

Факультет систем управління літальних апаратів
Кафедра систем управління літальних апаратів

з дисципліни «Алгоритмізація та програмування»
на тему «Розробка десктоп-застосунків в середовищі Visual Studio»

МЕТА РОБОТИ

Ознайомитися з основами розробки програм з використанням Windows Forms і навчитися розробляти десктоп-застосунки із графічним користувацьким інтерфейсом для введення/виведення даних на мові програмування C++ в середовищі Visual Studio.

ПОСТАНОВКА ЗАДАЧІ

Завдання 1. Вивчити алгоритм створення проекту Windows Forms в середовищі Visual Studio. Ознайомитись з налаштуваннями основних елементів для введення, виведення і управління. *Опрацювати навички створення і налаштування десктоп-застосунку у Visual Studio.

Завдання 2. Для вирішення завдання (табл.1) відповідно до варіанта:

А. Спроекувати і реалізувати в конструкторі форм графічний інтерфейс програми з об'єктами Label, TextBox і Button. *Використати інші елементи управління.

В. Додати програмний код для введення вхідних даних, обчислень і виведення результатів. *Відтестувати і налагодити десктоп-застосунок.

С*. Передбачити зчитування даних з файлу з використанням стандартного діалогу для вибору файлу, а також збереження результатів в файл із відповідним діалогом. Відтестувати і налагодити десктоп-застосунок.

10. Дано тризначне число. Вивести число, отримане при перестановці цифр сотень і десятків вихідного числа (наприклад, 123 перейде в 213).

ВИКОНАННЯ РОБОТИ

Завдання 1.

1. Візуальний дизайн:

За допомогою середовища розробки, такого як Visual Studio, створіть інтерфейс користувача (UI) програми.

Додайте текстове поле (textBox1) для введення користувачем тризначного числа.

Додайте кнопку (button1) для запуску обробки числа.

Додайте текстове поле (textBox5) для відображення результату (оберненого числа).

За бажанням додайте інші елементи UI, такі як мітки, кнопки та поля введення, відповідно до ваших потреб.

Переконайтеся, що елементи UI мають змістовні назви, які чітко відображають їхню функціональність.

2. Реалізація коду:

Створіть обробники подій для кнопок та текстових полів.

button1_Click:

Ця функція обробляє натискання кнопки "Обробити".

Її завдання:

Отримати значення, введене користувачем в textBox1.

Перевірити, чи є введене значення тризначним числом (від 100 до 999).

Якщо ні, відобразити повідомлення про помилку за допомогою MessageBox::Show.

Якщо так, розібрати число на сотні, десятки та одиниці.

Розрахувати обернене число, помінявши місцями цифри сотень, десятків та одиниць.

Відобразити обернене число в textBox5.

Додати обробку винятків для `FormatException` та `Exception`, щоб відобразити інформативні повідомлення про помилки користувачеві.

button2_Click:

Ця функція обробляє натискання кнопки "Відкрити файл".

Її завдання:

Відкрити діалогове вікно `OpenFileDialog`.

Дозволити користувачеві вибрати текстовий файл (".txt").

Якщо файл вибрано, прочитати перший рядок файлу за допомогою `StreamReader` та відобразити його в `textBox1`.

button3_Click:

Ця функція обробляє натискання кнопки "Зберегти результат".

Її завдання:

Відкрити діалогове вікно `SaveFileDialog`.

Дозволити користувачеві вказати ім'я та місце збереження файлу.

Якщо ім'я файлу вказано, записати вміст `textBox5` (обернене число) до вибраного файлу за допомогою `StreamWriter`.

3. Тестування та налагодження:

Ретельно протестуйте програму з різними вхідними даними, щоб переконатися, що вона працює коректно.

Перевірте, чи правильно обробляються тризначні числа.

Перевірте, чи видаються чіткі повідомлення про помилки при введенні невірних даних.

Перевірте, чи правильно зберігаються результати до файлів.

Алгоритм вирішення показано на рис. 1

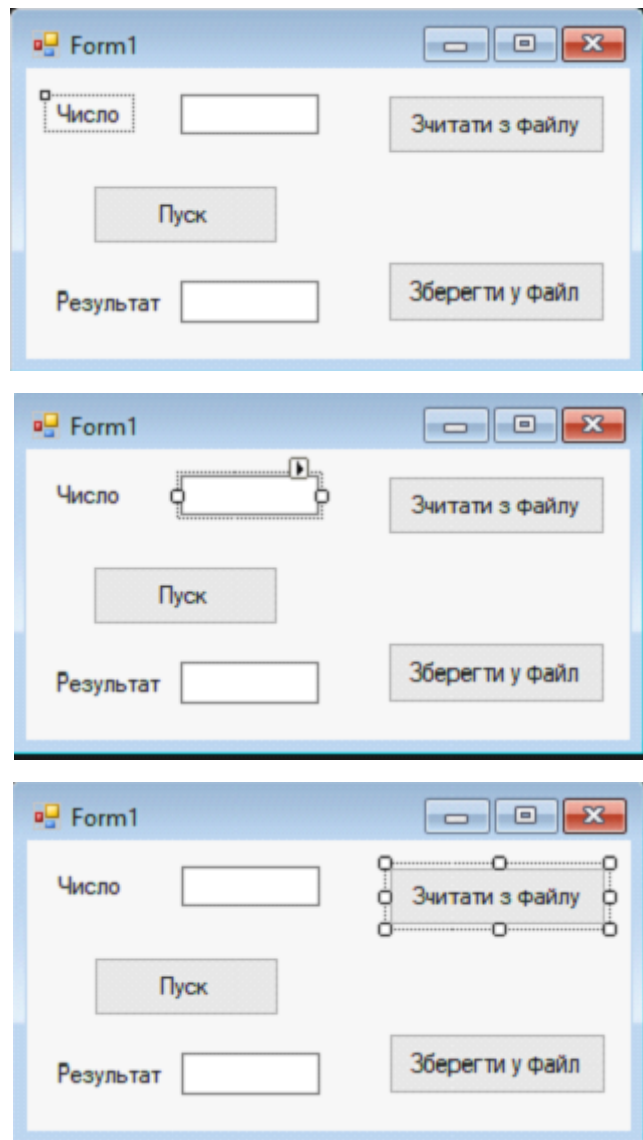


Рисунок 1

Лістинг коду вирішення задачі наведено в дод. А (стор. 9-10).

Завдання 2.

Алгоритм

Спроба зчитати значення з текстового поля та перетворити його на число

Перевірка на тризначність числа

Розбиття числа на цифри

Формування нового числа з перестановкою цифр

Виведення результату в інше текстове поле

Обробка можливих винятків

```
private: System::Void button1_Click(System::Object^ sender, System::EventArgs^ e)
{
    try {
        // Витягти число та обробити можливі помилки парсування
        int число;
        if (!Int32::TryParse(textBox1->Text, число)) {
            MessageBox::Show("Будь ласка, введіть дійсне числове значення.",
"Помилка вводу", MessageBoxButtons::OK, MessageBoxIcon::Error);
            return;
        }

        // Перевірка тризначного числа (необов'язково для чіткості)
        if (число < 100 || число > 999) {
            MessageBox::Show("Будь ласка, введіть тризначне число.", "Помилка
вводу", MessageBoxButtons::OK, MessageBoxIcon::Error);
            return;
        }

        // Ефективно перевернути число за допомогою одного циклу
        int перевернутеЧисло = 0;
        while (число > 0) {
            перевернутеЧисло = (перевернутеЧисло * 10) + (число % 10);
            число /= 10;
        }

        // Відобразити перевернуте число
        textBox5->Text = перевернутеЧисло.ToString();
    } catch (Exception^ ex) {
        MessageBox::Show("Сталася помилка: " + ex->Message, "Помилка",
MessageBoxButtons::OK, MessageBoxIcon::Error);
    }
}
```

Парсування та обробка помилок:

Код починається з спроби розібрати вхідні дані в textBox1 за допомогою Int32::TryParse.

Якщо розбір не вдається, він відображає зрозуміле повідомлення про помилку за допомогою MessageBox::Show та виходить із функції.

Цей підхід уникає потенційних винятків та елегантно обробляє недійсний ввід.

Необов'язкова перевірка тризначного числа (для чіткості):

Закomentовані рядки (`if (число < 100 || число > 999) { ... }`) забезпечують додаткову перевірку на тризначне число.

Ви можете розкоментувати цей розділ, якщо вам спеціально потрібно забезпечити тризначний ввід.

Ефективне перевернення числа:

Цикл `while` повторюється, поки початкове число (`число`) більше 0.

В середині циклу:

перевернутеЧисло обчислюється шляхом множення на 10 (зміщення на одне місце вліво) та додавання останньої цифри початкового числа (`число % 10`).

Потім початкове число ділиться на 10 (видаляється остання цифра) для підготовки до наступної ітерації.

Цей цикл ефективно створює перевернуте число за один прохід.

Відображення результату:

Перевернуте число (`перевернутеЧисло`) перетворюється на рядок за допомогою `ToString` і присвоюється `textBox5->Text` для відображення результату.

Алгоритм вирішення показано на рис. 2

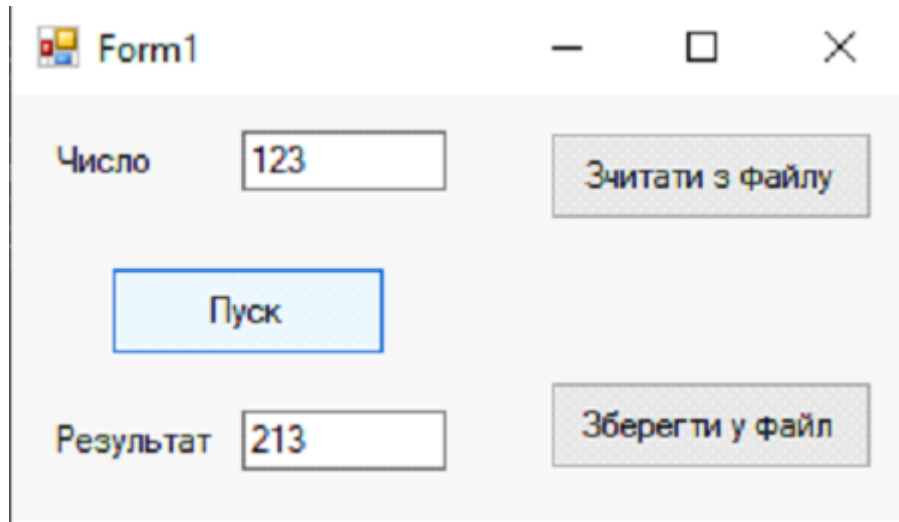


Рисунок 2 – 10

Лістинг коду вирішення задачі 10 наведено в дод. А (стор. 9-10).

Екран роботи програми показаний на рис. Б.2.

ВИСНОВКИ

Ознайомився з основами розробки десктопних програм на C++ з графічним інтерфейсом користувача для введення та виведення даних, використовуючи Windows Forms в середовищі Visual Studio. Вивчив основи Windows Forms: Я навчився створювати форми, додавати елементи керування та обробляти події. Ознайомився з інструментами Visual Studio для розробки Windows Forms: Я дізнався про Toolbox, Designer та Property Window.

ДОДАТОК А

Лістинг коду програми

```

private: System::Void button1_Click(System::Object^ sender, System::EventArgs^
e) {
    try {
        int number = Int32::Parse(textBox1->Text);

        if (number < 100 || number > 999) {
            MessageBox::Show("Please enter a three-digit number.",
"Input Error", MessageBoxButtons::OK, MessageBoxIcon::Error);
            return;
        }

        int hundreds = number / 100;
        int tens = (number / 10) % 10;
        int units = number % 10;

        int newNumber = tens * 100 + hundreds * 10 + units;

        textBox5->Text = newNumber.ToString();
    }
    catch (FormatException^ ex) {
        MessageBox::Show("Please enter a valid numeric value.", "Input
Error", MessageBoxButtons::OK, MessageBoxIcon::Error);
    }
    catch (Exception^ ex) {
        MessageBox::Show("An error occurred: " + ex->Message, "Error",
MessageBoxButtons::OK, MessageBoxIcon::Error);
    }
}

private: System::Void button2_Click(System::Object^ sender,
System::EventArgs^ e) {
    OpenFileDialog^ openFileDialog1 = gcnew OpenFileDialog();
    openFileDialog1->Filter = "Text Files (*.txt)|*.txt|All Files
(*.*)|*.*";
    openFileDialog1->Title = "Select a Text File";
    if (openFileDialog1->ShowDialog() ==
System::Windows::Forms::DialogResult::OK) {
        String^ filename = openFileDialog1->FileName;
        StreamReader^ reader = gcnew StreamReader(filename);
        textBox1->Text = reader->ReadLine();
        reader->Close();
    }
}

private: System::Void button3_Click(System::Object^ sender,
System::EventArgs^ e) {

```

```

SaveFileDialog^ saveFileDialog1 = gcnew SaveFileDialog();
saveFileDialog1->Filter = "Text Files (*.txt)|*.txt|All Files
(*.*)|*.*";
saveFileDialog1->Title = "Save Result to a Text File";

if (saveFileDialog1->ShowDialog() ==
System::Windows::Forms::DialogResult::OK) {
    String^ resultFilename = saveFileDialog1->FileName;
    StreamWriter^ writer = gcnew StreamWriter(resultFilename);
    writer->WriteLine(textBox5->Text);
    writer->Close();
}
}

```

ДОДАТОК Б

Скрін-шоти вікна виконання програми

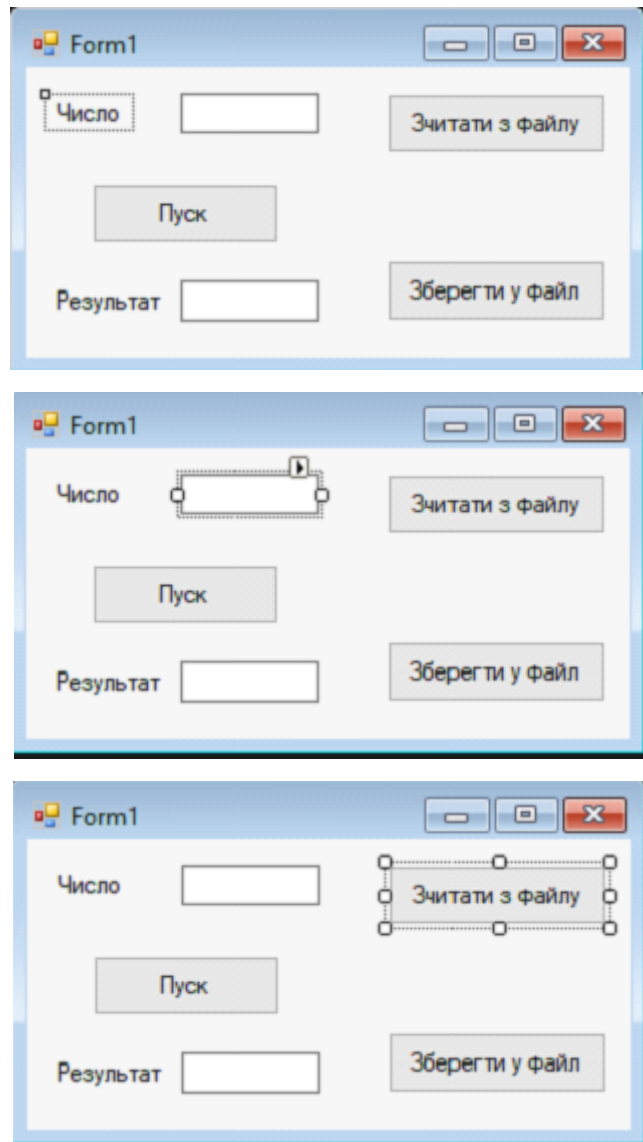


Рисунок Б.1 – Екран виконання програми для вирішення завдання

The screenshot shows a standard Windows application window titled "Form1". Inside the window, there is a user interface with the following elements:

- A label "Число" (Number) followed by a text input box containing the value "123".
- A button labeled "Зчитати з файлу" (Load from file) located to the right of the first input box.
- A button labeled "Пуск" (Start) in the center of the window.
- A label "Результат" (Result) followed by a text input box containing the value "213".
- A button labeled "Зберегти у файл" (Save to file) located to the right of the second input box.

Рисунок Б.2 – Екран виконання програми для вирішення завдання