

Федеральное агентство связи
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Сибирский государственный университет телекоммуникаций и
информатики»

Кафедра прикладной математики и кибернетики

Отчет по лабораторной работе
по дисциплине «Визуальное программирование и человеко-машинное
взаимодействие»

Лабораторная работа №6

Выполнил: студент 3 курса
группы ИП-811
Мироненко К. А

Проверил: доцент кафедры
ПМиК
Мерзлякова Е. Ю.

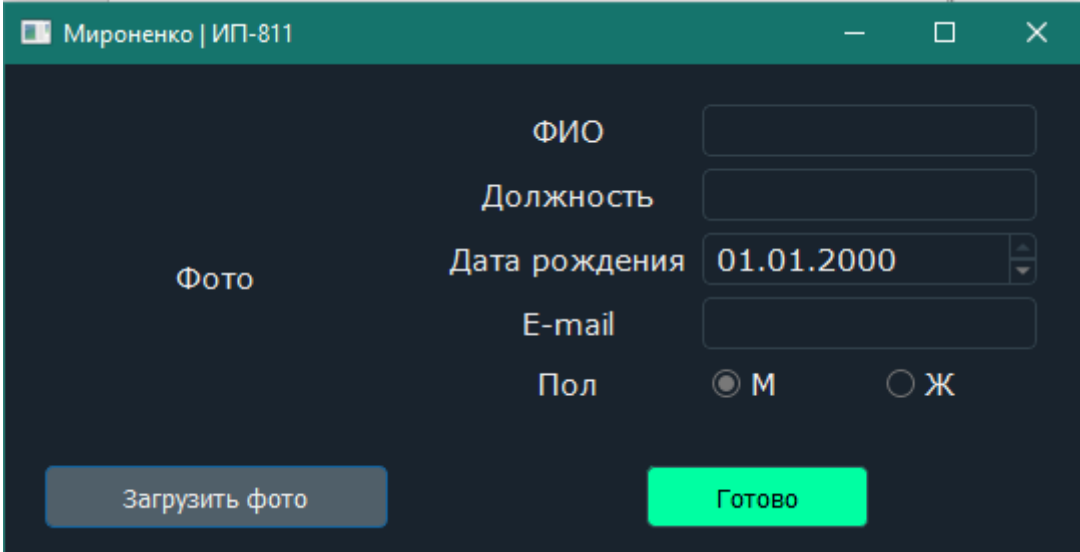
Оглавление

1. Постановка задачи.....	3
2. Примеры работы программы	4
<i>Приложение</i> Листинг.....	6

1. Постановка задачи

Цель: научиться обмену данными между формами. Использовать стили.

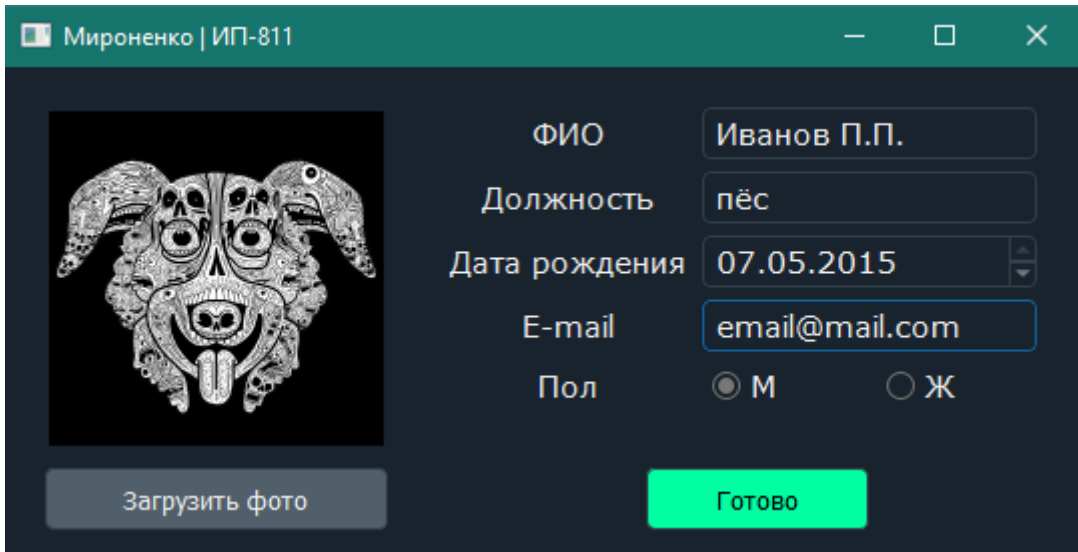
2. Примеры работы программы



The screenshot shows a window titled "Мироненко | ИП-811". The form contains the following fields and controls:

- Фото**: A label for the photo field.
- ФИО**: A text input field.
- Должность**: A text input field.
- Дата рождения**: A date picker showing "01.01.2000".
- E-mail**: A text input field.
- Пол**: Radio buttons for "М" (Male) and "Ж" (Female).
- Buttons**: "Загрузить фото" (Load photo) and "Готово" (Done).

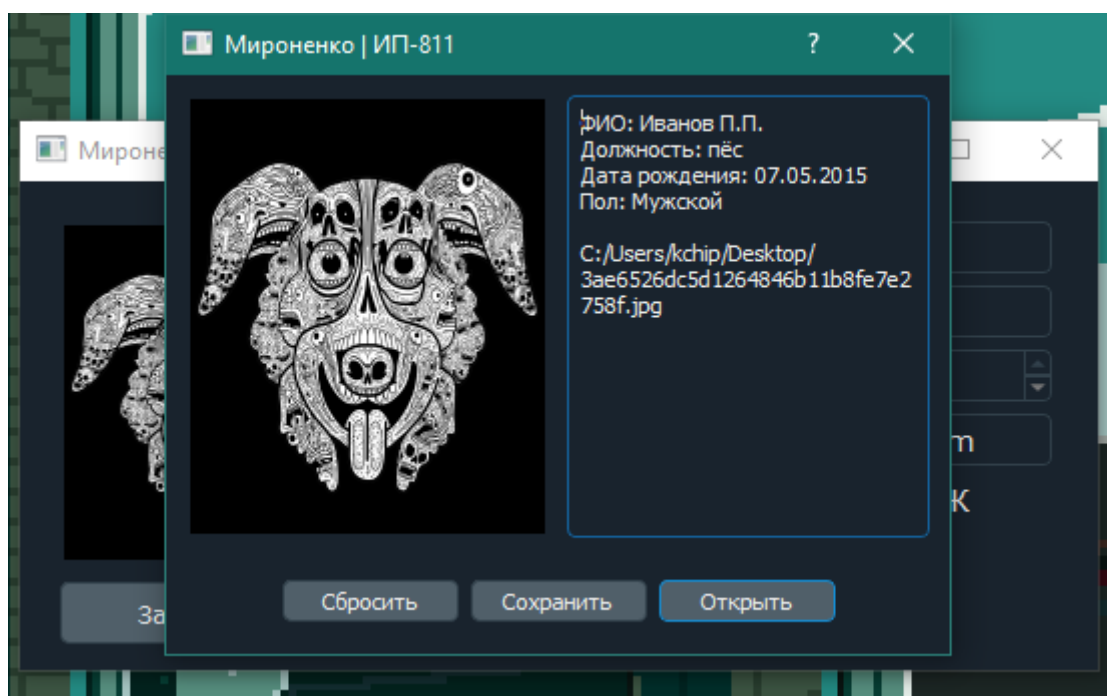
(Основная форма)



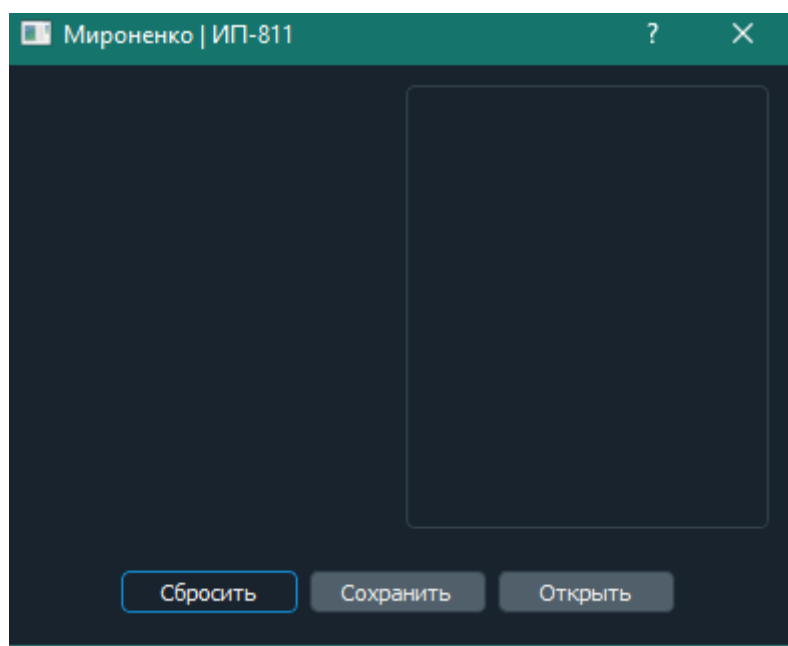
The screenshot shows the same window with the following data entered:

- Фото**: A placeholder image of a dog's head.
- ФИО**: "Иванов П.П."
- Должность**: "пёс"
- Дата рождения**: "07.05.2015"
- E-mail**: "email@mail.com"
- Пол**: "М" (Male) is selected.
- Buttons**: "Загрузить фото" and "Готово".

(Основная форма с заполненными данными)



(Вторая форма, в которую были переданы данные из основной)



(Вторая форма, после нажатия кнопки «Сбросить»)

Приложение Листинг

main.cpp

```
#include "mainwindow.h"
#include <QApplication>
#include <QFile>
#include <QTextStream>
#include <QTranslator>
#include <QLibraryInfo>

int main(int argc, char *argv[])
{
    QApplication a(argc, argv);
    QFile f(":/qdarkstyle/style.qss");
    if (!f.exists()) {
        printf("Unable to set stylesheet, file not found\n");
    }
    else {
        f.open(QFile::ReadOnly | QFile::Text);
        QTextStream ts(&f);
        qApp->setStyleSheet(ts.readAll());
    }
    QTranslator qtTranslator;
    qtTranslator.load("qt_ru", QLibraryInfo::location(QLibraryInfo::TranslationsPath));
    a.installTranslator(&qtTranslator);
    MainWindow w;
    w.show();

    return a.exec();
}
```

mainwindow.h

```
#ifndef MAINWINDOW_H
#define MAINWINDOW_H

#include <QMainWindow>
#include <QFileDialog>
#include <QMessageBox>
#include "mironenkoform.h"

namespace Ui {
class MainWindow;
}

class MainWindow : public QMainWindow
{
    Q_OBJECT

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

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

signals:
    void sendData(QString str);
private slots:
    void btnReadyClicked();
    void btnUploadPhotoClicked();
};

#endif // MAINWINDOW_H
```

mainwindow.cpp

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

MainWindow::MainWindow(QWidget *parent) :
    QMainWindow(parent),
    ui(new Ui::MainWindow)
{
    ui->setupUi(this);
    myform = new MironenkoForm();
    connect(ui->btnUploadPhoto, SIGNAL(clicked()), this, SLOT(btnUploadPhotoClicked()));
    connect(ui->btnReady, SIGNAL(clicked()), this, SLOT(btnReadyClicked()));
    connect(this, SIGNAL(sendData(QString)), myform, SLOT(recieveData(QString)));
}

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

void MainWindow::btnUploadPhotoClicked(){
    QString filename = QFileDialog::getOpenFileName(nullptr, "Выберите изображение", QDir::currentPath(),
    "*.png *.jpg *.gif *.jpeg");
    QImage image(filename);
    ui->photo->setPixmap(QPixmap::fromImage(image));
    ui->photoPath->setText(filename);
}

void MainWindow::btnReadyClicked()
{
    if(ui->photoPath->text() == "" || ui->lineName->text() == "" || ui->linePost->text() == "" || ui->lineEmail->text() == "")
    {
        QMessageBox msgBox;
        msgBox.setText("Все поля должны быть заполнены!");
        msgBox.setIcon(QMessageBox::Critical);
        msgBox.setDefaultButton(QMessageBox::Ok);
        msgBox.exec();
    }
    else {
        QString st = "ФИО: " + ui->lineName->text() +
```

```

        "\nДолжность: " + ui->linePost->text() +
        "\nДата рождения: " + ui->dateOfBirth->text();
        "\nEmail: " + ui->lineEmail->text();
        if (ui->radioButtonMan->isChecked())
            st += "\nПол: Мужской";
        else
            st += "\nПол: Женский";
        st += "*" + ui->photoPath->text();
        emit sendData(st);
    }
}

```

mirotenkoform.h

```

#ifndef MIRONENKOFORM_H
#define MIRONENKOFORM_H

#include <QWidget>
#include <QAbstractButton>
#include <QFileDialog>
#include <QTextDocumentWriter>

namespace Ui {
class MironenkoForm;
}

class MironenkoForm : public QDialog
{
    Q_OBJECT

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

private:
    Ui::MironenkoForm *ui;
public slots:
    void receiveData(QString str);
private slots:
    void btnBoxClicked(QAbstractButton *button);
};

#endif // MIRONENKOFORM_H

```

mirotenkoform.cpp

```

#include "mirotenkoform.h"
#include "ui_mironenkoform.h"

MironenkoForm::MironenkoForm(QWidget *parent) :
    QDialog(parent),
    ui(new Ui::MironenkoForm)
{
    ui->setupUi(this);
}

```



```

    connect(ui->btnBox, SIGNAL(clicked()), this, SLOT(btnBoxClicked()));
}

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

void MironenkoForm::recieveData(QString str){
    this->show();
    QStringList lst = str.split("*");
    ui->textEdit->setText(lst.at(0) + "\n\nPhoto: " + lst.at(1));
    QImage image(lst.at(1));
    ui->label->setPixmap(QPixmap::fromImage(image));
}

void MironenkoForm::btnBoxClicked(QAbstractButton *button){
    if (button->text() == "Сбросить"){
        ui->textEdit->clear();
        ui->label->clear();
    }
    else if(button->text() == "Сохранить"){
        QString filename = QFileDialog::getSaveFileName(nullptr, "Сохранить файл", QDir::currentPath(),
        "*.txt");
        QTextDocumentWriter writer;
        writer.setFileName(filename);
        writer.write(ui->textEdit->document());
    }
    else if(button->text() == "Открыть"){
        QString filename = QFileDialog::getOpenFileName(nullptr, "Открыть файл", QDir::currentPath(),
        "*.txt");
        QFile file(filename);
        if(file.open(QIODevice::ReadOnly | QIODevice::Text))
            ui->textEdit->setPlainText(file.readAll());
        QStringList inf = ui->textEdit->toPlainText().split("\n");
        QImage image(inf.at(6));
        ui->label->setPixmap(QPixmap::fromImage(image));
    }
}

```