Министерство образования и науки Российской Федерации  
Федеральное‌ ‌государственное‌ ‌бюджетное‌ ‌образовательное‌ ‌учреждение‌

высшего‌ ‌образования‌

**«Пермский национальный исследовательский  
политехнический университет»**

Кафедра «Информационные технологии и автоматизированные системы»

**О Т Ч Ё Т**

**по лабораторной работе (ТЗ)**

Тема: Разработка калькулятора

Выполнил работу

студент группы РИС-20-1б

Томилов В.А.

Проверила

Доцент кафедры ИТАС

к.т.н ПоляковаО.А.

Пермь, 2021

**Постановка задачи**

1. Разработать алгоритм калькулятора в соответствии с заданием.
2. Реализовать алгоритм в виде программы на алгоритмическом языке С++.
3. Разработать интерфейс средствами Qt.
4. Использование анимации, автоматизации, креативной визуализации повышает рейтинг автора проекта и влияет на экзаменационную оценку.

Вариант 17

|  |  |  |
| --- | --- | --- |
| 17 | Хранение капиталов в банке. | Банк предлагает своим клиентам условия:   1. Положить деньги на год под p - процентов годовых с начислением процентов по окончанию года 2. Положить деньги на год под p1 – процентов с начислением процентов ежемесячно 3. Положить деньги на год с ежедневным начислением p2 – процентов на каждую полную тысячу вклада 4. Положить деньги на год с ежедневным начислением p3 – процентов на сумму вклада   Банк может изменять уговорённые проценты. Калькулятор получает на вход величину вклада в рублях и копейках или в долларах и центах, а также вариант условий размещения капитала с вычислением величины вклада к концу срока.  Калькулятор должен иметь окна для исходных данных и результатов, для ввода данных, операций и служебных клавиш, клавиши изменения вида валюты. Должен позволять корректировать последнее введённое число и выдавать сообщения об ошибках. |

**Анализ задачи**

1. Для решения задачи необходимо:

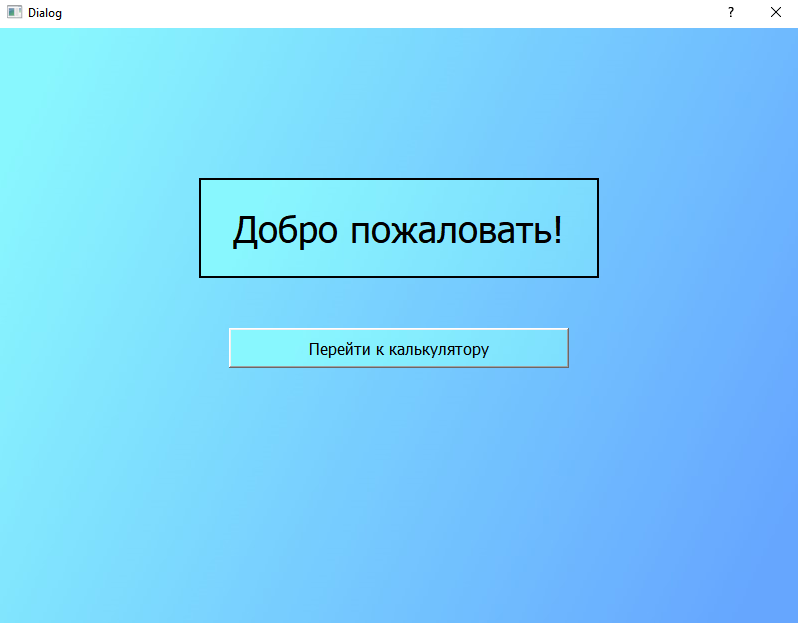
1.1. Создать три класса форм Qt Designer, для 3 окон работы калькулятора соответственно (начальный экран, окно калькулятора, дополнительная информация для вычислений):

homewindow.h/homewindow.cpp/homewindow.ui

mainwindow.h/mainwindow.cpp/mainwindow.ui

var.h/var.cpp/var.ui

1.2. Для каждой формы разработать дизайн, прописать для всех активных модулей Qt Designer (QButton, QLineEdits, etc) сигналы и их функции.

1) homewindow (Начальный экран)

В данной форме активна одна кнопка (“Перейти к калькулятору”). При нажатии на нее, появится основное окно калькулятора. Сигналом для данной операции является нажатие по иконке кнопки.

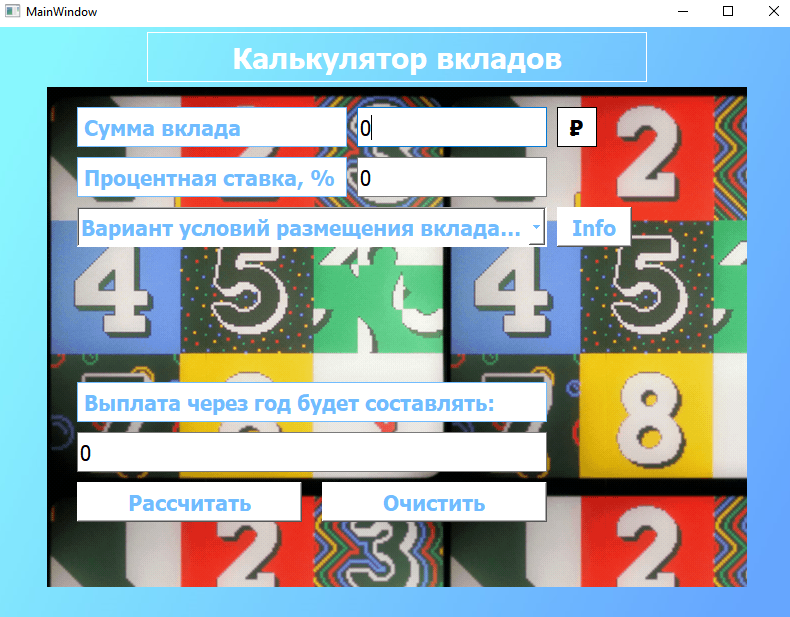
void homewindow::**on\_pushButton\_clicked**()

{

calculatorWindow->show();

this->close();

}

2) mainwindow (Окно калькулятора)

В данном окне присутствуют такие активные модули как:

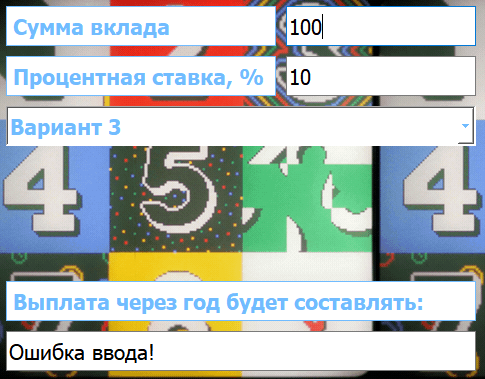
* 1. В данном поле пользователь вводит сумму вклада, который он хочет положить в банк. Реализовано данное поле с помощью виджета QLineEdit. Сигналом для данного модуля служит изменение текста в правой рамке:

void MainWindow::**on\_lineEdit\_textChanged**(const QString &arg1)

При вводе переменная преобразуется из QString в double. Производится проверка значения: если Сумма вклада (sum <= 0) выводится сообщение об ошибке ввода в дополнительное поле QLineEdit, в котором выводится результат вычеслений.



Также существует особый случай условий размещения вклада, при котором происходит ежедневное начисление процентов на каждую полную тысячу вклада. Следовательно, если сумма вклада (sum < 1000) и выбран 3 вариант размещения вклада, будет выводится сообщение об ошибке ввода.



void MainWindow::**on\_lineEdit\_textChanged**(const QString &arg1)

{

QString summa= arg1;

sum = summa.toDouble();

if ((sum <= 0))

ui->lineEdit\_3->setText("Ошибка ввода!");

else if ((sum < 1000) && (ui->comboBox->currentIndex()==3))

ui->lineEdit\_3->setText("Ошибка ввода!");

else ui->lineEdit\_3->setText("0");

}

void MainWindow::**on\_comboBox\_currentIndexChanged**(int index)

{

if ((sum < 1000) && (index==3))

ui->lineEdit\_3->setText("Ошибка ввода!");

else ui->lineEdit\_3->setText("0");

}

* 1. В данном поле пользователь вводит процентную ставку, под которую планируется положить вклад в банк. Реализовано данное поле с помощью виджета QLineEdit. Сигналом для данного модуля служит изменение текста в правой рамке:

void MainWindow::**on\_lineEdit\_2\_textChanged**(const QString &arg1)

При вводе переменная преобразуется из QString в double. Производится проверка значения: если Сумма вклада (percent <= 0) или (percent >100) выводится сообщение об ошибке ввода в дополнительное поле QLineEdit, в котором выводится результат вычеслений.

void MainWindow::**on\_lineEdit\_2\_textChanged**(const QString &arg1)

{

QString perc= arg1;

percent=perc.toDouble();

if ((percent <=0)||(percent>100))

ui->lineEdit\_3->setText("Ошибка ввода!");

else ui->lineEdit\_3->setText("0");

}

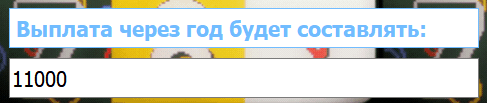
* 1. Данная кнопка отвечает за перевод суммы вклада и выплаты из одной валюты в другую (рубль – доллар/ доллар - рубль). Сигналом для данной операции является нажатие по иконке кнопки.

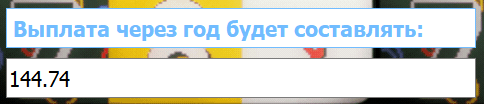
void MainWindow::**on\_pushButton\_clicked**(bool)

При нажатии на кнопку есть 2 возможных исхода:

1) Если нажать на иконку рубля, значит сумма вклада и выплата делятся на 76 (курс рубля к доллару), а также все значения выводятся в QLineEdit соответствующих полей. Иконка рубля меняется на иконку доллара.

2) Если нажать на иконку доллара, значит сумма вклада и выплата умножаются на 76 (курс рубля к доллару), а также все значения выводятся в QLineEdit соответствующих полей. Иконка доллара меняется на иконку рубля.

Сумма и выплата в рублях:

Сумма и выплата в долларах:

bool check=false;

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

{

check=(!check);

if (check)

{

sum /=76;

vklad /=76;

ui->lineEdit->setText(QString::number(sum));

ui->label\_4->setText("$");

ui->lineEdit\_3->setText(QVariant(round(vklad\*100)/100).toString());

}

else

{

sum \*=76;

vklad \*=76;

ui->lineEdit->setText(QString::number(sum));

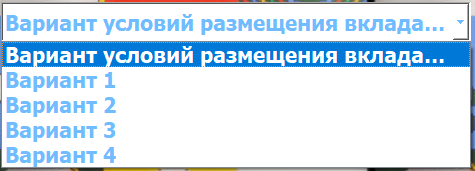
ui->label\_4->setText("₽");

ui->lineEdit\_3->setText(QVariant(round(vklad\*100)/100).toString());

}

}

* 1. Данное поле отвечает за выбор условий размещения вклада в банке. Реализовано с помощью QComboBox. Данное поле представляет собой выдвижной список, в котором можно выбрать определенный варианта размещения вклада.

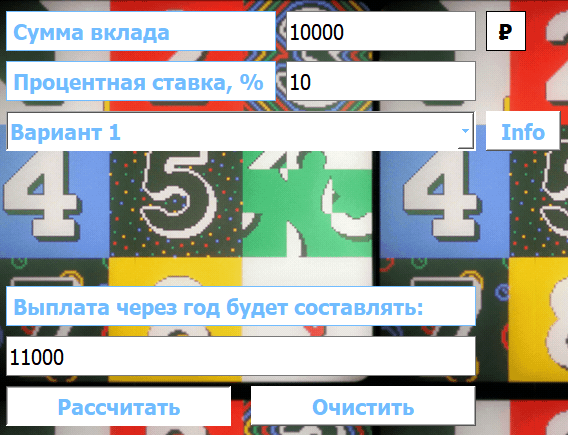


Сигналом для данного поля является выбор варианта из списка, каждый из которых имеет свой индекс.

ui->comboBox->currentIndex()

5. При нажатии на кнопку (“Рассчитать”) производится расчёт выплаты в зависимости от выбранного варианта размещения вклада и данное значение выводится в соответствующий QLineEdit. Сигналом для данной операции является нажатие по иконке кнопки.





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

{

switch (ui->comboBox->currentIndex())

{

case 1: {

vklad=sum\*(1+percent/100); break;

}

case 2: {

vklad=sum\*(pow((1+percent/100), 12)); break;

}

case 3: {

if (sum < 1000) ui->lineEdit\_3->setText("Ошибка ввода!");

else

{

vklad=sum;

for(int i=0; i<365; i++)

{

vklad+=int(vklad/1000)\*1000\*(percent/100);

}

}

break;

}

case 4: {

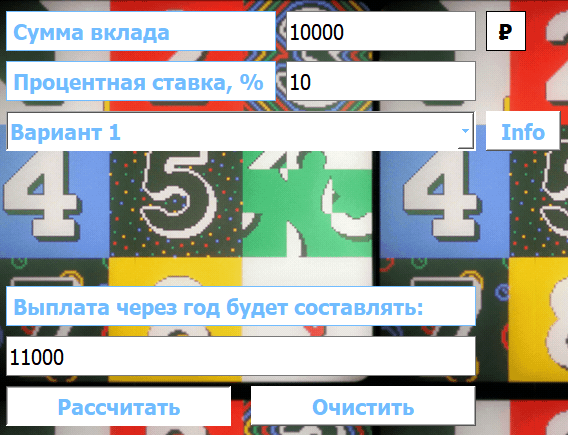
vklad=sum\*(pow((1+percent/100),365)); break;

}

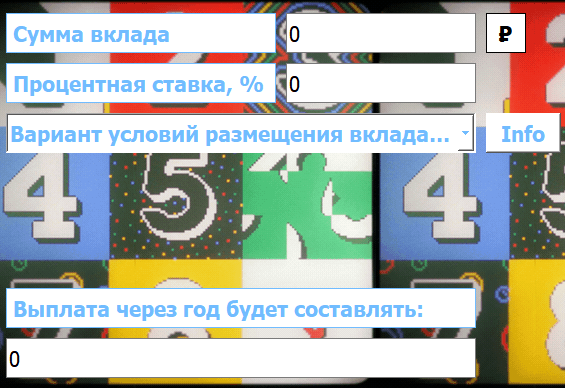
}

ui->lineEdit\_3->setText(QVariant(round(vklad\*100)/100).toString());

}

6. При нажатии на кнопку (“Очистить”) производится обнуление всех переменных, а также вывод всех дефолтных значений в окно калькулятора соответствующий поля. Сигналом для данной операции является нажатие по иконке кнопки.





void MainWindow::**on\_pushButton\_3\_clicked**()

{

sum=0;

ui->lineEdit->setText(QString::number(sum));

percent=0;

ui->lineEdit\_2->setText(QString::number(percent));

vklad=0;

ui->lineEdit\_3->setText(QString::number(vklad));

ui->comboBox->setCurrentIndex(0);

}

7. При нажатии кнопки (“Info”) появится дополнительное окно с информацией о условиях размещения вклада в банке. Сигналом для данной операции является нажатие по иконке кнопки.

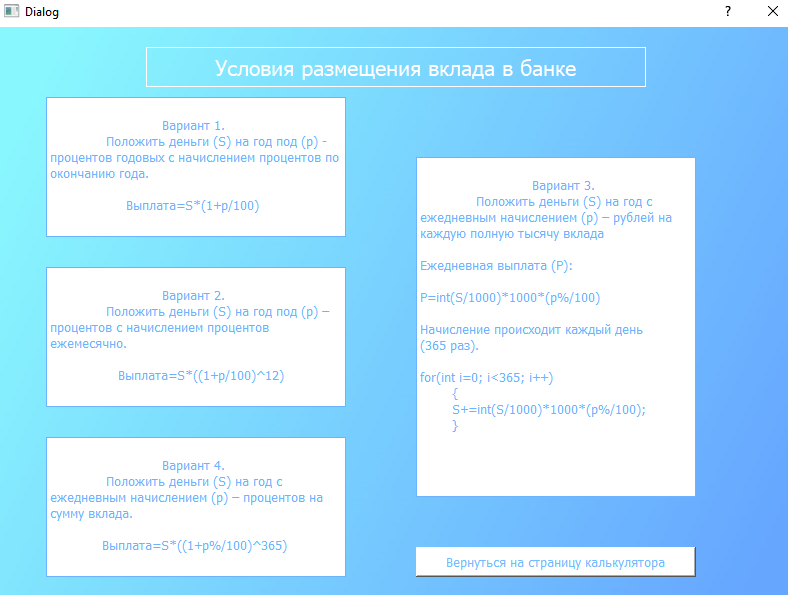
void MainWindow::**on\_pushButton\_4\_clicked**()

{

infoWindow ->show();

this->close();

}

3) var (Окно с дополнительной информацией)

В данном окне описаны все возможные условия размещения вклада в банке.

* + 1. При нажатии на кнопку (“Вернуться на страницу калькулятора”) появиться основное окно калькулятора. Сигналом для данной операции является нажатие по иконке кнопки.

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

{

this->close();

emit firstWindow();

}

**2.** В ходе работы были использованы следующие типы данных:

**2.1.** Глобальные переменные типа double для упрощённого применения в функциях.

extern double percent; - процентная ставка

extern double sum; - сумма вклада

extern double vklad; - сумма выплат

**3.** Для решения задачи данные были представлены в следующем виде:

**3.1.** Ввод данных производится с помощью полей QLineEdit:

При изменении значения в полях ввода QLineEdit, оно передается в переменные с помощью сигнала:

On\_lineEdit\_textChaned;

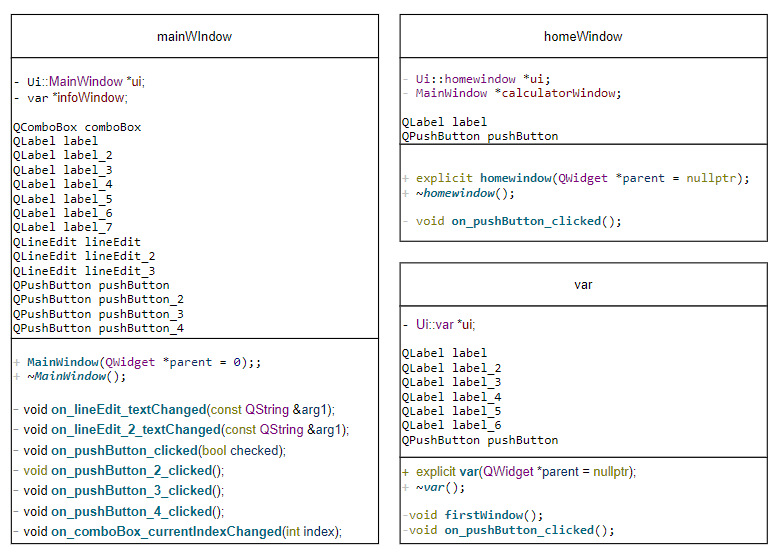
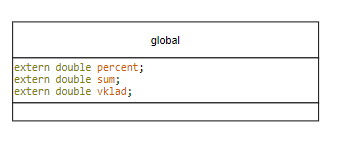
**3.2.** Вывод данных производится с помощью функции:

ui->lineEdit->setText(Qstring);

**3.3.** Преобразование типа QString к типу double производится с помощью функции:

QString.toDouble;

**UML-диаграмма**



**Код программы**

global.h

#ifndef GLOBAL\_H

#define GLOBAL\_H

extern double percent;

extern double sum;

extern double vklad;

class **global**

{

public:

**global**();

};

#endif // GLOBAL\_H

Global.cpp

#include "global.h"

double percent=0;

double sum=0;

double vklad;

global::**global**()

{

}

Mainwindow.h

#ifndef MAINWINDOW\_H

#define MAINWINDOW\_H

#include <QMainWindow>

#include <var.h>

QT\_BEGIN\_NAMESPACE

namespace Ui { class MainWindow; }

QT\_END\_NAMESPACE

class MainWindow : public QMainWindow

{

Q\_OBJECT

public:

MainWindow(QWidget \*parent = 0);

~MainWindow();

private slots:

void on\_lineEdit\_textChanged(const QString &arg1);

void on\_lineEdit\_2\_textChanged(const QString &arg1);

void on\_pushButton\_clicked(bool checked);

void on\_pushButton\_2\_clicked();

void on\_pushButton\_3\_clicked();

void on\_pushButton\_4\_clicked();

void on\_comboBox\_currentIndexChanged(int index);

private:

Ui::MainWindow \*ui;

var \*infoWindow;

};

#endif // MAINWINDOW\_H

Mainwindow.cpp

#include "mainwindow.h"

#include "ui\_mainwindow.h"

#include "global.h"

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

{

ui->setupUi(this);

infoWindow = new var();

connect(infoWindow, &var::firstWindow,this, &MainWindow::show);

}

MainWindow::~MainWindow()

{

delete ui;

}

void MainWindow::on\_lineEdit\_2\_textChanged(const QString &arg1)

{

QString perc= arg1;

percent=perc.toDouble();

if ((percent <=0)||(percent>100))

ui->lineEdit\_3->setText("Ошибка ввода!");

else ui->lineEdit\_3->setText("0");

}

void MainWindow::on\_lineEdit\_textChanged(const QString &arg1)

{

QString summa= arg1;

sum = summa.toDouble();

if ((sum <= 0))

ui->lineEdit\_3->setText("Ошибка ввода!");

else if ((sum < 1000) && (ui->comboBox->currentIndex()==3))

ui->lineEdit\_3->setText("Ошибка ввода!");

else ui->lineEdit\_3->setText("0");

}

bool check=false;

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

{

check=(!check);

if (check)

{

sum /=76;

vklad /=76;

ui->lineEdit->setText(QString::number(sum));

ui->label\_4->setText("$");

ui->lineEdit\_3->setText(QVariant(round(vklad\*100)/100).toString());

}

else

{

sum \*=76;

vklad \*=76;

ui->lineEdit->setText(QString::number(sum));

ui->label\_4->setText("₽");

ui->lineEdit\_3->setText(QVariant(round(vklad\*100)/100).toString());

}

}

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

{

switch (ui->comboBox->currentIndex())

{

case 1: {

vklad=sum\*(1+percent/100); break;

}

case 2: {

vklad=sum\*(pow((1+percent/100), 12)); break;

}

case 3: {

if (sum < 1000) ui->lineEdit\_3->setText("Ошибка ввода!");

else

{

vklad=sum;

for(int i=0; i<365; i++)

{

vklad+=int(vklad/1000)\*1000\*(percent/100);

}

}

break;

}

case 4: {

vklad=sum\*(pow((1+percent/100),365)); break;

}

}

ui->lineEdit\_3->setText(QVariant(round(vklad\*100)/100).toString());

}

void MainWindow::on\_pushButton\_3\_clicked()

{

sum=0;

ui->lineEdit->setText(QString::number(sum));

percent=0;

ui->lineEdit\_2->setText(QString::number(percent));

vklad=0;

ui->lineEdit\_3->setText(QString::number(vklad));

ui->comboBox->setCurrentIndex(0);

}

void MainWindow::on\_pushButton\_4\_clicked()

{

infoWindow ->show();

this->close();

}

void MainWindow::on\_comboBox\_currentIndexChanged(int index)

{

if ((sum < 1000) && (index==3))

ui->lineEdit\_3->setText("Ошибка ввода!");

else ui->lineEdit\_3->setText("0");

}

Homewindow.h

#ifndef HOMEWINDOW\_H

#define HOMEWINDOW\_H

#include <QDialog>

#include <mainwindow.h>

namespace Ui {

class homewindow;

}

class homewindow : public QDialog

{

Q\_OBJECT

public:

explicit homewindow(QWidget \*parent = nullptr);

~homewindow();

private slots:

void on\_pushButton\_clicked();

private:

Ui::homewindow \*ui;

MainWindow \*calculatorWindow;

};

#endif // HOMEWINDOW\_H

Homewindow.cpp

#include "homewindow.h"

#include "ui\_homewindow.h"

homewindow::homewindow(QWidget \*parent) :

QDialog(parent),

ui(new Ui::homewindow)

{

ui->setupUi(this);

calculatorWindow = new MainWindow();

}

homewindow::~homewindow()

{

delete ui;

}

void homewindow::on\_pushButton\_clicked()

{

calculatorWindow->show();

this->close();

}

Var.h

#ifndef VAR\_H

#define VAR\_H

#include <QDialog>

namespace Ui {

class var;

}

class var : public QDialog

{

Q\_OBJECT

public:

explicit var(QWidget \*parent = nullptr);

~var();

signals:

void firstWindow();

private slots:

void on\_pushButton\_clicked();

private:

Ui::var \*ui;

};

#endif // VAR\_H

Var.cpp

#include "var.h"

#include "ui\_var.h"

var::var(QWidget \*parent) :

QDialog(parent),

ui(new Ui::var)

{

ui->setupUi(this);

}

var::~var()

{

delete ui;

}

void var::on\_pushButton\_clicked()

{

this->close();

emit firstWindow();

}

Main.cpp

#include "mainwindow.h"

#include <QApplication>

#include <homewindow.h>

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

{

QApplication a(argc, argv);

homewindow w;

w.show();

return a.exec();

}