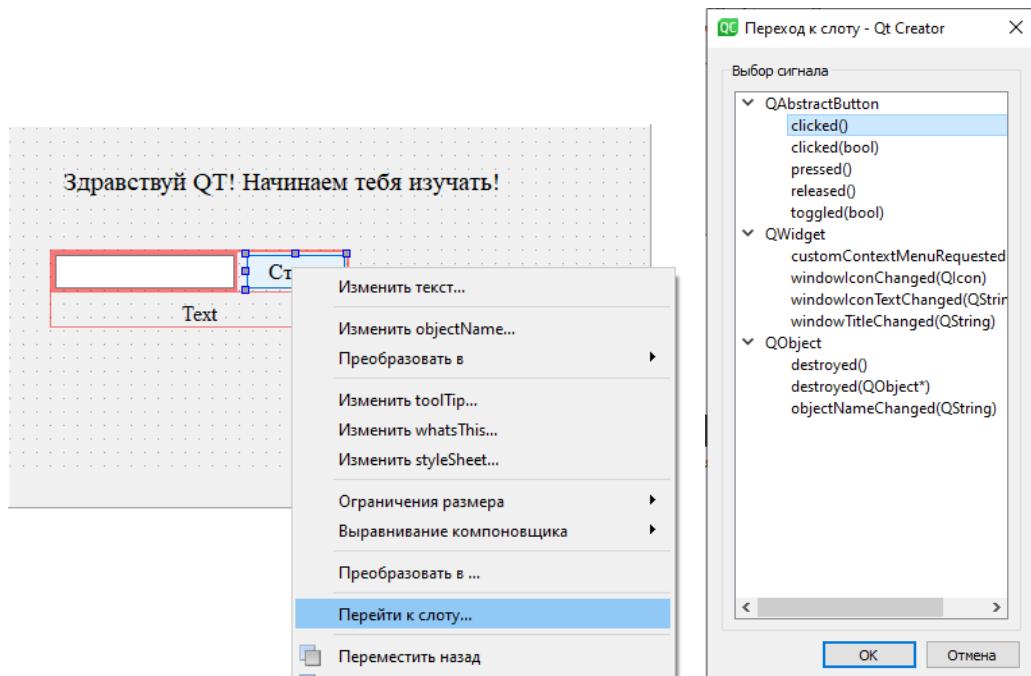
15. Предоставим пользователю возможность самому ввести текст приветствия. Для этого добавим на форму виджеты QLineEdit (поле для ввода), QPushButton (кнопка) и еще одну метку. В результате получим:



16. Выделите виджеты `lineEdit` и `pushButton` и выберите действие в меню над формой «Скомпоновать по горизонтали», затем выделите `label` и скомпонуйте элементы по вертикали. Применение вертикальной и горизонтальной компоновок обеспечивает масштабирование интерфейса приложения при различных разрешениях экрана;
17. В свойствах главного окна, выберите свойство `WindowTitle` и переименуйте окно в «Widgets»;
18. Используя свойство `alignment` виджета `label` и свойство `styleSheet` виджетов `lineEdit` и `pushButton`, отредактируйте форму, чтобы получить такой вид:



19. Перейдем к слоту, который будет вызываться в ответ на событие `clicked()`. При нажатии кнопки `Start` вы будете использовать механизм сигналов и слотов Qt. Сигнал вырабатывается, когда происходит определённое событие, а слот – это функция, которая вызывается в ответ на определённый сигнал. У виджетов Qt есть предопределённые сигналы и слоты которые вы можете использовать прямо из режима Дизайна. Чтобы перейти к слоту: – нажмите правой кнопкой мыши на кнопке `Start` для открытия контекстного меню; – выберите Перейти к слоту... > `clicked()` и нажмите OK



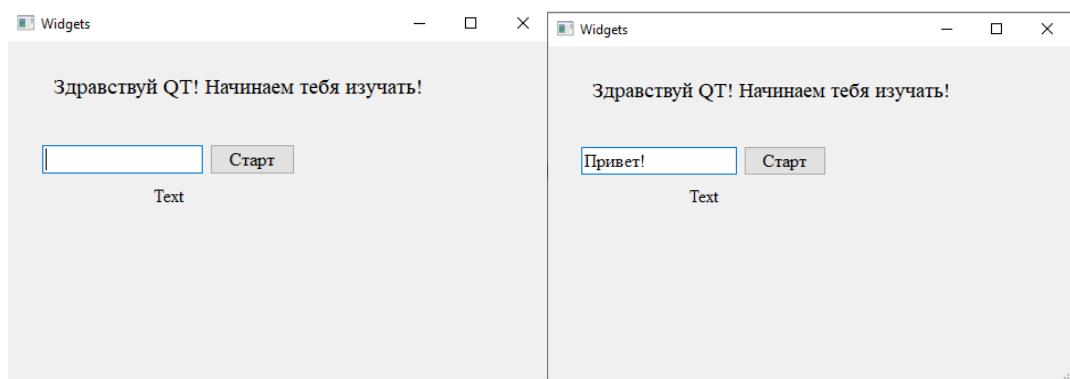
20. В результате Закрытый слот `on_pushButton_clicked()` будет добавлен в заголовочный файл `mainwindow.h`, и закрытый метод `Mainwindow::on_pushButton_clicked()` будет добавлен в файл исходных кодов `mainwindow.cpp`

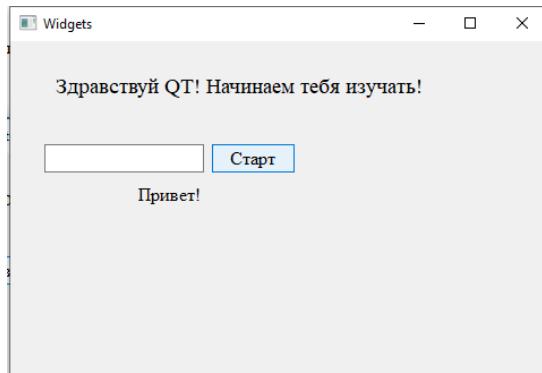
```
void MainWindow::on_pushButton_clicked()
{
}
```

21. Добавляем в метод строки:

```
void MainWindow::on_pushButton_clicked()
{
    ui->label_2->setText(ui->lineEdit->text());
    ui->lineEdit->clear();
}
```

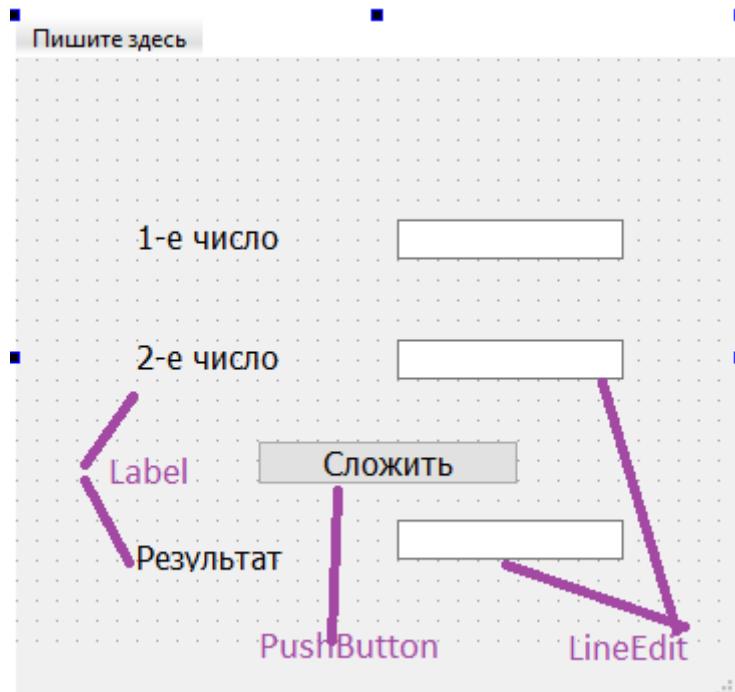
22. Запускаем программу и получаем следующее:





### 23. Задание:

- А) Разработать форму «Калькулятор» подобную данной, дополнив ее кнопками для операций «Вычесть», «Разделить», «Умножить», «Получить остаток» и другими (по желанию разработчика). В заголовке формы прописать ФИО и номер группы студента!
- Б) Используя механизм сигналов и слотов реализовать все операции, предусмотренные на форме. **ВСЕ СОЕДИНЕНИЯ ДОЛЖНЫ БЫТЬ ПРОПИСАНЫ П В ПРОГРАММНОМ КОДЕ!!!**
- В) Дополнительно – обработать исключение деления на НОЛЬ
- Г) При внесении изменений на форме все результаты расчетов должны быть удалены (т.е на форме всегда должна отображаться корректная информация)!!!



Фильтр	
Объект	Класс
MainWindow	QMainWindow
centralwidget	QWidget
label	QLabel
label_2	QLabel
label_3	QLabel
lineEdit	QLineEdit
lineEdit_2	QLineEdit
lineEdit_3	QLineEdit
pushButton	QPushButton
menubar	QMenuBar
statusbar	QStatusBar

24. Максимальная оценка за работу «ХОРОШО». Для получения дополнительного балла **(ДОПОЛНИТЕЛЬНЫЙ БАЛЛ могут получить только те студенты, которые сдают работу и загружают отчет в числе первых)** добавить на выбор студента:
- обработка нажатия клавиши (т.е., допустим разрешить вводить только цифры и если используются вещественные числа, то еще и символ точки или запятой);
  - использование валидации
  - если данные не введены, то кнопки не доступны

**Содержание отчета (соблюдать порядок следования пунктов в отчете):**

- 1) Разработанная Вами форма для приветствия (из п.18)
- 2) Скриншоты этой формы с результатами работы.
- 3) Скриншот разработанной Вами формы «Калькулятор» в режиме дизайнера.
- 4) Описание к форме «Калькулятор» (перечислить все используемые на форме виджеты с их программными именами и семантикой)
- 5) Текст программы (содержимое файлов \*.h и \*.cpp).
- 6) Результаты работы программы (скриншоты, демонстрирующие выполнение всех предусмотренных операций)
- 7) Выводы
- 8) Файл отчета должен быть в формате \*.doc или \*.pdf. Имя файла с отчетом задаем следующим образом : Группа\_ФИО\_ЛР№ (например 4431\_Иванов\_ЛР1).