

Лабораторна робота 2. Робота з елементами управління

- Вправа 1.** Обробка подій Click і MouseMove (2 бали)
- Вправа 2.** Робота зі списками (2 бали)
- Вправа 3.** Створення і використання елемента управління ToolStrip (4 бали)
- Вправа 4.** Використання елемента управління StatusStrip (2 бали)
- Вправа 5.** Робота з контейнерними елементами управління (5 балів)
- Вправа 6.** Елементи з підтримкою відображення тексту (3 бали)
- Вправа 7.** Елементи з підтримкою редагування тексту (2 бали)
- Вправа 8.** Додавання і видалення елементів управління в режимі роботи застосування (2 бали)
- Вправа 9.** Перевірка значень, які вводяться. Подія KeyPress і Validating. Елемент управління ErrorProvider (3 бали)

Мета роботи: Вивчення способів використання елементів управління та отримання навичок з обробки подій.

Вправа 1. Обробка подій Click і MouseMove (2 бали)

Елементи управління - це компоненти, які об'єднують графічний інтерфейс із задалегідь розробленою функціональністю. Елементи управління є багаторазово використовуваними блоками коду, призначеними для виконання певних завдань. Всі елементи управління є похідними базового класу **Control**, а значить, теж використовують різні властивості, які задають розмір, розташування та інші основні аспекти елементів управління.

Виконавши першу вправу цього завдання, ви створите просте застосування, яке відслідковує події миші, які відбуваються у конкретного елемента управління.

Розміщення на формі елементів управління

1. Створіть нове **Windows**-застосування. Назвіть його **WinQuestion**.
2. Розташуйте на формі дві кнопки **Button** і напис **Label**, розмістіть їх на свій розсуд.
3. Встановіть наступні властивості елементів управління і форми:

| Об'єкт | Властивість | Значення |
|---------|-----------------|---------------------------------|
| Form1 | FormBorderStyle | Fixed3D |
| | Size | 350; 200 |
| | Text | Суттєве питання |
| label1 | Text | Ви задоволені своєю стипендією? |
| Button1 | Name | btnyes |
| | Text | Так |
| Button2 | Name | btnno |
| | Text | Ні |

4. Клацніть двічі по кнопці "Так". В обробнику цієї кнопки додайте наступний код:

```
MessageBox.Show("Ми і не сумнівалися, що Ви так думаєте!");
```

- Виділіть кнопку "Hi". У вікні **Properties** переключіться у вікно подій і двічі клацніть в полі MouseMove.
- В обробнику цієї події додайте код для зв'язування руху миші з координатами кнопки і зазначення координат, куди кнопка переміщатиметься:

```
btnno.Top -= e.Y;
btnno.Left += e.X;
if (btnno.Top < - 10 || btnno.Top > 100)
    btnno.Top = 60;
if (btnno.Left < - 80 || btnno.Left > 250)
    btnno.Left = 120;
```

- Запустіть застосування і натисніть на кожну з кнопок.

Вправа 2. Робота зі списками (2 бали)

Основними елементами управління списком є **ListBox**, **ComboBox** і **CheckedListBox**. Незважаючи на деякі відмінності в зовнішньому вигляді і різні функціональні можливості, вони однаково формують і представляють списки даних і включають колекцію **Items**, яка систематизує елементи, які містять один з цих елементів управління.

| Елемент управління | Опис |
|--------------------|---|
| ListBox | Найпростіший елемент управління списку - служить головним чином для відображення простого списку елементів в інтерфейсі користувача, яким легко переміщатися. |
| CheckedListBox | Список із позначками - є різновидом простого списку. Його додаткова позитивна характеристика - в наявності чекбоксі поряд з кожним елементом списку. Користувач може відмітити один або декілька елементів списку, виставивши навпроти нього прапорець. |
| ComboBox | Випадний список . Постійно на формі представлено тільки одне значення цього списку. При потребі користувач може розкрити список і вибрати інше значення, яке цікавить його, або ввести власне. |

Створення застосування, яке використовує список

- Створіть новий проект **Windows Forms**, вкажіть ім'я **TestList**.
- Додайте на форму наступні елементи управління:
 - GroupBox**,
 - CheckedListBox** (помістіть в **GroupBox**)
 - ComboBox**
 - три елементи **Button**.
- Встановіть наступні властивості форми та елементів управління:

| Об'єкт | Властивість | Значення |
|----------------|-----------------|--------------------|
| Form1 | FormBorderStyle | Fixed3D |
| | Text | Робота зі списками |
| | Size | 410; 310 |
| groupBox1 | Text | Список учасників |
| CheckedListBox | Name | memberList |
| ComboBox | Name | peopleList |
| | Text | |
| Button1 | Name | buttonAdd |
| | Text | Додати |
| Button2 | Name | buttonDelete |

| | | |
|---------|------|------------|
| | Text | Видалити |
| Button3 | Name | buttonSort |
| | Text | Сортувати |

4. Проініціалізуйте елемент управління **ComboBox** списком можливих учасників. Для цього у вікні властивостей **peopleList** виберіть властивість **Items**. Відкрийте вікно **String Collection Editor**, натиснувши на кнопку з трьома точками в полі **Items (Изменить элементы)**. Додайте у вікно П.І.Б, декількох учасників. Натисніть **ОК**.
5. Додайте обробники для кнопок **Додати** і **Видалити**, двічі клацнувши лівою кнопкою миші по кожній з кнопок.
6. У тіло обробника події кнопки **Додати** вставте наступний код:

```
if (peopleList.Text.Length != 0)
{
    memberList.Items.Add(peopleList.Text);
}
else MessageBox.Show("Виберіть елемент зі списку або введіть новий");
```

7. Для реалізації видалення елементів зі списку введіть код в тіло обробника події кнопки **Видалити**:

```
while (memberList.CheckedIndices.Count > 0)
    memberList.Items.RemoveAt(memberList.CheckedIndices[0]);
```

8. Для реалізації сортування елементів введіть код в тіло обробника події кнопки **Сортувати**:

```
memberList.Sorted = true;
```

9. Відкомпілюйте і запустіть застосування. Заповніть список учасників, вибираючи їх з елемента **ComboBox**. Запишіть нові дані в цей елемент і додайте їх в список. Відсортуйте список учасників.

Вправа 3. Створення і використання елемента управління **ToolStrip** (4 бали)

| Елемент управління | Опис |
|--------------------|--|
| ToolStrip | Елемент управління, розроблений з метою спрощення створення панелей інструментів користувача, які виглядають і працюють, як панелі інструментів Microsoft Office і Microsoft Internet Explorer. Використовуючи елемент управління ToolStrip, ви можете швидко розробляти панелі інструментів професійного вигляду, які легко налаштовуються. |

Додавання на форму шаблону панелі інструментів

1. Відкрийте проект **MdiApplication**.
2. Відкрийте форму в режимі конструктора.
3. Додайте на форму **Parent Form** елемент управління **ToolStrip**.
4. На формі відкрийте випадний список елемента управління **ToolStrip** і виберіть **Button** – буде додано елемент **toolStripButton1**. В панелі інструментів він представлений у вигляді кнопки з малюнком, який означає функцію, яку цей елемент містить.
5. Знову відкрийте випадний список елемента управління **ToolStrip** і виберіть **Separator** - додасться елемент, який відділяє одні елементи панелі інструментів від інших.

- Праворуч від роздільника додайте ще дві кнопки – елементи **toolStripButton2** і **toolStripButton3**.
- У результаті ви повинні отримати три кнопки, відокремлені одним роздільником.

Відображення малюнка на елементах панелі управління

- Виберіть першу кнопку. Переконайтеся, що у вікні **Properties** властивості **DisplayStyle** задано значення **Image**.
- У вікні **Properties** виберіть зображення елемента управління, клацнувши властивість **Image** і вибравши відповідне зображення або шлях до нього в діалоговому вікні **Select Resource (Выбор ресурса)**. Якщо у вас є готові файли зображень, то виберіть їх, інакше знайдіть файл із зображеннями.
- Повторіть попередні два пункти для інших кнопок.

Створення графічних зображень кнопок панелі інструментів

- Якщо вас не влаштовує вигляд готових малюнків, то можна для надання кнопкам графічних зображень самостійно їх створити. Створювати зображення можна з використанням середовища розробки **Visual Studio Image**. Воно дозволяє створити зображення з використанням простих інструментів.
- Для опанування роботи з редактором графічних зображень **Visual Studio Image** можна звернутися до MSDN за електронною адресою <https://msdn.microsoft.com/en-us/library/hh315744.aspx>.
- Для створення файла із зображенням відкрийте контекстне меню проекту, клацнувши на імені проекту. Із контекстного меню виберіть **Додати** → **Створити елемент**. У вікні **Доданий новий елемент** яке з'явилося, виберіть **Графіка**, а потім виберіть відповідний формат файла для зображення. Далі вказуємо ім'я для файла зображення і місце де ми його збережемо. Для відображення панелі з палітрою компонент достатньо вибрати в панелі інструментів потрібний інструмент. Маємо можливість вибрати бажану палітру, а також інші параметри, які стосуються розміру та ін.
- Створіть на власний розсуд зображення для кнопки **New**, наприклад, у вигляді синього листа і збережіть зображення у файл з ім'ям **Icon_New.bmp** в каталог з рішенням.
- Повторіть дії для створення іконок для інших кнопок, наприклад, у вигляді букв **C** і **T**. Збережіть зображення в каталог з рішенням у файли з іменами **Icon_Windows_Cascade.bmp** та **Icon_Windows_Title.bmp** відповідно.
- Виконайте дії для вибору нових зображень цим кнопкам.
- Збережіть і запустіть проект. Перевірте вигляд і працездатність кнопок.

Додавання обробників подій для кнопок

- Додайте обробник події **Click** об'єкта **toolStrip1**, клацнувши двічі покажчиком миші на імені події **ItemClicked** на закладці подій у вікні властивостей. У програму додається функція **toolStrip1_ItemClicked** як обробник події, яка відбувається при натисненні кнопки на панелі інструментів.
- У вікні **Properties** для **toolStripButton1** у властивості **Tag** запишіть **NewDoc**. Аналогічно вкажіть для **toolStripButton2** і **toolStripButton3** для властивості **Tag** значення **Cascade** і **Tile** відповідно.
- Вкажіть для кнопок спливаючі підказки у властивості **ToolTipText: Create new document, Windows cascade і Windows tile**.
- В обробнику події **Click** об'єкта **toolStrip1_ItemClicked** додайте код, який реалізовуватиме різні дії в залежності від кнопки, яка натискається:

```
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);
        break;
    case "Tile":
        this.LayoutMdi(System.Windows.Forms.MdiLayout.TileHorizontal);
        break;
}

```

22. Відкомпілюйте і запустіть застосування. Перевірте працездатність кнопок.

Вправа 4. Використання елемента управління **StatusStrip** (2 бали)

Елемент управління **StatusStrip** застосовується в програмах для виведення інформації в *рядок стану* - невелику смужку, розташовану внизу застосування. У цій вправі ви додасте до застосування **MdiApplication** рядок стану, на якому показується варіант орієнтації вікон і виводиться поточна дата.

1. Відкрийте проект **MdiApplication**.
2. Збільшіть розмір форми до значення **(450;350)**.
3. Додайте на форму елемент управління **StatusStrip**.
4. Видаліть вміст поля властивості **Text**.
5. Клацніть на кнопку випадного списку панелі і виберіть **StatusLabel**. Додасться елемент **toolStripStatusLabel1** - перша панель для відображення.
6. Створіть ще одну панель аналогічним чином - **toolStripStatusLabel2** і встановіть їм наступні властивості:

| Об'єкт | Властивість | Значення |
|--------------|-------------|----------|
| Перша панель | Text | Status |
| | Name | spWin |
| Друга панель | Text | Data |
| | Name | spData |

7. Для відображення інформації на першій панелі вставте у відповідні обробники команд меню і кнопок на панелі інструментів наступний рядок коду:
 - а. Для каскадної орієнтації:


```
spWin.Text="Windows is cascade";
```
 - б. Для горизонтальної орієнтації:


```
spWin.Text="Windows is horizontal";
```
8. Для відображення дати на другій панелі в конструкторі форми **ParentForm** додайте код:

```

public ParentForm()
{
    InitializeComponent();
    // Властивості Text панелі spData встановлюється поточна дата
    spData.Text = Convert.ToString(System.DateTime.Today.ToLongDateString());
}

```

}

- Відкомпілюйте і запустіть застосування. Перевірте працездатність панелі стану.

Вправа 5. Робота з контейнерними елементами управління (5 балів)

Контейнерні елементи управління - це спеціалізовані елементи управління, які виступають в ролі сховища, яке налаштовується для інших елементів управління. До контейнерних елементів управління відносяться **Panel** і **GroupBox**. Вони надають формі логічні і фізичні підрозділи, які можуть групувати інші елементи управління в однакові підгрупи інтерфейсу користувача. Наприклад, елемент управління **GroupBox** містить в собі набір пов'язаних елементів управління **RadioButton**. Контейнерні елементи управління допоможуть вам створити відчуття стилю або інформаційного потоку в інтерфейсі користувача і дозволять узгоджено управляти елементами управління, які містяться в них.

Виконавши цю вправу, ви навчитеся створювати форми Windows з використанням різних контейнерних елементів управління.

Створення проекту з можливістю групування елементів на вкладках

- Відкрийте Visual Studio і створіть новий проект Windows Forms. Назвіть його **WinContainer**.
- Перетягніть з **Toolbox** у форму елемент управління **TabControl**. У вікні **Properties** задайте властивості **Dock** значення **Fill**.
- У вікні **Properties** виберіть властивість **TabPageCollectionEditor**. Додайте вкладки так, щоб їх стало всього п'ять. Задайте властивостям **Text** цих п'яти елементів управління **TabPage** значення **GroupBox**, **Panel**, **FlowLayoutPanel**, **TableLayoutPanel** і **SplitContainer**. Клацніть **ОК**.

Налаштування контейнерного елемента GroupBox

- У формі виберіть вкладку **GroupBox**. Перетягніть елемент управління **GroupBox** з **Toolbox** в елемент управління **TabPage** з іменем **GroupBox**.
- Перетягніть в **GroupBox** два елементи управління **RadioButton**.
- Додайте на вкладку **GroupBox** поза елементом управління **GroupBox** кнопку (елемент управління **Button**). Для кнопки властивість **Text** зробіть порожньою, а властивості **Name** вкажіть значення **but**.
- Двічі клацніть по кнопці і додайте код обробника події встановлення напису на кнопці залежно від вибраного перемикача (**RadioButton**):

```
if (radioButton1.Checked == true)
    this.but.Text = "First";
else if (radioButton2.Checked == true)
    this.but.Text = "Second";
```

Налаштування елемента Panel

- Виберіть у формі вкладку **Panel**. Перетягніть елемент управління **Panel** з **Toolbox** в елемент управління **TabPage** з іменем **Panel**. Для елемента **Panel** задайте властивості **Dock** значення **Fill**.
- Перетягніть чотири елементи управління **Button** з **Toolbox** в елемент управління **Panel**.
- Властивості **AutoScroll** елемента управління **Panel** встановіть значення **True**, у цьому випадку елемент управління **Panel** відображатиме смуги прокручування, якщо елементи знаходяться за межами видимих меж.

Налаштування елемента FlowLayoutPanel

11. Виберіть у формі вкладку **FlowLayoutPanel**. Перетягніть елемент управління **FlowLayoutPanel** з **Toolbox** в елемент управління **TabPage** з іменем **FlowLayoutPanel**. Задайте значення **Fill** властивості **Dock** елемента управління **FlowLayoutPanel**.
12. Перетягніть чотири елементи управління **Button** з **Toolbox** в елемент управління **FlowLayoutPanel**. Зверніть увагу на розміщення елементів, які додаються: за замовчуванням порядок розміщення елементів управління в **FlowLayoutPanel** - зліва направо. Це означає, що елементи управління, розташовані в **FlowLayoutPanel**, знаходяться в лівому верхньому куті і розміщуються управо до тих пір, поки не досягнуть краю панелі. Така поведінка контролюється властивістю **FlowDirection**, якій може бути задане чотири значення заливки у **FlowLayoutPanel**: **LeftToRight** - за замовчуванням, **RightToLeft** - справа наліво, **TopDown** - зверху вниз і **BottomUp** - знизу вгору.
13. Двічі клацніть кнопку **button5** і додайте в обробник події **button5_Click** наступний код:

```
flowLayoutPanel1.SetFlowBreak(button6, true);
```

Налаштування елемента TableLayoutPanel

14. Виберіть конструктор форми (якщо це потрібно). У формі виберіть вкладку **TableLayoutPanel**. Перетягніть елемент управління **TableLayoutPanel** з **Toolbox** в **TabPage** з іменем **TableLayoutPanel**. Задайте властивості **CellStyle** (визначає вигляд елементів таблиці та їх поведінку) значення **Inset**, а властивості **AutoScroll** - **True**.
15. Перетягніть елемент управління **Button** з **Toolbox** в ліву верхню комірку елемента управління **TableLayoutPanel**.
16. Двічі клацніть **Button9** і додайте в обробник події **Button9_Click** наступний код:

```
Button aButton = new Button();
tableLayoutPanel1.Controls.Add(aButton, 1, 1);
```

Налаштування елемента SplitContainer

17. У конструкторі виберіть вкладку **SplitContainer**. Перетягніть елемент управління **SplitContainer** з **Toolbox** в **TabPage** з іменем **SplitContainer**. Задайте властивості **BorderStyle** значення **Fixed3D**.
18. Перетягніть два елементи управління **Button** з **Toolbox** в **Panel1** елемента управління **SplitContainer**. Задайте властивостям **Text** цих кнопок значення **Fix/Unfix Panel1** і **Fix/Unfix Splitter**. Змініть розміри кнопок так, щоб відображався текст.
19. Додайте кнопку в **Panel2** і задайте властивості **Text** значення **Collapse/Uncollapse Panel1**. Змініть розміри кнопок так, щоб відображався текст.
20. Двічі клацніть кнопку **Fix/Unfix Panel1** і додайте в обробник події **Click** наступний код:

```
if (splitContainer1.FixedPanel == FixedPanel.Panel1)
    splitContainer1.FixedPanel = FixedPanel.None;
else
    splitContainer1.FixedPanel = FixedPanel.Panel1;
```

21. Двічі клацніть кнопку **Fix/Unfix Splitter** і додайте в обробник події **Click** наступний код:

```
splitContainer1.IsSplitterFixed = !splitContainer1.IsSplitterFixed;
```

22. Двічі клацніть кнопку **Collapse/Uncollapse Panel1** і додайте в обробник події **Click** наступний код:

```
splitContainer1.Panel1Collapsed = !(splitContainer1.Panel1Collapsed);
```

23. Побудуйте і запустіть застосування.
24. На вкладці **GroupBox** по черзі вибирайте перемикачі і клацнувши на кнопці стежте за зміною напису на кнопці.
25. На вкладці **Panel** змініть розмір форми за допомогою миші. Перевірте, чи з'явилися смуги прокручування.
26. На вкладці **FlowLayoutPanel** змініть розмір форми з допомогою миші. Перегляньте, чи змінилося автоматично компонування. Клацніть кнопку **button5** і перевірте, чи перервалася послідовність на елементі управління **button6** (це було реалізовано викликом методу **SetFlowBreak**).
27. На вкладці **TableLayoutPanel** клацніть кнопку **button9**, додасться нова кнопка.
28. На вкладці **SplitContainer** змініть розміри форми, а також розміри кожної панелі, пересунувши **Splitter**. По черзі клацайте кожну кнопку і дивіться, як це відбивається на можливості елемента управління змінювати свої розміри.

Вправа 6. Елементи з підтримкою відображення тексту (3 бали)

Виконавши цю вправу, ви навчитеся використовувати елемент управління **LinkLabel** у формі і налаштовувати його так, щоб відкривалося діалогове вікно, яке запитує у користувача ім'я.

Налаштування елемента управління **LinkLabel**

1. Відкрийте Visual Studio і створіть новий проект Windows Forms. Назвіть його **WinLinkLabel**.
2. Перетягніть два елементи управління **LinkLabel** з **Toolbox** в форму.
3. У вікні **Properties** для першого елемента задайте властивості **Text** значення **Open Form**, для другого - значення **Microsoft**.

Створення діалогової форми

4. У меню **Project** виберіть **Add Windows Form** і додайте у свій проект нову форму Windows з ім'ям **Form2**.
5. У конструкторі перетягніть два елементи управління **Button** у форму **Form2**. Задайте властивості **Text** цих кнопок значення **Accept** і **Cancel**. Розташуйте їх у правому нижньому кутку форми.
6. Задайте властивості **DialogResult** кнопки **Accept** значення **OK**, а властивості **DialogResult** кнопки **Cancel** - значення **Cancel**.
7. Перетягніть два елементи управління **TextBox** з **Toolbox** у форму.
8. Задайте властивості **Modifiers** кожного елемента управління **TextBox** значення **Internal**. Властивість **Text** для них залиште порожньою.
9. Перетягніть два елементи управління **Label** з **Toolbox** у форму і розмістіть їх поряд з елементом управління **TextBox**.
10. Задайте властивостям **Text** елементів управління **Label** значення **&First Name** і **&Last Name**.
11. Перевірте, що властивості **UseMnemonic** усіх написів встановлене значення **True**.
12. Встановіть властивість **TabIndex** у вікні **Properties**, як показано нижче.

| Елемент управління | Індекс закладки |
|--------------------|-----------------|
| label1 | 0 |
| textBox1 | 1 |
| label2 | 2 |
| textBox2 | 3 |

| | |
|---------|---|
| button1 | 4 |
| button2 | 5 |

Реалізація обробника події виклику діалогового вікна

13. Виберіть у конструкторі закладку для форми **Form1**. Двічі клацніть перший елемент управління **linkLabel1** для створення обробника події **linkLabel1_LinkClicked**. Додайте наступний код:

```
DialogResult aResult;
Form2 aForm = new Form2();
aResult = aForm.ShowDialog();
if (aResult == System.Windows.Forms.DialogResult.OK)
{
    MessageBox.Show("Your name is " + aForm.textBox1.Text + " " + aForm.textBox2.Text);
}
linkLabel1.LinkVisited = true;
```

Реалізація обробника події виклику веб-сторінки

14. Виберіть в конструкторі закладку для форми **Form1**. Двічі клацніть другий елемент управління **linkLabel2** для створення обробника події **linkLabel2_LinkClicked**. Додайте наступний код:

```
System.Diagnostics.Process.Start("www.limtu.com");
linkLabel2.LinkVisited = true;
```

15. Побудуйте і запустіть застосування.
 16. Клацніть елемент управління **linkLabel - Open Form**, щоб відкрити форму. Введіть відповідну інформацію в поля введення і перевірте кнопки **Accept** і **Cancel**.
 17. Клацніть елемент управління **linkLabe2 - Microsoft**, щоб відкрити сайт, наприклад, факультету.

Вправа 7. Елементи з підтримкою редагування тексту (2 бали)

TextBox - це основний елемент управління, за допомогою якого можна приймати текст, який вводиться користувачем, а також відображати текст для користувача. Існує можливість створювати як поля, які відображають багаторядковий текст, так і поля, які відображають знак пароля замість реально введеного тексту.

Елемент управління **MaskedTextBox** - це модифікований елемент управління **TextBox**, який дозволяє задавати заздалегідь встановлений шаблон для прийняття введення користувача або відмови від нього. За допомогою властивості **Mask** можна вказати обов'язкові або необов'язкові символи або тип символів (букви або цифри), які вводяться і застосувати форматування для відображення рядків.

1. Відкрийте виконане вами в попередній вправі рішення **WinLinkLabel**.
2. Відобразіть конструктор для форми **Form2**.
3. Додайте елемент управління **TextBox** на форму під розташованими раніше елементами. Перетягніть елемент управління **Label** у форму і розмістіть зліва від цього елемента.
4. Задайте властивості **Text** елемента управління **Label** значення **Address**.
5. Для елемента управління **TextBox** задайте наступні властивості:

| Властивості | Значення | Коментар |
|-------------|----------|---------------------------------------|
| Multiline | True | багаторядковий |
| WordWrap | False | перехід слова з одного рядка на інший |

| | | |
|------------|------|---------------------------------|
| ScrollBars | Both | відображення смуг прокручування |
|------------|------|---------------------------------|

- Змініть розміри елемента управління **TextBox** так, щоб в ньому поміщалася адреса. При потребі збільште розміри форми і перемістіть кнопки **Accept** і **Cancel**.
- Перетягніть елементи управління **MaskedTextBox** і **Label** з **Toolbox** на форму і розмістіть їх під раніше введені елементи.
- Властивості **Text** елемента управління **Label** задайте значення **Phone Number**.
- Задайте значення **(999)-000-0000** властивості **Mask** елемента управління **MaskedTextBox**.
- Задайте значення **Internal** властивості **Modifiers** для останніх елементів управління **TextBox** і **MaskedTextBox**.
- Відкрийте вікно коду форми **Form1**.
- В обробнику події **linkLabel1_LinkClicked** додайте у блок **if**, розташований під кодом, який ви додали в попередній вправі, наступний код:

```
MessageBox.Show("Your address is " + aForm.textBox3.Text);
MessageBox.Show("Your phone number is " + aForm.maskedTextBox1.Text);
```

- Побудуйте і запустіть застосування. Введіть в текстове поле свій телефон. Перевірте, що номер відображається згідно з потрібним форматом.

Вправа 8. Додавання і видалення елементів управління в режимі роботи застосування (2 бали)

При розміщенні на формі елемента управління в режимі дизайну, середовище створює код, який описує цей елемент. Якщо призначити в обробнику заданого елемента управління генерацію аналогічного коду, то в запущеному застосуванні можна буде додавати на форму або видаляти елементи, активізуючи цей обробник.

Для роботи з елементами управління використовується об'єкт **ControlsCollection**, який містить ряд методів, основні з яких будуть використані в цій вправі.

- Створіть нове застосування і назвіть його **RegistrationForm**.
- Додайте на форму елемент **GroupBox**, три написи, два текстові поля, кнопку, елемент **CheckBox**.
- Встановіть наступні значення властивостей форми та елементів управління:

| Об'єкт | Властивість | Значення |
|----------|-----------------|-------------------------|
| Form1 | FormBorderStyle | Fixed3D |
| | Text | Реєстрація |
| | Size | 294; 269 |
| label1 | Location | 30; 20 |
| | Text | Виберіть тип реєстрації |
| label2 | Location | 30; 43 |
| | Text | Name |
| label3 | Location | 30; 73 |
| | Text | PIN |
| button1 | Location | 187; 191 |
| | Text | Реєстрація |
| textBox1 | Location | 80; 40 |
| | Text | |
| | Size | 184; 22 |
| textBox2 | Location | 80; 70 |

| | | |
|-----------|----------|---------------------------|
| | Size | 184; 20 |
| | Text | |
| checkBox1 | Location | 40; 109 |
| | Size | 80; 17 |
| | Text | Розширені можливості |
| groupBox1 | Text | Введіть реєстраційні дані |
| | Location | 0; 0 |
| | Size | 284; 237 |

4. Для реалізації можливості додавання і видалення елементів у процесі виконання програми реалізуйте обробник події **CheckedChanged**: клацніть двічі на елементі **checkBox1** і додайте наступний код:

```

if (checkBox1.Checked == true)
{
    Label lbl = new Label();
    lbl.Location = new System.Drawing.Point(30, 160);
    lbl.Size = new System.Drawing.Size(32, 23);
    lbl.Name = "label1";
    lbl.TabIndex = 2;
    lbl.Text = "PIN2";
    groupBox1.Controls.Add(lbl);
    TextBox txt = new TextBox();
    txt.Location = new System.Drawing.Point(80, 160);
    txt.Size = new System.Drawing.Size(184, 20);
    txt.Name = "textBox1";
    txt.TabIndex = 1;
    txt.Text = "";
    groupBox1.Controls.Add(txt);
}
else
{
}

```

5. Відкомпілюйте і запустіть застосування. Перевірте, що при встановленні прапорця в елементі управління **checkBox** *"Розширені можливості"* на формі з'являється напис і поле введення для додаткових даних.
6. Для видалення елементів управління з форми можуть застосовуватися методи: **Clear** (видалення всіх елементів з колекції), **Remove** (видалення елемента з колекції) і **RemoveAt** (видалення елемента за заданим індексом). У тіло оператора **else** додайте код для видалення елементів управління за індексом:

```

int lcv;
lcv = groupBox1.Controls.Count; // визначається кількість
while (lcv > 4)
{
    groupBox1.Controls.RemoveAt(lcv - 1);
    lcv -= 1;
}

```

- Запустіть застосування. Переконайтеся, що при включенні *"Розширені можливості"* додаткові елементи з'являються на формі, а при виключенні - зникають.

Вправа 9. Перевірка значень при введенні. Події KeyPress і Validating. Елемент управління ErrorProvider (3 бали)

При внесенні значень параметрів користувачем у багатьох випадках вимагається перевіряти текст, який вводиться, за заданими критеріями. Наприклад, реєстраційний номер, телефон не повинні містити букв, поле імені - цифр. У цій вправі розглядаються реалізації перевірок, які можна здійснювати, використовуючи вбудовані події текстового поля.

Використання події KeyPress

- Відкрийте застосування **RegistrationForm**.
- Виділіть по черзі текстові поля **TextBox1** і **TextBox2**, у вікні **Properties** створіть обробники події **KeyPress**, яка виникає при натисненні будь-якої клавіші в полі.
- У тіло обробника події **KeyPress** для текстового поля **TextBox1** введіть наступний код (для елемента **TextBox1** неприпустимими значеннями будуть цифри):

```
if (char.IsDigit(e.KeyChar))
{
    e.Handled = true;
    MessageBox.Show("Поле Name не може містити цифри");
}
```

- Для елемента **TextBox2**, навпаки, неприпустимими значеннями будуть букви, в обробнику події **KeyPress** для текстового поля **TextBox2** вкажіть код:

```
if (!char.IsDigit(e.KeyChar))
{
    e.Handled = true;
    MessageBox.Show("Поле PIN не може містити букви");
}
```

- Відкомпілюйте і запустіть застосування. Спробуйте ввести в поле **Name** цифри, в поле **PIN** - букви.
- Для захисту текстового поля, яке з'являється при встановленні галочки в чекбоксі *"Розширені можливості"*, потрібно вручну визначити подію **KeyPress**. В обробнику події **CheckedChanged** для елемента **CheckBox1** вкажіть код:

```
txt.KeyPress+= new System.Windows.Forms.KeyPressEventHandler(this.textBox2_KeyPress);
```

- Запустіть і протестуйте застосування.

Використання події Validating

Подія **KeyPress** блокує частину клавіатури. Іншим способом перевірки є подія **Validating**, яка дозволяє працювати з клавіатурою, блокуючи при цьому інші дії користувача.

- Закоментуйте обробник елемента **TextBox2**.
- У режимі дизайну форми у вікні **Properties** елемента **TextBox2** створіть обробник події **Validating** і запишіть наступний код:

```
if(textBox2.Text == "")
{
```

```

        e.Cancel=false;
    }
    else
    {
        try
        {
            double.Parse(textBox2.Text);
            e.Cancel = false;
        }
        catch
        {
            e.Cancel = true;
            MessageBox.Show("Поле PIN не може містити букви");
        }
    }
}

```

10. Запустіть застосування. При перемиканні фокуса введення або натисненні на кнопку **"реєстрація"** відбувається подія **Validating**.

Використання елемента управління **ErrorProvider**.

Елемент управління **ErrorProvider** зручно застосовувати, коли треба виводити невелику іконку у разі помилки введення.

11. У режимі дизайну з вікна **ToolBox** перенесіть на форму елемент управління **ErrorProvider**.
12. У коді форми в обробнику **textBox1_KeyPress** додайте наступний рядок:

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

13. Запустіть застосування. При помилці введення з'являється миготлива іконка сповіщення, при наведенні на неї спливає, пояснююче повідомлення про помилку.