**Практическая работа 4**

**Выполнила:** Шестакова Д.Н.

**Группа:** ИС213

1. Счетчик кликов: Разработайте приложение с кнопкой и меткой. Каждый раз, когда пользователь нажимает кнопку, счётчик (отображаемый в метке) увеличивается на 1. Далее сделать условия, что при n количестве кликов, появляются новые возможности на форме.

Для разработки приложения потребуется разместиться в форме некоторые детали с помощью панели элементов:

* Label – строчка текста для отображения количества нажатий
* Button – кнопка, с помощью которой происходят нажатия
* PictureBox – изображение, которые будет размещено для внешнего вида приложения

Внешний вид интерфейса выглядит следующим образом:



Далее идет разработка кода, основных и дополнительных действий:

namespace cliker

{

public partial class Form1 : Form

{

private int clickCount = 0;

public Form1()

{

InitializeComponent();

}

private void label1\_Click(object sender, EventArgs e)

{ }

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

{ }

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

{

clickCount++;

label1.Text = $"Кликнуто: {clickCount}";

// Изменяем цвет метки каждые кликов

if (clickCount % 5 == 0)

{

ChangeLabelColor();

if (ShowMouseWarning()) // Показ уведомления каждые 10 кликов

{

this.Close(); // Закрытие программы, если нажали "Да"

}

}

}

private void ChangeLabelColor()

{

Random random = new Random();

label1.BackColor = Color.FromArgb(random.Next(256), random.Next(256), random.Next(256));

}

private void ChangeLabelText()

{

}

private bool ShowMouseWarning()

{

DialogResult result = MessageBox.Show("Мышку не жалко?(", "Предупреждение", MessageBoxButtons.YesNo, MessageBoxIcon.Warning);

return result == DialogResult.Yes; // если нажали "Да"

}

private void pictureBox1\_Click(object sender, EventArgs e)

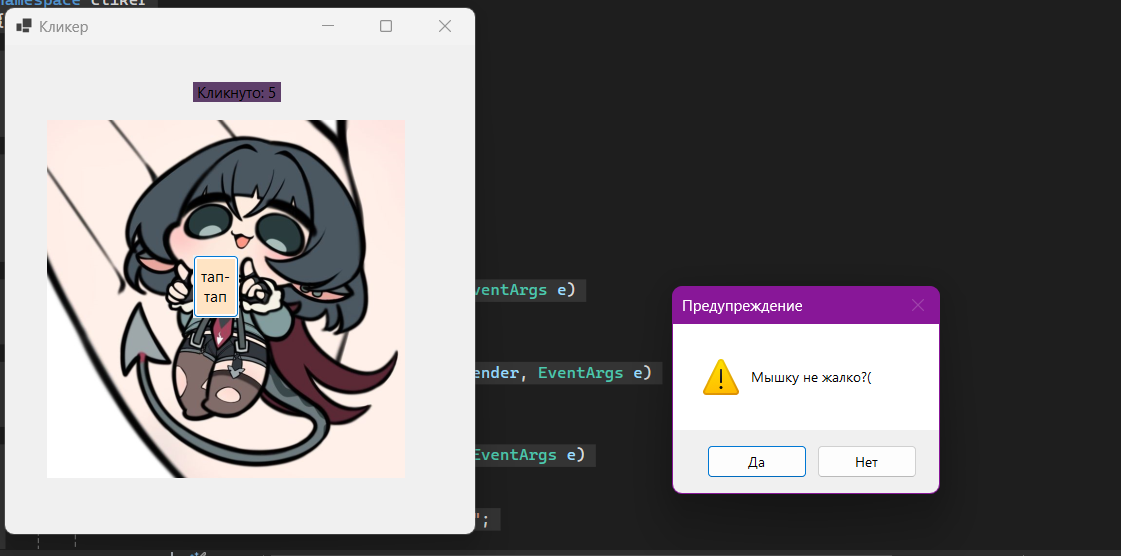
{ }

}

}

Проверяем программу на корректность работы:

После каждого 10-ого нажатия строка отсчета нажатий меняет свой цвет, а также появляется окно уведомления:



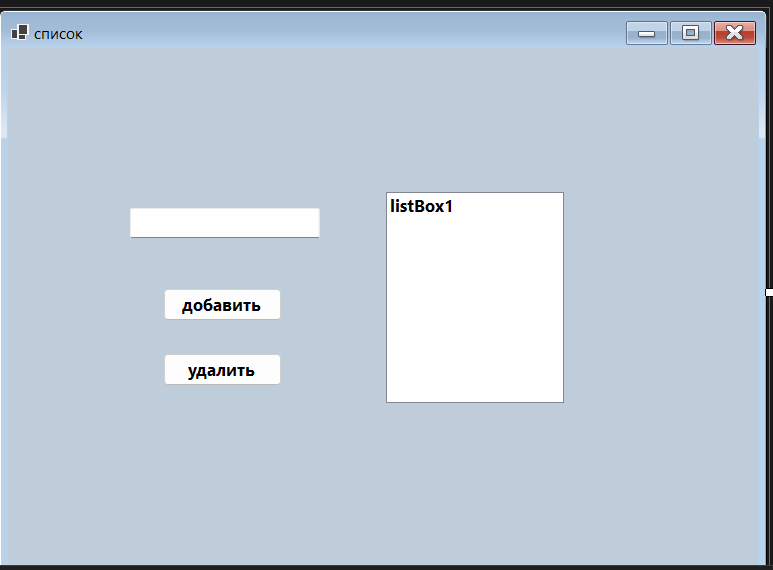
При нажатии «Нет» программа продолжает работать, а цвета в строке меняются, если нажать «Да» программа закрывается.

2. Создайте простую форму, где пользователь может вводить название товара в текстовое поле и добавлять его в ListBox. Сделайте возможность удаления выбранного товара из списка. Сделайте вывод отдельного окна, с именем выбранного элемента списка.

Для разработки приложения потребуется разместиться в форме некоторые детали с помощью панели элементов:

* Button – кнопка, с помощью которой происходят нажатия, добавления и удаления товара
* ListBox – элемент, отображающий список
* TextBox – элемент, отображающий строку ввода товара

Внешний вид интерфейса выглядит следующим образом:



Далее разрабатывается код работы и взаимодействия элементов в форме друг с другом:

namespace tovar

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void listBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

}

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

{

}

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

{

string productName = textBox1.Text;

// Проверка на пустое значение

if (string.IsNullOrWhiteSpace(productName))

{

MessageBox.Show("Пожалуйста, введите название товара.");

return;

}

// Добавление товара в ListBox

listBox1.Items.Add(productName);

// Очистка TextBox

textBox1.Clear();

textBox1.Focus(); // Ставим фокус обратно на текстовое поле

}

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

{

if (listBox1.SelectedItem != null)

{

listBox1.Items.Remove(listBox1.SelectedItem);

}

else

{

MessageBox.Show("Пожалуйста, выберите товар для удаления. ");

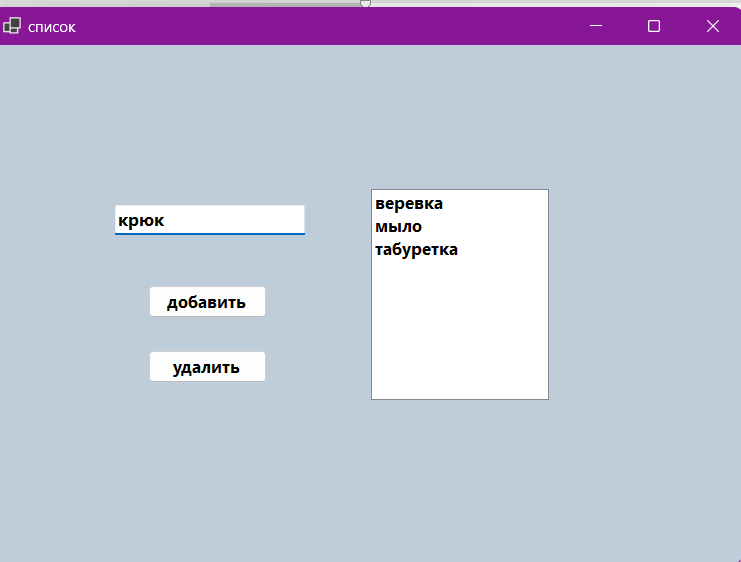
}

}

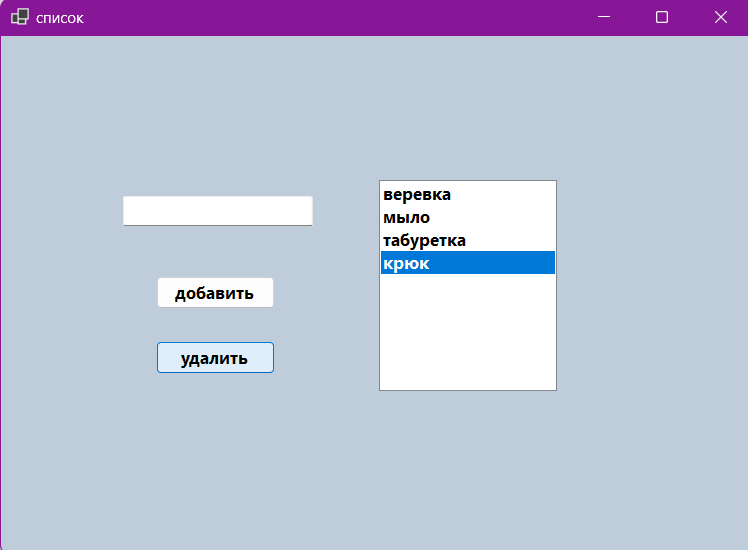
}

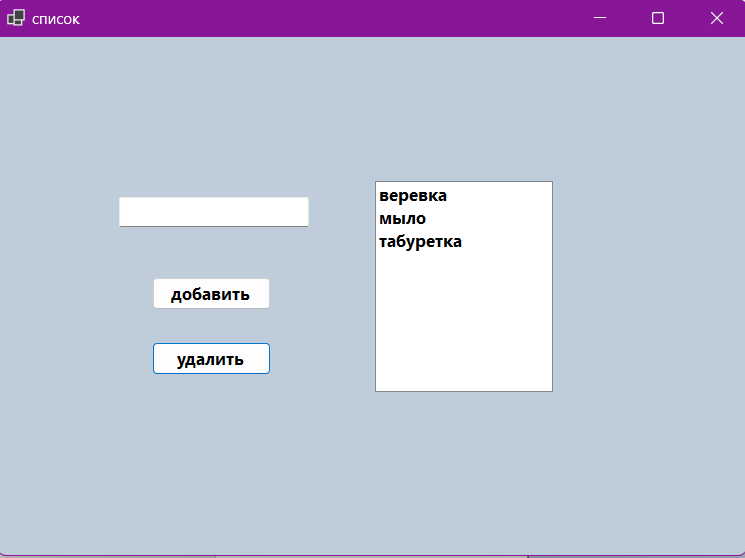
}

Проверка на корректность работы программы. В TextBox вводим тот или иной товар, при нажатии на кнопку «Добавить» данный элемент попадает в список ListBox:



При указании элемента в списке, возможно удалить его:





Если ничего не вводить и не выбирать элемент в списке, срабатывает уведомление:

