|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РЕСПУБЛИКИ КАЗАХСТАН  СЕВЕРО-КАЗАХСТАНСКИЙ УНИВЕРСИТЕТ ИМ. М. КОЗЫБАЕВА  ФАКУЛЬТЕТ ИНЖЕНЕРИИ И ЦИФРОВЫХ ТЕХНОЛОГИЙ  КАФЕДРА «ИНФОРМАЦИОННО-КОММУНИКАЦИОННЫЕ ТЕХНОЛОГИИ» | | | | |
|  | | | | |
|  | | | |  |
| Объектно-ориентированное программирование  МОДУЛЬ 2  Вариант 2 | | | | |
| Выполнил студент  группы ВТиПО-22 |  | | Белокопытов Я.Ю. | |
| старший преподаватель, магистр информационных систем |  | | Долматова Л.В. | |
|  |  |  | | |

Петропавловск, 2024

ЛАБОРАТОРНАЯ РАБОТА №7

1.1 Событие мыши DoubleClick элемента управления ImageBox изменяет

FormBorderStyle формы.

Листинг кода:

public partial class Form1 : Form

{

private bool isTriangle = false;

public Form1()

{

InitializeComponent();

}

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

{

if (!isTriangle)

{

GraphicsPath trianglePath = new GraphicsPath();

Point[] points =

{

new Point(this.Width / 2, 0),

new Point(0, this.Height),

new Point(this.Width, this.Height)

};

trianglePath.AddPolygon(points);

this.Region = new Region(trianglePath);

isTriangle = true;

}

else

{

this.Region = null;

isTriangle = false;

}

}

private void textBox1\_GotFocus(object sender, EventArgs e)

{

this.Opacity = 0.75;

}

private void textBox1\_LostFocus(object sender, EventArgs e)

{

this.Opacity = 1;

}

}

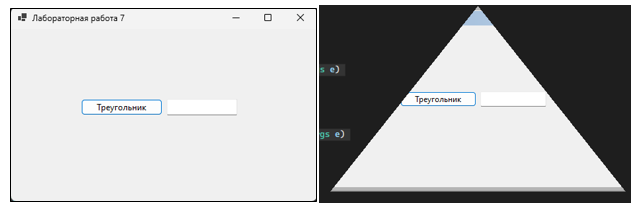


Рисунок 1 Результат

2 Лабораторная работа 8

2.1. Создать приложение, в котором используются свойства кнопок. На форме определить несколько кнопок; при щелчке на одной кнопке какие-то кнопки становятся невидимыми, другие недоступны, меняется цвет, размер и т. д. (минимум пять свойств).

public partial class Form1 : Form

{

bool isClicked = false;

public Form1()

{

InitializeComponent();

}

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

{

if (!isClicked)

{

button1.BackColor = Color.Aqua;

button2.Height = 100;

button3.Visible = false;

button4.Text = "Текст";

button4.Enabled = false;

button5.Location = new Point(315, 53);

isClicked = true;

}

else

{

button1.BackColor = Color.White;

button2.Height = 23;

button3.Visible = true;

button4.Text = "button 4";

button4.Enabled = true;

button5.Location = new Point(315, 103);

isClicked = false;

}

}

private void button2\_Click(object sender, EventArgs e)

{

}

}

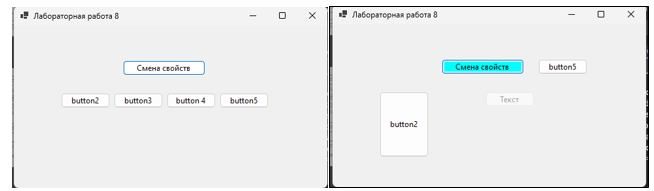


Рисунок 2 Результат

2.2. Создать приложение с использованием панелей, имеющее внешний вид, соответствующий приведенному на рисунках по вариантам (рисунки 8.1–8.5)

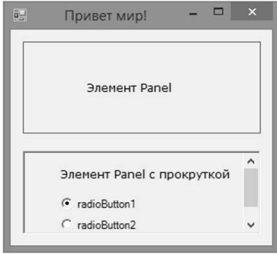


Рисунок 3 Задание

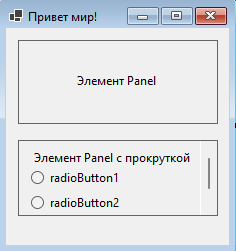


Рисунок 4 Результат

# 3 Лабораторная работа 9

3.1. Создать меню с командами Input, Work, Exit.

При выборе команды Exit приложение завершает работу. При выборе команды Input открывается диалоговое окно, содержащее:

− три поля ввода типа TextBox с метками Radius, Height, Density;

− группу из двух флажков (Volume, Mass) типа Checkbox;

− кнопку типа Button.

Обеспечить возможность:

− ввода радиуса, высоты и плотности конуса;

− выбора режима с помощью флажков: подсчет объема и/или массы конуса.

При выборе команды Work открывается окно сообщений c результатами.

Листинг кода:

public partial class Form1 : Form

{

private float radius = 0;

private float height = 0;

private float density = 0;

private double volume = 0;

private double mass = 0;

Form dilog1 = new Form();

public Form1()

{

InitializeComponent();

}

private void button2\_Click(object sender, EventArgs e)

{

if (checkBoxDialog1.Checked && checkBoxDialog2.Checked)

{

MessageBox.Show($"Объем конуса: {volume} кубических единиц\nМасса конуса: {mass} единиц массы", "Результат", MessageBoxButtons.OK, MessageBoxIcon.Information);

} else if (checkBoxDialog1.Checked && !checkBoxDialog2.Checked)

{

MessageBox.Show($"Объем конуса: {volume} кубических единиц", "Результат", MessageBoxButtons.OK, MessageBoxIcon.Information);

} else if (!checkBoxDialog1.Checked && checkBoxDialog2.Checked)

{

MessageBox.Show($"Массу конуса невозможно рассчитать без объема", "Результат", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

}

private void buttonDialog1\_Click(object sender, EventArgs e)

{

try

{

radius = float.Parse(textBox1.Text);

height = float.Parse(textBox2.Text);

density = float.Parse(textBox3.Text);

if (radius < 0)

{

MessageBox.Show("Радиус не может быть отрицательным!", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

if (density <= 0)

{

MessageBox.Show("Плотность должна быть положительной!", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

return;

}

volume = (1.0 / 3.0) \* Math.PI \* Math.Pow(radius, 2) \* height;

mass = volume \* density;

dilog1.Close();

}

catch (FormatException)

{

MessageBox.Show("Введите корректные числовые значения! Возможны вы оставили поля пустыми", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void button3\_Click(object sender, EventArgs e)

{

Application.Exit();

}

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

{

dilog1.Text = "Input";

dilog1.Size = new Size(390, 190);

dilog1.Controls.Add(textBox1);

dilog1.Controls.Add(textBox2);

dilog1.Controls.Add(textBox3);

dilog1.Controls.Add(labelDialog1);

dilog1.Controls.Add(labelDialog2);

dilog1.Controls.Add(labelDialog3);

dilog1.Controls.Add(buttonDialog1);

dilog1.Controls.Add(panelDialog1);

dilog1.ShowDialog();

}

}

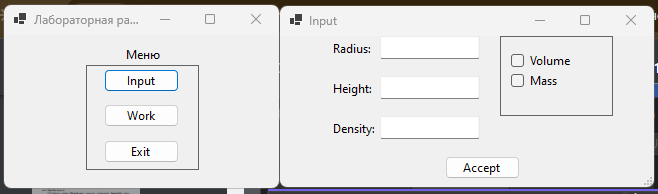


Рисунок 5 Результат

# 4 Лабораторная работа 2

4.1. Создание и использование элемента управления ToolStrip и использование элемента управления StatusStrip.

Листинг кода:

public partial class Form1 : Form

{

public partial class ChildForm : Form

{

public ChildForm()

{

}

}

public Form1()

{

InitializeComponent();

spData.Text = Convert.ToString(System.DateTime.Today.ToLongDateString());

IsMdiContainer = true;

}

private void toolStrip1\_ItemClicked(object sender, ToolStripItemClickedEventArgs e)

{

int openDocuments = 0;

switch (e.ClickedItem.Tag.ToString())

{

case "NewDoc":

ChildForm newChild = new ChildForm();

newChild.MdiParent = this;

newChild.Show();

newChild.Text = newChild.Text + " " +

++openDocuments;

break;

case "Cascade":

this.LayoutMdi(System.Windows.Forms.MdiLayout.Cascade);

spWin.Text = "Windows is cascade";

break;

case "Title":

this.LayoutMdi

(System.Windows.Forms.MdiLayout.TileHorizontal);

spWin.Text = "Windows is horizontal";

break;

}

}

}

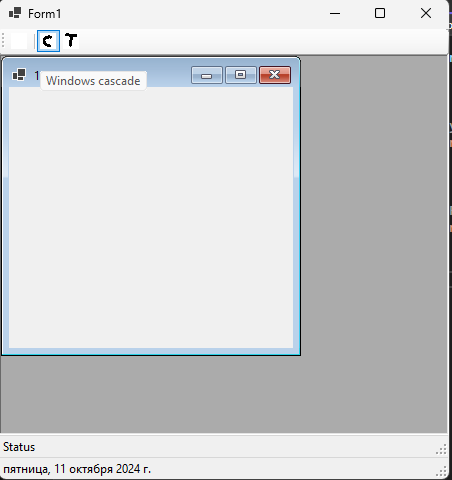


Рисунок 6 Результат

4.2. Добавление и удаление элементов управления в режиме работы приложения и проверка вводимых значений. События KeyPress и Validating. Элемент управления ErrorProvider.

Листинг кода:

public Form1()

{

InitializeComponent();

}

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

{

}

private void checkBox1\_CheckedChanged(object sender, EventArgs e)

{

if (checkBox1.Checked == true)

{

Label lbl = new Label();

lbl.Location = new System.Drawing.Point(16, 96);

lbl.Size = new System.Drawing.Size(32, 23);

lbl.Name = "labelll";

lbl.TabIndex = 2;

lbl.Text = "PIN2";

groupBox1.Controls.Add(lbl);

TextBox txt = new TextBox();

txt.KeyPress += new System.Windows.Forms.KeyPressEventHandler(this.textBox2\_KeyPress);

txt.Location = new System.Drawing.Point(96, 96);

txt.Size = new System.Drawing.Size(184, 20);

txt.Name = "textboxx";

txt.TabIndex = 1;

txt.Text = "";

groupBox1.Controls.Add(txt);

}

else

{

int lcv;

lcv = groupBox1.Controls.Count;

while (lcv > 4)

{

groupBox1.Controls.RemoveAt(lcv - 1);

lcv -= 1;

}

}

}

private void textBox1\_KeyPress(object sender, KeyPressEventArgs e)

{

if (char.IsDigit(e.KeyChar))

{

e.Handled = true;

MessageBox.Show("Поле Name не может содержать цифры!");

}

errorProvider1.SetError(textBox1, "Must be letter");

}

private void textBox2\_KeyPress(object sender, KeyPressEventArgs e)

{

/\*if (!char.IsDigit(e.KeyChar))

{

e.Handled = true;

MessageBox.Show("Поле PIN не может содержать буквы!");

}\*/

}

private void textBox2\_Validating(object sender, CancelEventArgs e)

{

if (textBox2.Text == "")

{

e.Cancel = false;

}

else

{

try

{

double.Parse(textBox2.Text);

e.Cancel = false;

}

catch

{

e.Cancel = true;

MessageBox.Show("Поле PIN не может содержать буквы!");

}

}

}

}

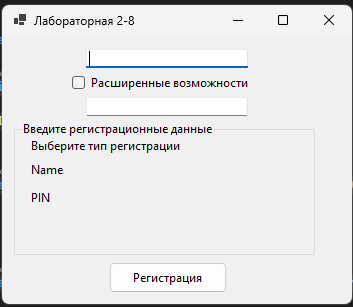


Рисунок 7 Результат

ВЫВОД

В модуле 2 было выполнено 7 заданий, улучшены навыки в работе с Windows Forms.