
МЕТОДИЧНА РОЗРОБКА ЗАНЯТТЯ

Дисципліна: ІНСТРУМЕНТАЛЬНІ ЗАСОБИ
ВІЗУАЛЬНОГО ПРОГРАМУВАННЯ

Група: КН-31

Дата: 18.11.2021 р.



Windows Forms

**ТЕМА: РОЗРОБКА ЗАСТОСУНКІВ НА БАЗІ ПЛАТФОРМИ .NET. ВВЕДЕННЯ /
ВИВЕДЕННЯ ДАНИХ З ВИКОРИСТАННЯМ ТЕХНОЛОГІЇ WINDOWS FORMS ТА
МОВИ C#**

МЕТА ЗАНЯТТЯ:

ФОРМУВАННЯ ПРАКТИЧНИХ НАВИКІВ створення проєктів Windows Forms, конструювання інтерфейсу програми з допомогою візуального конструктора та елементів керування, налаштування властивостей елементів керування, використання методів і програмування обробників подій для організації введення та виведення даних засобами мови програмування C#.

Тип заняття: формування практичних умінь та навичок.

Форма заняття: практичне.

Базові поняття й терміни: об'єктно-орієнтоване програмування, бібліотека класів .NET, форма, елемент управління, програмний код, властивість, метод, подія, обробник події.

Технології та методи: мультимедійні, інформаційно-комунікаційні, інтелект-карта, використання робочого зошита студента, робота в групах, «шпаргалка», словесні методи, мозковий штурм, демонстрація та інші.

Обладнання: мультимедійний проєктор, екран, мультимедійні презентації, відео, робочий зошит, інтелект-карта та інші дидактичні матеріали на платформі Moodle.gi.edu.ua, інтерактивні вправи на сайтах LaerningApps.org, Mindmeister.com, Mentimeter.com.

Міждисциплінарні зв'язки:

Дисципліни, які забезпечують: Алгоритмізація та програмування, Об'єктно-орієнтоване програмування, Інформатика.

Дисципліна, яка забезпечується Технології створення програмних продуктів, ОБДЗ.

СТРУКТУРА ЗАНЯТТЯ

- I. Організаційний етап.
- II. Актуалізація опорних знань.
- III. Формування практичних навичок.
Виконання практичних завдань:
 1. Виконаних завдань групою з допомогою викладача.
 2. Самостійне виконання завдань.
- IV. Підбиття підсумків заняття, рефлексія.
- V. Домашнє завдання.

Хід заняття

ДІЯЛЬНІСТЬ ВИКЛАДАЧА

ДІЯЛЬНІСТЬ СТУДЕНТІВ

І. ОРГАНІЗАЦІЙНИЙ ЕТАП

Привітання. Виявлення відсутніх студентів.

Привітання. Оголошення старостою відсутніх студентів.

ІІ. МОТИВАЦІЯ НАВЧАЛЬНОЇ ДІЯЛЬНОСТІ СТУДЕНТІВ

➔ Методи та прийоми навчання

Розповідь, пояснення, мозковий штурм, інтерактивна презентація, демонстрація, виконання інтерактивних вправ

Щоденно для вирішення практичних завдань на ПК ми використовуємо різноманітне прикладне програмне забезпечення, так звані «настільні» або «декстопні» програми – текстові та табличні редактори, графічні додатки, програми для перегляду фотографій, файлові менеджери, калькулятор, браузер тощо.

А якщо подивиться на це програмне забезпечення не з точки зору кінцевого користувача, а з точки зору програміста, розробника програмного забезпечення? На яких мовах програмування, з використанням яких технологій, платформ, бібліотек класів написані популярні Windows-застосунки?

Огляд ПЗ, встановленого на комп'ютерах

Файловий менеджер, текстовий редактор, калькулятор... – проаналізуємо ці програми з точки зору програміста, **що нам потрібно, щоб створити таку програму?**

Спробуємо розробити власний застосунок із використанням вивчених на попередніх заняттях технологій та інструментів.

Студенти розглядають програми, аналізують їх та дають відповідь на запитання.

Очікувані відповіді:

Потрібно знати мови програмування / вміти програмувати.

Потрібно мати середовище програмування.

Потрібно знати, яка технологія використовується для створення Windows-додатків.

Потрібно знати бібліотеку класів для створення візуальних додатків

ОГЛОШЕННЯ ТЕМИ ТА МЕТИ ЗАНЯТТЯ

➔ ФОРМУЛЮВАННЯ ЦІЛЕЙ ЗАНЯТТЯ ТА РЕЗУЛЬТАТІВ НАВЧАННЯ

Цілі заняття

- сформулювати загальний алгоритм створення візуального застосунку;
- навчитись використовувати бібліотеку класів .NET для створення візуальних додатків;
- спроектувати інтерфейс Windows-застосунку з використанням елементів керування;
- налаштувати властивості елементів керування, застосувати методи та реалізувати

обробники подій для введення / виведення даних;

- перевірити коректність роботи створеного застосунку, навчитися аналізувати та усувати недоліки.

РЕЗУЛЬТАТИ НАВЧАННЯ:

- знати алгоритм створення візуальних застосунків;
- вміти створювати об'єкти на основі базових класів візуальних компонентів;
- аналізувати необхідність використання тих чи інших компонентів для створення візуального застосунку;
- вміти налаштовувати властивості візуальних компонентів програми, використовувати методи для опрацювання даних програми;
- вміти створювати обробники подій візуальних компонентів для обчислень та введення / виведення даних;
- вміти виконувати тестування програми;
- аналізувати помилки та знаходити оптимальні методи їх вирішення.

Виконання ІНТЕРАКТИВНОЇ ВПРАВИ «ЩО ПОТРІБНО ЗНАТИ / ВМІТИ, ЩОБ ДОСЯГНУТИ ЦІЛЕЙ ЗАНЯТТЯ?»

За QR-кодом або посиланням на Moodle студенти відкривають інтерактивну вправу (сервіс Mentimeter) та форму хмаринку слів, які на їхню думку знання та навик потрібні для досягнення мети і цілей заняття.

III. АКТУАЛІЗАЦІЯ ОПОРНИХ ЗНАЬ

➔ Методи та прийоми навчання

ПОЯСНЕННЯ, ДЕМОНСТРУВАННЯ, ІНТЕРАКТИВНА ПРЕЗЕНТАЦІЯ, РІШЕННЯ СИТУАЦІЙНИХ ЗАВДАНЬ, МОЗКОВИЙ ШТУРМ, ФРОНТАЛЬНА БЕСІДА, ВИКОНАННЯ ІНТЕРАКТИВНИХ ВПРАВ.

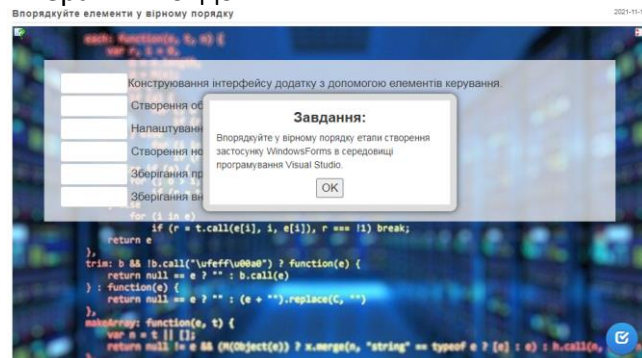
Для того, щоб перейти до виконання практичних завдань, необхідно пригадати та систематизувати знання, необхідні для їх виконання.

1. ЯКИЙ ЗАГАЛЬНИЙ АЛГОРИТМ СТВОРЕННЯ ВІЗУАЛЬНОГО ЗАСТОСУНКУ В СЕРЕДОВИЩІ ПРОГРАМУВАННЯ VISUAL STUDIO ?

Завдання: Вкажіть правильний порядок етапів створення візуального застосунку в середовищі програмування Visual Studio.
(виконання інтерактивної вправи 1)

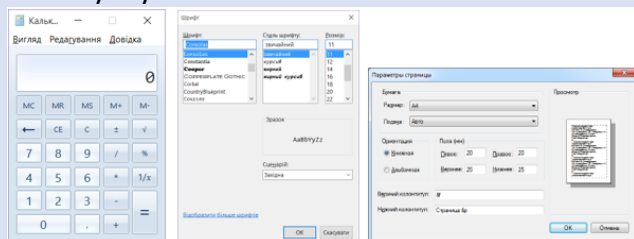
Виконання ІНТЕРАКТИВНОЇ ВПРАВИ НА MOODLE

Студенти завантажують Moodle на мобільних пристроях або ПК, переходять до інтерактивної вправи 1 і виконують її спільно групою, один із студентів працює біля інтерактивної дошки.



2. З яких складових чи компонентів складається інтерфейс типового вікна Windows-застосунку?

Задання: Перелічіть основні елементи графічного інтерфейсу даного вікна Windows-застосунку.



Студенти розглядають запропоновані вікна Windows-додатків, діляться на команди, аналізують перелік елементів інтерфейсу. Представник кожної групи оголошує результати.

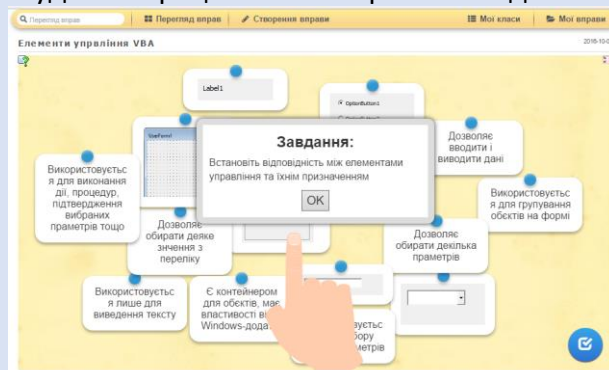
Отже, для конструювання візуального інтерфейсу додатку використовуються візуальні компоненти, які розміщуються на «заготовці» вікна – формі. Пригадаємо назви та призначення основних візуальних компонентів, які використовуються для проектування інтерфейсу типового Windows-застосунку.

3. Яке призначення мають візуальні компоненти?

Завдання: З'єднайте зображення та призначення елементів управління (виконання інтерактивної вправи 2 на Moodle)

Виконання інтерактивної вправи на Moodle

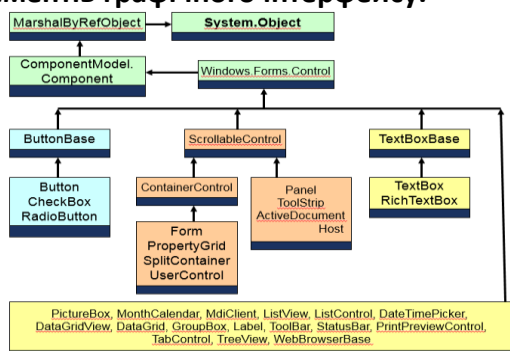
Студенти завантажують Moodle на мобільних пристроях або ПК, переходять до інтерактивної вправи, виконують її, один із студентів працює біля інтерактивної дошки.



4. Пригадайте класи елементів керування

Для розробки застосунків WindowsForms використовується ієрархія класів .Net.

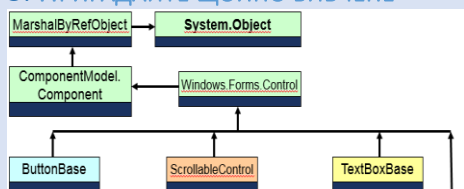
✓ Базовим класом для елементів керування є клас **Control** – компонента з візуальним представленням. Пригадайте загальні риси для всіх класів, що відносяться до елементів графічного інтерфейсу.



Студенти усно називають спільні риси елементів класу. Очікувані відповіді:

- Зовнішній вигляд
- Взаємодія
- Два режими роботи – режим розробки (Design mode) та режим роботи програми (Run-Time mode)

5. ПРИГАДАЙТЕ ЩОЙНО ВИВЧЕНЕ



Назвіть перелік класів-нащадків елементів керування відповідно базових категорій класів в ієрархії.

Студенти усно пригадують, які класи-нащадки належать до відповідних категорій в ієрархії класів .Net.

6. ЯКІ ХАРАКТЕРИСТИКИ МАЮТЬ ОБ'ЄКТИ?

Властивості – певні параметри об'єкта, які визначають його стан, зовнішній вигляд тощо. Наприклад, діалогове вікно може бути видимим чи невидимим в даний момент на екрані. За це відповідає властивість Visible.

Для встановлення значення властивості існує два методи:

Програмно	3 використанням вікна властивостей Properties
Необхідно ввести ім'я об'єкта, поставити крапку і за нею – ім'я властивості. Дальше ставиться знак дорівнює і значення властивості. Наприклад: <pre>Form1.ForeColor = blue;</pre>	

Крім властивостей, у об'єктів є **методи**, тобто команди, які можна застосувати до об'єкта. Наприклад, у форми є метод Close(), який закриває вікно форми.

Подія являє собою певну дію, на яку реагує об'єкт (наприклад, клацання мишею чи натискання на клавішу), для котрої можна запрограмувати відповідь. Події виникають в результаті дій користувача, програми, або вони можуть бути викликані системою.

Наприклад, процедура натискання на кнопку:

```
private void button1_Click (object sender, EventArgs e)
{ string s = textBox1.Text; }
```

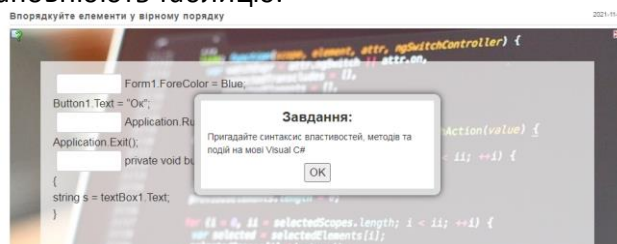
7. ПРИГАДАЙТЕ ЩОЙНО ВИВЧЕНЕ / СИНТАКСИС ВЛАСТИВОСТЕЙ, МЕТОДІВ ТА ОБРОБНИКІВ ПОДІЙ НА МОВІ VISUAL C#

Завдання: подано приклади використання властивостей, методів та подій, потрібно визначити який із записів відноситься до котрої з характеристик об'єкта.

(виконання інтерактивної вправи 3)

Виконання інтерактивної вправи на MOODLE

Студенти завантажують Moodle, переходять до інтерактивної вправи 4 і виконують її спільно групою, один із студентів працює біля інтерактивної дошки. В робочих зошитах заповнюють таблицю.



8. ВСТАНОВІТЬ ХАРАКТЕРИСТИКИ ОБ'ЄКТІВ ВІДПОВІДНО ДО НАВЕДЕНИХ ПРИКЛАДІВ (ВЛАСТИВІСТЬ / МЕТОД / ОБРОБНИК ПОДІЇ)

НА ІНТЕРАКТИВНІЙ ДОШЦІ З'ЯВЛЯЮТЬСЯ ПРИКЛАДИ ХАРАКТЕРИСТИК, ЯКІ ПОТРІБНО КЛАСИФІКУВАТИ

Узагальнення

Отже, ми пригадали технологію роботи в середовищі візуального програмування, базові класи, які необхідні для створення Windows-застосунку, загальний алгоритм створення візуальних додатків та набір необхідних елементів для проектування інтерфейсу програми.

Тепер можемо переходити до практичної частини заняття, тобто розробки візуального застосунку в середовищі програмування.

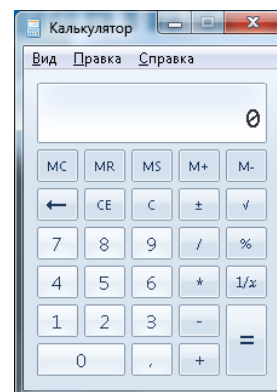
IV. ФОРМУВАННЯ ПРАКТИЧНИХ НАВИКІВ

Методи та прийоми навчання

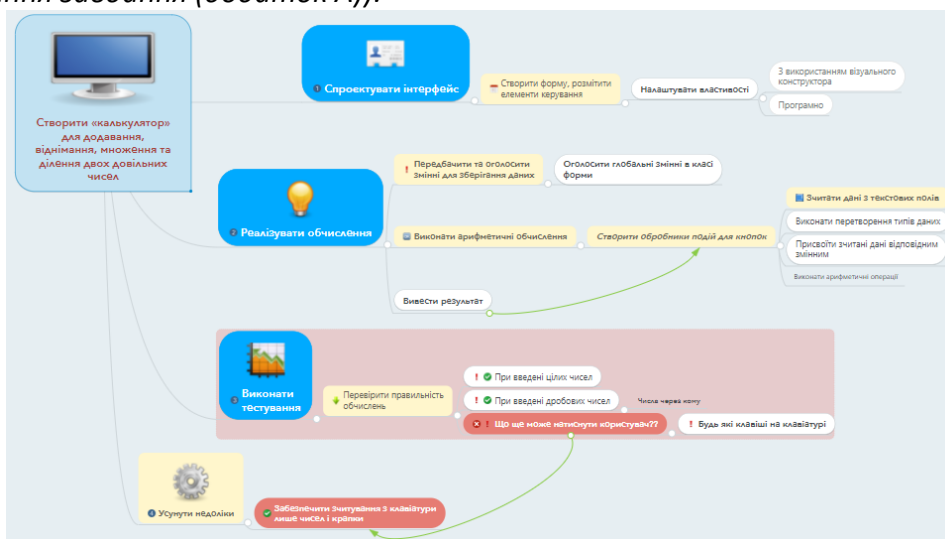
ПОЯСНЕННЯ, ДЕМОНСТРУВАННЯ, ВИКОНАННЯ ПРАКТИЧНИХ ЗАВДАНЬ, МОЗКОВИЙ ШТУРМ, АНАЛІЗ ПРАКТИЧНИХ СИТУАЦІЙ, ВІДПРАЦЮВАННЯ НАВИЧОК

Постановка завдання

Створити «калькулятор» для додавання, віднімання, множення та ділення довільних чисел (інтерфейс спроектувати схожим до «стандартного» калькулятора Windows).



Сформуємо алгоритм виконання завдання. Необхідно виділити основні етапи роботи над завданням та деталізувати їх (робота з картою знань на Mindmeister.com та опорною схемою виконання завдання (додаток А)):



- 1) **Спроекувати інтерфейс додатку:** створити форму, розмістити елементи керування, налаштувати властивості:
 - а) з використанням можливостей візуального конструктора;
 - б) програмно;
- 2) **Реалізувати обчислення – створити обробники подій** для виконання дій додавання, віднімання, множення та ділення довільних чисел:
 - забезпечити зчитування необхідних даних:
 - зчитати дані з елементів керування, присвоїти значення деяким змінним;
 - виконати перетворення типів даних з використанням відповідних функцій;
 - виконати арифметичні обчислення.
 - забезпечити виведення даних у зручному вигляді.
- 3) **Виконати перевірку коректності роботи створеного застосунку:**
 - перевірка правильності виконання обчислень;
 - перевірка коректності вхідних даних.
- 4) **Усунути недоліки.**

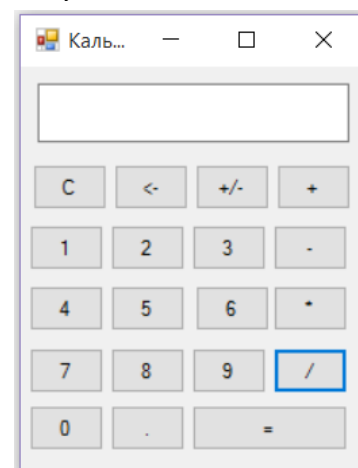
1) ПРОЄКТУВАННЯ ІНТЕРФЕЙСУ, НАЛАШТУВАННЯ ВЛАСТИВОСТЕЙ ЕЛЕМЕНТІВ КЕРУВАННЯ

Проаналізуємо завдання: перший пункт стосується проектування інтерфейсу користувача. Отже, нам потрібно створити новий проект Windows Worms у середовищі програмування Visual Studio та розмістити на формі елементи керування.

Які елементи керування потрібно використати, щоб користувач мав можливість вводити числа, виконувати обчислення та побачити результат?

Насправді варіантів проектування інтерфейсу такого додатку може бути декілька. Спробуємо спроектувати інтерфейс, який забезпечить виконання базових арифметичних операцій.

!Студенти можуть пропонувати інші варіанти розміщення елементів інтерфейсу!



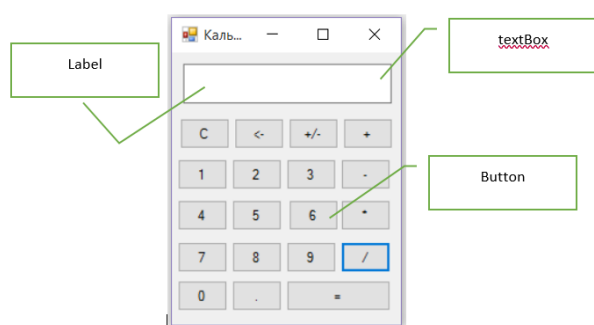
Розміщувати елементи керування на форму можна двома способами: з допомогою візуального конструктора, або програмно – з допомогою написання програмного коду.

Якщо використовувати візуальний конструктор, то алгоритм роботи такий:

- 1) Обираємо потрібний елемент на панелі елементів керування
- 2) поміщаємо його на форму
- 3) налаштовуємо необхідні властивості у вікні властивостей

Якщо робити це все програмно, то потрібно:

- 1) прописати код для створення нових елементів на основі базових класів (у функції *Form_Load()* – *завантаження форми*, яка виконується першою при запуску проєкту),
- 2) задати їм властивості
- 3) додати ці елементи в колекцію елементів керування форми



```
//Приклад створення кнопки в функції
Form_Load()
//1) створюємо кнопку
Button b = new Button();
//2) задаємо властивості
//позиція на екрані
b.Top = 50;
b.Left = 60;
//текст кнопки
b.Text = "1";
//колір кнопки
b.BackColor = Color.FromArgb(200, 12, 15);
//розмір
b.Size = new System.Drawing.Size(45, 45);
// шрифт -> гарнітура - розмір - накреслення
b.Font = new Font("Arial", 14,
FontStyle.Regular);
//3) додаємо на форму кнопку
this.Controls.Add(b);
```

2) РЕАЛІЗАЦІЯ ОБЧИСЛЕНЬ – СТВОРЕННЯ ОБРОБНИКІВ ПОДІЙ

Перш за все, необхідно оголосити змінні, які будуть використовуватись в програмі для зберігання даних.

А) Код для оголошення змінних прописуємо в класі форми, щоб забезпечити доступ до них з усіх функцій.

Студенти відкривають програмний код форми (файл Form1.cs)

Приклад:

```
//оголошення змінних
float a = 0; //перше число
float b = 0; //друге число
float op = 0; //арифм_операція
```

Б) Створюємо процедури обробки подій натискання на цифрові кнопки та крапку, які повинні відображатися в текстовому полі калькулятора.

Приклад:

```
private void button5_Click(object sender,
EventArgs e)
{//виведення натиснутої цифри в текстове поле
калькулятора
textBox1.Text = textBox1.Text + "1"; }...
```

В) Створюємо процедури обробки подій натискання на кнопки арифметичних операцій.

Приклад:

```
a = float.Parse(textBox1.Text);
textBox1.Clear();
op = 1; //зберігаємо номер арифм_операції
label1.Text = a.ToString() + "+";
...
```

Пригадуємо:

- *функція `float.Parse()` перетворює текст в число з плаваючою комою;
- *функція `Clear()` очищує текстове поле;
- *функція `ToString()` перетворює числові дані в текстові.

Г) Створюємо обробник події натискання на кнопку «дорівнює».	Приклад: <pre>case 1: b = a + float.Parse(textBox1.Text); textBox1.Text = b.ToString(); break; }...</pre>
--	---

ДЕМОНСТРАЦІЯ ВИКОНАНОГО ЗАВДАННЯ Викладач переглядає роботи студентів, аналізує, при потребі допомагає виконати ті чи інші завдання, вказує на недоліки.	<i>Студенти демонструють результати своєї роботи. Задають питання при виникненні проблемних чи незрозумілих ситуацій.</i>
--	---

3. ПЕРЕВІРКА КОРЕКТНОСТІ РОБОТИ СТОРЕНИХ ЗАСТОСУНКІВ

Аналіз типових помилок, які виникають при перевірці коректності роботи застосунків.	<i>Студенти тестують роботу програми, досліджують програмний код, аналізують записані команди. Запитують викладача про незрозумілі моменти.</i>
--	---

АНАЛІЗ ПРОБЛЕМ, ЩО МОЖУТЬ ВИНИКНУТИ ПРИ ТЕСТУВАННІ

- 1) Чи завжди користувачі використовують програмне забезпечення передбачувано?
- 2) Які можуть виникнути проблеми і помилки?
- 3) Хто винен при виникненні помилок – програміст чи користувач?
- 4)

1. Неправильні результати обчислень 2. Не виводяться дані у потрібному форматі 3. Аварійне завершення роботи програми через неправильні дії користувача!	<i>Студенти аналізують можливі проблемні ситуації.</i> ✦ 1 – провина програміста ✦ 2 – провина програміста ✦ 3? – також провина програміста! Чому?
---	--

4. УСУНЕННЯ НЕДОЛІКІВ ТА ПОМИЛОК

ЯК ПРОГРАМІСТ МОЖЕ УБЕЗПЕЧИТИ РОЗРОБЛЮВАНЕ ПЗ ВІД НЕПРАВИЛЬНИХ ДІЙ КОРИСТУВАЧА? АНАЛІЗ МЕТОДІВ УСУНЕННЯ ПОМИЛОК.

- ✓ При введенні даних необхідно їх перевіряти на коректність
- ✓ Унеможливити введення некоректних даних – найкращий варіант

САМОСТІЙНЕ ВИКОНАННЯ ЗАВДАННЯ 1) Реалізувати програмний код для видалення останнього введенного символу (кнопка "<-“ калькулятора). <i>*Властивість Length зберігає кількість символів в текстовому полі.</i>	Приклад: <pre>int lenght = textBox1.Text.Length - 1; string text = textBox1.Text; textBox1.Clear(); for (int i = 0; i < lenght; i++) { textBox1.Text = textBox1.Text + text[i]; }</pre>
2) Реалізувати обробник події для кнопки очищення поля введення калькулятора (кнопка «C»).	

V. ПІДБИТТЯ ПІДСУМКІВ ЗАНЯТТЯ, РЕФЛЕКСІЯ

➔ Методи та прийоми навчання

ФРОНТАЛЬНА БЕСІДА, МОЗКОВИЙ ШТУРМ, ДИСКУСІЯ, ТЕСТУВАННЯ, ВИКОНАННЯ ІНТЕРАКТИВНИХ ВПРАВ

НАВЧАЛЬНИЙ ТЕСТ (ТЕСТ РОЗМІЩЕНО НА MOODLE)

Завдання тестів у додатку Б.

Аналіз типових помилок після виконання тесту

Студенти проходять тестування на сайті Moodle.gi.edu.ua.

ПОВЕРНЕННЯ ДО ЦІЛЕЙ ЗАНЯТТЯ ТА РЕЗУЛЬТАТІВ НАВЧАННЯ

Які цілі було досягнуто, що потрібно доопрацювати на наступних заняттях?

Виконання вправи «Що вивчено на занятті?» (сервіс Mentimeter.com)

VI. ДОМАШНЄ ЗАВДАННЯ

1. Вдосконалити Калькулятор таким чином, щоб була можливість виконувати піднесення до ступеня, знаходження кореня, додати математичні функції (sin, cos, exp ...).
2. Оформити звіт по виконаній роботі.
3. Повторити теоретичні відомості про організацію циклічних обчислень, синтаксис операторів циклу, базові алгоритми опрацювання масивів.