Министерство образования Республики Беларусь

Управление образования

«Белорусский государственный университет информатики и радиоэлектроники»

Отчёт

По лабораторной работе №8

«**Исключительные ситуации.**

**Запуск Qt приложений .exe**»

|  |  |
| --- | --- |
|  | Выполнил:  студент группы 262101  Калинков Даниил Русланович  Проверил:  Ловецкий Михаил Юрьевич |

Минск, 2023

Цель работы: Изучить принцип обработки исключительных ситуаций.

Научится создавать исполняемый файл .exe.

Вариант 6

Задание 1:



Реализация:

#include <iostream>

#include <stdexcept>

class Budget {

private:

int number;

public:

Budget() : number(0) {} // Инициализируем number значением по умолчанию 0

void check(int value) {

try {

if (value < 1000) {

number = value;

}

else {

throw std::invalid\_argument("Number must be less than 1000");

}

}

catch (const std::invalid\_argument& e) {

std::cerr << "Error: " << e.what() << std::endl;

}

}

int getNumber() const {

return number;

}

};

int main() {

Budget budget;

int inputValue;

std::cout << "Enter a number: ";

std::cin >> inputValue;

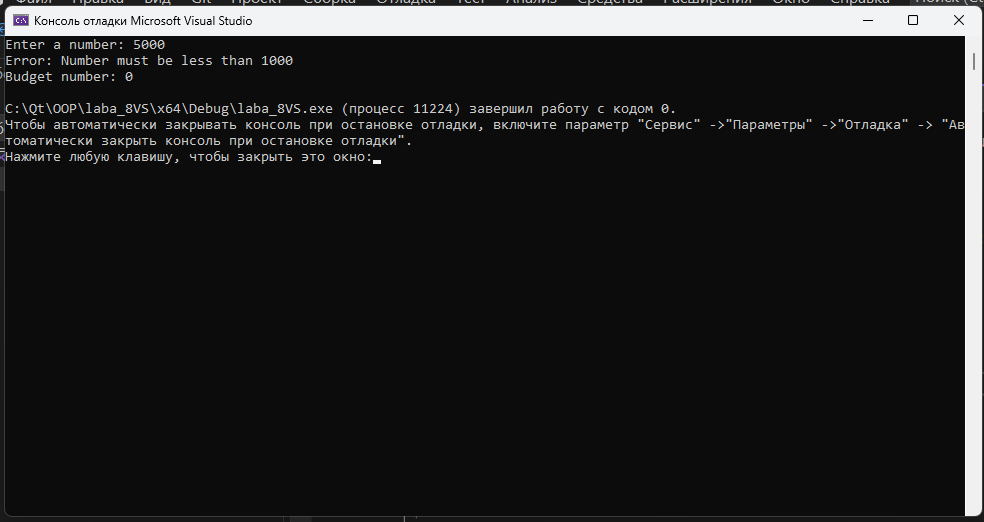
budget.check(inputValue);

std::cout << "Budget number: " << budget.getNumber() << std::endl;

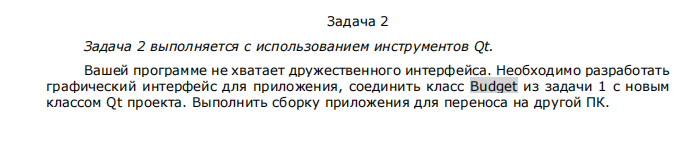
return 0;

}

результат:



Задание 2:



Задание выпонено:

**mainwindow.h**

#ifndef MAINWINDOW\_H

#define MAINWINDOW\_H

#include <second.h>

#include <QMainWindow>

#include <QMessageBox>

#include <QKeyEvent>

QT\_BEGIN\_NAMESPACE

namespace **Ui** { class **MainWindow**; }

QT\_END\_NAMESPACE

class **MainWindow** : public QMainWindow

{

Q\_OBJECT

public:

**MainWindow**(QWidget \*parent = nullptr);

~***MainWindow***();

public slots:

void **test**();

void ***keyPressEvent***(QKeyEvent\* e);

signals:

void **secondSignal**(QString str);

private slots:

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

void **on\_pushButton\_2\_clicked**();

private:

Ui::MainWindow \*ui;

Second \*second;

};

#endif // MAINWINDOW\_H

class **ErrorMessage1** {

public:

static void **showError**(QWidget\* parent) {

QMessageBox::critical(*parent*, "НИ\*\*\*СЕБЕ", "НЕВДЕНГАХсчаСТЬЕ");

}

};

class **ErrorMessage2** {

public:

static void **showError**(QWidget\* parent, const QString& message) {

QMessageBox::critical(*parent*, "ЧИТАЙ","ТАМсказаноЧИСЛО,АТЫЧТОНАПИСАЛ");

}

};

**mainwindow.cpp**

#include "mainwindow.h"

#include "ui\_mainwindow.h"

#include <QDebug>

MainWindow::**MainWindow**(QWidget \*parent)

: QMainWindow(*parent*)

, ui(new Ui::MainWindow)

{

ui->setupUi(this);

second = new Second(this);

connect(ui->pushButton, SIGNAL(clicked()),this,SLOT(test()));

connect(this, &MainWindow::secondSignal,second,&Second::secondSlot);

}

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

{

delete ui;

}

void MainWindow::***keyPressEvent***(QKeyEvent \*e)

{

if(e->key()==Qt::Key\_Escape)

{

ui->label\_2->setText("Клавиша ESC");

}

}

void MainWindow::**test**()

{

qDebug() << "слот тест запусчен";

emit secondSignal(ui->lineEdit->text());

}

void MainWindow::**on\_pushButton\_clicked**()

{

QString text = ui->lineEdit->text();

bool ok;

int number = text.toInt(*&ok*);

if (!ok) {

ErrorMessage2::showError(this, "Неверный формат числа");

} else if (number > 1000) {

ErrorMessage1::showError(this);

} else {

// Ваш код для обработки числа, если оно прошло проверку

}

}

void MainWindow::**on\_pushButton\_2\_clicked**()

{

QMessageBox::critical(this, "CriticalDamage","это гг братан");

QMessageBox::question(this, "medicine is powerless here","хоЧешь продолжИть?");

}

**second.h**

#ifndef SECOND\_H

#define SECOND\_H

#include <QObject>

#include <QMainWindow>

#include <QWidget>

#include <QMessageBox>

#include <QKeyEvent>

#include <QMainWindow>

#include <QMessageBox>

#include <QKeyEvent>

class **Second** : public QObject

{

Q\_OBJECT

public:

explicit **Second**(QObject \*parent = nullptr);

signals:

public slots:

void **secondSlot**(QString str);

};

#endif // SECOND\_H

**second.cpp**

#include <QObject>

#include <QDebug>

#include <iostream>

#include <stdexcept>

#include <QMainWindow>

#include <QMessageBox>

#include <QKeyEvent>

class **Budget**; // Декларация класса Budget

class **Second** : public QObject {

Q\_OBJECT

public:

**Second**(QObject\* parent = nullptr);

public slots:

void **secondSlot**(QString str);

private:

Budget\* budget; // Указатель на объект Budget

};

class **Budget** {

private:

int number;

public:

**Budget**() : number(0) {}

void **check**(int value) {

try {

if (value < 1000) {

number = value;

} else {

throw std::invalid\_argument("Number must be less than 1000");

}

} catch (const std::invalid\_argument& e) {

std::cerr << "Error: " << e.*what*() << std::endl;

}

}

int **getNumber**() const {

return number;

}

friend void Second::**secondSlot**(QString str); // Объявление secondSlot как друга

};

Second::**Second**(QObject \*parent)

: QObject(*parent*) {

budget = new Budget(); // Создание объекта Budget

}

void Second::**secondSlot**(QString str) {

bool ok; // Флаг для проверки успешности преобразования

int intValue = str.toInt(*&ok*);

if (ok) {

// Преобразование прошло успешно, intValue содержит целое число

if (intValue>1000){

// QMessageBox::critical(this, "CriticalDamage","это гг братан");

// QMessageBox::question(this, "medicine is powerless here","хоЧешь продолжИть?");

}

budget->check(intValue);

std::cout << "Budget number: " << budget->getNumber() << std::endl;

} else {

// Преобразование не удалось

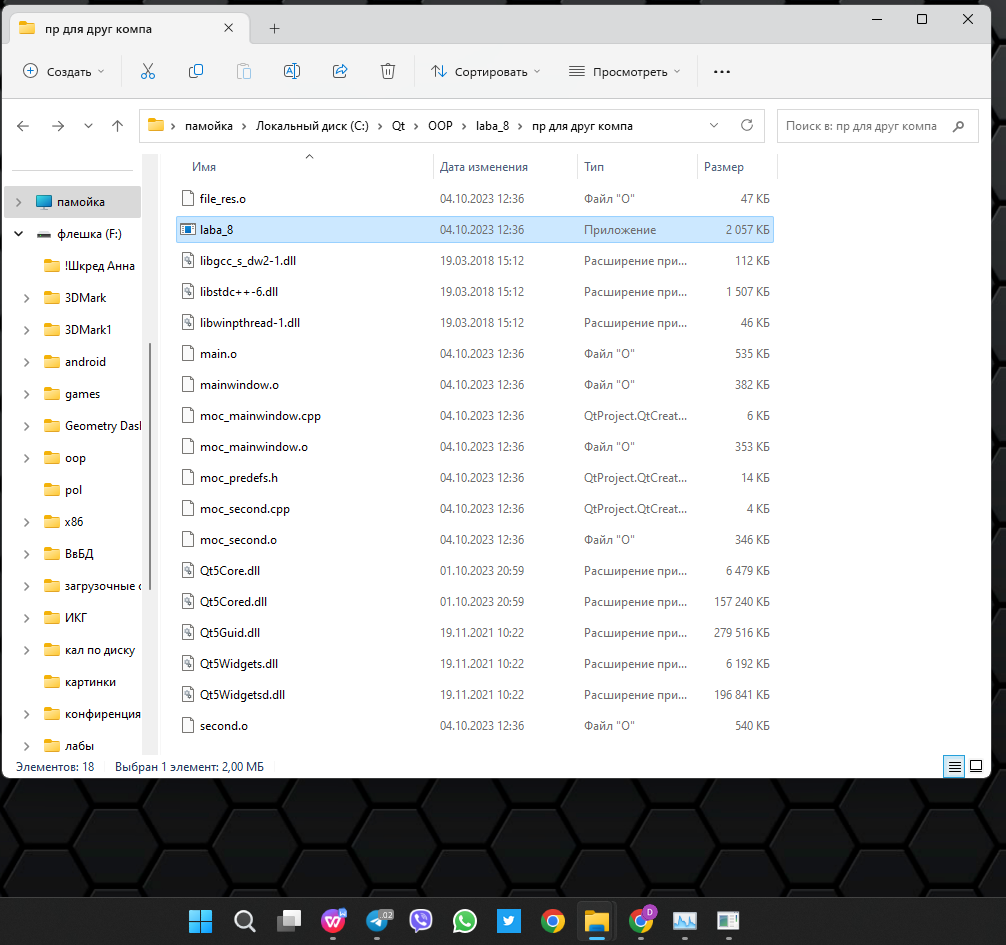
qDebug() << "Invalid input: not an integer";

}

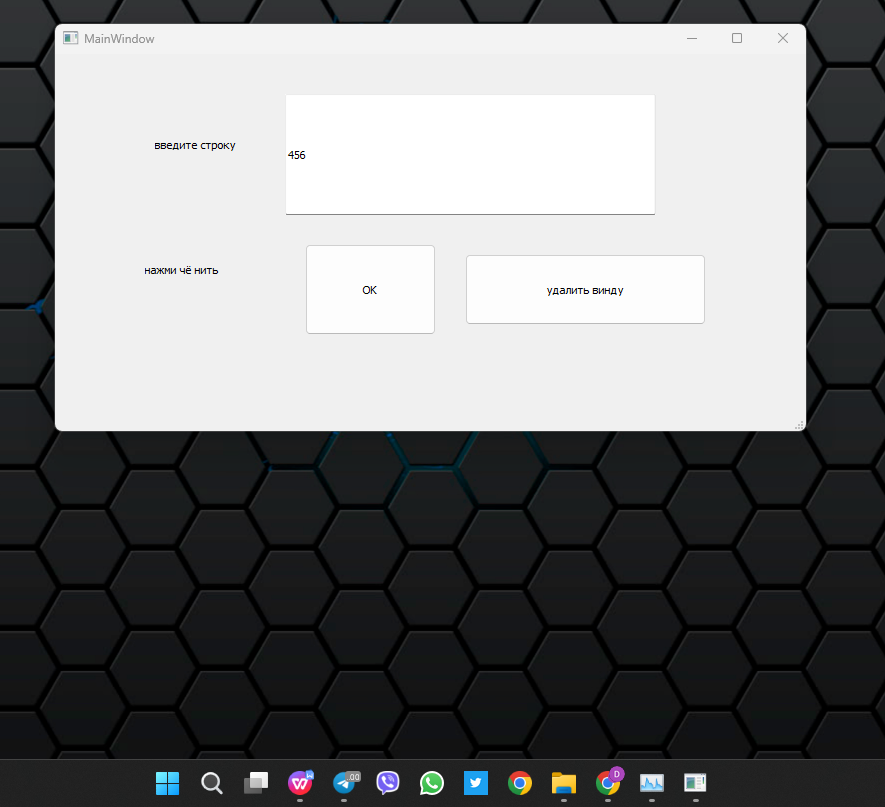
qDebug() << "Переданная строка: " << str;

}

**Оно работает:Все функции работают несмотря на то что QT не запусчен ,приложение готово для запуска на другом компе так же все библиотеки подключены**







После нажания клавиши ESC   
