

Лабораторная работа 1,2

Задание

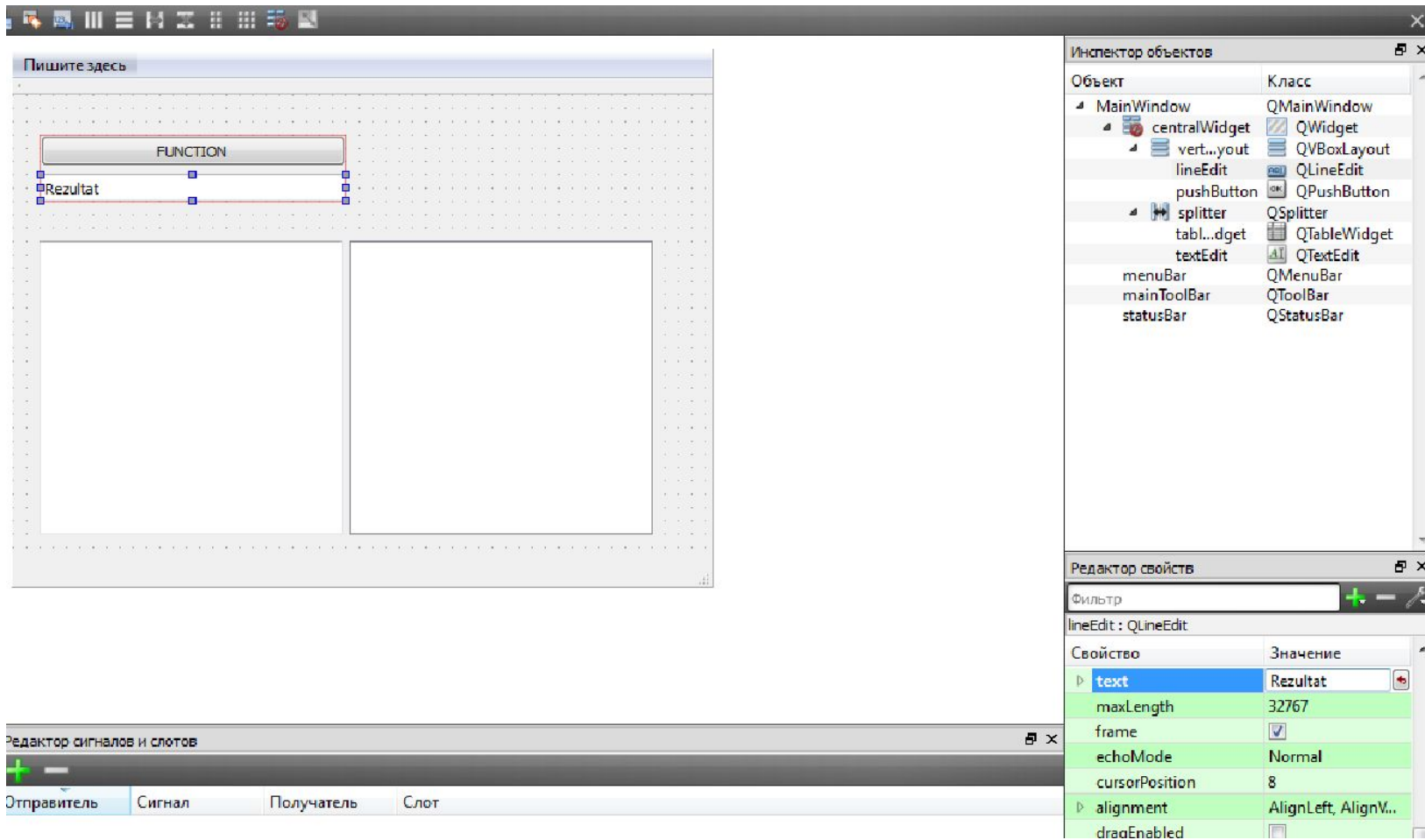
Создать кнопки:

- Вывода результата функции в элемент TextLabel;
- Вывода сформированной матрицы в TextEdit;
- Вывод матрицы в таблицу;
- Очистки содержимого полей;
- Закрытия формы (режим дизайна – Редактор сигналов и слотов).

Настроить свойства виджетов!

Файл проекта

```
ZF_Laba.pro
1 #-----
2 #
3 # Project created by QtCreator 2017-01-24T15:22:13
4 #
5 #-----
6
7 QT      += core gui
8
9 greaterThan(QT_MAJOR_VERSION, 4): QT += widgets
10
11 TARGET = ZF_Laba
12 TEMPLATE = app
13
14
15 SOURCES += main.cpp\
16         /mainwindow.cpp
17
18 HEADERS  +=/mainwindow.h
19
20 FORMS    +=mainwindow.ui
21
```

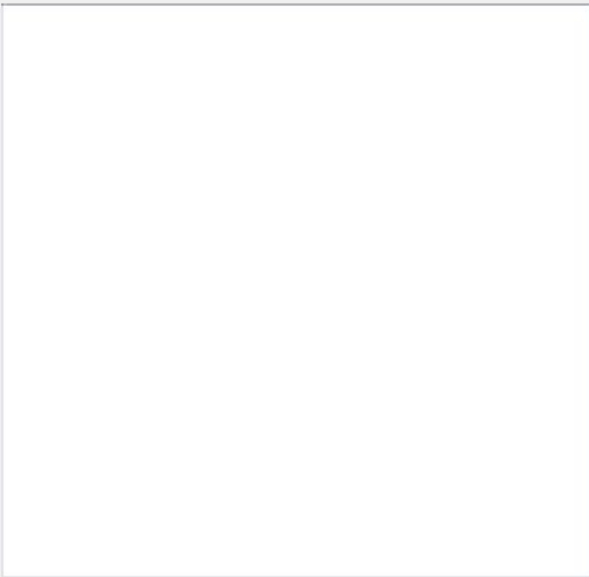
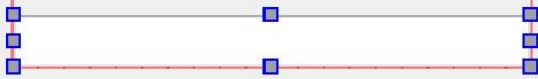


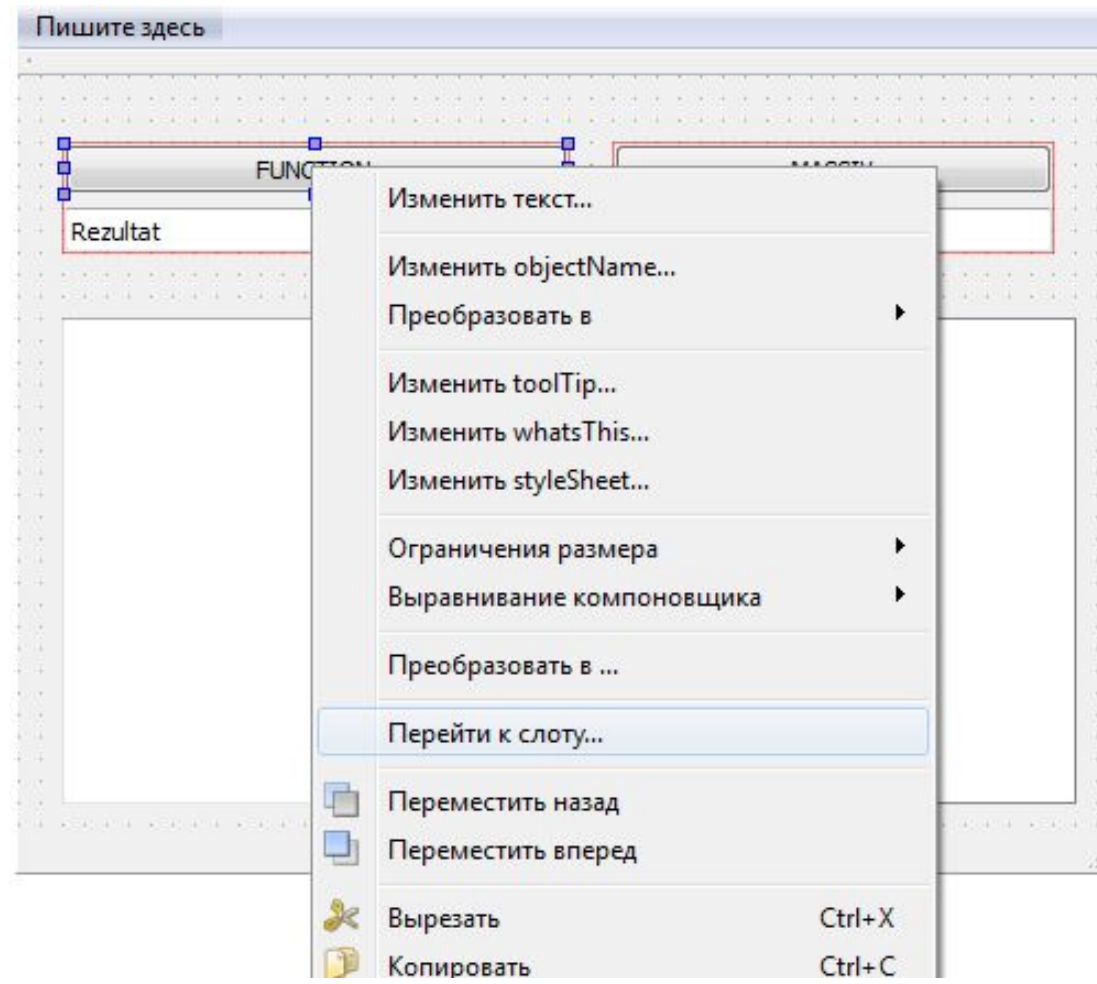
Пишите здесь

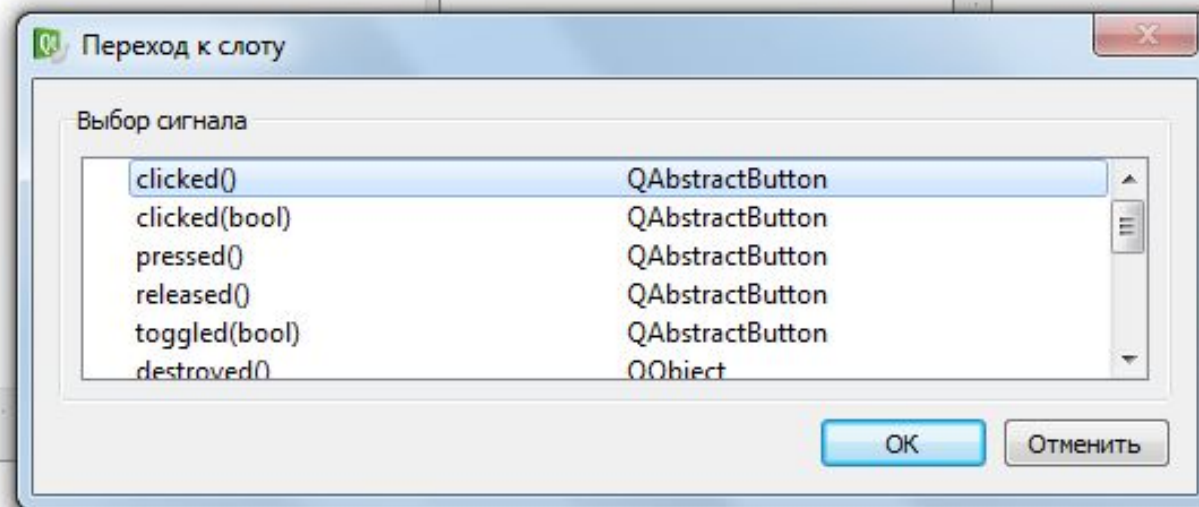
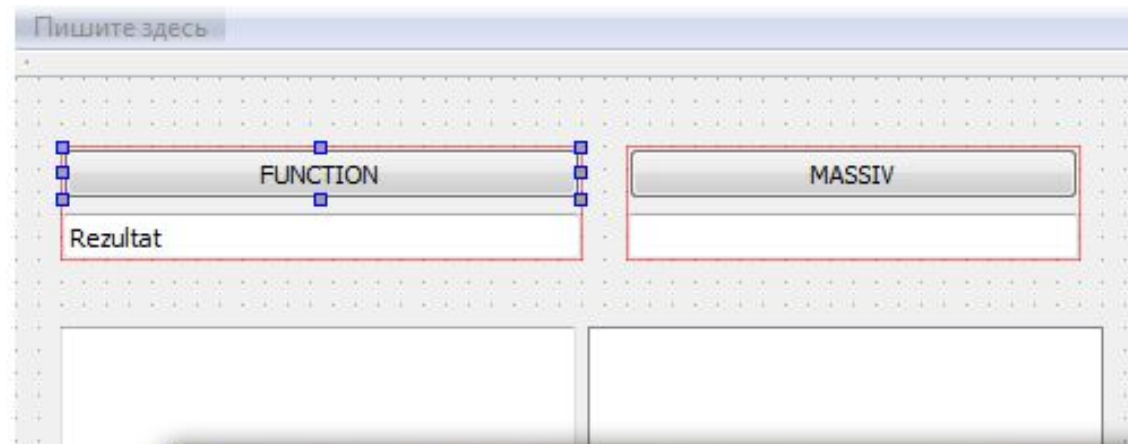
FUNCTION

Rezultat

MASSIV





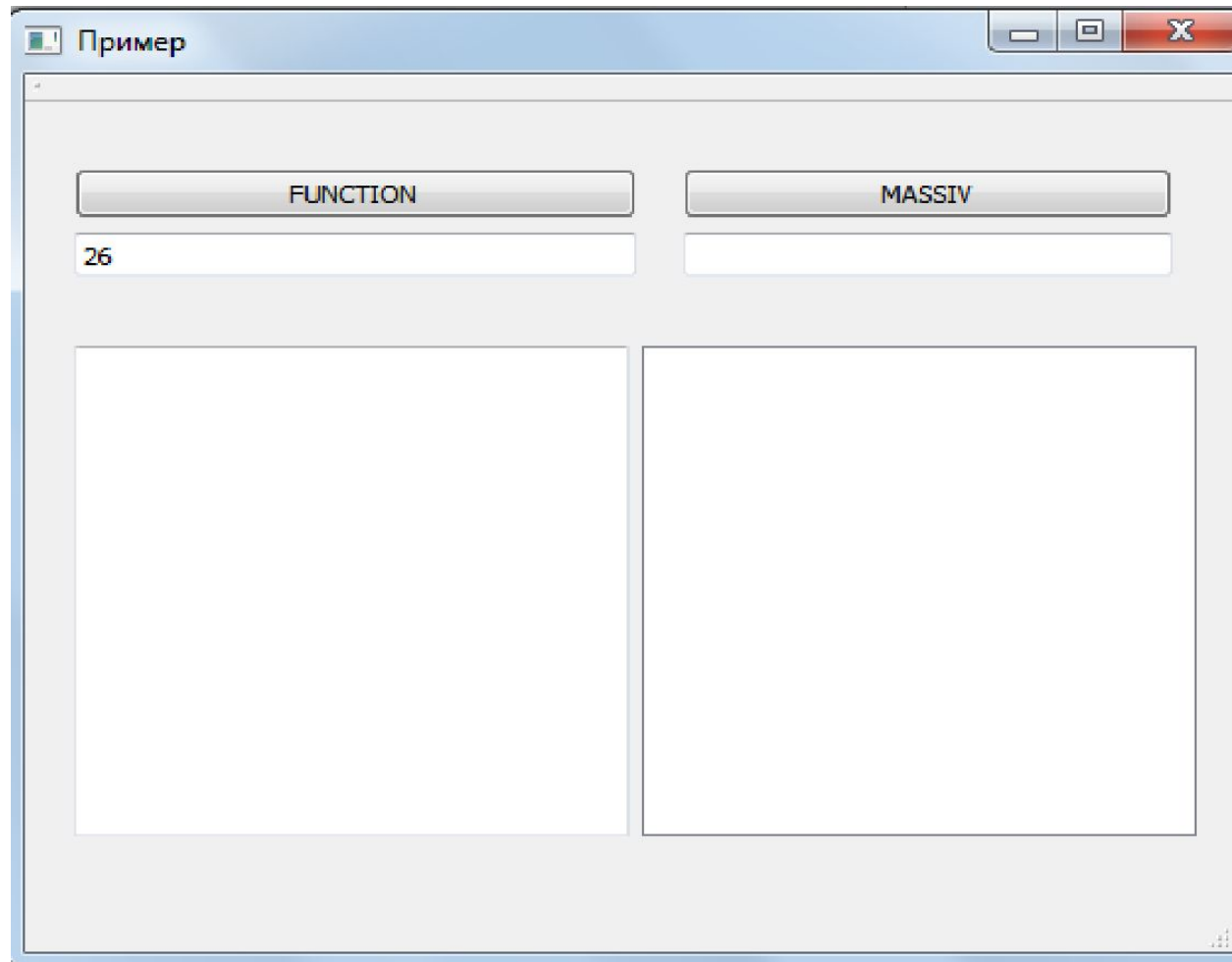


```
main.cpp*
1  #include "mainwindow.h"
2  #include <QApplication>
3
4  int main(int argc, char *argv[])
5  {
6      QApplication a(argc, argv);
7      MainWindow w;
8      w.setWindowTitle("Пример");
9      w.show();
10
11     return a.exec();
12 }
13
```



```
1 #ifndef MAINWINDOW_H
2 #define MAINWINDOW_H
3 #include <QMainWindow>
4 #include <QLabel>
5 #include <QLineEdit>
6 namespace Ui {
7     class MainWindow;
8 }
9 class MainWindow : public QMainWindow
10 {
11     Q_OBJECT
12
13 public:
14     explicit MainWindow(QWidget *parent = 0);
15     ~MainWindow();
16
17 private slots:
18     void on_pushButton_clicked();
19
20 private:
21     Ui::MainWindow *ui;
22     QLineEdit *lineEdit;
23 };
24
25 #endif // MAINWINDOW_H
```

```
16 void MainWindow::on_pushButton_clicked()
17 {   int i;
18     lineText=ui->lineEdit;
19     QString str="";
20     for(i=1;i<=5;i++)
21     {
22         i=i*i;
23     }
24     str.append(QString("%1").arg(i));
25
26     ui->lineEdit->setText(str);
27 }
28
```



Добавим

```
#include <QTextEdit>
```

И

```
private: Ui::MainWindow *ui;
```

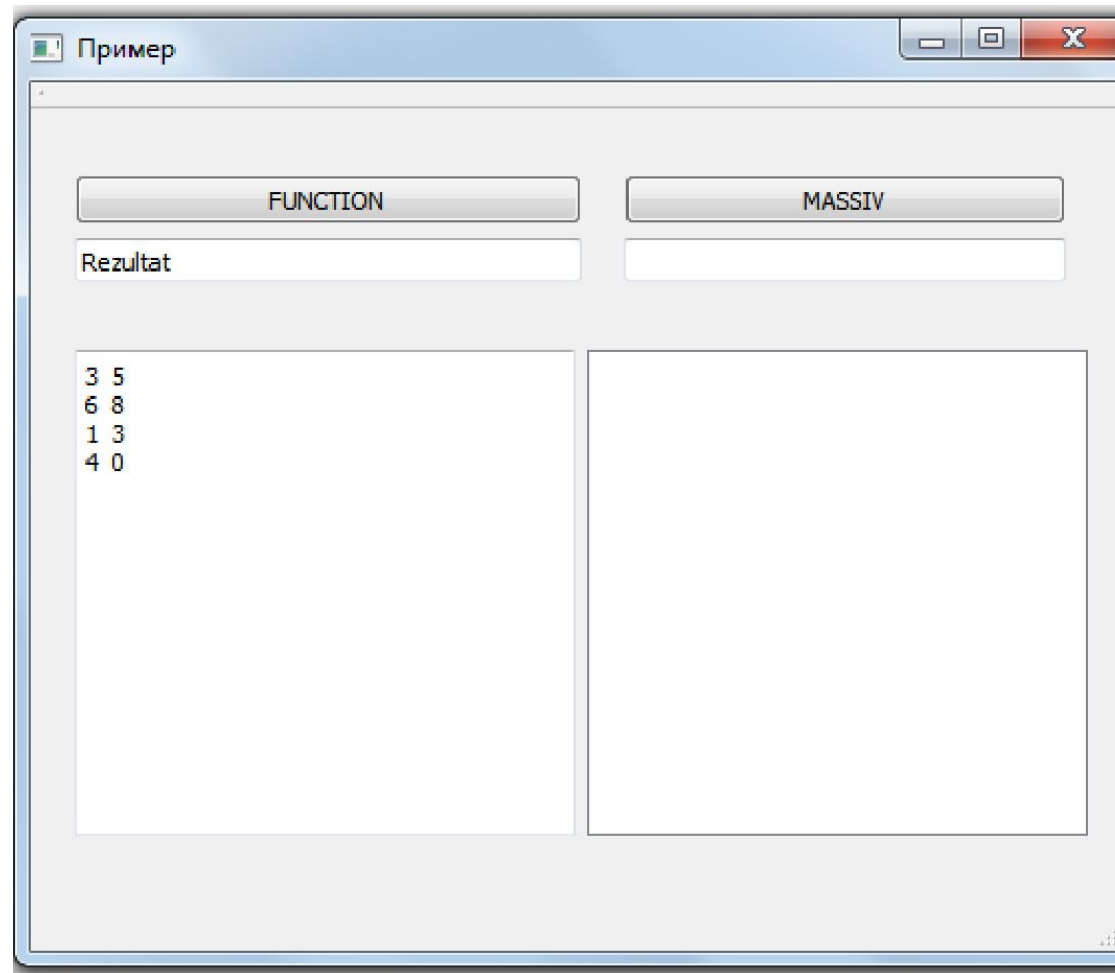
```
QLineEdit *lineText;
```

```
QTextEdit *textEdit;
```

В файле реализации

```
29 void MainWindow::on_pushButton_2_clicked()
30 {
31     QString str;
32     int a[4][2]={{3,5},{6,8},{1,3},{4,0}};
33     for( int i = 0; i <4; i++ )
34     {
35         for( int j = 0; j <2; j++ )
36             str.append(QString("%1 ").arg(a[i][j]));
37         str.append(QString("\n"));
38     }
39     ui->textEdit->setText(str);
40 }
```

Результат



Добавим вывод матрицы в таблицу

```
QString str;
int a[4][2]={{3,5},{6,8},{1,3},{4,0}};
QFont font; ui->tableWidget->setColumnCount(2);
// Указываем число строк и столбцов
ui->tableWidget->setRowCount(4);
// Включаем сетку
ui->tableWidget->setShowGrid(true);
ui->tableWidget->setHorizontalHeaderLabels(QStringList() << tr("A") << tr("B"));
// Разрешаем выделение только одного элемента
ui->tableWidget->setSelectionMode (QAbstractItemView::SingleSelection);
// Разрешаем выделение построчно
ui->tableWidget-> setSelectionBehavior (QAbstractItemView::SelectRows);
// Растягиваем последнюю колонку на всё доступное пространство
ui->tableWidget->horizontalHeader()->setStretchLastSection(true);
```

продолжение

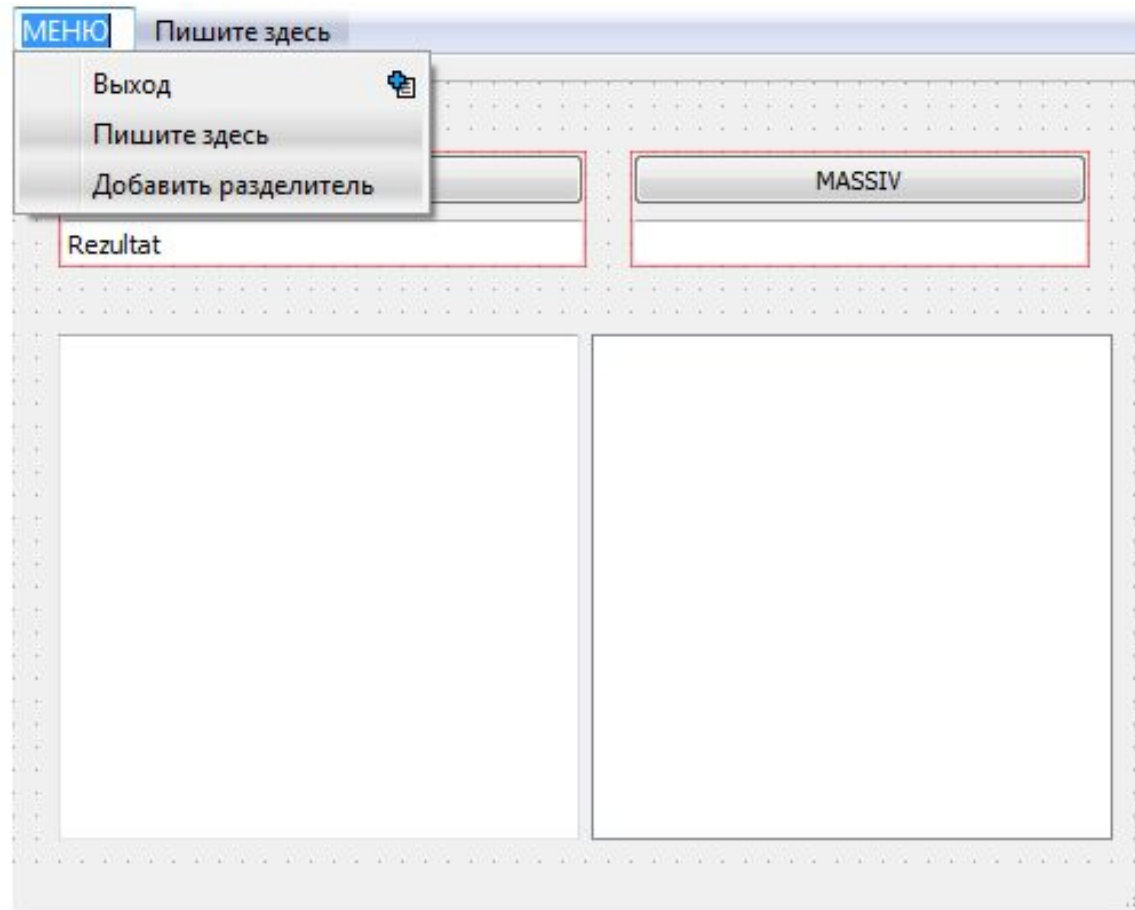
```
for( int i = 0; i <4; i++ )
{ for( int j = 0; j <2; j++ )
{
str.append(QString("%1 ").arg(a[i][j]));
// Добавим вывод в таблицу
QTableWidgetItem *item=new QTableWidgetItem;
item->setText(QString::number(a[i][j]));
ui->tableWidget->setItem(i, j, item); }
str.append(QString("\n"));
}
ui->textEdit->setText(str);
```


Конечный результат

The screenshot shows a window titled "Пример" with two buttons: "FUNCTION" and "MASSIV". The "FUNCTION" button has a text input field containing the number "26". Below the buttons are two data displays. The left display is a text area containing a 4x2 grid of numbers: 3 5, 6 8, 1 3, 4 0. The right display is a table with two columns labeled "A" and "B", containing the same 4x2 grid of numbers.

	A	B
1	3	5
2	6	8
3	1	3
4	4	0

Добавим меню



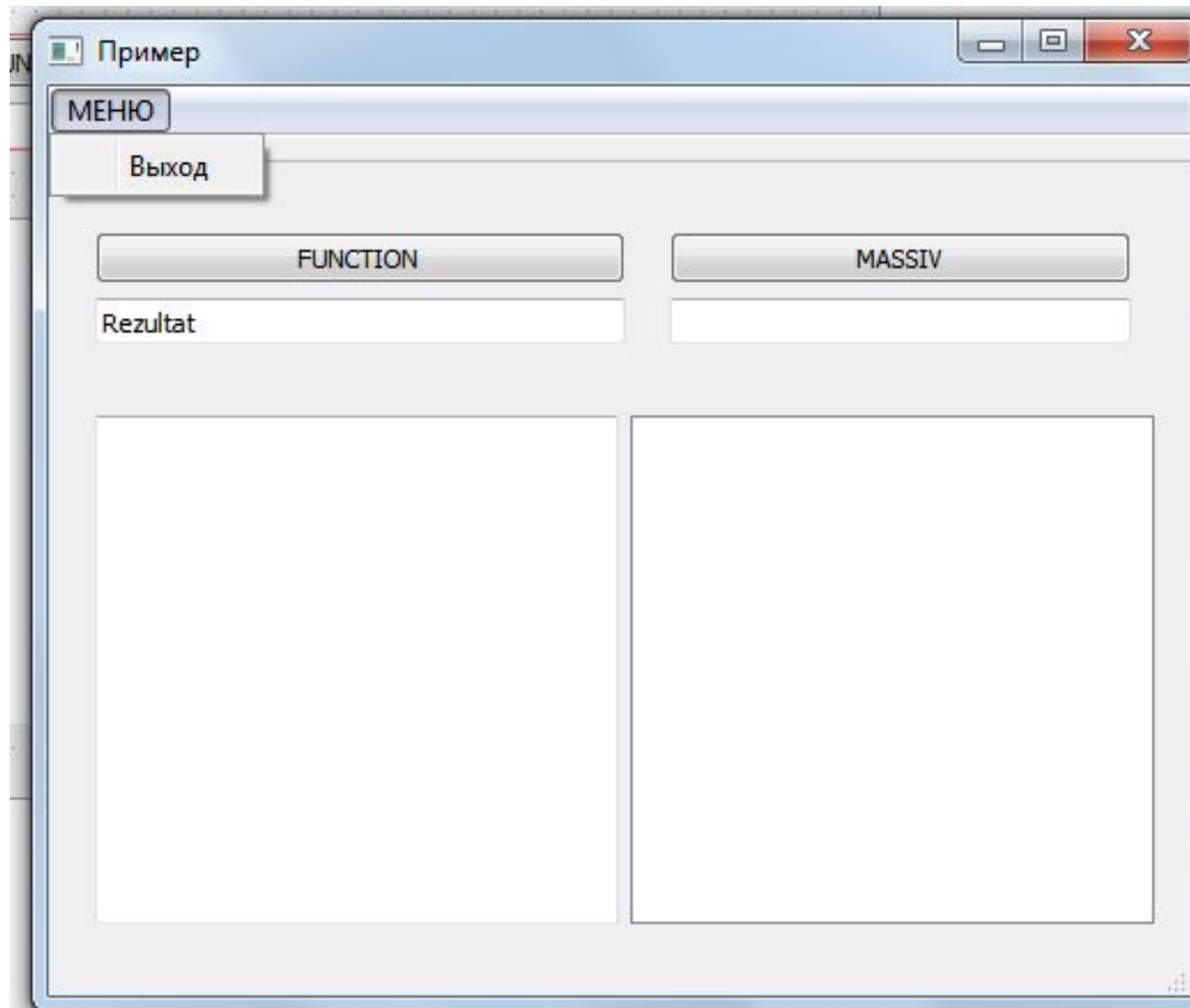
III

Редактор сигналов и слотов

+ -

Отправитель	Сигнал	Получатель	Слот
action	triggered()	MainWindow	close()

Результат



Добавим пункт меню «Очистка»

The image shows a Qt application window with a menu bar containing 'МЕНЮ' and 'Пишите здесь'. The 'МЕНЮ' menu is open, showing items: 'Выход', 'Очистка', 'Пишите здесь', and 'Добавить разделитель'. The 'Очистка' item is highlighted. The main window contains a toolbar with a 'MASSIV' button and a text input field. Below the toolbar is a QTableWidget with two columns and two rows. The 'Пишите здесь' label is positioned above the text input field.

Инспектор объектов

Объект	Класс
MainWindow	QMainWindow
centralWidget	QWidget
vert...yout	QVBoxLayout
lineEdit	QLineEdit
pushButton	QPushButton
vert...ut_2	QVBoxLayout
lineEdit_2	QLineEdit
push...on_2	QPushButton
splitter	QSplitter
tabl...dget	QTableWidget
textEdit	QTextEdit
menuBar	QMenuBar
menu	QMenu
action	QAction
action_2	QAction
mainToolBar	QToolBar
statusBar	QStatusBar

Редактор сигналов и слотов



Отправитель	Сигнал	Получатель	Слот
action_2	triggered()	textEdit	clear()
action_2	triggered()	lineEdit	clear()
action_2	triggered()	tableWidget	clear()