Лабораторна робота № 27

Тема. Розширення можливостей візуальних елементів керування за рахунок наслідування їх класів. Контейнери елементів керування.

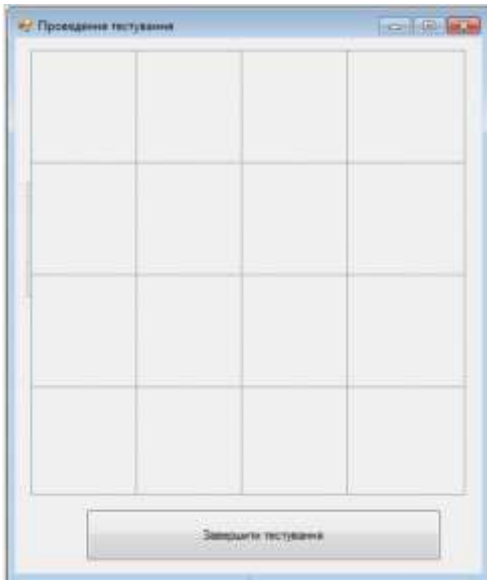
Мета. Формування вмінь і навиків створення та використання наслідуваних класів, контейнерів елементів керування. Закріплення вмінь і навиків використання властивостей та методів об’єктів. Застосування вмінь і навиків програмування алгоритмів лінійної, розгалуженої та циклічної структури.

Контрольні запитання

1. Для чого використовується наслідування візуальних елементів керування?
2. Як звернутися до полів, властивостей чи методів об’єкта, який викликав процедуру обробки події?
3. Що таке контейнери? Які різновиди контейнерів використовуються в Visual Studio?
4. Які два основні методи підтримують контейнери?
5. Як визначається положення елементів керування, доданих в контейнер?
6. Як забезпечити пропорційне масштабування розмірів елементів керування, розміщених в контейнері, при зміні розміру самого контейнера? Які дві властивості для цього використовуються? У чому різниця між ними?

Завдання

1. Ознайомтеся з третім розділом навчального курсу по створенню Windows Forms на сайті https://metanit.com/, починаючи, з сторінки https://metanit.com/sharp/windowsforms/3.1.php.
2. Для програмної організації форми тестування створіть спочатку проект Windows Form та дайте йому назву, яка б містила ваше прізвище латинськими літерами та номер лабораторної без пробілів.
3. Переробіть форму, створену автоматично, за зразком, наведеним нижче. **В середині форми вставте контейнер класу *TableLayoutPanel* та дайте йому назву *tableControl*** (таблиця елементів керування). Використовуючи контекстне меню контейнера, задайте для нього чотири рядки та чотири стовпці елементів однакового масштабованого розміру. За допомогою властивості контейнера *Anchor* забезпечте сталі його відступи від всіх країв в процесі масштабування. Забезпечте початкове відображення цієї форми по центру екрана.



4. Для ідентифікації кожної кнопки, яка має програмно вставлятися у створений контейнер, опишіть в модулі форми **під класом форми** клас *buttonTest*, породжений від класу *Button*, передбачивши у ньому публічне числове поле *index*.
5. В класі форми створіть універсальну процедуру обробки натиснення кнопок, передбачивши в ній видачу повідомлення з номером кнопки та видалення кнопки з контейнера.
6. В процедурі обробки події завантаження форми забезпечте створення кнопки класу *buttonTest* для кожної комірки контейнера, встановіть для кожної з них унікальний індекс, надпис з аналогічним номером питання, прив’язку розмірів до комірок контейнера та додавання в контейнер. Після цього код класу форми буде приблизно таким:

```
public partial class Form1 : Form
 {  public Form1()
     {InitializeComponent();
     }

    private void Form1_Load(object sender, EventArgs e)
     {int nomer=1;
      for (int i=0; i<tableControl.RowCount; i++)
          for (int j = 0; j < tableControl.ColumnCount; j++)
          {   buttonTest pb = new buttonTest();
              pb.index = nomer;
              pb.Text = "Питання № "+(nomer++).ToString();
              pb.Dock = System.Windows.Forms.DockStyle.Fill;
              pb.Click += new System.EventHandler(this.buttonTest_Click);
              tableControl.Controls.Add(pb);
          }
    }

    private void buttonTest_Click(object sender, EventArgs e)
     {MessageBox.Show("Натискається кнопка № " + ((buttonTest)sender).index);
      tableControl.Controls.Remove((Button)sender);
     }
 }

public class buttonTest : Button
 {public int index;
 }
```

7. Самостійно відредагуйте універсальну процедуру обробки натиснення кнопок *buttonTest_Click* так, щоб після натиснення кожної з них задавалося тестове питання залежно від індекса кнопки, а залежно від правильності відповіді додавалися бали (ваші тестові питання мають відрізнятися від питань одногрупників). В процедурі обробки події натиснення кнопки завершення роботи передбачте спочатку вивід набраних балів, а потім – виклик методу *close();* для забезпечення завершення роботи програми.