МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

 «БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра ИИТ

\

ЛАБОРАТОРНАЯ РАБОТА №1-2

По дисциплине: «ОСИСП»

Выполнил:

Студент ФЭИС

3-го курса, группы ПО-5

Пищик А.В

Проверила:

Дряпко А.В.

Брест 2021

**Цель работы:** приобрести практические навыки проектирования и разработки приложений с графическим

пользовательским интерфейсом в ОС Windows средствами Qt

Вариант 15

*Задание:*

Приложение для отображения информации о системе. Реализовать возможности по выводу

информации о процессоре, модулях памяти и материнской плате, жестком диске, установленной

ОС.

***mainwindow.h:***

#ifndef MAINWINDOW\_H

#define MAINWINDOW\_H

#include <QMainWindow>

#include <QDebug>

#include <QLocale>

#include <iostream>

#include <fstream>

#include <string>

using namespace std;

QT\_BEGIN\_NAMESPACE

namespace Ui { class MainWindow; }

QT\_END\_NAMESPACE

class MainWindow : public QMainWindow

{

Q\_OBJECT

public:

MainWindow(QWidget \*parent = nullptr);

~***MainWindow***();

private slots:

void **on\_processor\_clicked**(bool checked);

void **result\_update**(int size\_cols, QStringList header, QStringList mass);

void **update\_str**(string s);

void **read\_file\_info**();

void **on\_mother\_clicked**();

void **on\_os\_clicked**();

void **on\_info\_programm\_clicked**();

void **on\_ram\_clicked**();

void **on\_hard\_clicked**();

private:

Ui::MainWindow \*ui;

};

#endif // MAINWINDOW\_H

***main.cpp:***

#include "mainwindow.h"

#include <QApplication>

#include <QLabel>

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

{

QApplication a(argc, argv);

MainWindow w;

w.show();

return a.exec();

}

***mainwindow.cpp***

#include "mainwindow.h"

#include "ui\_mainwindow.h"

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

{

ui->setupUi(this);

bool checked = true;

on\_processor\_clicked(checked);

}

string \*res = new string[100];

QStringList header, result\_data, mass;

int size\_str = 0;

MainWindow::~***MainWindow***()

{

delete ui;

}

void MainWindow::**result\_update**(int size\_cols, QStringList header, QStringList mass) {

string result[size\_cols];

string s;

ifstream f1;

f1.open("test1.txt");

int i = 0;

while (!f1.eof()) {

getline(f1, s);

result[i] = s;

i++;

}

f1.close();

for (int i = 0; i < size\_cols; i++)

{

res[i] = result[i];

}

ui->tableWidget->setRowCount(size\_cols); // Устанавливаем количество строк на size\_cols

ui->tableWidget->setColumnCount(1);

for(int i = 0; i < size\_cols; i++) {

ui->tableWidget->verticalHeader()->setSectionResizeMode(i, QHeaderView::Stretch);

}

ui->tableWidget->horizontalHeader()->setSectionResizeMode(0, QHeaderView::Stretch);

ui->tableWidget->setVerticalHeaderLabels(header);

ui->tableWidget->horizontalHeader()->*setVisible*(false);

for(int i = 0; i < size\_cols; i++) {

QString data = QString::fromStdString(res[i]);

result\_data.append(data);

}

int countt = 0;

for (int i = countt; i < ui->tableWidget->rowCount(); ++i) {

QTableWidgetItem \* item;

for (int j = 0; j < ui->tableWidget->columnCount(); ++j) {

item = new QTableWidgetItem;

for(int s = 0; s < size\_str; s++) {

if(i==s)

item->setText(result\_data.value(mass[s].toInt()));

ui->tableWidget->setItem(i, j, item);

}

}

}

countt++;

ui->tableWidget->resizeColumnsToContents();

ui->tableWidget->resizeRowsToContents();

}

void MainWindow::**update\_str**(string s) {

string new\_str;

bool flag = false;

int flag1 = 0;

for (int i = 0; i < s.length(); i++) {

if (s[i] == '|') {

flag = false;

flag1++;

}

if (flag1 == 1) {

new\_str += '\n';

flag1++;

}

if (flag == true) {

new\_str += s[i];

}

if (s[i] == '\n' || (s[i] == ' ' && s[++i] == ' ')) {

flag = false;

}

if (s[i] == '=') {

size\_str++;

flag = true;

}

}

ofstream ff;

ff.open("test1.txt");

ff << new\_str;

ff.close();

}

void MainWindow::**read\_file\_info**() {

ifstream f;

string s;

f.open("test.txt");

while (!f.eof()) {

char c = f.get();

s += c;

}

f.close();

remove("test.txt");

update\_str(s);

}

void MainWindow::**on\_processor\_clicked**(bool checked) {

size\_str = 0;

ui->tableWidget->*setVisible*(true);

ui->switch\_info->setText("Информация о процессоре");

header.clear(), result\_data.clear(), mass.clear();

system("wmic CPU get Name, NumberOfCores, NumberOfLogicalProcessors, Maxclockspeed /format:list >> test.txt"); //proc

read\_file\_info();

ui->tableWidget->setRowCount(0);

ui->tableWidget->setColumnCount(0);

for(int i = 0; i < size\_str; i++) {

QString m = QVariant(i).toString();

mass.append(m);

}

result\_update(size\_str, header<<"Частота на ядро"<<"Имя процессора"<<"Количество ядер"<<"Количество потоков", mass);

}

void MainWindow::**on\_mother\_clicked**() {

size\_str = 0;

ui->tableWidget->*setVisible*(true);

header.clear(), result\_data.clear(), mass.clear();

ui->switch\_info->setText("Информация о Материнской плате");

system("wmic baseboard get manufacturer, product /format:list >> test.txt"); //mather

read\_file\_info();

ui->tableWidget->setRowCount(0);

ui->tableWidget->setColumnCount(0);

for(int i = 0; i < size\_str; i++) {

QString m = QVariant(i).toString();

mass.append(m);

}

result\_update(size\_str, header<<"Производитель"<<"Модель", mass);

}

void MainWindow::**on\_os\_clicked**() {

size\_str = 0;

ui->tableWidget->*setVisible*(true);

header.clear(), result\_data.clear(), mass.clear();

ui->switch\_info->setText("Информация об Операционной системе");

system("wmic OS get Manufacturer, Version /format:list >> test.txt"); //OS

read\_file\_info();

ui->tableWidget->setRowCount(0);

ui->tableWidget->setColumnCount(0);

for(int i = 0; i < size\_str; i++) {

QString m = QVariant(i).toString();

mass.append(m);

}

result\_update(size\_str, header<<"Производитель OS"<<"Версия", mass);

}

void MainWindow::**on\_info\_programm\_clicked**() {

ui->tableWidget->*setVisible*(false);

ui->switch\_info->setText("Программу сделал студент группы ПО-5, Пищик Алексей");

}

void MainWindow::**on\_ram\_clicked**() {

size\_str = 0;

ui->tableWidget->*setVisible*(true);

header.clear(), result\_data.clear(), mass.clear();

ui->switch\_info->setText("Информация об Модулях оперативной памяти");

system("wmic memorychip get speed, PartNumber /format:list >> test.txt"); //ram

read\_file\_info();

ui->tableWidget->setRowCount(0);

ui->tableWidget->setColumnCount(0);

int r = size\_str/2;

for(int i = 0; i < r; i++) {

header<<"Партийный номер"<<"Частота";

}

for(int i = 0; i < size\_str; i++) {

QString m = QVariant(i).toString();

mass.append(m);

}

result\_update(size\_str, header, mass);

}

void MainWindow::**on\_hard\_clicked**() {

size\_str = 0;

ui->tableWidget->*setVisible*(true);

header.clear(), result\_data.clear(), mass.clear();

ui->switch\_info->setText("Информация об Накопителях");

system("wmic diskdrive get model, size /format:list >> test.txt"); //hard

read\_file\_info();

ui->tableWidget->setRowCount(0);

ui->tableWidget->setColumnCount(0);

int r = size\_str/2;

for(int i = 0; i < r; i++) {

header<<"Модель"<<"Размер";

}

for(int i = 0; i < size\_str; i++) {

QString m = QVariant(i).toString();

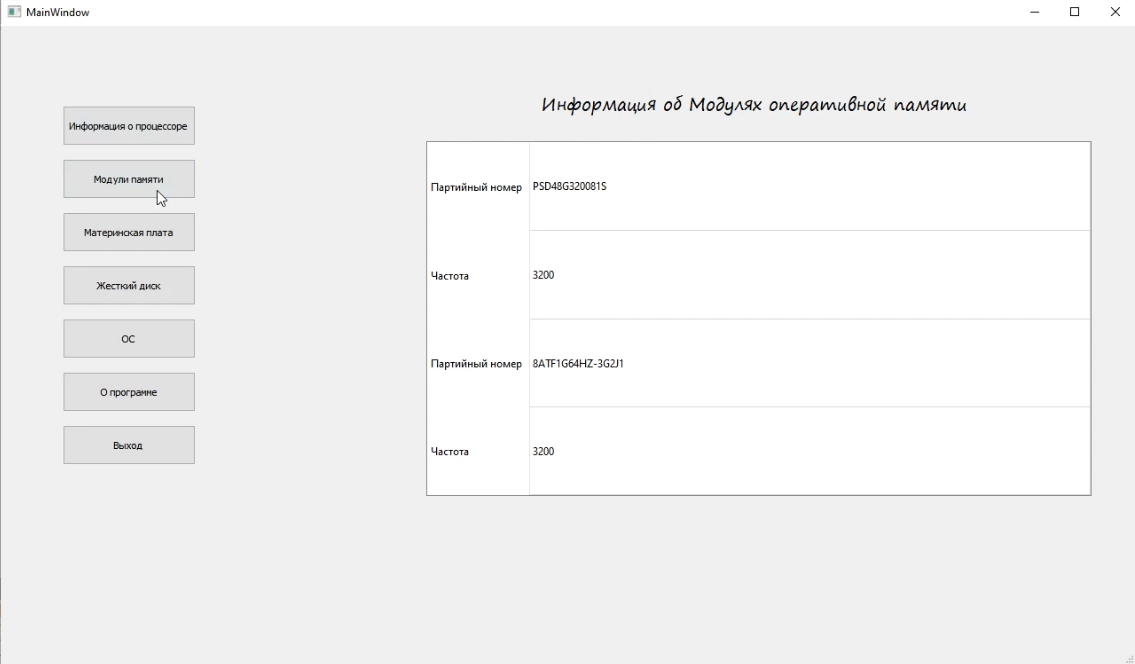
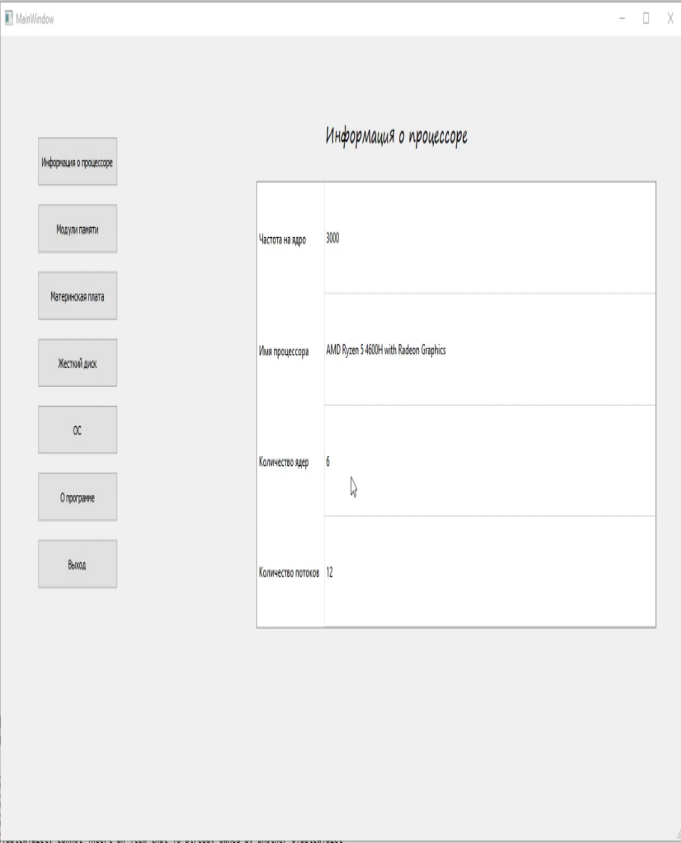
mass.append(m);

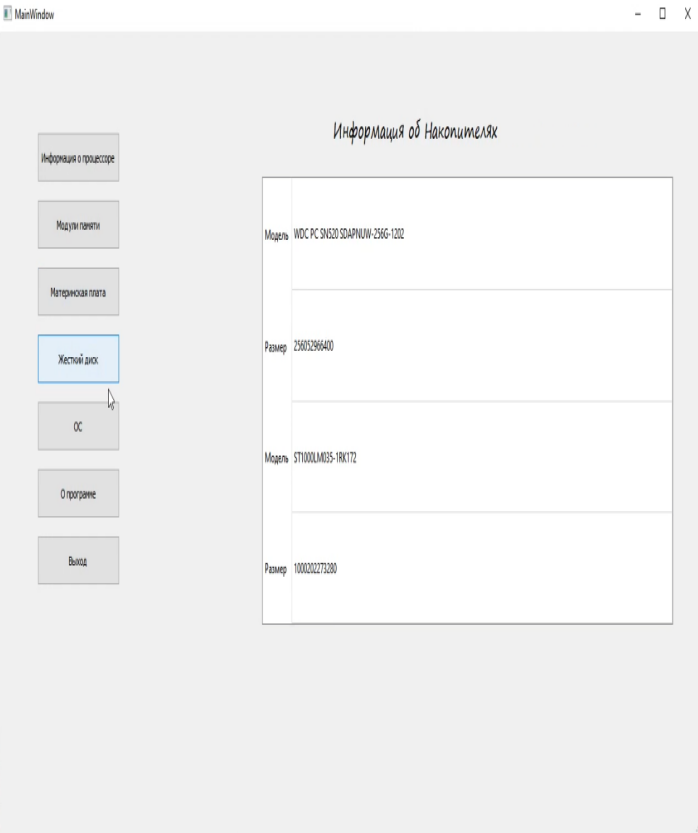
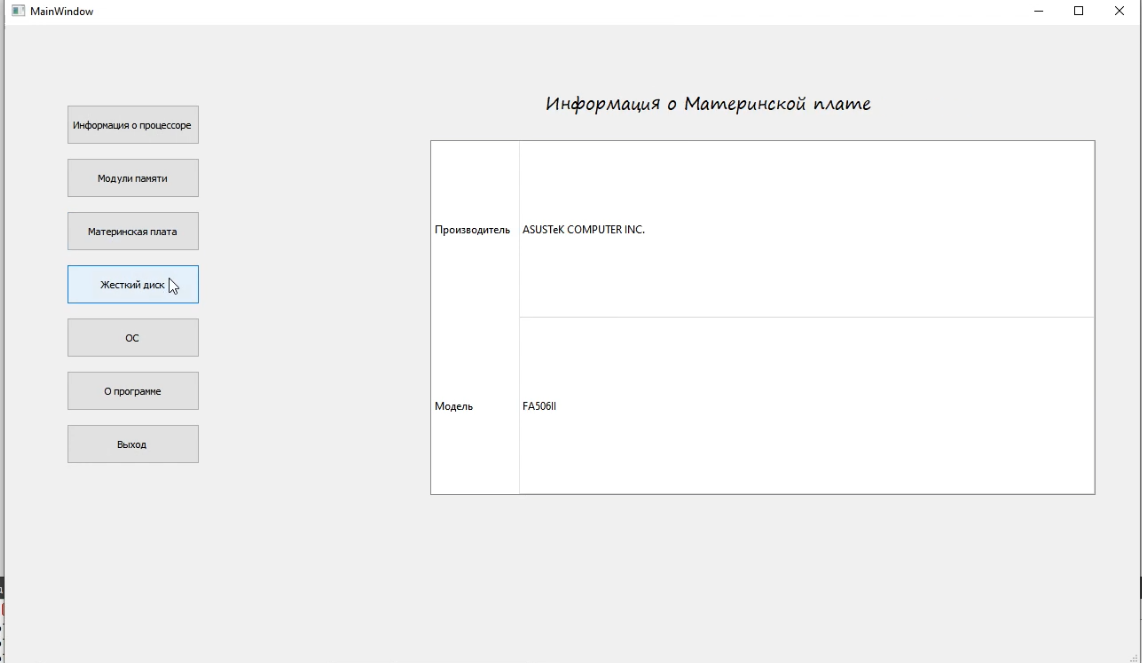
}

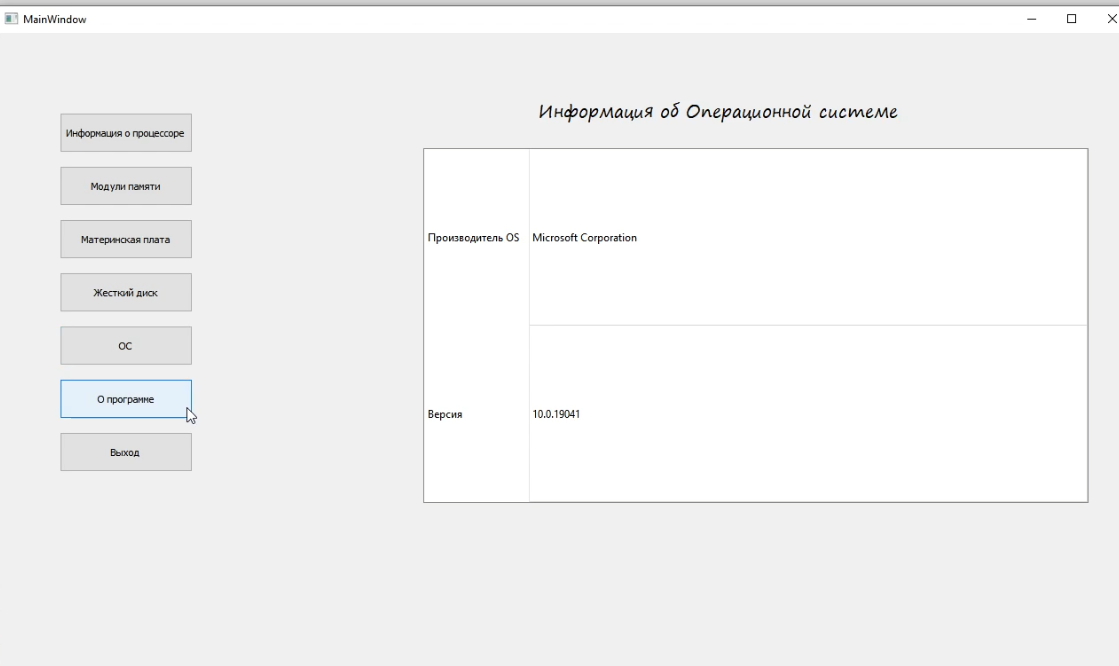
result\_update(size\_str, header, mass);

}

*Работа программы:*







Вывод: приобрел практические навыки проектирования и разработки приложений с графическим пользовательским интерфейсом в ОС Windows средствами Qt.