

Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Сибирский государственный университет телекоммуникаций
и информатики» (СибГУТИ)

09.03.01 "Информатика и вычислительная техника"
профиль "Программное обеспечение средств
вычислительной техники и
автоматизированных систем"

ОТЧЕТ

по дисциплине «Визуальное программирование и
человеко-машинное взаимодействие»

Практическое занятие №1

Выполнил:
Разумов Д. Б. студент группы ИП-811

Проверил:
преподаватель Мерзлякова Е.Ю.

Новосибирск 2020

Оглавление

Постановка задачи.....	3
Листинг main.cpp.....	3
Листинг mainwindow.cpp.....	3
Листинг mainwindow.cpp.....	5
Листинг mainwindow.h.....	5
Листинг auth.h.....	6
Скриншоты.....	7

Постановка задачи

Практическое занятие номер 1. Написать программу, которая может открывает файлы типов *.cpp, *.txt, *.c, редактировать их и сохранять.

Листинг main.cpp

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

int main(int argc, char *argv[])
{
    QApplication a(argc, argv);
    MainWindow w;
    w.show();

    return a.exec();
}
```

Листинг mainwindow.cpp

```
#include "mainwindow.h"
#include "ui_mainwindow.h"
#include "auth.h"
#include "ui_auth.h"
#include "QFileDialog"
#include <QTextDocumentWriter>

void MainWindow::About_Lab1()
{
    auth *dg = new auth();
    dg->show();
}

void MainWindow::slotOpen()
{
    QString filename = QFileDialog::getOpenFileName(0,
        "Открыть файл", QDir::currentPath(), "*.cpp *.txt *.c");
}
```

```

    QFile file(filename);
    if (file.open(QIODevice::ReadOnly | QIODevice::Text))
        ui->textEdit->setPlainText(file.readAll());
}

void MainWindow::slotSave()
{
    QString filename = QFileDialog::getSaveFileName(0,
        "Сохранить файл", QDir::currentPath(), "*.cpp *.txt *.c");
    //QTextDocumentWriter writer;
    //writer.setFileName(filename);
    //writer.write(ui->textEdit->document());

    QFile file(filename);
    file.open(QIODevice::WriteOnly);
    file.write(ui->textEdit->toPlainText().toUtf8());
    file.close();
}

void MainWindow::slotClear()
{
    ui->textEdit->clear();
}

MainWindow::MainWindow(QWidget *parent) :
    QMainWindow(parent),
    ui(new Ui::MainWindow)
{
    ui->setupUi(this);
    connect(ui->action, SIGNAL(triggered()), this, SLOT(About_Lab1()));

    //пункт 11
    QAction* pactOpen = new QAction("file open action", 0);
    pactOpen->setText("&Открыть");
    pactOpen->setShortcut(QKeySequence("CTRL+O"));
    pactOpen->setToolTip("Открытие документа");
    pactOpen->setStatusTip("Открыть файл");
    pactOpen->setWhatsThis("Открыть файл");
    pactOpen->setIcon(QPixmap("1.png"));
    connect(pactOpen, SIGNAL(triggered()), SLOT(slotOpen()));
    QMenu* pmnuFile=new QMenu("&Файл");
    pmnuFile->addAction(pactOpen);
    menuBar()->addMenu(pmnuFile);
    //пункт 14
    QAction* pactSave = new QAction("file save action", 0);
    pactSave->setText("&Сохранить");
    pactSave->setShortcut(QKeySequence("CTRL+S"));
    pactSave->setToolTip("Сохранение документа");
    pactSave->setStatusTip("Сохранить файл");
    pactSave->setWhatsThis("Сохранить файл");
    pactSave->setIcon(QPixmap("1.png"));
    connect(pactSave, SIGNAL(triggered()), SLOT(slotSave()));
    pmnuFile->addAction(pactSave);
    //пункт 15
    QAction* pactClear = new QAction("clear action", 0);
    pactClear->setText("&Очистить");
    pactClear->setShortcut(QKeySequence("CTRL+L"));
    pactClear->setToolTip("Очистить поле");
    pactClear->setStatusTip("Очистить поле");
    pactClear->setWhatsThis("Очистить поле");

```

```

    pactClear->setIcon(QPixmap("1.png"));
    connect(pactClear, SIGNAL(triggered()), SLOT(slotClear()));
    pmnuFile->addAction(pactClear);
    //punkt 16
    ui->mainToolBar->addAction(pactOpen);
    ui->mainToolBar->addAction(pactSave);
    ui->mainToolBar->addAction(pactClear);

    ui->mainToolBar->setStyleSheet("background-color: rgb(220, 220, 0)");
}

MainWindow::~MainWindow()
{
    delete ui;
}

```

Листинг mainwindow.cpp

```

#include "auth.h"
#include "ui_auth.h"

auth::auth(QWidget *parent) :
    QMainWindow(parent),
    ui(new Ui::auth)
{
    ui->setupUi(this);
    //ui->labela->addAction(QIcon(":/imgcat/5b81PMWCKgo.jpg"));
    ui->label_2->setStyleSheet("background-color: rgb(220, 220, 0)");
}

auth::~auth()
{
    delete ui;
}

```

Листинг mainwindow.h

```

#ifndef MAINWINDOW_H
#define MAINWINDOW_H

#include <QMainWindow>

namespace Ui {
class MainWindow;
}

```

```

class MainWindow : public QMainWindow
{
    Q_OBJECT

public:
    explicit MainWindow(QWidget *parent = nullptr);
    ~MainWindow();

private:
    Ui::MainWindow *ui;

private slots:
    void About_Lab1();
    void slotOpen();
    void slotSave();
    void slotClear();
};

#endif // MAINWINDOW_H

```

Листинг auth.h

```

#ifndef AUTH_H
#define AUTH_H

#include <QMainWindow>

namespace Ui {
class auth;
}

class auth : public QMainWindow
{
    Q_OBJECT

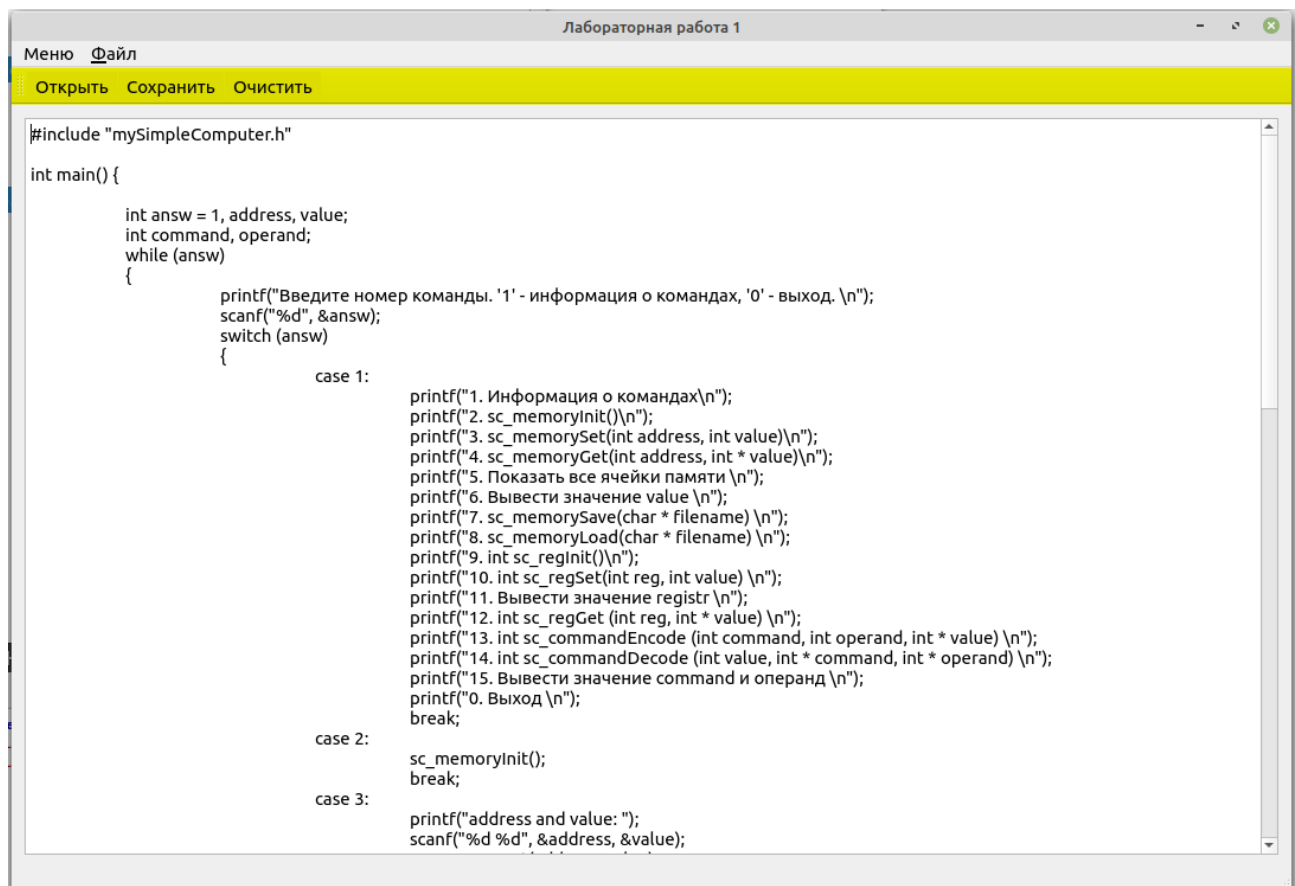
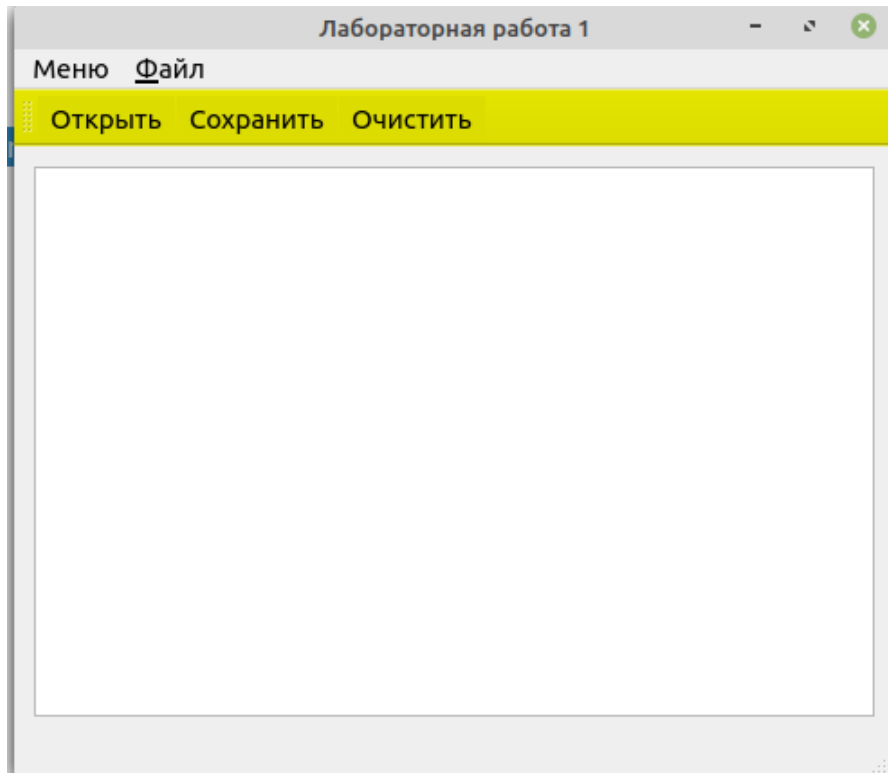
public:
    explicit auth(QWidget *parent = nullptr);
    ~auth();

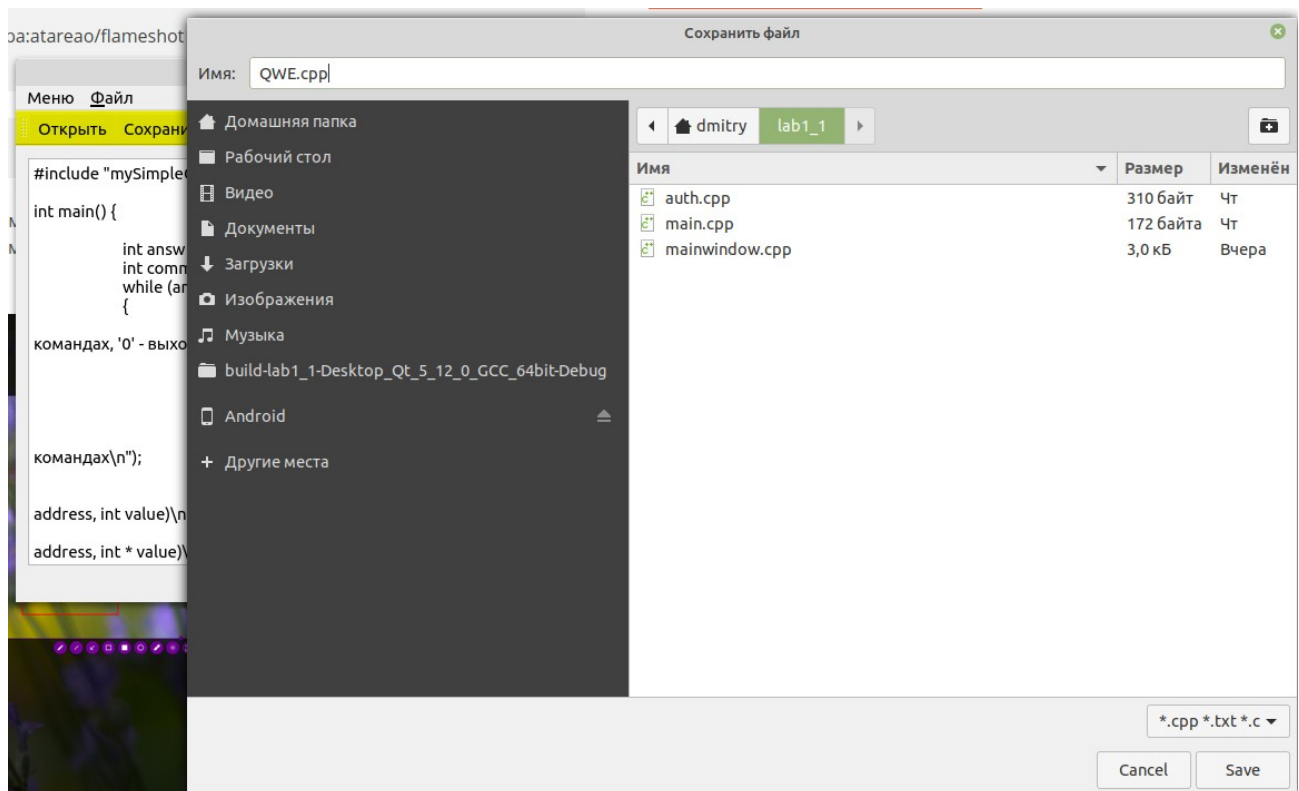
private:
    Ui::auth *ui;
};

#endif // AUTH_H

```

Скриншоты






```
QWE.cpp - Code::Blocks 16.0
Edit View Search Project Build Debug Tools Plugins Settings Help

global>
QWE.cpp
1  #include "mySimpleComputer.h"
2
3  int main() {
4
5      int answ = 1, address, value;
6      int command, operand;
7      while (answ)
8      {
9          printf("Введите номер команды. '1' - информация о командах, '0' - выход. \n");
10         scanf("%d", &answ);
11         switch (answ)
12         {
13             case 1:
14                 printf("1. Информация о командах\n");
15                 printf("2. sc_memoryInit()\n");
16                 printf("3. sc_memorySet(int address, int value)\n");
17                 printf("4. sc_memoryGet(int address, int * value)\n");
18                 printf("5. Показать все ячейки памяти \n");
19                 printf("6. Вывести значение value \n");
20                 printf("7. sc_memorySave(char * filename) \n");
21                 printf("8. sc_memoryLoad(char * filename) \n");
22                 printf("9. int sc_regInit()\n");
23                 printf("10. int sc_regSet(int reg, int value) \n");
24                 printf("11. Вывести значение registr \n");
25                 printf("12. int sc_regGet (int reg, int * value) \n");
26                 printf("13. int sc_commandEncode (int command, int operand, int * value) \n");
27                 printf("14. int sc_commandDecode (int value, int * command, int * operand) \n");
28                 printf("15. Вывести значение command и операнд \n");
29                 printf("0. Выход \n");
30                 break;
31             case 2:
32                 sc_memoryInit();
33                 break;
34             case 3:
35                 printf("address and value: ");
36                 scanf("%d %d", &address, &value);
37                 sc_memorySet(address, value);
38                 break;
39             case 4:
40                 printf("address: ");
41                 scanf("%d", &address);
42                 sc_memoryGet(address, &value);
43                 break;
44             case 5:
45                 sc_memoryShow();
46                 break;
47             case 6:
48                 printf("value == %d \n", value);
49                 break;
```