

# Описание программы для дисциплины «Технологии разработки ПО»

Выполнил Малышев Дмитрий, группа 3530201/90101

**Название программы:** «Поиск нужных реквизитов»

## Постановка задачи:

1. Организовать отображение пользовательского интерфейса
2. Реализовать возможность выбора файла для чтения и записи
3. Реализовать поиск и передачу данных из одного файла в другой
4. Обеспечить пользователя информацией, необходимой для работы с программой

## Ограничения:

1. Тип файла для чтения должен быть .doc или .txt
2. Тип файла для записи должен быть .xls
3. Необходимо закрыть все файлы, с которыми будет осуществляться работа

## Особенности реализации:

Программа реализована на языке программирования C++ с помощью фреймворка Qt. Реализация состоит из окна, в котором размещены кнопки выбора файла для чтения, выбора файла для записи, а также кнопка дополнительной информации.

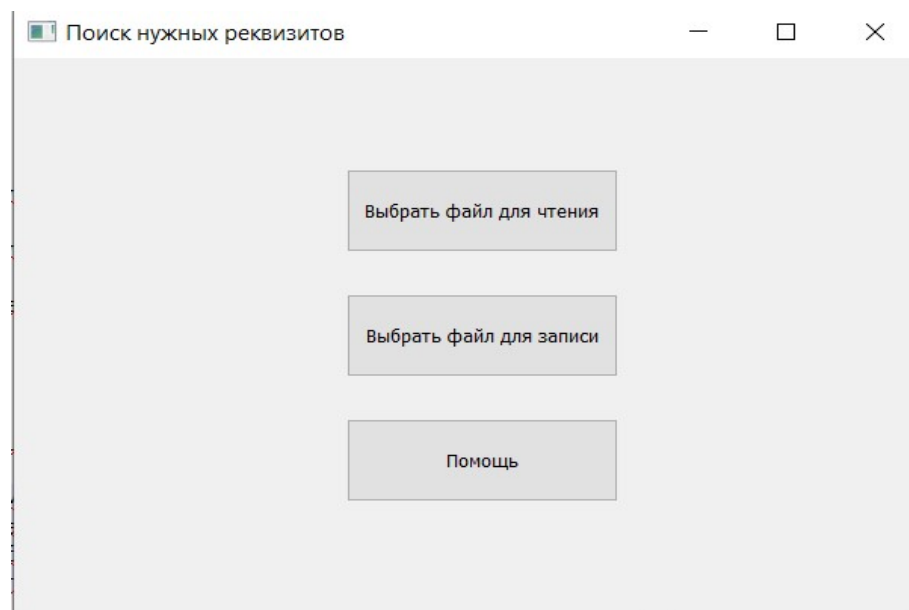


рис. 1. Главное меню

При нажатии кнопки «Выбрать файл для чтения» открывается всплывающее

окно, в котором пользователю предоставлена возможность выбора документа для чтения данных.

При нажатии кнопки «Выбрать файл для записи» открывается всплывающее окно, в котором пользователю предоставлена возможность выбора документа для записи данных из документа, выбранного как файл для чтения.

При нажатии кнопки «Помощь» открывается всплывающее окно, в котором отображена вся информация, необходимая для работы с данной программой

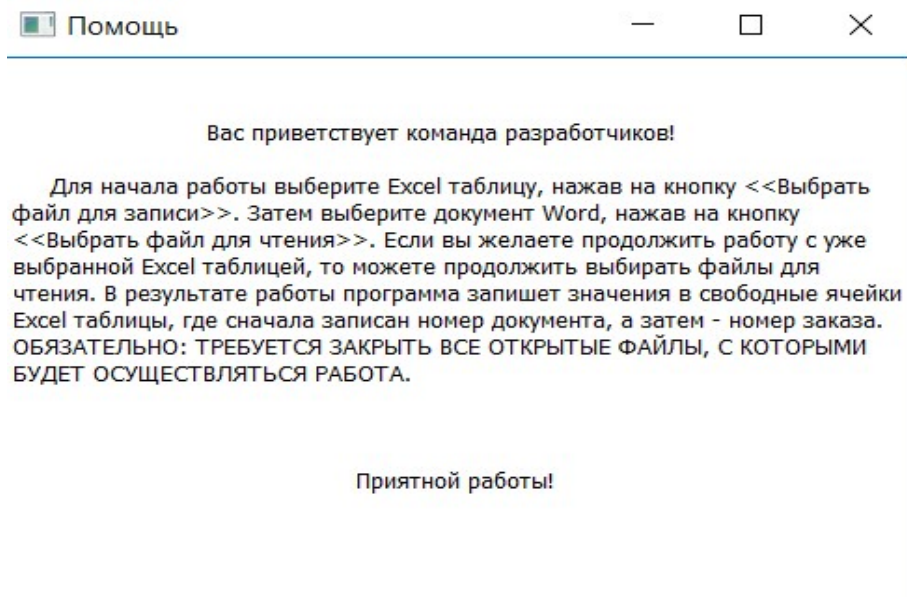


рис. 2. Кнопка «Помощь»

## Приложение:

---

```
#include "mainwindow.h"
#include "ui_mainwindow.h"

MainWindow::MainWindow(QWidget *parent)
: QMainWindow(parent)
//, ui(new Ui::MainWindow)
{

    OpenFile= new QPushButton("Выбрать файлдлячтения", this);
    chooseFile=new QPushButton("Выбрать файлдлязаписи",this);
    info=new QPushButton("Помощь", this);

    QString phrase = "Поиск нужныхреквизитов";
    this->setWindowTitle(phrase);
    this->setGeometry(600, 350, 600, 400);
    QVBoxLayout* vbox = new QVBoxLayout(this);
    QHBoxLayout* hbox = new QHBoxLayout();
    //QObject* tmp = new QObject;

    hbox->addWidget(OpenFile, 1, Qt::AlignVCenter);
    hbox->addWidget(info, 1);

    hbox->addWidget(chooseFile, 1);
    vbox->addLayout(hbox);
    // tmp->setLayout(hbox);
    // setCentralWidget(tmp);
    OpenFile->setGeometry(220, 80, OpenFile->geometry().width()+80,
        OpenFile->geometry().height()+30);
    info->setGeometry(220, 260, info->geometry().width()+80,
        info->geometry().height()+30);
    chooseFile->setGeometry(220, 170, chooseFile->geometry().width()+80,
        chooseFile->geometry().height()+30);
    connect(info, SIGNAL(clicked()), this, SLOT(OpenInfo()));
    connect(OpenFile, SIGNAL(clicked()), this, SLOT(open()));
    connect(chooseFile, SIGNAL(clicked()), this, SLOT(openWrite()));
    Information = new QTextEdit;
    Information->setWindowTitle("Помощь");
    QString phrase2 = "\n\n Васприветствуеткомандаразработчиков
        !\n\n Дляначалаработывыберите      Excel таблицу, нажавнакнопкуВыбрать <<
        файлдлязаписи >>. Затемвыберитедокумент Word, нажавнакнопкуВыбрать <<
        файлдлячтения >>. Есливыжелаетепродолжитьработусужевыбранной      Excel
        таблицей, томожетепродолжитьвыбиратьфайлыдлячтения      .
        Врезультатеработыпрограммазапишетзначениявсвободныеячейки      Excel таблицы,
        гдесначалазаписанномердокумента      , азатем - номерзаказа.\ОБЯЗАТЕЛЬНОn:
        ТРЕБУЕТСЯЗАКРЫТЬВСЕОТКРЫТЫЕФАЙЛЫ      , СКТОРЫМИБУДЕТОСУЩЕСТВЛЯТЬСЯРАБОТА
```

```

        .\n\n\n\n Приятной работы        !";
Information->append(phrase2);
Information->setGeometry(700,400,Information->geometry().width()-150,Information->geometry().height()-150);
Information->setReadOnly(1);
// ui->setupUi(this);
}

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

void MainWindow::OpenInfo()
{
    Information->show();
}

void MainWindow::open()
{
    fileName = QFileDialog::getOpenFileName(this,
        tr("Open Text file"), "", tr("Text Files (*.docx *.txt *.doc)"));

    try {
        openfile();
    }
    catch (...) {
        QMessageBox::critical(this,
            "Ошибка!",
            "Не удалось открыть файл : " + fileName);
        return;
    }
}

void MainWindow::openfile()
{
    QString textResult;
    QString type= typeFile(fileName);

    if(type == "doc" || type=="docx")
    {
        m_file.close();
        m_file.setFileName(fileName);
        const QString& file =m_file.fileName();
        QAxObject* wordApplication=new QAxObject("Word.Application");
        QAxObject *documents = wordApplication->querySubObject("Documents()");
        QAxObject *document = documents->querySubObject("Add(QVariant)", file);
        QAxObject *words = document->querySubObject("Words");

        int countWord = words->dynamicCall("Count()").toInt();

        for (int a = 1; a <= countWord; a++)

```

```

{
    textResult.append(words->querySubObject("Item(int)",
        a)->dynamicCall("Text()").toString());
}
textResult.replace('\r', '\n');
wordApplication->dynamicCall("Quit()");
delete wordApplication;
fileName= QDir::currentPath ()+"\\tmp.txt";
m_file.setFileName(fileName);
if (m_file.openMode() != QIODevice::WriteOnly)
{
    m_file.close();
    m_file.open(QIODevice::WriteOnly);
}
QTextStream fstream(&m_file);
fstream << textResult;
m_file.close();

}

m_file.setFileName(fileName);
if (m_file.openMode() != QIODevice::ReadOnly)
{
    m_file.close();
    m_file.open(QIODevice::ReadOnly);
}
QTextStream fstream(&m_file);
fstream.setAutoDetectUnicode(1);

int mod=0;

while (fstream.readLineInto(&string)) {

    //fstream >> string;
    string1=search(string,properties[(mod+1)%2]);
    string = search(string,properties[mod%2]);
    if (string == NULL && string1==NULL)
    {
        continue;
    }
    if(string!=NULL)
    {
        props.first=string;
    }
    if(string1!=NULL)
    {
        props.second=string1;
        break;
    }
}

```

```

    }

}

if(props.first=="&&props.second=="")
{
    QMessageBox::critical(this,
        "Ошибка!",
        "В файлеотсутствуютреквизиты : " + m_file.fileName());
}
else
{
    write();
}
m_file.close();

//

}

void MainWindow::openWrite()
{
    fileNameWrite = QFileDialog::getOpenFileName(this,
        tr("Open file"), "", tr("Excel file (*.xlsx *.xls)"));

    m_file1.close();
    m_file1.setFileName(fileNameWrite);

}

void MainWindow::write()
{
    /*
    m_file1.open(QIODevice::Append);
    QTextStream fstream(&m_file1);
    fstream << endl;
    fstream << string;
    m_file.close();
    */
    QString file = m_file1.fileName();
    if(file=="")
    {
        QMessageBox::critical(this,
            "Ошибка!",
            "Не выбранфайлдлязаписи  " );
    }
}

```

```

}
else
{
    QAxObject* mexcel= new QAxObject("Excel.Application",this);
    mexcel->dynamicCall("setVisible(bool)",false);
    QAxObject* workbooks=mexcel->querySubObject("Workbooks");
    QAxObject* workbook=workbooks->querySubObject("Open(const QString&)",file);
    QAxObject* sheets= workbook->querySubObject("worksheets");
    QAxObject* sheet= sheets->querySubObject("Item(int)",1);
    int indexr=1;

    while(1)
    {
        QAxObject* cell2=sheet->querySubObject("Cells(int,int)",indexr,1);
        QString val=cell2->dynamicCall("Value()").toString();
        if(val.size()==0)
        {
            cell2=sheet->querySubObject("Cells(int,int)",indexr,2);
            val=cell2->dynamicCall("Value()").toString();
            if(val.size()==0)
            {
                break;
            }
            else
            {
                indexr++;
                continue;
            }
        }
        else
        {
            indexr++;
            continue;
        }
    }
    QAxObject* cell=sheet->querySubObject("Cells(int,int)",indexr,1);
    QVariant value=cell->dynamicCall("SetValue(const QVariant&)",props.first);
    QAxObject* cell1=sheet->querySubObject("Cells(int,int)",indexr,2);
    QVariant value1=cell1->dynamicCall("SetValue(const QVariant&)",props.second);
    delete cell;
    delete cell1;
    props.first="";
    props.second="";
    workbook->dynamicCall("Save()");
    delete workbook;
    delete workbooks;
    mexcel->dynamicCall("Quit()");
    delete mexcel;
    QMessageBox::warning(this,

```

```

        "Завершено",
        "Передача завершена");
    }
}

```

```

QString MainWindow::search(QString& str,QString Document )
{
    QString res;
    int index=0;
    int check = 0;
    for (int i = 0; i < str.size(); i++){
        if (str[i] == Document[0]) {
            index = i;
            break;
        }
    }

    for (int i = 0; i < Document.size(); i++) {
        if (str[index++] == Document[i])check++;
    }

    //if ((index - 1 > 0) && (str[index] == ' ')) check = 0;

    if (check == Document.size()) {
        for (int i = index; i < str.size(); i++)
            if(str[i]!=' ')
                res.push_back(str[i]);
        return res;
    }
    else return NULL;
}

```

```

}
QString MainWindow::typeFile(QString str)
{
    QString res;
    int index;
    for(int i =0;i<str.size();i++)
    {
        if(str[i]==".")
        {
            index=i;
            break;
        }
    }
    for(int i =index+1;i<str.size();i++)
    {
        res.push_back(str[i]);
    }
}

```



```
}  
    return res;  
}
```

---