



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 №3

«Работа с элементами управления List Box и Combo Box»

ДИСЦИПЛИНА: "Программирование в среде Windows"

Выполнил: студент гр. ИУК2-51.Б:

(Подпись)

(Мелкумян Д.Т.)
(Ф.И.О.)

Проверил:

(Подпись)

(Крысин И. А.)
(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 компонентов CListBox и CComboBox. Освоить приёмы работы с данными элементами.

Задачи: Разработать программу с использованием элементов ListBox и ComboBox. Осуществить заполнение элементов цветами радуги. Программа должна обеспечивать удаление, изменение и добавление элементов.

Теоретическая часть

Класс CListBox

Класс *CListBox* производный от класса *CWnd*:

class *CListBox* : public *CWnd*

Элемент управления **List Box** выводит на экран список элементов, которые пользователь может просмотреть и выбрать.

В поле простого списка пользователь может выбрать только один элемент. В поле списка с множественным выбором может быть выбран диапазон элементов. Когда пользователь выбирает элемент, он выделен, и поле списка отправляет уведомление в родительское окно.

Возможно, создание поле списка или из диалогового шаблона или непосредственно в коде программы. Чтобы создать его непосредственно необходимо создать объект *CListBox*, затем вызвать *Create* функцию класса, чтобы создать управление поля списка Windows и присоединить его к объекту *CListBox*. Чтобы использовать поле списка в диалоговом шаблоне, необходимо объявить переменную поля списка в своем классе диалогового окна, затем используйте *DDX_Control* в классе диалогового окна функцию *DoDataExchange*, чтобы соединить задействованную переменную с управлением.

Класс CComboBox

Класс *CComboBox* производный от класса *CWnd*:

class *CComboBox* : public *CWnd*

Поле Combo Box состоит из поля списка, объединенного или со статическим элементом управления или со средствами редактирования. Часть поля списка управления может быть выведена на экран в любом случае, или может только выпадающий, когда пользователь выбирает стрелку выпадающего списка рядом с управлением.

Выбранный пункт (если таковые имеются) в поле списка выведен на экран в статическом элементе управления или средствах редактирования. Кроме того, если у поля комбинированного списка есть стиль выпадающего списка, пользователь может ввести начальный символ одного из элементов в списке, и поле списка, если видимый, выделит следующий элемент с тем начальным символом.

Исходный код

```
#include "mainwindow.h"
#include "ui_mainwindow.h"
#include <QStandardItemModel>

MainWindow::MainWindow(QWidget *parent)
    : QMainWindow(parent)
    , ui(new Ui::MainWindow)
{
    ui->setupUi(this);
    ui->temp->setText(ui->comboBox->currentText());

    for (int i=0; i<ui->comboBox->count(); i++) {
        ui->listWidget->addItem(QString(ui->comboBox->itemText(i)));
    }
    ui->listWidget->setCurrentItem(ui->listWidget->item(0));
}

MainWindow::~MainWindow()
{
    delete ui;
}

void MainWindow::on_comboBox_currentIndexChanged(int index)
{
    if (ui->listWidget->currentRow() >= 0) {
        ui->temp->setText(ui->comboBox->currentText());
        ui->listWidget->setCurrentRow(index);
    }
}

void MainWindow::on_listWidget_currentRowChanged(int currentRow)
{
    if (ui->listWidget->currentRow() >= 0) {
        ui->comboBox->setCurrentIndex(currentRow);
        ui->temp->setText(ui->comboBox->currentText());
    }
}

void MainWindow::on_pushButton_clicked()
{
    int index = ui->listWidget->currentRow();
    ui->listWidget->takeItem(index);
    ui->comboBox->removeItem(index);
    if (ui->listWidget->currentRow() < 0) ui->temp->setText("");
}

void MainWindow::on_pushButton_2_clicked()
{
    ui->listWidget->addItem(ui->lineEdit->text());
    ui->comboBox->addItem(ui->lineEdit->text());
    ui->lineEdit->setText("");
}
```

```

void MainWindow::on_pushButton_4_clicked()
{
    ui->listWidget->item(ui->listWidget->currentRow())->setText(ui->lineEdit-
>text());
    ui->comboBox->setItemText(ui->comboBox->currentIndex(), ui->lineEdit-
>text());
    ui->lineEdit->setText("");
}

void MainWindow::on_pushButton_5_clicked()
{
    QApplication::quit();
}

void MainWindow::on_pushButton_3_clicked()
{
    QListWidgetItem *item = new QListWidgetItem;
    item->setText(ui->lineEdit->text());
    if (ui->listWidget->currentRow() < ui->listWidget->count() - 1)
        ui->listWidget->insertItem(ui->listWidget->currentRow() + 1, item);
    else
        ui->listWidget->insertItem(ui->listWidget->count(), item);
    ui->comboBox->insertItem(ui->comboBox->currentIndex() + 1, ui->lineEdit-
>text());
    ui->lineEdit->setText("");
}

```

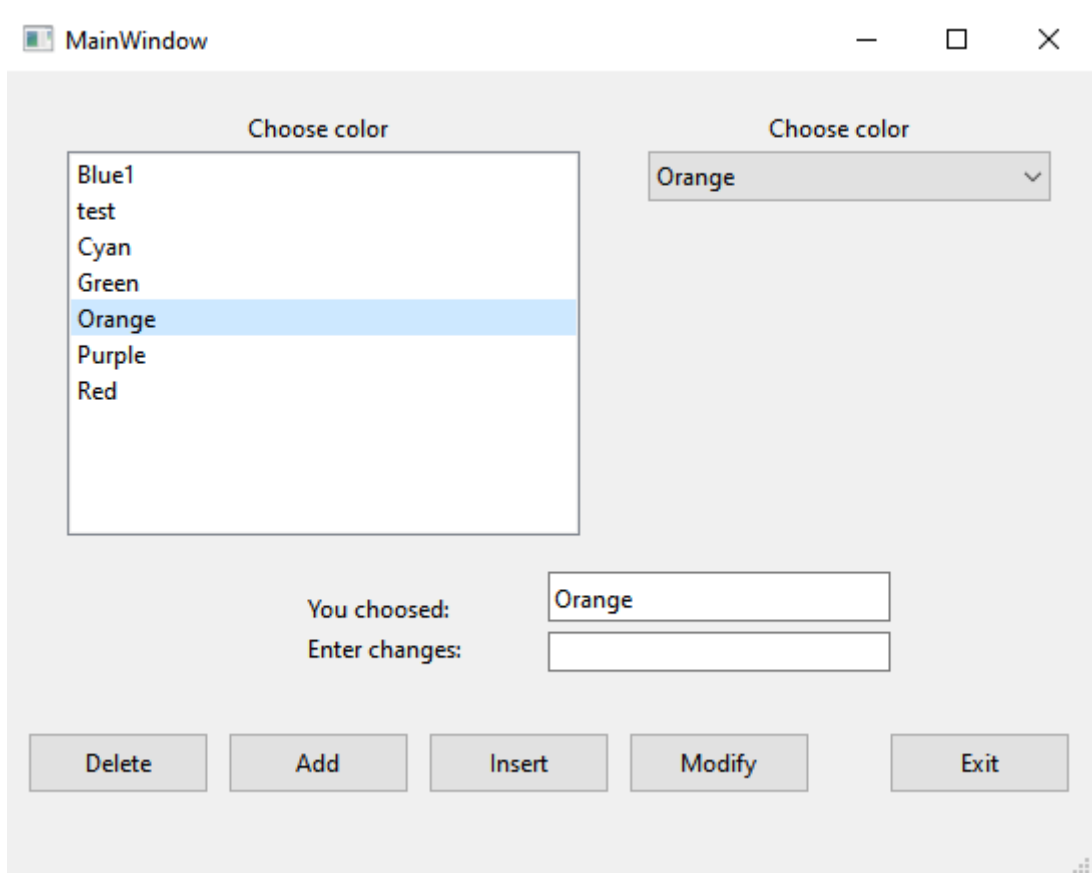


Рис.1-Результат работы команды

Вывод: были приобретены практические навыки работы с основными функциями компонентов CListBox и CComboBox. Освоены приёмы работы с данными элементами.

СПИСОК ЛИТЕРАТУРЫ

1. **Подбельский В.В.** Язык СИ++: учебное пособие. М.: Финансы и статистика, 2008.
2. **Павловская Т.А.** С/С++ Программирование на языке высокого уровня: учебник. СПб.:Питер, 2010.
3. **Перова В. И.** Программирование на С++ 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>.