|  |  |
| --- | --- |
|  | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет  имени Н. Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н. Э. Баумана)** |

|  |  |
| --- | --- |
| ФАКУЛЬТЕТ | «Информатика и системы управления» (ИУ) |

|  |  |
| --- | --- |
| КАФЕДРА | «Информационная безопасность» (ИУ8) |

Домашнее задание №2

ПО КУРСУ

«Алгоритмические языки»

на тему «Разработка приложений с использованием библиотеки QT»

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Студент | ИУ8-23 |  |  |  | В. С. Ажгирей  М. Ю. Григорьева |
|  | (Группа) |  |  |  | (И. О. Фамилия) |
|  |  |  |  |  |  |
| Преподаватель: |  |  |  |  | М. В. Малахов |
|  |  |  |  |  | (И.О. Фамилия) |

2024

**Введение**

**Общее задание**: необходимо реализовать графическое приложение на языке C++ с использованием Qt.

**Вариант 6.** Разработайте приложение – кликер. В приложении вводится идентификатор запущенной сторонней программы (рекомендуется запускать блокнот) и текстовая строка. При запуске приложения каждый символ из заданной строки должен с паузой в 1 секунду отображаться в запущенной сторонней программе (имитация ввода с клавиатуры). По окончании ввода заданной текстовой строки необходимо вывести всплывающее окно с надписью, что операция успешно выполнена.

**Цели и задачи:**

- создать программы для приложения на языке С++;

- изучить библиотеку QT для разработки приложения;

- разработать дизайн и функционал приложения;

- отладить программы;

**-** выполнить решение контрольного примера небольшой размерности с помощью программы;

- подготовить отчет по лабораторной работе.

**Основная часть**

**Исходный код программы**

# Файл заголовка mainwindow.h:

#ifndef MAINWINDOW\_H

#define MAINWINDOW\_H

#include <QMainWindow>

#include <QProcess>

QT\_BEGIN\_NAMESPACE

namespace Ui {

class MainWindow;

}

QT\_END\_NAMESPACE

class MainWindow : public QMainWindow

{

Q\_OBJECT

public:

MainWindow(QWidget \*parent = nullptr);

void addProgramPath(const QString &);

~MainWindow();

private slots:

void on\_pushButtonSelectApplication\_clicked();

void on\_pushButtonCompleteTask\_clicked();

void on\_pushButtonSelectFile\_clicked();

void on\_pushButton\_clicked();

private:

Ui::MainWindow \*ui;

};

#endif // MAINWINDOW\_H

# Файл описания mainwindow.cpp:

#include "mainwindow.h"

#include "./ui\_mainwindow.h"

#include <QFileDialog>

#include <QProcess>

#include <QMessageBox>

#include <QTimer>

#include <QDebug>

MainWindow::MainWindow(QWidget \*parent)

: QMainWindow(parent)

, ui(new Ui::MainWindow)

{

ui->setupUi(this);

this->setWindowTitle("Clicker");

this->setWindowIcon(QIcon("E:/LaboratoryWorks\_IU8/HomeWork\_2\_02/Clicker/icon.png"));

addProgramPath("C:/Windows/notepad.exe");

}

void MainWindow::addProgramPath(const QString & path)

{

ui->comboBoxProgramPath->addItem(path);

if (!path.isEmpty()) ui->comboBoxProgramPath->setCurrentText(path);

//qDebug() << ui->comboBoxProgramPath->currentText(); // Debug

}

MainWindow::~MainWindow()

{

delete ui;

}

void MainWindow::on\_pushButtonSelectApplication\_clicked()

{

QString programPath = QFileDialog::getOpenFileName(this, "Выберите приложение", QDir::homePath(), "Приложения (\*.exe)");

addProgramPath(programPath);

//qDebug() << programPath; // Debug

}

void MainWindow::on\_pushButtonCompleteTask\_clicked()

{

QString inputText = ui->textEditInputText->toPlainText();

QString filePath = ui->lineEditFilePath->text();

QString programPath = ui->comboBoxProgramPath->currentText();

if (inputText.isEmpty() || filePath.isEmpty())

{

QMessageBox::warning(this, "Предупреждение", inputText.isEmpty() ? "Пожалуйста, введите текст для отправки в программу." : "Пожалуйста, выберете файл для записи.");

return;

}

this->hide();

QTimer \*timer = new QTimer(this);

connect(timer, &QTimer::timeout, this, [=]() mutable {

if (inputText.isEmpty())

{

timer->stop();

QMessageBox::information(this, "Успех", "Операция успешно выполнена.");

this->show();

delete timer;

return;

}

QString symbol = inputText.at(0);

QFile file(filePath);

if (!file.open(QIODevice::WriteOnly | QIODevice::Append))

{

QMessageBox::critical(this, "Ошибка", "Не удалось открыть файл для дозаписи.");

return;

}

QTextStream out(&file);

out << symbol;

file.close();

QProcess \*process = new QProcess(this);

process->setProcessChannelMode(QProcess::ForwardedChannels);

process->start(programPath, {filePath});

connect(process, &QProcess::finished, process, &QProcess::deleteLater);

QFile::copy(filePath, filePath);

inputText.remove(0, 1);

});

timer->start(1000);

}

void MainWindow::on\_pushButtonSelectFile\_clicked()

{

QString filePath = QFileDialog::getOpenFileName(this, "Выберите приложение", QDir::homePath(), "Текстовые файлы (\*.txt)");

ui->lineEditFilePath->setText(filePath);

}

void MainWindow::on\_pushButton\_clicked()

{

QMessageBox::information(this, "Справка", "Инструкция:\n"

"1) Выберете приложение в котором будет открыт текстовый файл.\n"

"2) Нажмите кнопку \"Выбрать приложение\"\n"

"3) В поле для ввода текста введите текст. Нажмите кнопку \"Выполнить задачу\"\n"

"4) Выберите файл в который будет записываться текст, "

"для этого пропишите абсолюный путь к файлу или нажмите на кнопку \"Выбрать файл\"");

}

**Исполняемый файл main.cpp:**

#include "mainwindow.h"

#include <QApplication>

int main(int argc, char \*argv[])

{

QApplication a(argc, argv);

MainWindow w;

w.show();

return a.exec();

}

**Файл CMakeLists.txt:**

cmake\_minimum\_required(VERSION 3.5)

project(Clicker VERSION 0.1 LANGUAGES CXX)

set(CMAKE\_AUTOUIC ON)

set(CMAKE\_AUTOMOC ON)

set(CMAKE\_AUTORCC ON)

set(CMAKE\_CXX\_STANDARD 17)

set(CMAKE\_CXX\_STANDARD\_REQUIRED ON)

find\_package(QT NAMES Qt6 Qt5 REQUIRED COMPONENTS Widgets)

find\_package(Qt${QT\_VERSION\_MAJOR} REQUIRED COMPONENTS Widgets)

set(PROJECT\_SOURCES

main.cpp

mainwindow.cpp

mainwindow.h

mainwindow.ui

)

if(${QT\_VERSION\_MAJOR} GREATER\_EQUAL 6)

qt\_add\_executable(Clicker

MANUAL\_FINALIZATION

${PROJECT\_SOURCES}

.gitignore CMakeLists.txt dialog.cpp dialog.h dialog.ui main.cpp mainwindow.cpp mainwindow.h mainwindow.ui

)

# Define target properties for Android with Qt 6 as:

# set\_property(TARGET Clicker APPEND PROPERTY QT\_ANDROID\_PACKAGE\_SOURCE\_DIR

# ${CMAKE\_CURRENT\_SOURCE\_DIR}/android)

# For more information, see https://doc.qt.io/qt-6/qt-add-executable.html#target-creation

else()

if(ANDROID)

add\_library(Clicker SHARED

${PROJECT\_SOURCES}

)

# Define properties for Android with Qt 5 after find\_package() calls as:

# set(ANDROID\_PACKAGE\_SOURCE\_DIR "${CMAKE\_CURRENT\_SOURCE\_DIR}/android")

else()

add\_executable(Clicker

${PROJECT\_SOURCES}

)

endif()

endif()

target\_link\_libraries(Clicker PRIVATE Qt${QT\_VERSION\_MAJOR}::Widgets)

# Qt for iOS sets MACOSX\_BUNDLE\_GUI\_IDENTIFIER automatically since Qt 6.1.

# If you are developing for iOS or macOS you should consider setting an

# explicit, fixed bundle identifier manually though.

if(${QT\_VERSION} VERSION\_LESS 6.1.0)

set(BUNDLE\_ID\_OPTION MACOSX\_BUNDLE\_GUI\_IDENTIFIER com.example.Clicker)

endif()

set\_target\_properties(Clicker PROPERTIES

${BUNDLE\_ID\_OPTION}

MACOSX\_BUNDLE\_BUNDLE\_VERSION ${PROJECT\_VERSION}

MACOSX\_BUNDLE\_SHORT\_VERSION\_STRING ${PROJECT\_VERSION\_MAJOR}.${PROJECT\_VERSION\_MINOR}

MACOSX\_BUNDLE TRUE

WIN32\_EXECUTABLE TRUE

)

include(GNUInstallDirs)

install(TARGETS Clicker

BUNDLE DESTINATION .

LIBRARY DESTINATION ${CMAKE\_INSTALL\_LIBDIR}

RUNTIME DESTINATION ${CMAKE\_INSTALL\_BINDIR}

)

if(QT\_VERSION\_MAJOR EQUAL 6)

qt\_finalize\_executable(Clicker)

endif()

**Графический интерфейс**

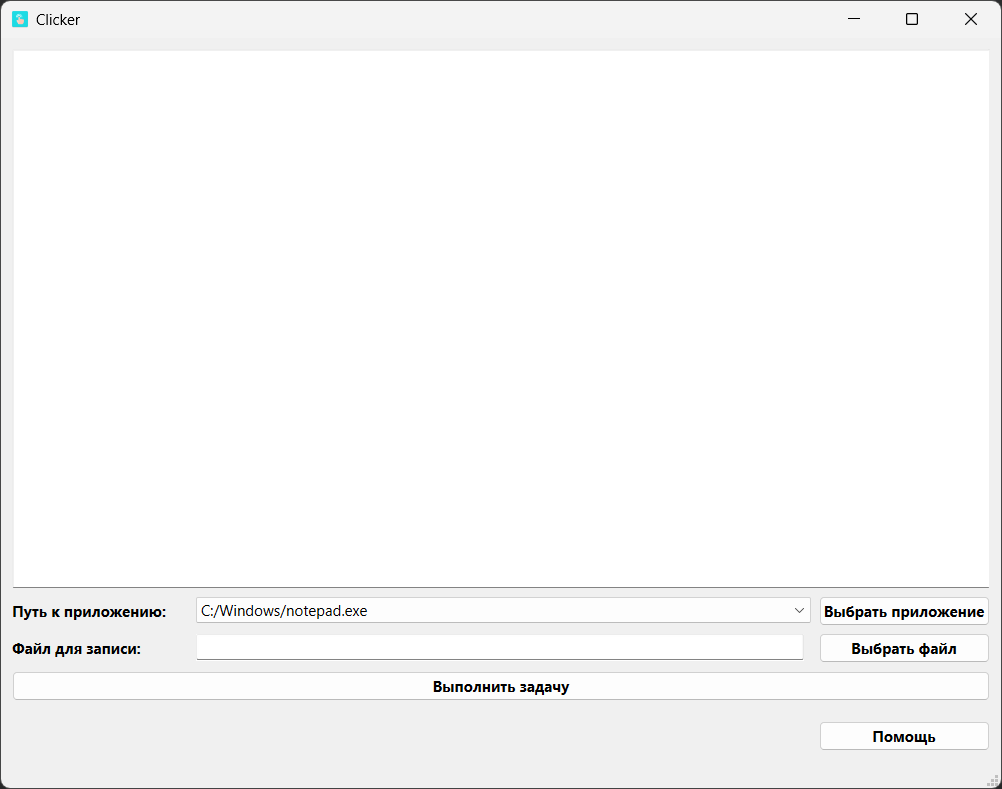
****

Рисунок 1 – Главное окно приложения

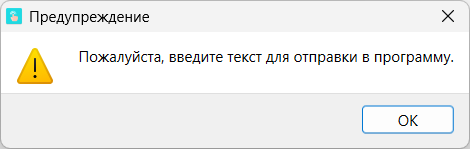


Рисунок 2 – Окно-предупреждение «Введите текст»

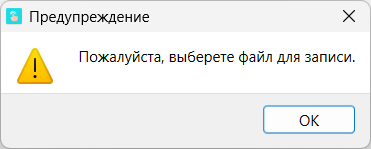


Рисунок 3 – Окно-предупреждение «Выберите файл»

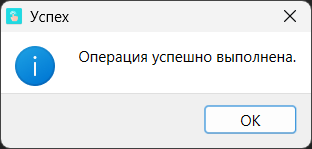


Рисунок 4 – Окно-успех «Операция успешно выполнена»

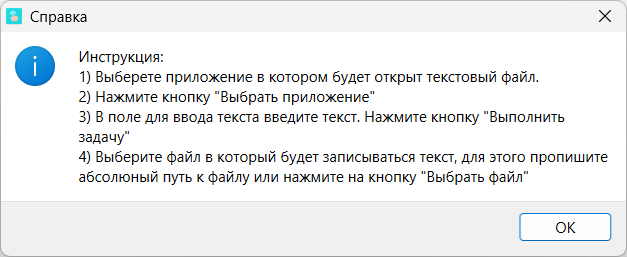


Рисунок 5 – Окно-справка «Инструкция»

**Снимки выполнения работы программы**

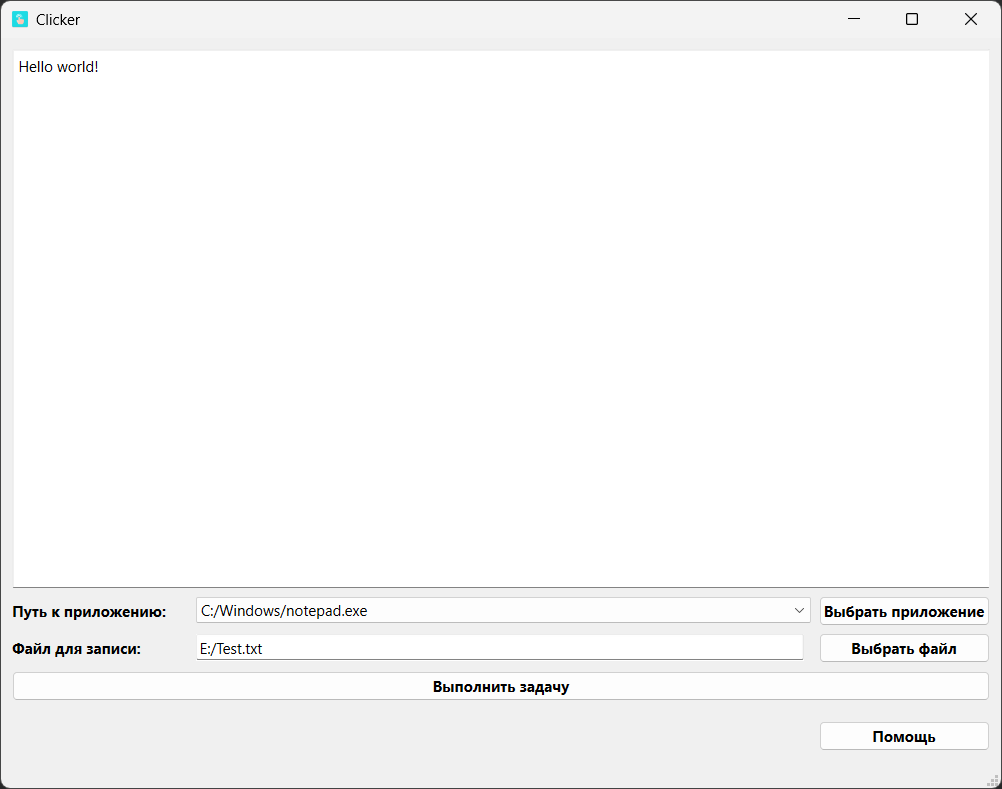
****

Рисунок 6 – Входные данные

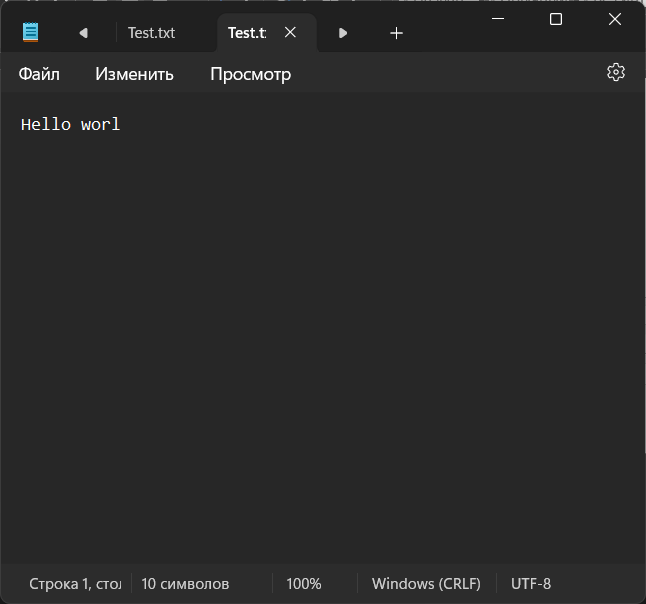


Рисунок 7 – Работа программы

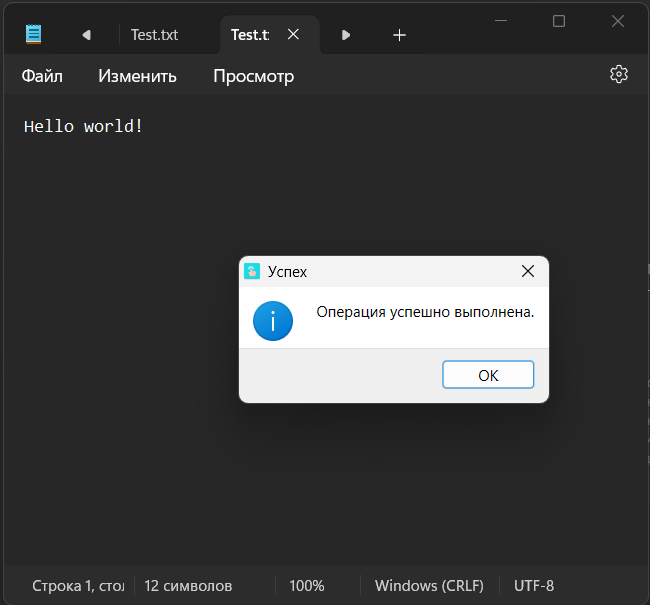


Рисунок 8 – Результат работы программы

**Заключение**

Задачи домашнего задания были решены, результаты проверены. Изучена на практике библиотека QT, разработка приложения в программе QT Creator, создание различных окон приложения и их внешнего вида, а также вывод текста в стороннюю программу.