|  |  |
| --- | --- |
| **Gerb-BMSTU_01** | **Министерство науки и высшего образования Российской Федерации**  Калужский филиал федерального государственного бюджетного  образовательного учреждения высшего образования  ***«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»***  ***(КФ МГТУ им. Н.Э. Баумана)*** |

**ФАКУЛЬТЕТ** **Информатика и управление**

**КАФЕДРА** **Информационные системы и сети**

##### 

**ЛАБОРАТОРНАЯ РАБОТА №4**

«**Работа с элементом CListCtrl**»

|  |  |
| --- | --- |
| **ДИСЦИПЛИНА:** | "Программирование в среде Windows" |

|  |  |  |
| --- | --- | --- |
| Выполнил: студент гр. ИУК2-51.Б: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | (Мелкумян Д.Т.) |
|  | (Подпись) | (Ф.И.О.) |
|  |  |  |
| Проверил: | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | (Крысин И. А.) |
|  | (Подпись) | (Ф.И.О.) |
|  |  |  |

Дата сдачи (защиты): \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Результаты сдачи (защиты):

-Бальная оценка: \_\_\_\_

-Оценка: \_\_\_\_\_

Калуга, 2021 г.

**Цель лабораторной работы:** формирование практических навыков работы с основными функциями компонента CListCtrl. Освоить приёмы работы с данным элементом.

**Задачи:** Разработать программу с использованием элемента List Control формирования списка студентов. Программа должна обеспечивать удаление, изменение и сортировку списка. Необходимой информацией о студентах являются следующие поля: Фамилия, Имя, отчество, год рождения, пол (Radio Button) и гражданство (Check Box) в диалоге CDialogInput. Данный диалог переделать самостоятельно. При сортировке указатель текущего элемента должен перемещаться вместе с элементом.

**Теоретическая часть**

Класс CListCtrl производный от класса CWnd:

class CListCtrl : public CWnd

Класс CListCtrl - элемент "список управления" отображает коллекцию элемен-тов каждый состоящий из значков (из списка изображений) и метки. В дополнение к значкам и меткам каждый элемент может иметь информацию, отображаемую в колонках справа от значка и меток.

Элементы управления представления списка может отображать их содержимое четырьмя различными способами:

 Вид значков. Каждый элемент отображается в виде полноразмерного значка (32 x 32 пикселей) с подписью под ним. Пользователь может перетаскивать элементы в любое место в окне представления списка.

 Вид мелких значков. Каждый элемент отображается как мелкого значка (16 x 16 точек) с подписью справа от него. Пользователь может перетаскивать элементы в любое место в окне представления списка.

 Представление списка. Каждый элемент отображается в виде небольшого значка с меткой справа от него. Элементы расположены в столбцах и нельзя перетащить в любое место в окне представления списка.

 Представление отчета. Каждый элемент отображается в отдельной строке с дополнительной информацией, в столбцах справа. Крайний левый столбец содержит небольшой значок и метку, а последующие столбцы содержат подэлементов как указанным приложением. Встроенный заголовок элемента управления (класс CHeaderCtrl) реализует эти столбцы.

**Исходный код**

**Mainwindow**

#include "mainwindow.h"

#include "ui\_mainwindow.h"

#include "secondwindow.h"

#include <QMessageBox>

#include "QStandardItemModel"

#include "QStandardItem"

#include <QApplication>

QStandardItemModel \*model = new QStandardItemModel;

QStandardItem \*item;

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

: QMainWindow(*parent*)

, ui(new Ui::MainWindow)

{

ui->setupUi(this);

QStringList horizontalHeader;

horizontalHeader.append("Фамилия");

horizontalHeader.append("Имя");

horizontalHeader.append("Отчество");

horizontalHeader.append("Год рождения");

horizontalHeader.append("Пол");

horizontalHeader.append("Гражданство РФ");

model->setHorizontalHeaderLabels(horizontalHeader);

ui->tableView->*setModel*(*model*);

ui->tableView->verticalHeader()->hide();

ui->tableView->resizeRowsToContents();

ui->tableView->resizeColumnsToContents();

//Выделение строки ui->tableView->selectRow(0);

}

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

{

delete ui;

}

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

{

secondwindow window;

window.setModal(true);

if (window.*exec*())

{

int i = ui->tableView->model()->*rowCount*();

item = new QStandardItem(QString(window.value()[0]));

model->setItem(i, 0, *item*);

item = new QStandardItem(QString(window.value()[1]));

model->setItem(i, 1, *item*);

item = new QStandardItem(QString(window.value()[2]));

model->setItem(i, 2, *item*);

item = new QStandardItem(QString(window.value()[3]));

model->setItem(i, 3, *item*);

item = new QStandardItem(QString(window.value()[4]));

model->setItem(i, 4, *item*);

item = new QStandardItem(QString(window.value()[5]));

model->setItem(i, 5, *item*);

ui->tableView->*setModel*(*model*);

}

}

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

{

secondwindow window;

window.setModal(true);

QString familia = ui->tableView->model()->*data*(ui->tableView->model()->*index*(ui->tableView->currentIndex().row(),0)).toString();

QString name = ui->tableView->model()->*data*(ui->tableView->model()->*index*(ui->tableView->currentIndex().row(),1)).toString();

QString otch = ui->tableView->model()->*data*(ui->tableView->model()->*index*(ui->tableView->currentIndex().row(),2)).toString();

QString gr = ui->tableView->model()->*data*(ui->tableView->model()->*index*(ui->tableView->currentIndex().row(),3)).toString();

bool pol;

bool graj;

if (ui->tableView->model()->*data*(ui->tableView->model()->*index*(ui->tableView->currentIndex().row(),4)).toString()=="М"){

pol=true;

}

else{

pol=false;

}

if (ui->tableView->model()->*data*(ui->tableView->model()->*index*(ui->tableView->currentIndex().row(),5)).toString()=="Да"){

graj = true;

}

else{

graj = false;

}

window.setchanges(familia, name, otch, gr, pol, graj);

if (window.*exec*())

{

int i = ui->tableView->currentIndex().row();

item = new QStandardItem(QString(window.value()[0]));

model->setItem(i, 0, *item*);

item = new QStandardItem(QString(window.value()[1]));

model->setItem(i, 1, *item*);

item = new QStandardItem(QString(window.value()[2]));

model->setItem(i, 2, *item*);

item = new QStandardItem(QString(window.value()[3]));

model->setItem(i, 3, *item*);

item = new QStandardItem(QString(window.value()[4]));

model->setItem(i, 4, *item*);

item = new QStandardItem(QString(window.value()[5]));

model->setItem(i, 5, *item*);

ui->tableView->*setModel*(*model*);

}

}

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

{

ui->tableView->model()->removeRow(ui->tableView->currentIndex().row());

}

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

{

secondwindow window;

window.setModal(true);

if (window.*exec*())

{

int i = ui->tableView->currentIndex().row()+1;

item = new QStandardItem(QString(window.value()[0]));

model->insertRow(i, *item*);

item = new QStandardItem(QString(window.value()[1]));

model->setItem(i, 1, *item*);

item = new QStandardItem(QString(window.value()[2]));

model->setItem(i, 2, *item*);

item = new QStandardItem(QString(window.value()[3]));

model->setItem(i, 3, *item*);

item = new QStandardItem(QString(window.value()[4]));

model->setItem(i, 4, *item*);

item = new QStandardItem(QString(window.value()[5]));

model->setItem(i, 5, *item*);

ui->tableView->*setModel*(*model*);

}

}

void MainWindow::**on\_pushButton\_5\_clicked**()

{

QApplication::quit();

}

**Secondwindow**

#include "secondwindow.h"

#include "ui\_secondwindow.h"

secondwindow::secondwindow(QWidget \*parent) :

QDialog(parent),

ui(new Ui::secondwindow)

{

ui->setupUi(this);

}

secondwindow::~secondwindow()

{

delete ui;

}

QString\* secondwindow::value()

{

QString\* addtext = new QString[6];

addtext[0] = ui->lineEdit->text();

addtext[1] = ui->lineEdit\_2->text();

addtext[2] = ui->lineEdit\_3->text();

addtext[3] = ui->lineEdit\_4->text();

if (ui->radioButton->isChecked()){

addtext[4] = "М";

}

else if (ui->radioButton\_2->isChecked()){

addtext[4] = "Ж";

}

else{

addtext[4] = "Неив.";

}

if (ui->checkBox->isChecked()){

addtext[5] = "Да";

}

else{

addtext[5] = "Нет";

}

return addtext; //текст передаваемый в форму

}

void secondwindow::setchanges(QString familia, QString name, QString otch, QString gr, bool pol, bool grajd)

{

ui->lineEdit->setText(familia);

ui->lineEdit\_2->setText(name);

ui->lineEdit\_3->setText(otch);

ui->lineEdit\_4->setText(gr);

if (pol)

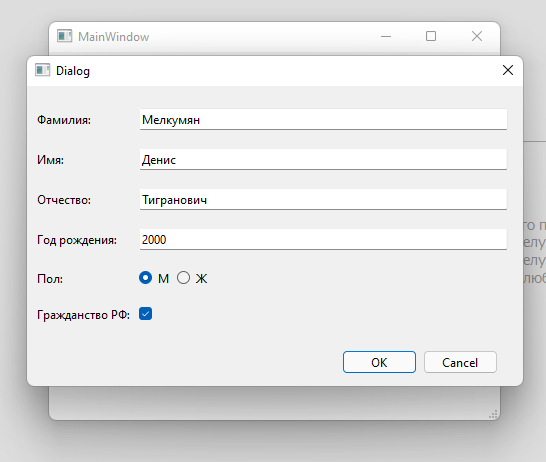
ui->radioButton->setChecked(true);

else

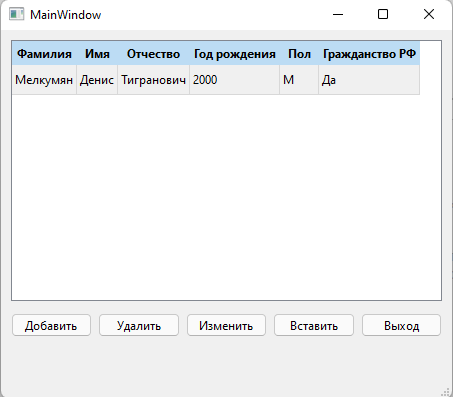
ui->radioButton\_2->setChecked(true);

if (grajd) ui->checkBox->setChecked(true);

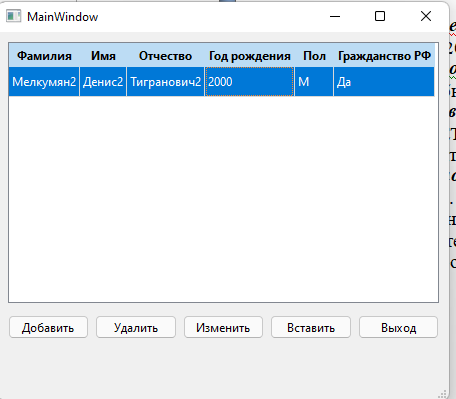
}

****

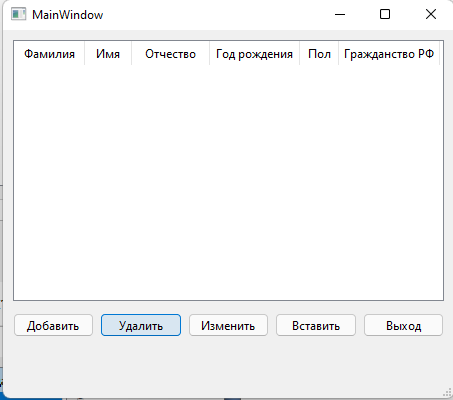
**Рис 1. Окно добавление элемента**

****

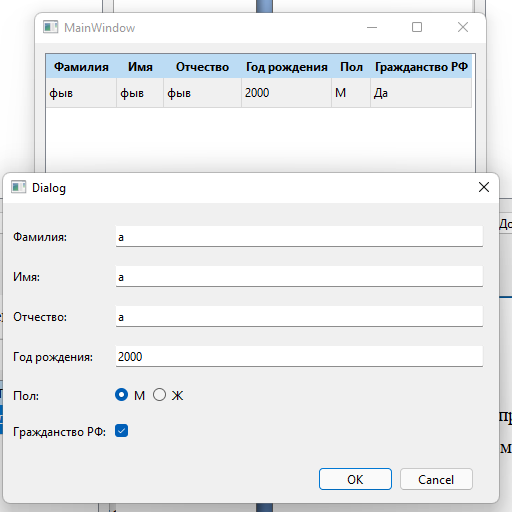
**Рис 2. Результат добавления элемента**

****

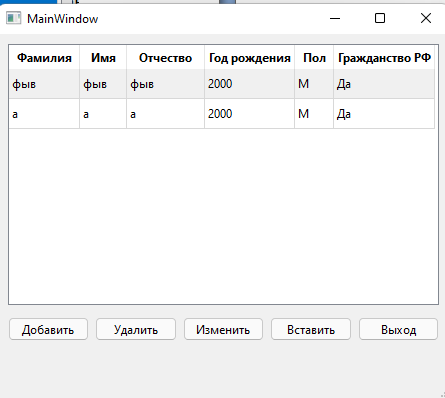
**Рис 3. Результат изменения элемента**

****

**Рис 4. Результат удаления элемента**

****

**Рис 5. Окно вставки элемента**

****

**Рис 6. Результат вставки элемента**

**Вывод**: были приобретены практические навыки работы с основными функциями компонента CListCtrl. Освоены приёмы работы с данным элементом.

**СПИСОК ЛИТЕРАТУРЫ**

1. ***Подбельский В.В***. Язык СИ++: учебное пособие. М.: Финансы и статистика, 2008.

2. ***Павловская Т.А.*** С/C++ Программирование на языке высокого уровня: учебник. СПб.:Питер, 2010.

3. ***Перова В. И.*** Программирование на C++ в среде VISUAL STUDIO.NET. Нижний Новгород: Издательство Нижегородского госуниверситета, 2010., http://elibrary.ru/item.asp?id=19597408

4. ***Крылов Е.В.*** Техника разработки программ: учебник. М.: Высшая школа, 2008.

5. Научная электронная библиотека: http://eLIBRARY.RU.

6. Издательство «Лань». Электронно-библиотечная система. http://e.lanbook.com.