09.02.07 Информационные системы и программирование

Проверил:

\_\_\_\_\_\_\_\_\_\_\_ Антонов М.Б.

Дата:”\_\_\_” \_\_\_\_\_\_\_20\_\_г.

Оценка: \_\_\_\_\_

Выполнил:

студент группы Y2331

\_\_\_\_\_\_\_\_\_ Кирпиченко Д.А.

Практическая работа №7

Тема: «Работа с базой данных»

по дисциплине: «Разработка программных модулей»

Санкт-Петербург 2019

УНИВЕРСИТЕТ ИТМО

Факультет среднего профессионального образования

1. Постановка задачи

Целью работы является познакомиться с основами работы с базами данных в приложениях Qt/

1. Диаграмма классов

На рисунке 1 представлена диаграмма классов:

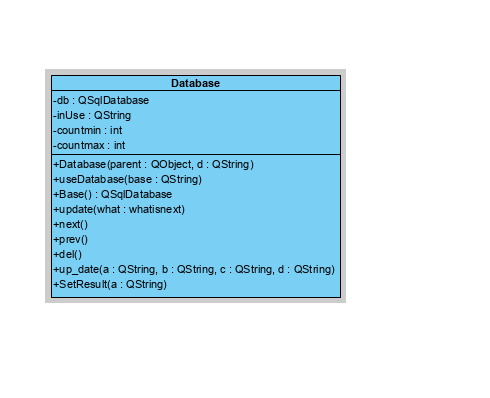


Рисунок 1 – Диаграмма классов

1. код

/\*\*

/\*\*

\* @file database.h

\* @author Kirpichenko Daniil <ko1lmain4@gmail.com>

\*/

#ifndef DATABASE\_H

#define DATABASE\_H

#include <QObject>

#include <QSqlDatabase>

#include <QSqlQuery>

#include <QDebug>

#include <QString>

enum **whatisnext**{

*next*,*prev*,*now*

};

class **DataBase** : public QObject

{

Q\_OBJECT

public:

explicit **DataBase**(QObject \*parent = nullptr,const QString& d = "QMYSQL");

public slots:

QSqlDatabase **Base**();

//Смена таблицы

void **useTable**(const QString&);

//Обновление

void **update**(const whatisnext&);

//Вставка

void **insert**(const QString&,const QString&);

//Next

void **next**();

//Prev

void **prev**();

//Del

void **del**();

private:

QSqlDatabase\* db;

QString inUse;

int countmin = 1,countmax;

signals:

void **up\_date**(QString,QString,QString,QString);

void **SetResult**(QString);

};

#endif // DATABASE\_H

/\*\*

\* @file mainwindow.h

\* @author Kirpichenko Daniil <ko1lmain4@gmail.com>

\*/

#ifndef MAINWINDOW\_H

#define MAINWINDOW\_H

#include <QMainWindow>

QT\_BEGIN\_NAMESPACE

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

QT\_END\_NAMESPACE

class **MainWindow** : public QMainWindow

{

Q\_OBJECT

public:

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

~***MainWindow***();

private:

Ui::MainWindow \*ui;

};

#endif // MAINWINDOW\_H

/\*\*

\* @file database.cpp

\* @author Kirpichenko Daniil <ko1lmain4@gmail.com>

\*/

#include "database.h"

DataBase::**DataBase**(QObject \*parent,const QString& name) : QObject(parent)

{

db = new QSqlDatabase();

\*db = QSqlDatabase::addDatabase(name);

db->setDatabaseName("/Qt/base.sqlite");

db->open();

}

QSqlDatabase DataBase::**Base**()

{

return \*db;

}

void DataBase::**useTable**(const QString& table)

{

QSqlQuery query;

inUse = table;

query.exec(" CREATE TABLE '"+table+"' ('Name' TEXT,'LastName' TEXT ,'id' INTEGER PRIMARY KEY AUTOINCREMENT);");

update(whatisnext::*now*);

emit SetResult(table);

}

void DataBase::**update**(const whatisnext& value)

{

switch (value) {

case 0:{

//Next

if(countmin != countmax){

countmin++;

QSqlQuery count;

qInfo() << count.exec("select Name,LastName from " + inUse +" where id = " + QString::number(countmin));

if (count.next()){

emit up\_date(count.value(0).toString(),count.value(1).toString(),QString::number(countmin),QString::number(countmax));

};

}

break;};

case 1:{

//Prev

if(countmin != 1){

countmin--;

QSqlQuery count;

qInfo() << count.exec("select Name,LastName from " + inUse +" where id = " + QString::number(countmin));

if (count.next()){

emit up\_date(count.value(0).toString(),count.value(1).toString(),QString::number(countmin),QString::number(countmax));

};

}

break;};

case 2:{

//Update

QSqlQuery count;

qInfo() << count.exec("select count(\*) from " + inUse);

count.next();

countmax = count.value(0).toInt();

qInfo() << count.exec("select Name,LastName from " + inUse +" where id = " + QString::number(countmin));

if (count.next()){

emit up\_date(count.value(0).toString(),count.value(1).toString(),QString::number(countmin),QString::number(countmax));

};

break;};

}

}

void DataBase::**insert**(const QString& first, const QString& second)

{

QSqlQuery insert("INSERT INTO "+inUse+"(Name,LastName) VALUES('"+first+"','"+second+"');");

update(whatisnext::*now*);

}

void DataBase::**next**()

{

update(whatisnext::*next*);

}

void DataBase::**prev**()

{

update(whatisnext::*prev*);

}

void DataBase::**del**(){

QSqlQuery del;

qInfo() << del.exec("delete from " + inUse + " where id = " + QString::number(countmin));

if(countmin != 1){

countmin--;

}

update(whatisnext::*now*);

}

/\*\*

\* @file main.cpp

\* @author Kirpichenko Daniil <ko1lmain4@gmail.com>

\*/

#include "mainwindow.h"

#include <QApplication>

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

{

QApplication a(*argc*, argv);

MainWindow w;

w.show();

return a.exec();

}

/\*\*

\* @file mainwindow.cpp

\* @author Kirpichenko Daniil <ko1lmain4@gmail.com>

\*/

#include "mainwindow.h"

#include "ui\_mainwindow.h"

#include "database.h"

#include <QInputDialog>

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

: QMainWindow(parent)

, ui(new Ui::MainWindow)

{

ui->setupUi(this);

ui->tablename->setReadOnly(true);

ui->infotext->setReadOnly(true);

DataBase\* bd = new DataBase(this,"QSQLITE");

connect(ui->tablebtm,&QPushButton::clicked,[bd,this](){ //Задать вопрос

bool connect;

const QString Name = QInputDialog::getText(this,

"Table Name",

"Name:",

QLineEdit::*Normal*,

"",

&connect,

Qt::*CustomizeWindowHint* | Qt::*WindowStaysOnTopHint* | Qt::*WindowMinimizeButtonHint*,

Qt::InputMethodHint::*ImhLowercaseOnly*);

if(connect == true) {

bd->useTable(Name);

}

});

connect(bd,&DataBase::SetResult,ui->tablename,&QLineEdit::setText);

connect(ui->addbtm,&QPushButton::pressed,[=](){

bd->insert(ui->namevalue->text(),ui->lastnamevalue->text());

});

connect(bd,&DataBase::up\_date,[&](QString first,QString second,QString min,QString max){

ui->namevalue->setText(first);

ui->lastnamevalue->setText(second);

ui->infotext->setText(min+"/"+max);

});

connect(ui->nextbtm,&QPushButton::clicked,bd,&DataBase::next);

connect(ui->prevbtm,&QPushButton::clicked,bd,&DataBase::prev);

connect(ui->deletebtm,&QPushButton::clicked,bd,&DataBase::del);

}

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

{

delete ui;

}

1. листинг

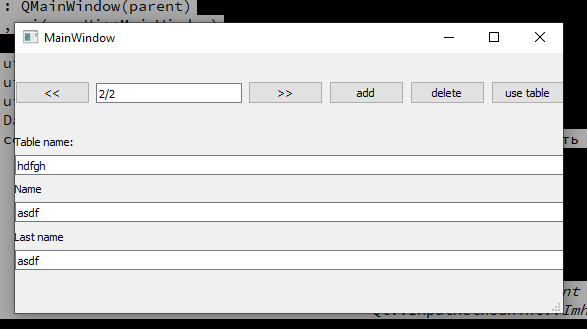


Рисунок 2 – листинг программы.

1. Вывод

В ходе проделанной работы была успешно разработана программа, которая удовлетворяет поставленным задачам.