#### Практическое занятие 2. Интерфейс приложения «Конвертор чисел»

#### *Цель: Сформировать практические навыки реализации графических интерфейсов пользователя (GUI) на основе библиотеки визуальных компонентов.*

#### Задание

1. Реализовать «Интерфейс» приложения «Конвертер р1\_р2», используя библиотечный класс формы и визуальные компоненты.
2. Протестировать методы класса.

Спецификация класса «Интерфейс».

Интерфейс приложения представлен на рисунке 1.

**ADT TPanel\_p\_p**

**Данные**

«Интерфейс» конвертера действительных чисел из системы счисления с основанием p1 в систему счисления с основанием p2 предназначен:

1. Выбора оснований систем счисления p1, p2 из диапазона от 2..16;
2. Ввода и редактирования действительного числа со знаком в системе счисления с выбранным основанием p1;
3. Отображения результата – представления ввёдённого числа в системе счисления с основанием p2;
4. Отображения справки о приложении;
5. Отображения истории текущего сеанса работы пользователя с приложением.

«Интерфейс» несёт на себе визуальные компоненты, реализующие выполнения команд преобразователя и объект «Управление» класса Control\_.

Операции. Операции представлены в таблице 1.

Таблица 1 - Операции

|  |  |
| --- | --- |
| **Наименование** | **Пояснение** |
| **trackBar1\_Scroll** | Обработчик события Scroll для компонента trackBar1. |
| Вход: | object sender, EventArgs e.  sender – указатель на объект, который явился инициатором события Scroll. e - это объект базового класса для классов, содержащих данные о событии. |
| Предусловия: | Пользователь перетаскивает бегунок компонента trackBar1. |
| Процесс: | Обновляет свойства визуальных компонентов формы, связанных с изменением основания системы счисления р1 исходного числа. Устанавливает новое значение основания системы счисления. Обновляет состояние командных кнопок. |
| Постусловия: | Обновления выполнены. |
| Выход: | Нет. |
| **numericUpDown1\_ValueChanged** | Обработчик события ValueChanged для компонента numericUpDown1. |
| Вход: | object sender, EventArgs e. sender – указатель на объект, который явился инициатором события ValueChanged. |
| Предусловия: | Пользователь изменяет р1 с помощью компонента numericUpDown1. |
| Процесс: | Обновляет свойства визуальных компонентов формы, связанных с изменением основания системы счисления р1 исходного числа. Устанавливает новое значение основания системы счисления. Обновляет состояние командных кнопок. |
| Постусловия: | Обновления выполнены. |
| Выход: | Нет. |
| **trackBar2\_Scroll** | Обработчик события Scroll для компонента trackBar2. |
| Вход: | object sender, EventArgs e. sender – указатель на объект, который явился инициатором события Scroll. |
| Предусловия: | Пользователь перетаскивает бегунок компонента trackBar2. |
| Процесс: | Обновляет свойства визуальных компонентов формы, связанных с изменением основания системы счисления p2 исходного числа. Устанавливает новое значение основания системы счисления. |
| Постусловия: | Обновления выполнены. |
| Выход: | Нет. |
| **numericUpDown2\_ValueChanged** | Обработчик события ValueChanged для компонента numericUpDown2. |
| Вход: | object sender, EventArgs e. sender – указатель на объект, который явился инициатором события ValueChanged. |
| Предусловия: | Пользователь изменяет р2 с помощью компонента numericUpDown2. |
| Процесс: | Обновляет свойства визуальных компонентов формы, связанных с изменением основания системы счисления р2 результата. Устанавливает новое значение основания системы счисления. Обновляет состояние командных кнопок. |
| Постусловия: | Обновления выполнены. |
| Выход: | Нет. |
| **numericUpDown1\_ValueChanged** | Обработчик события ValueChanged для компонента numericUpDown1. |
| Вход: | object sender, EventArgs e. sender – указатель на объект, который явился инициатором события ValueChanged. |
| Предусловия: | Пользователь изменяет р1 с помощью компонента numericUpDown1. |
| Процесс: | Обновляет свойства визуальных компонентов формы, связанных с изменением основания системы счисления р1 результата. Устанавливает новое значение основания системы счисления. Обновляет состояние командных кнопок. |
| Постусловия: | Обновления выполнены. |
| Выход: | Нет. |
| **TPanelp\_p\_Load** | Обработчик события Load для компонента TPanelp\_p. |
| Вход: | (object sender, EventArgs e). sender – указатель на объект, который явился инициатором события Load. |
| Предусловия: | Форма загружается в память. |
| Процесс: | Устанавливает начальные значения свойств визуальных компонентов формы после загрузки формы. |
| Выход: | Нет. |
| Постусловия: | Установка свойств выполнена. |
| **DoCmnd(int j)** | Выполнить команду. |
| Вход: | j значение целого типа – номер команды преобразователя. |
| Предусловия: | Пользователь нажал командную кнопку команды с номером j. |
| Процесс: | Передаёт сообщение объекту Управление и отображает возвращаемый им результат. Вызывается метод объекта Управление и передаётся номер набранной пользователем команды Конвертора. |
| Выход: | Нет. |
| Постусловия: | Обновляется состояние Интерфейса. |
| **button\_Click** | Обработчик события Click для командных кнопок. |
| Вход: | object sender, EventArgs e. sender: object – указатель на объект, который явился инициатором события Click. |
| Предусловия: | Пользователь нажал командную кнопку. |
| Процесс: | Извлекает из свойства Tag командной кнопки номер соответствующей ей команды. Вызывает метод DoCmnd Интерфейса и передаёт в него номер команды. |
| Выход: | Нет. |
| Постусловия: | Нет. |
| **TPanelp\_p\_KeyPress** | Обработчик события KeyPress для алфавитно-цифровых клавиш клавиатуры. |
| Вход: | object sender, KeyPressEventArgs e. |
| Предусловия: | Пользователь нажал алфавитно-цифровую клавишу клавиатуры. |
| Процесс: | Определяет по нажатой алфавитно-цифровой клавише номер соответствующей ей команды. Вызывает метод DoCmnd Интерфейса и передаёт в него номер команды. |
| Выход: | Нет. |
| Постусловия: | Команда пользователя вызвана. |
| **TPanelp\_p\_KeyDown** | Обработчик события KeyDown для клавиш управления клавиатуры. |
| Вход: | (object sender, KeyEventArgs e) |
| Предусловия: | Пользователь нажал клавишу управления клавиатуры. |
| Процесс: | Определяет по нажатой клавише управления номер соответствующей ей команды. Вызывает метод DoCmnd Интерфейса и передаёт в него номер команды. |
| Выход: | нет. |
| Постусловия: | Команда пользователя вызвана. |
| **UpdateP1** | Выполнить обновления связанные с изменением р1. |
| Вход: | Нет. |
| Предусловия: | Изменено основание с.сч. р1 исходного числа. |
| Процесс: | Выполняет необходимые обновления при смене ос. с. сч. р1. |
| Выход: | Нет. |
| Постусловия: | Состояние кнопок обновлено. |
| **UpdateP2** | Выполнить обновления связанные с изменением р2. |
| Вход: | Нет. |
| Предусловия: | Изменено основание с.сч. р2 результата. |
| Процесс: | Выполняет необходимые обновления при смене ос. с. сч. р2. |
| Выход: | Нет. |
| Постусловия: | Состояние кнопок обновлено. |
| **UpdateButtons** |  |
| Вход: | Нет. |
| Предусловия: | Изменено основание с.сч. р1 исходного числа. |
| Процесс: | Обновляет состояния командных кнопок предназначенных для ввода цифр выбранной системы счисления p1. |
| Выход: | Нет. |
| Постусловия: | Состояние кнопок обновлено. |
| **выходToolStripMenuItem\_Click** | Команда Выход основного меню класса TPanelp\_p формы. |
| Вход: | object sender, EventArgs e. |
| Предусловия: | Пользователь кликает мышью на пункте Выход основного меню формы. |
| Процесс: | Завершает работу приложения. |
| Выход: | Нет. |
| Постусловия: | Приложение завершено. |
| **справкаToolStripMenuItem\_Click** | Команда Справка основного меню класса TPanelp\_p формы. |
| Вход: | object sender, EventArgs e |
| Предусловия: | Пользователь кликает мышью на пункте Справка основного меню формы. |
| Процесс: | Показывает окно со справкой по приложению. |
| Выход: | Нет. |
| Постусловия: | Отображено окно справки. |
| **историяToolStripMenuItem\_Click** | Команда История основного меню класса TPanelp\_p формы. |
| Вход: | object sender, EventArgs e |
| Предусловия: | Пользователь кликает мышью на пункте История основного меню формы. |
| Процесс: | Открывает окно История. |
| Выход: | Нет. |
| Постусловия: | Окно История - открыто. |

**end TPanel\_p\_p**

Рекомендации к выполнению.

1. Интерфейс приложения будет состоять из трёх форм: основная форма класс TPanel\_p\_p, HistoryForm – форма для отображения истории, AboutBox – форма, информирующая о приложении. Все они наследники класса Form.
2. Для реализации интерфейса приложения разместите на форме компоненты, описанные в таблице 2.

Таблица 2 - Компоненты

|  |  |  |
| --- | --- | --- |
| **Имя компонента** | **Имя класса** | **Назначение** |
| label1 | Label | Исходное число. |
| label2 | Label | Результат. |
| label3 | Label | Подпись к исходному числу. |
| label4 | Label | Подпись к результату. |
| trackBar1 | TrackBar | Изменять основание с. сч. р1. |
| trackBar2 | TrackBar | Изменять основание с. сч. р2. |
| numericUpDown1 | NumericUpDown | Изменять основание с. сч. р1. |
| numericUpDown2 | NumericUpDown | Изменять основание с. сч. р2. |
| button1 – button10 | Button | Ввод цифр от 0 – 9. |
| button11 – button16 | Button | Ввод цифр от A – F. |
| button17 | Button | Разделитель целой и дробной частей (,). |
| button18 | Button | Забой крайнего правого символа (BS). |
| button19 | Button | Удалить исходное число (CL). |
| button20 | Button | Выполнить (Execute). |
| menuStrip1 | menuStrip1 | Основное меню главной формы. |
| выходToolStripMenuItem | ToolStripMenuItem | Завершение работы приложения. |
| историяToolStripMenuItem | ToolStripMenuItem | Просмотр журнала сеанса работы. |
| справкаToolStripMenuItem | ToolStripMenuItem | Справка по приложению. |

Для этого перейдите на вкладку Конструктор формы окна редактора и сделайте форму активной. Добавьте на форму из вкладки «Панели элементов» из раздела «Все формы Windows Forms» командные кнопки Button. Кнопки помещайте на форму снизу вверх слева направо, начиная с 0. Порядок добавления кнопок на форму должен совпадать с порядком их нумерации. Выделите все кнопку и в окне Свойства на закладке Свойства раскройте свойство Font (Шрифт) и установите требуемые вам свойства для шрифта. Затем в свойство Text (Текст) занесите соответствующие цифры. Затем в свойство Tag каждой кнопки занесите целое число соответствующее кнопке: кнопки для ввода цифр нумеруйте от 0 до 15; кнопка разделитель целой и дробной части – 16; удалить крайний символ слева (BS) - 17; очистить всё (CL) – 18; выполнить ( Enter) - 19. Измените размер кнопок (свойство Size), если это необходимо.

Добавьте на форму два компонента TrackBar из вкладки «Панель элементов» из раздела «Все формы Windows Forms». С помощью окна «Свойства» установите в их свойства Minimum значение 2, а в Maximum - значение 16. Рядом с каждым из них разместите компонент NumericUpDown «числовое поле со стрелками вверх/вниз». Добавьте на форму два компонента Label. В одном будем отображать основание системы счисления р1, в другом – подпись к нему. Добавьте на форму ещё два компонента Label для основания системы счисления результата р2.

Добавьте на форму компонент MenuStrip из вкладки «Панель элементов» из раздела «Все формы Windows Forms». Добавьте в компонент MenuStrip три пункта меню MenuItem. С помощью окна «Свойства» установите в их свойства Text значение Выход, История, Справка. Полученная форма будет иметь вид, как представлено на рис.18 .

Для настройки отклика приложения на действия пользователя необходимо создать обработчики событий для тех компонент интерфейса, которыми будет манипулировать пользователь. Обработчик события для выделенного компонента создаётся с помощью окна «Свойства» закладки «События». Правее выбранного события в свободном поле необходимо сделать двойной клик мышью и в классе формы появится текст шаблона обработчика этого события. Затем вам необходимо описать в обработчике необходимые вам действия. Обработчики событий для компонентов главной формы приведены в таблице 3.

Таблица 3 - Обработчики событий для компонентов главной формы

|  |  |  |
| --- | --- | --- |
| **Имя компонента** | **Событие** | **Обработчик** |
| TPanelp\_p (главная форма) | Load | TPanelp\_p\_Load |
| TPanelp\_p (главная форма) | KeyDown | TPanelp\_p\_KeyDown |
| TPanelp\_p (главная форма) | KeyPress | TPanelp\_p\_KeyPress |
| trackBar1 | Scroll | trackBar1\_Scroll |
| trackBar2 | Scroll | trackBar1\_Scroll |
| numericUpDown1 | ValueChanged | numericUpDown1\_ValueChanged |
| numericUpDown2 | ValueChanged | numericUpDown1\_ValueChanged |
| button1 – button19 | Click | button\_Click |
| выходToolStripMenuItem | Click | выходToolStripMenuItem\_Click |
| историяToolStripMenuItem | Click | историяToolStripMenuItem1\_Click |
| справкаToolStripMenuItem | Click | справкаToolStripMenuItem\_Click |

Теперь можно запустить приложение и протестировать работу командных кнопок для ввода цифр действительного числа, представленного в выбранной системе счисления. А также для изменения оснований систем счисления исходного числа и результата. Ниже приведён текст модуля главного окна приложения.

namespace Конвертор

{

public partial class Form1 : Form

{

//Объект класса Управление.

Control\_ ctl = new Control\_();

public Form1()

{

InitializeComponent();

}

private void Form1\_Load(object sender, EventArgs e)

{

label1.Text = ctl.ed.Number;

//Основание с.сч. исходного числа р1.

trackBar1.Value = ctl.Pin;

//Основание с.сч. результата р2.

trackBar2.Value = ctl.Pout;

label3.Text = "Основание с. сч. исходного числа " + trackBar1.Value;

label4.Text = "Основание с. сч. результата " + trackBar2.Value;

label2.Text = "0";

//Обновить состояние командных кнопок.

this.UpdateButtons();

}

//Обработчик события нажатия командной кнопки.

private void button1\_Click(object sender, EventArgs e)

{

//ссылка на компонент, на котором кликнули мышью

Button but = (Button)sender;

//номер выбранной команды

int j = Convert.ToInt16(but.Tag.ToString());

DoCmnd(j);

}

//Выполнить команду.

private void DoCmnd(int j)

{

if (j == 19) { label2.Text = ctl.DoCmnd(j); }

else

{

if (ctl.St == Control\_.State.Преобразовано)

{

//очистить содержимое редактора

label1.Text = ctl.DoCmnd(18);

}

//выполнить команду редактирования

label1.Text = ctl.DoCmnd(j);

label2.Text = "0";

}

}

//Обновляет состояние командных кнопок по основанию с. сч. исходного числа.

private void UpdateButtons()

{

//просмотреть все компоненты формы

foreach (Control i in this.Controls)

{

if (i is Button)//текущий компонент - командная кнопка

{

int j = Convert.ToInt16(i.Tag.ToString());

if (j < trackBar1.Value)

{

//сделать кнопку доступной

i.Enabled = true;

}

if ((j >= trackBar1.Value) && (j <= 15))

{

//сделать кнопку недоступной

i.Enabled = false;

}

}

}

}

//Изменяет значение основания с.сч. исходного числа.

private void trackBar1\_Scroll(object sender, EventArgs e)

{

numericUpDown1.Value = trackBar1.Value;

//Обновить состояние командных кнопок.

this.UpdateP1();

}

//Изменяет значение основания с.сч. исходного числа.

private void numericUpDown1\_ValueChanged(object sender, EventArgs e)

{

//Обновить состояние.

trackBar1.Value = Convert.ToByte(numericUpDown1.Value);

//Обновить состояние командных кнопок.

this.UpdateP1();

}

//Выполняет необходимые обновления при смене ос. с.сч. р1.

private void UpdateP1()

{

label3.Text = "Основание с. сч. исходного числа " + trackBar1.Value;

//Сохранить р1 в объекте управление.

ctl.Pin = trackBar1.Value;

//Обновить состояние командных кнопок.

this.UpdateButtons();

label1.Text = ctl.DoCmnd(18);

label2.Text = "0";

}

//Изменяет значение основания с.сч. результата.

private void trackBar2\_Scroll(object sender, EventArgs e)

{

//Обновить состояние.

numericUpDown2.Value = trackBar2.Value;

this.UpdateP2();

}

//Изменяет значение основания с.сч. результата.

private void numericUpDown2\_ValueChanged(object sender, EventArgs e)

{

trackBar2.Value = Convert.ToByte(numericUpDown2.Value);

this.UpdateP2();

}

//Выполняет необходимые обновления при смене ос. с.сч. р2.

private void UpdateP2()

{

//Копировать основание результата.

ctl.Pout = trackBar2.Value;

//Пересчитать результат.

label2.Text = ctl.DoCmnd(19);

label4.Text = "Основание с. сч. результата " + trackBar2.Value;

}

//Пункт меню Выход.

private void выходToolStripMenuItem\_Click(object sender, EventArgs e)

{

Close();

}

//Пункт меню Справка.

private void справкаToolStripMenuItem\_Click(object sender, EventArgs e)

{

AboutBox1 a = new AboutBox1();

a.Show();

}

//Пункт меню История.

private void историяToolStripMenuItem1\_Click(object sender, EventArgs e)

{

Form2 history = new Form2();

history.Show();

if (ctl.his.Записей() == 0)

{

history.textBox1.AppendText("История пуста");

return;

}

//Ообразить историю.

for (int i = 0; i < ctl.his.Записей(); i++)

{

history.textBox1.AppendText(ctl.his[i].ToString());

}

}

//Обработка алфавитно-цифровых клавиш.

private void Form1\_KeyPress(object sender, KeyPressEventArgs e)

{

int i = -1;

if (e.KeyChar >= 'A' && e.KeyChar <= 'F') i = (int)e.KeyChar - 'A' + 10;

if (e.KeyChar >= 'a' && e.KeyChar <= 'f') i = (int)e.KeyChar - 'a' + 10;

if (e.KeyChar >= '0' && e.KeyChar <= '9') i = (int)e.KeyChar - '0';

if (e.KeyChar == '.') i = 16;

if ((int)e.KeyChar == 8) i = 17;

if ((int)e.KeyChar == 13) i = 19;

if ((i < ctl.Pin)||(i >= 16)) DoCmnd(i);

}

//Обработка клавиш управления.

private void Form1\_KeyDown(object sender, KeyEventArgs e)

{

if (e.KeyCode == Keys.Delete)

//Клавиша Delete.

DoCmnd(18);

if (e.KeyCode == Keys.Execute)

//Клавиша Execute Separator.

DoCmnd(19);

if (e.KeyCode == Keys.Decimal)

//Клавиша Decimal.

DoCmnd(16);

}

}

}

#### Содержание отчета

1. Задание.
2. Текст программы.
3. Тестовые наборы данных для тестирования методов класса.

#### Контрольные вопросы

1. Назначения компонентов класса Button?
2. Назначения компонентов класса Label?
3. Назначения компонентов класса TextBox?
4. Назначения компонентов класса TrackBar?
5. Назначения компонентов класса numericUpDown?
6. Когда возникает событие Load?
7. Когда возникает событие Click?
8. Когда возникает событие Scroll?
9. Когда возникает событие ValueChanged?
10. Когда возникает событие KeyPress?
11. Когда возникает событие KeyDown?