\_\_\_\_\_\_*Національний університет харчових технологій*\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*Інформаційних систем*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**КУРСОВА РОБОТА**

з дисципліни *Об'єктно орієнтоване програмування*

(назва дисципліни)

на тему *Розробка системи тестування знань: тести з української літератури*

Студент \_**2\_** курсу **КН-2-2** групи

спеціальності *122 “Комп’ютерні науки”*

\_\_\_\_\_\_\_\_*Захарченко А.А.*\_\_\_\_\_\_\_\_\_\_\_\_

Керівник *\_доцент, канд. фіз-мат. наук\_*

\_\_\_\_\_\_\_\_\_\_\_\_*Андріюк О.П*\_\_\_\_\_\_\_\_\_\_\_\_

Національна шкала\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кількість балів: \_\_\_\_\_

Члени комісії

\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_  
 (підпис) (прізвище та ініціали)

\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_  
 (підпис) (прізвище та ініціали)

\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_  
 (підпис) (прізвище та ініціали)

Київ-2022

\_\_\_\_\_\_НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ХАРЧОВИХ ТЕХНОЛОГІЙ\_\_\_\_\_\_  
(назва вищого навчального закладу)

Кафедра \_\_*інформаційних систем*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дисципліна \_\_*Основи програмування та алгоритмічні мови*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Спеціальність \_\_122 «Комп’ютерні науки»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Курс \_\_\_2\_\_\_ Група \_\_2\_\_

**ЗАВДАННЯ  
на курсову роботу студента**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_*Захарченко Альона Анатоліївна*\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
(прізвище, ім'я , по-батькові)

1. Тема роботи Розробка системи тестування знань: тести з української літератури\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
2. Термін здачі студентом закінченої роботи \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
3. Вихідні дані до роботи \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
   \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
   \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
   \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
   \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
   \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
4. Зміст розрахунково-пояснювальної записки (перелік питань, які підлягають розробці)

Розділ 1 «Постанова задачі»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
Розділ 2 «Методи вирішення задачі»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
Розділ 3 «Інструкція користувача системи »\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Перелік графічного матеріалу (з точним зазначенням обов’язкових креслень)  
   1. Блок-схеми алгоритмів\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
   2. Фотокадри інтерфейсу користувача\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
   \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
   \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
   \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
2. Дата видачі завдання \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**АНОТАЦІЯ**

**Захарченко А.А. Створення системи тестування знань з української літератури**

Курсова робота складається з 39 сторінок, 4 таблиць, 15 рисунків та

1 додатку.

В даній курсовій роботі створена система тестування знань з української літератури, що забезпечує оцінювання знань користувачів із сфери української літератури. Під час виконання курсової роботи використовувались матеріали для створення різних питань в сфері української літератури.

Інформаційна система створена у середовищі Visual Studio та з використанням мови C#.

Ключові слова: тест, українська література , система створення тестування , адміністрування

**SUMMARY**

**Zakharchenko A.A. Creation of a ukrainian literature knowledge testing system**

The term paper consists of 39 pages, 4 tables, 15 figures and 1 appendicе.

In this coursework, a system of testing knowledge of ukrainian literature was created, which provides an assessment of one's knowledge in the field of ukrainian literature. During the course work, various issues related to ukrainian literature were created

The information system was created in the Visual Studio environment and using C# language.

Keywords: test, ukrainian literature, system of creating a tests, administration

Зміст

[ВСТУП 6](#_Toc121866722)

[РОЗДІЛ 1. ПОСТАНОВА ЗАДАЧІ НА СТВОРЕННЯ СИСТЕМИ 7](#_Toc121866723)

[РОЗДІЛ 2. МЕТОДИ ВИРІШЕННЯ ЗАДАЧІ 8](#_Toc121866724)

[2.1. Опис основних структур та змінних 8](#_Toc121866725)

[2.2. Блок-схеми реалізації основних алгоритмів 9](#_Toc121866726)

[2.3. Опис програмних одиниць та фактичних/формальних 11](#_Toc121866727)

[параметрів 11](#_Toc121866728)

[РОЗДІЛ 3. ІНСТРУКЦІЯ КОРИСТУВАЧА СИСТЕМИ 13](#_Toc121866729)

[ВИСНОВОК 22](#_Toc121866730)

[ДОДАТОК. ТЕКСТИ ПРОГРАМНИХ ФАЙЛІВ 23](#_Toc121866731)

## ВСТУП

Зараз, в нинішньому часі дуже велика перевага надається дистанційному навчанню . Це відбувається через те, що інтернет зробив можливим доступ до освіти з будь-якої точки світу, в будь-який час та за будь-яких умов.

Переваги дистанційного навчання полягають у тому, що учням не потрібно відвідувати заняття, і вони можуть навчатися у власному темпі та з будь-якого міста чи країни. Їм також не потрібно витрачати гроші на транспорт та інші витрати, пов’язані з відвідуванням занять. Дистанційне навчання також вигідне тим, що створює у студентів відчуття свободи. Це дозволяє їм бути більш прогресивними в своєму навчанні та отримувати менше стресу. Дистанційне навчання також може бути корисним для студентів, які не можуть відвідувати заняття через стан здоров’я чи роботу.

Дистанційне навчання – це процес навчання студента, його повний робочий день без будь-якої необхідності для студента офіційно відвідувати заняття фізично. Багато світових університетів використовують дистанційне навчання з метою пропонування альтернативних методів начання, адже навчальний матеріал особисто не навчить всіх студентів, незалежно від їх наявності та місця розташування.

Мета моєї курсової роботи – створення системи тестування знань з української літератури, що є дуже актуальним, враховуючи поширеність та популярність дистанційного навчання на сьогоднішній день.

C# — об'єктно-орієнтована мова програмування з безпечною системою типізації для платформи .NET.

## РОЗДІЛ 1. ПОСТАНОВА ЗАДАЧІ НА СТВОРЕННЯ СИСТЕМИ

Під час створення даної курсової роботи була поставлена задача розробити інтерфейс системи тестування знань з української літератури, із можливістю вибору, проходження та створення адміністрування тестів , а також збереження результатів тестування.

Для створення системи тестування знань з української літератури на C#, потрібно реалізувати наступні задачі:

* Запис та зчитування питань з файлу.
* Створення файлу з питаннями.
* Меню, для проходження тесту.
* Вивід кінцевого результату.
* Збереження результату тестів
* Меню , для керування програмою
* Адміністрування

Основна структура даних буде містити такі параметри:

* Назва тесту
* Масив питань
* Двомірний масив відповідей
* Тип відповідей
* Масив правильних відповідей

Додатково потрібно реалізувати функцію для визначення кількості та типу варіантів відповідей.

## РОЗДІЛ 2. МЕТОДИ ВИРІШЕННЯ ЗАДАЧІ

### 2.1. Опис основних структур та змінних

Структура **gt** – основна структура програми , використовується для запису , зчитування , формування , усіх тестів:

* Size – числова змінна для запису розміру масиву структур *allT*.
* allT[] - массив структур (*Test*) , для запису тестів.

Структура **Test** – структура програми , використовуєтсья для збереження інформації про тест , та усіх запитань.

* Name – текстова змінна для запису назви тесту.
* Size – числова змінна для запису розміру масиву структур *questions*.
* Banksize – числова змінна , для запису кількості виведених питань у тесті.
* questions[] - массив структур (*Question*) , для запису банку запитань та відповідей.

Структура **Question** – структура програми , використовуєтсья для збереження банку запитань та відповідей.

* qx – текстова змінна для запису назви тесту.
* type – числова змінна для запису типу питання.
* Image – логічна ( булева) змінна для запису використання зображення у тесті.
* imgurl – текстова змінна для запису посилання на зображення.
* q1 – текстова змінна для запису першого запитання.
* q2 – текстова змінна для запису другого запитання.
* q3 – текстова змінна для запису третього запитання.
* q4 – текстова змінна для запису четвертого запитання.
* r1 – текстова змінна для запису першої відповіді.
* r2 – текстова змінна для запису другої відповіді.
* r3 – текстова змінна для запису третьої відповіді.
* r4 – текстова змінна для запису четвертої вдіповіді.

### 2.2. Блок-схеми реалізації основних алгоритмів

Початок

textBox2.Text == "admin" && textBox1.Text == "11111111"

Close();

TeacherCabinet t = new TeacherCabinet();

t.Show();

t.Location = this.Location;

label3.Text = "ПОМИЛКА, СПРОБУЙТЕ ЩЕ РАЗ";

Кінець

Рис 2.1 Блок-схема входу адміна

Початок

int i = 0; i < TestClass.qq.Size; i++

Button btn = sender as Button;

TestClass.btest[i] = i;

TestClass.goS = int.Parse(btn.Name.Remove(0, 1)) - 1;

TestClass.curTest = 0;

TestClass.qq = TestClass.T.allT[TestClass.goS];

Shuffle(TestClass.btest);

TestClass.Answers[TestClass.goS].Mark = 0;

Close();

Array.Resize(ref TestClass.btest, TestClass.qq.Size);

TestGo t = new TestGo();

t.Show();

t.Location = this.Location;

Кінець

Рис 2.2 Блок-схема функції g\_button\_click класу StudentCabinet.

### 2.3. Опис програмних одиниць та формальних параметрів

Структура **Question** відповідає для збереження запитання , та всієї інформації про нього ( варіанти відповіді , правильна відповідь , тип питання)

Таблиця 2.1. Призначення змінних у структурі Question

|  |  |  |
| --- | --- | --- |
| **Змінна** | **Тип** | **Призначення змінної** |
| qx | string | Запитання |
| type | int | Тип питання |
| Image | bool | Для ввімкнення зображення |
| q1 | string | Варіант відповіді №1 |
| q2 | string | Варіант відповіді №2 |
| q3 | string | Варіант відповіді №3 |
| q4 | string | Варіант відповіді №4 |
| r1 | string | Правильна відповідь №1 |
| r2 | string | Правильна відповідь №1 |
| r3 | string | Правильна відповідь №1 |
| r4 | string | Правильна відповідь №1 |

Залежно від типу запитання , у змінні r1,r2,r3,r4 записуються різні данні ( 0 або 1 , при одній та кілька правильних відповідей , текст – при відповідності)

Структура **Test** відповідає для збереження тесту , та всієї інформації про нього ( назва, запитання , розмір банку запитань ,розмір тесту)

Таблиця 2.2. Призначення змінних у структурі Test

|  |  |  |
| --- | --- | --- |
| **Змінна** | **Тип** | **Призначення змінної** |
| Name | string | Назва тесту |
| Size | int | Загальна кількість запитань |
| banksize | int | Розмір банку запитань |
| questions | Question[] | Масив запитань |

Структура **gt** відповідає для збереження усіх тестів.

Таблиця 2.3. Призначення змінних у структурі gt

|  |  |  |
| --- | --- | --- |
| **Змінна** | **Тип** | **Призначення змінної** |
| Size | int | Кількість тестів |
| allT | Test[] | Масив тестів |

Структура **ans** відповідає для збереження кінцевого результату проходження тесту.

Таблиця 2.4. Призначення змінних у структурі ans

|  |  |  |
| --- | --- | --- |
| **Змінна** | **Тип** | **Призначення змінної** |
| AllM | int | Загальна кількість можливих балів |
| Mark | Int | Результат |
| selected | int | Індекс тесту |

## РОЗДІЛ 3. ІНСТРУКЦІЯ КОРИСТУВАЧА СИСТЕМИ

Запустивши створену програму , ми бачимо меню з трьома кнопками.

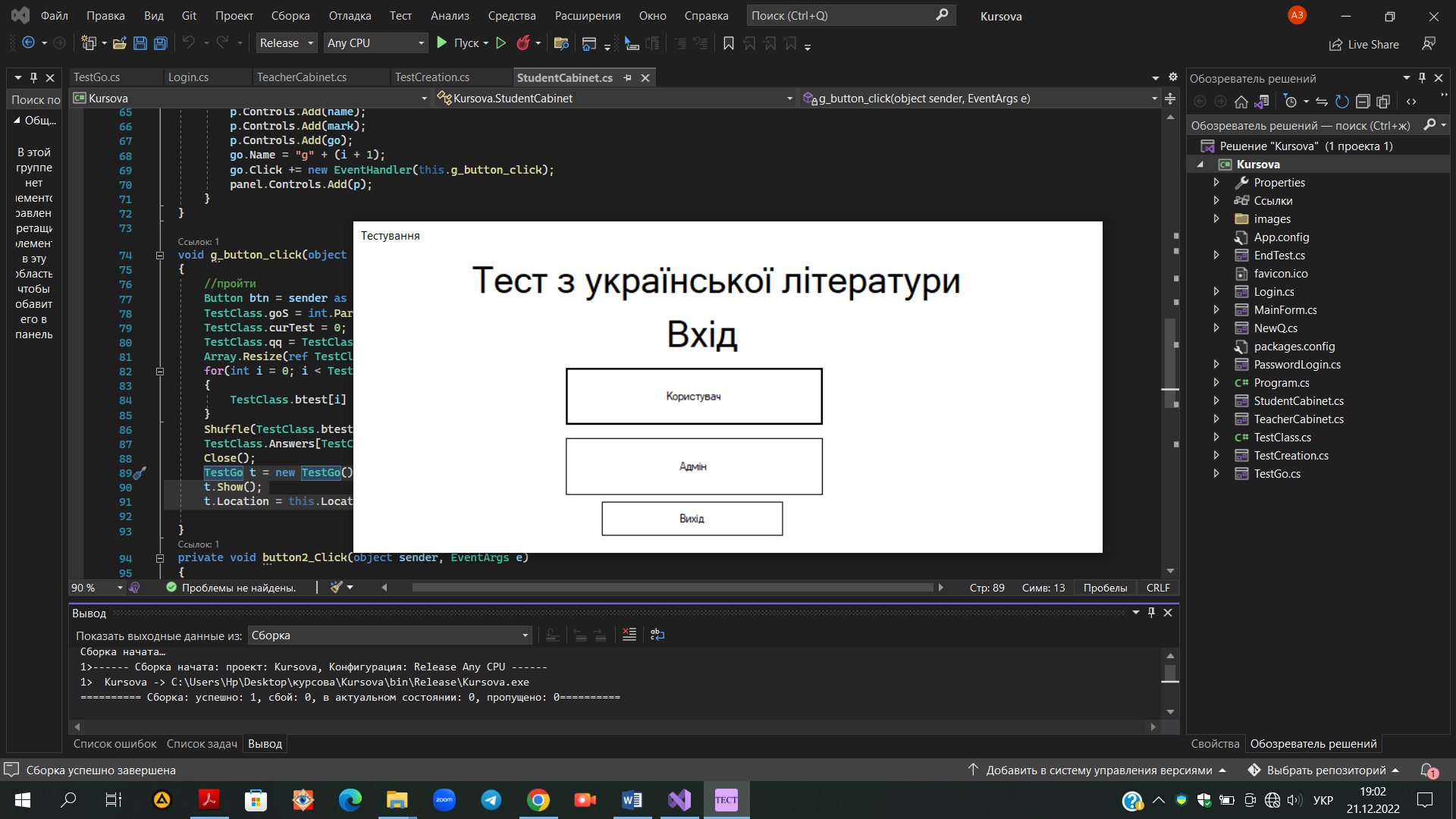


Рис Д.1 Головне меню інтерфейсу користувача.

Далі у нас є два варіанти , увійти в панель користувача , або в панель адміна.

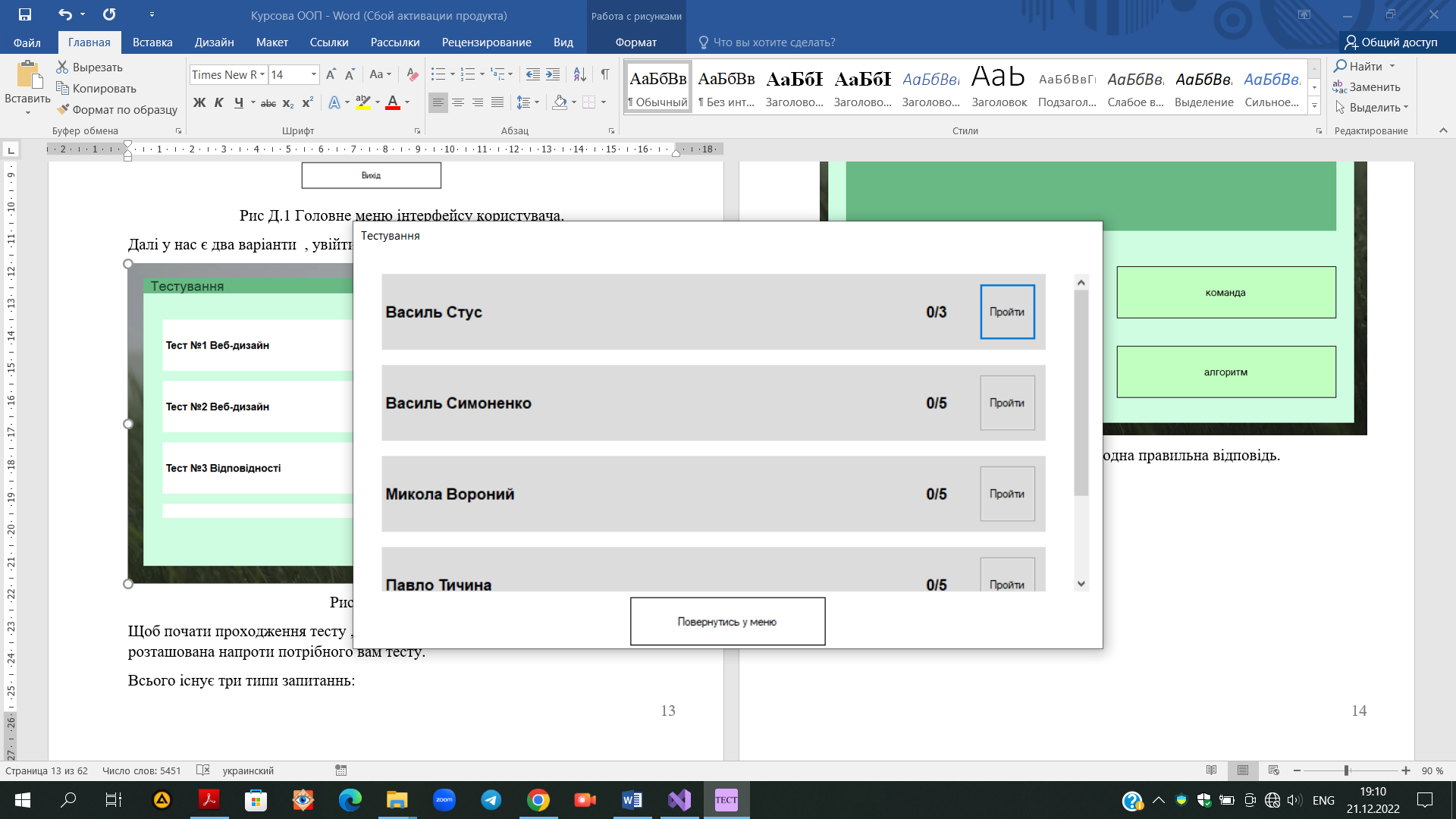


Рис Д.2 Панель учня.

Щоб почати проходження тесту , потрібно натиснути кнопку Пройти , яка розташована напроти потрібного вам тесту.

Всього існує три типи запитаннь:

* З однією правильною відповіддю.
* З кількома правильними відповідями.
* Відповідність.

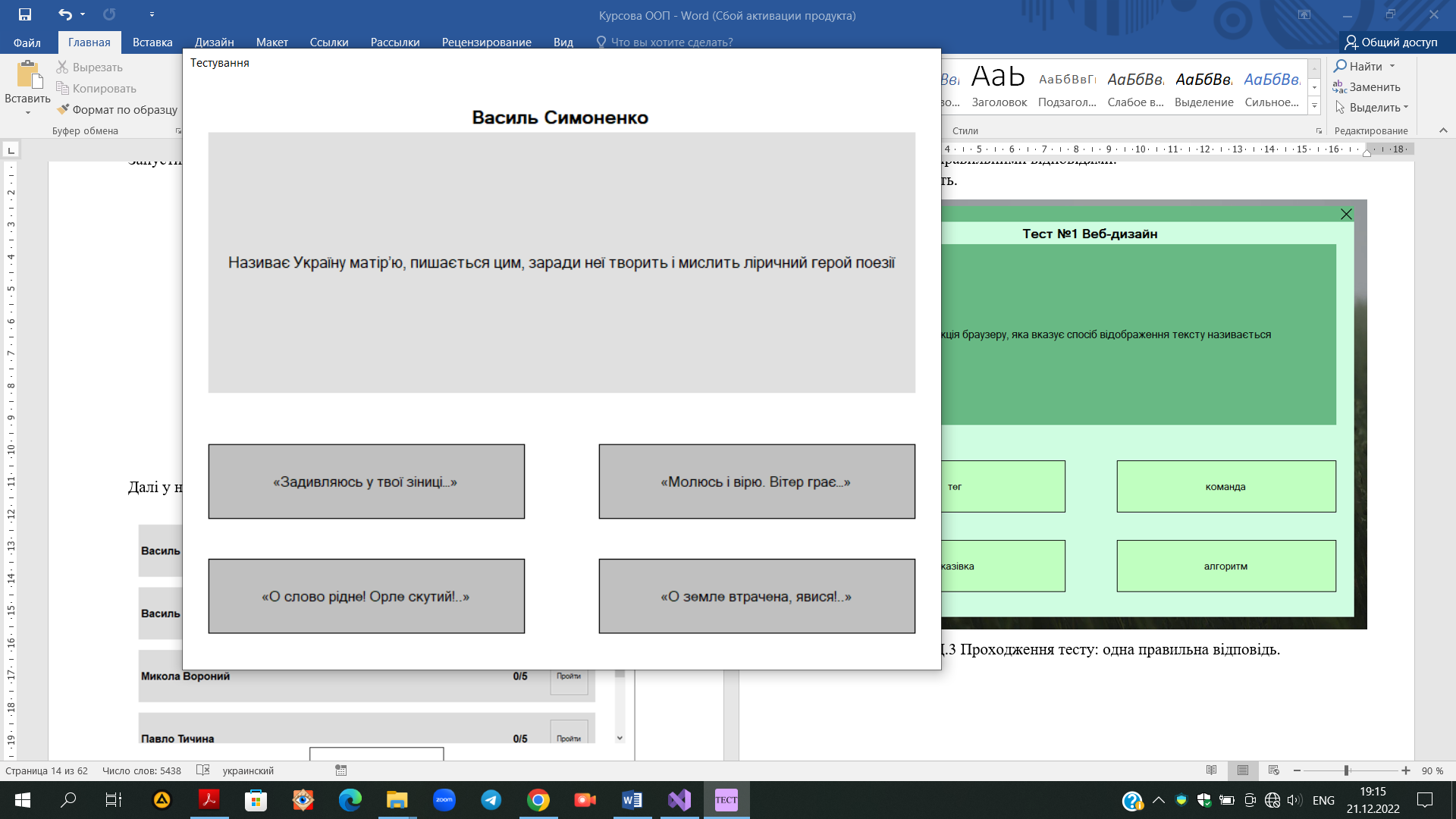


Рис Д.3 Проходження тесту: одна правильна відповідь.

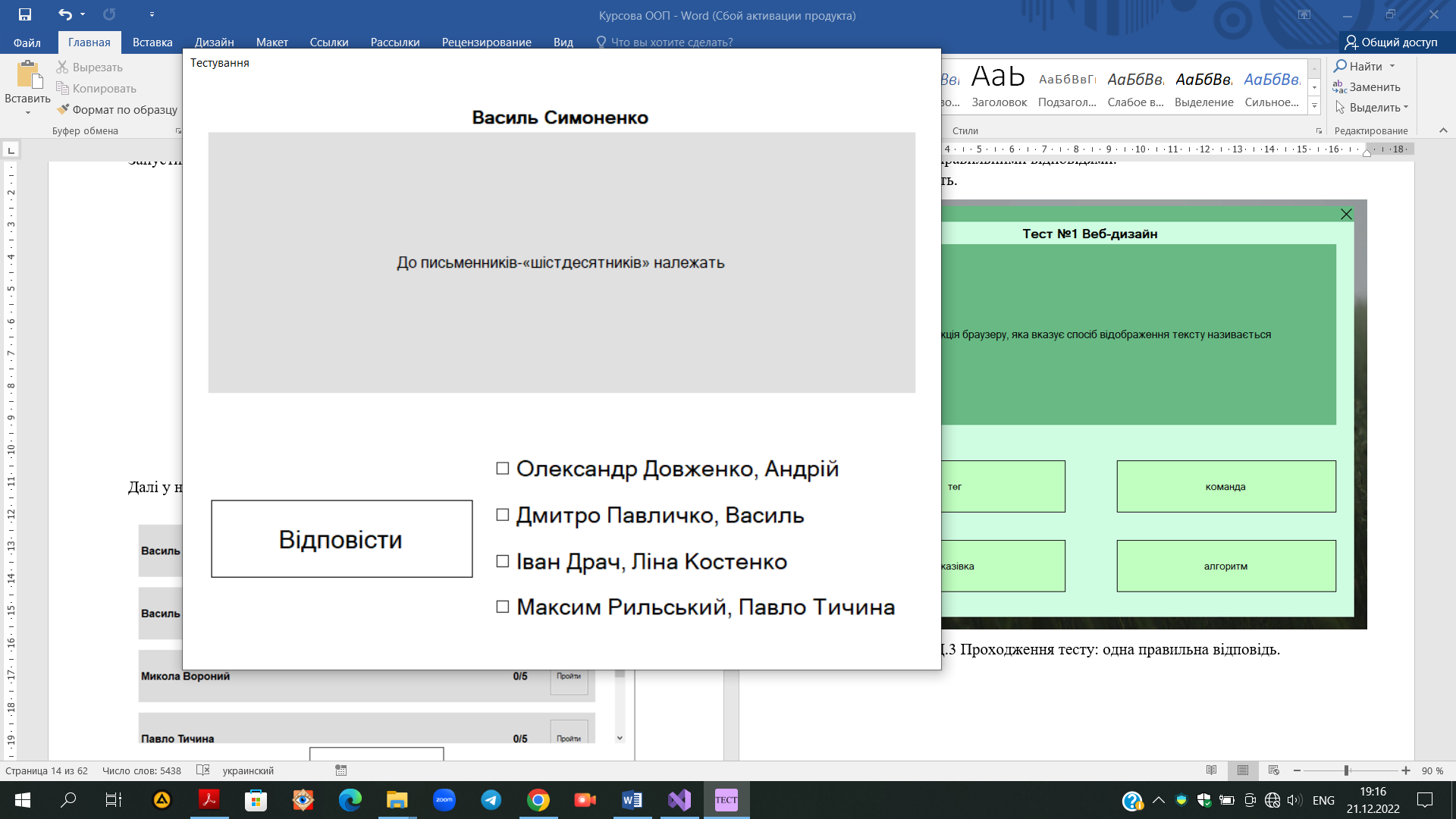


Рис Д.4 Проходження тесту: кілька правильних відповідей.

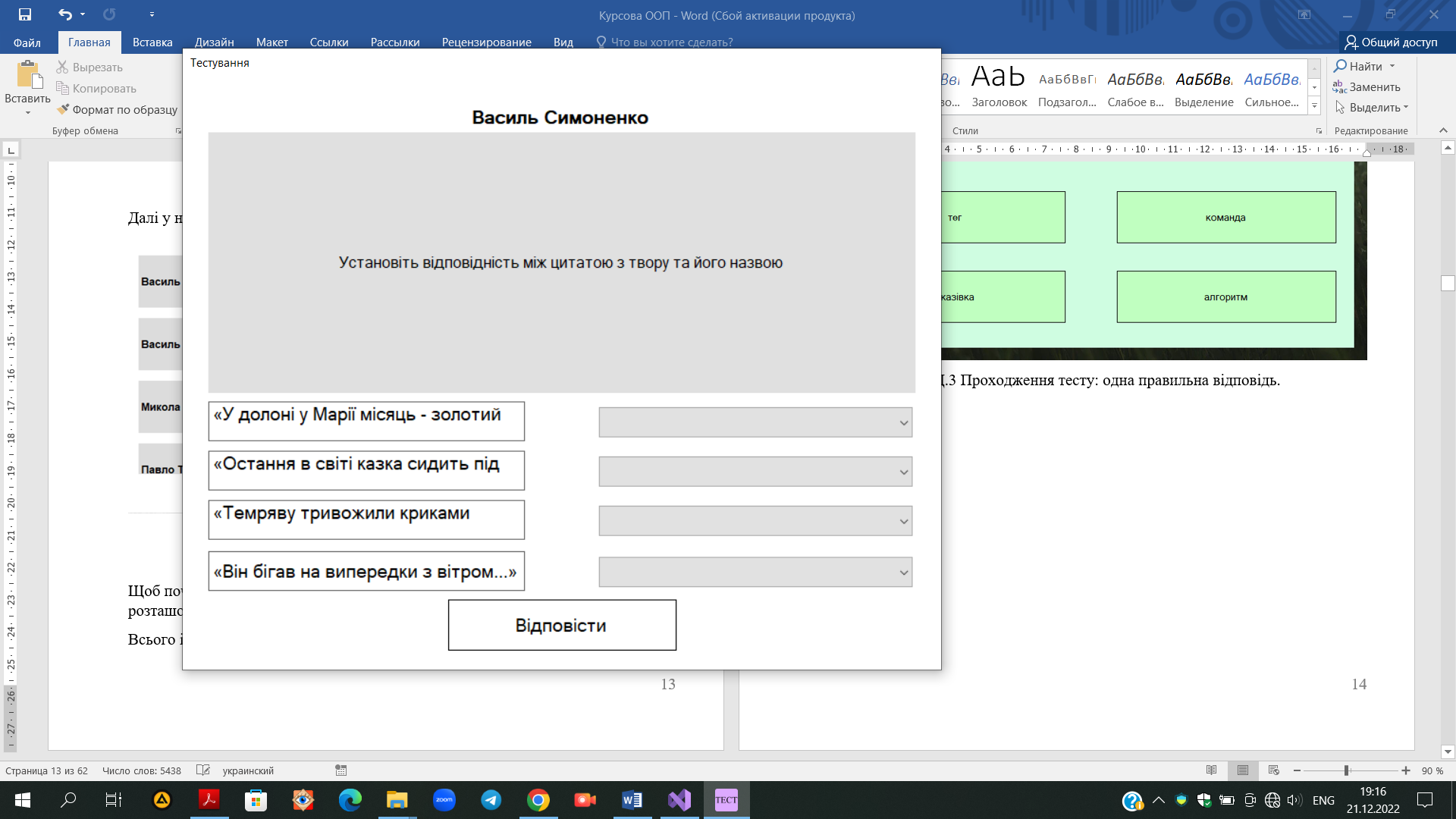


Рис Д.5 Проходження тесту: Відповідність.

Після того , як ви відповіли на останнє запитання , ви перейдете до меню результату.

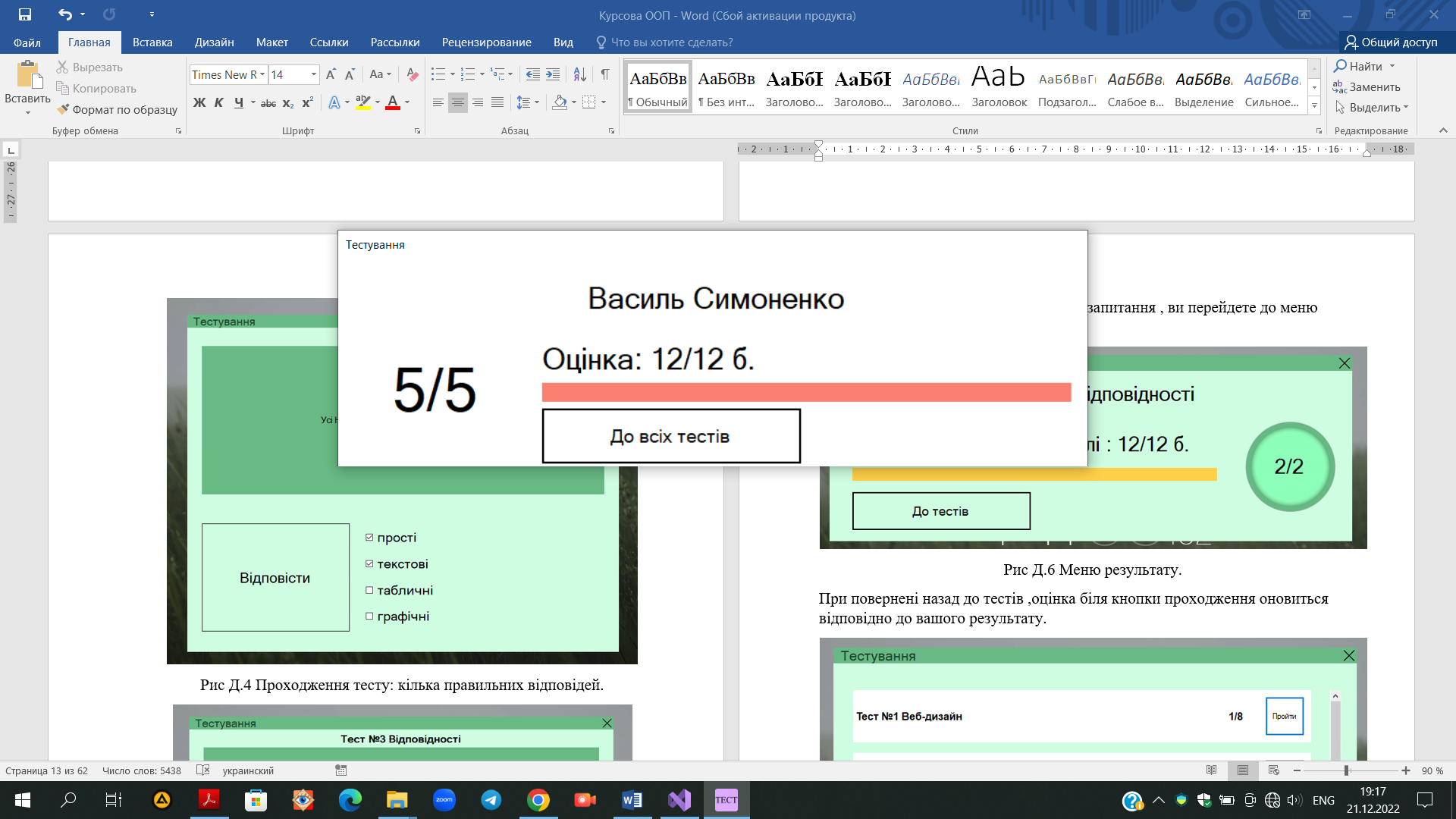


Рис Д.6 Меню результату.

При повернені назад до тестів ,оцінка біля кнопки проходження оновиться відповідно до вашого результату.

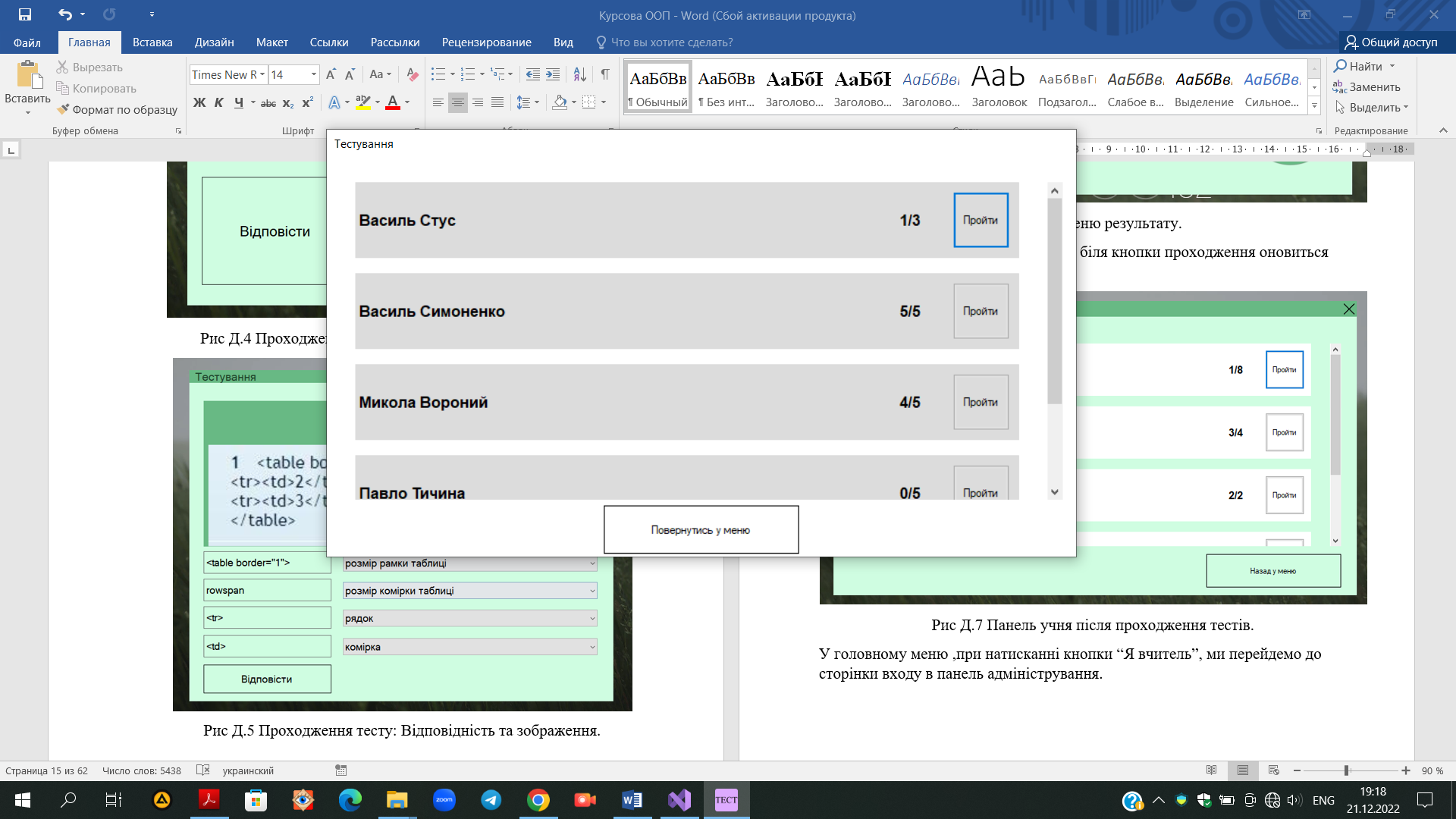


Рис Д.7 Панель учня після проходження тестів.

У головному меню ,при натисканні кнопки “адмін”, ми перейдемо до сторінки входу в панель адміністрування.

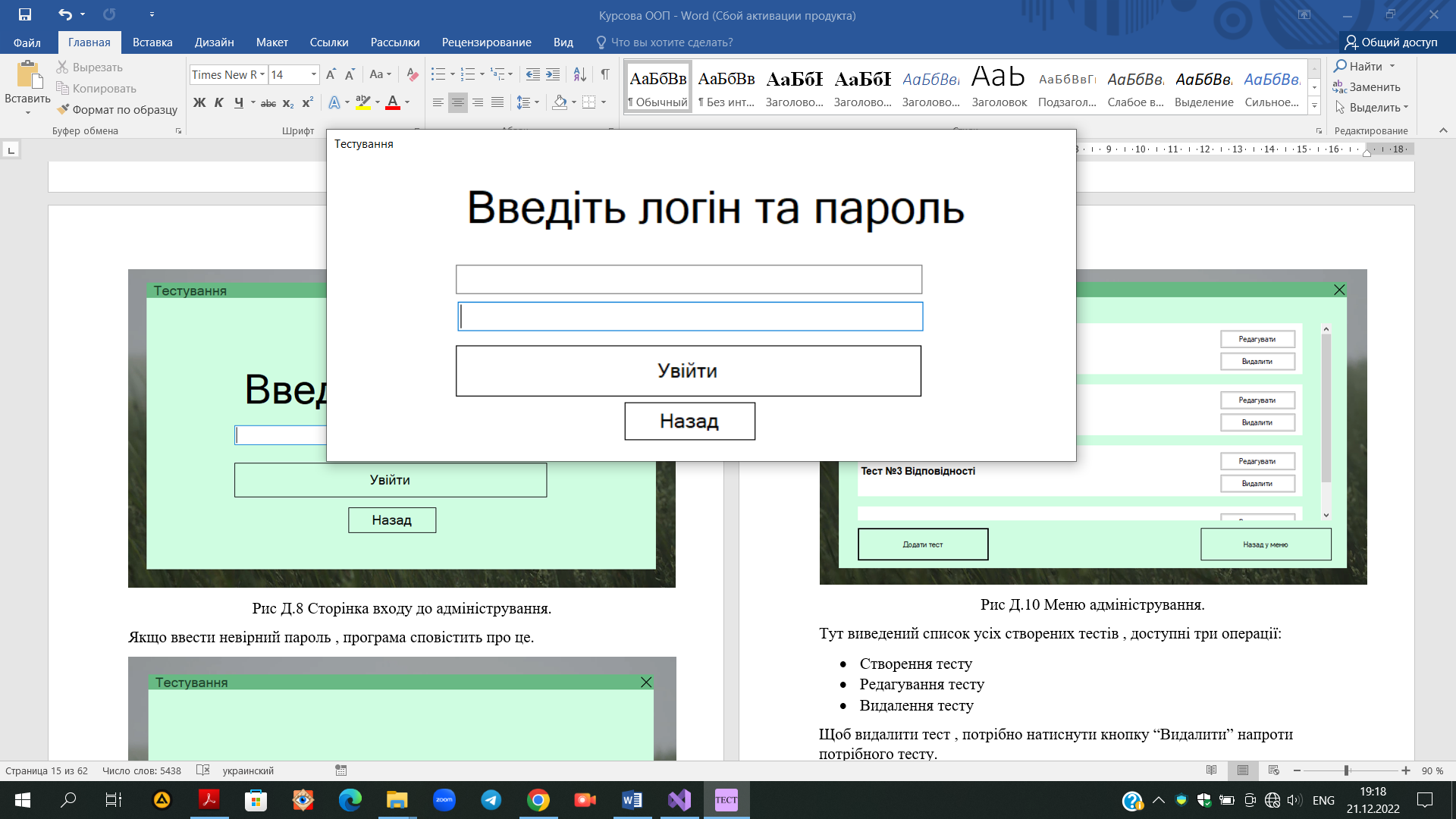


Рис Д.8 Сторінка входу до адміністрування.

Якщо ввести невірний пароль , програма сповістить про це.

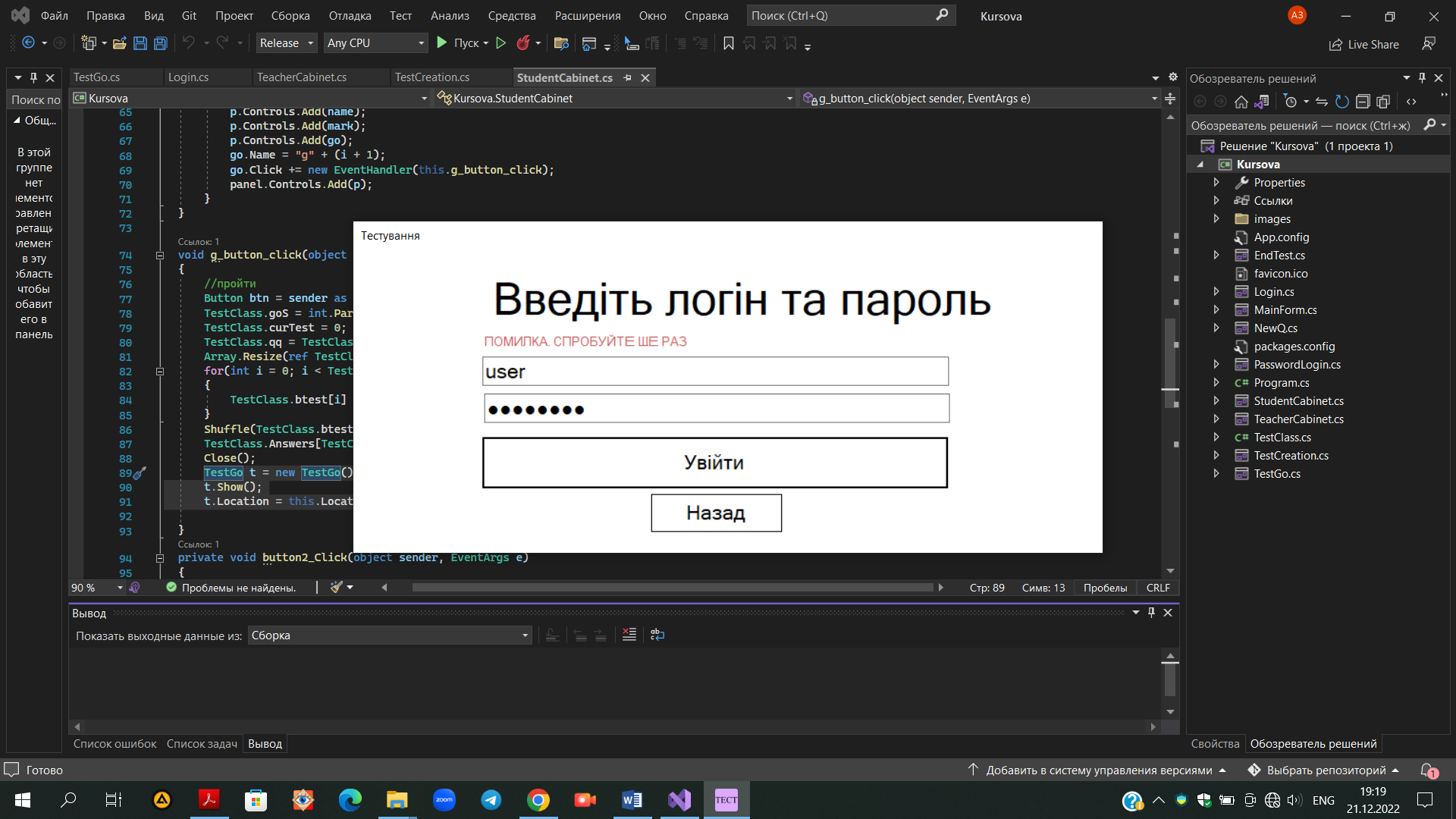


Рис Д.9 Вхід при невірному паролі.

При введені правильного логіну (admin) та паролю ( 11111111 ), ми зайдемо в панель адміністрування.

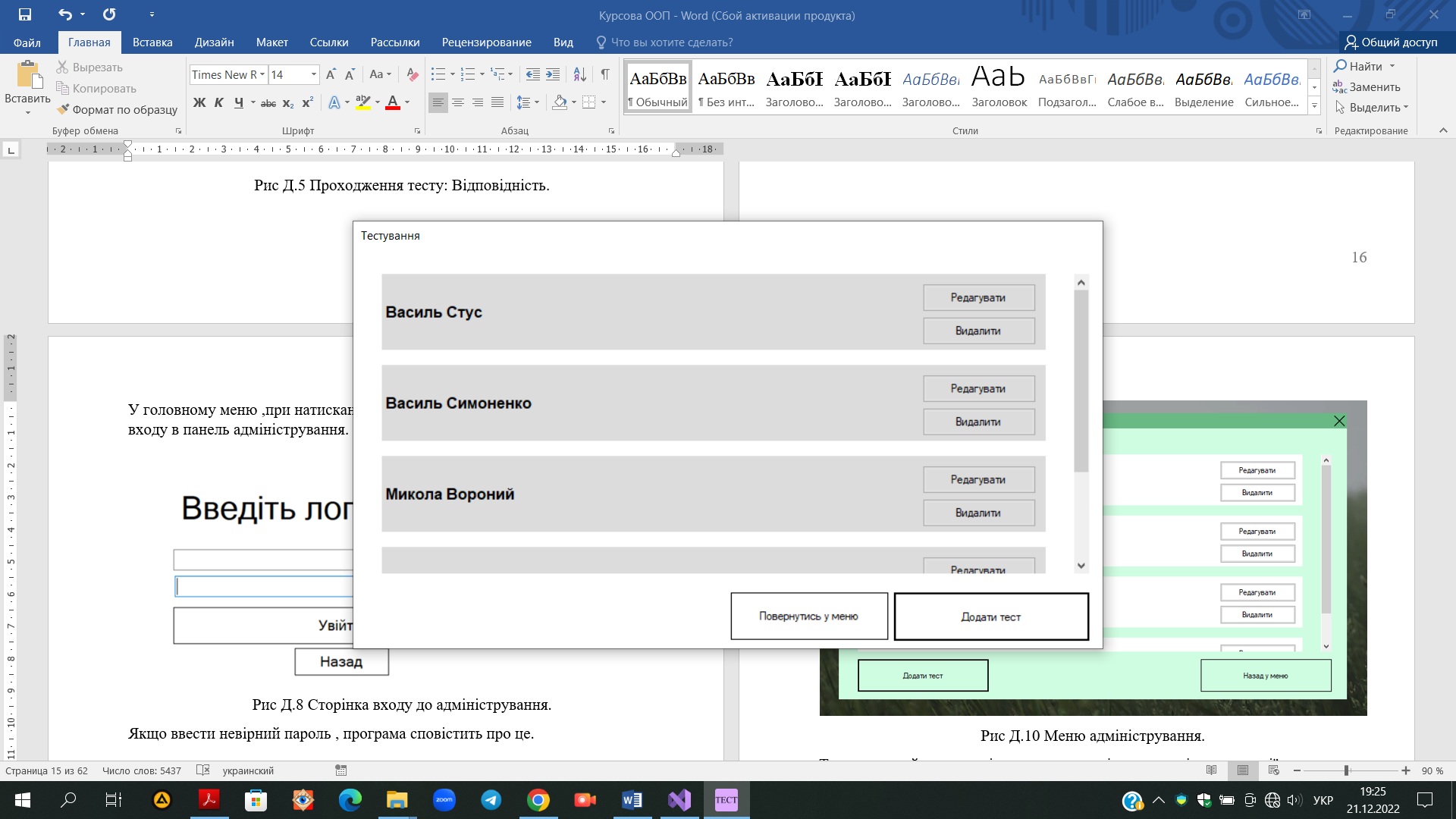


Рис Д.10 Меню адміністрування.

Тут виведений список усіх створених тестів , доступні три операції:

* Створення тесту
* Редагування вже створеного тесту
* Видалення вже створеного тесту

Щоб видалити тест , потрібно натиснути кнопку “Видалити” напроти потрібного тесту.

Щоб редагувати тест , потрібно натиснути кнопку “Редагувати” напроти потрібного тесту. При редагуванні тесту , відкривається меню , аналогічне створенню тесту , із наявною кнопкою назад.

Щоб створити новий тест ,потрібно натиснути кнопку “Додати тест”.

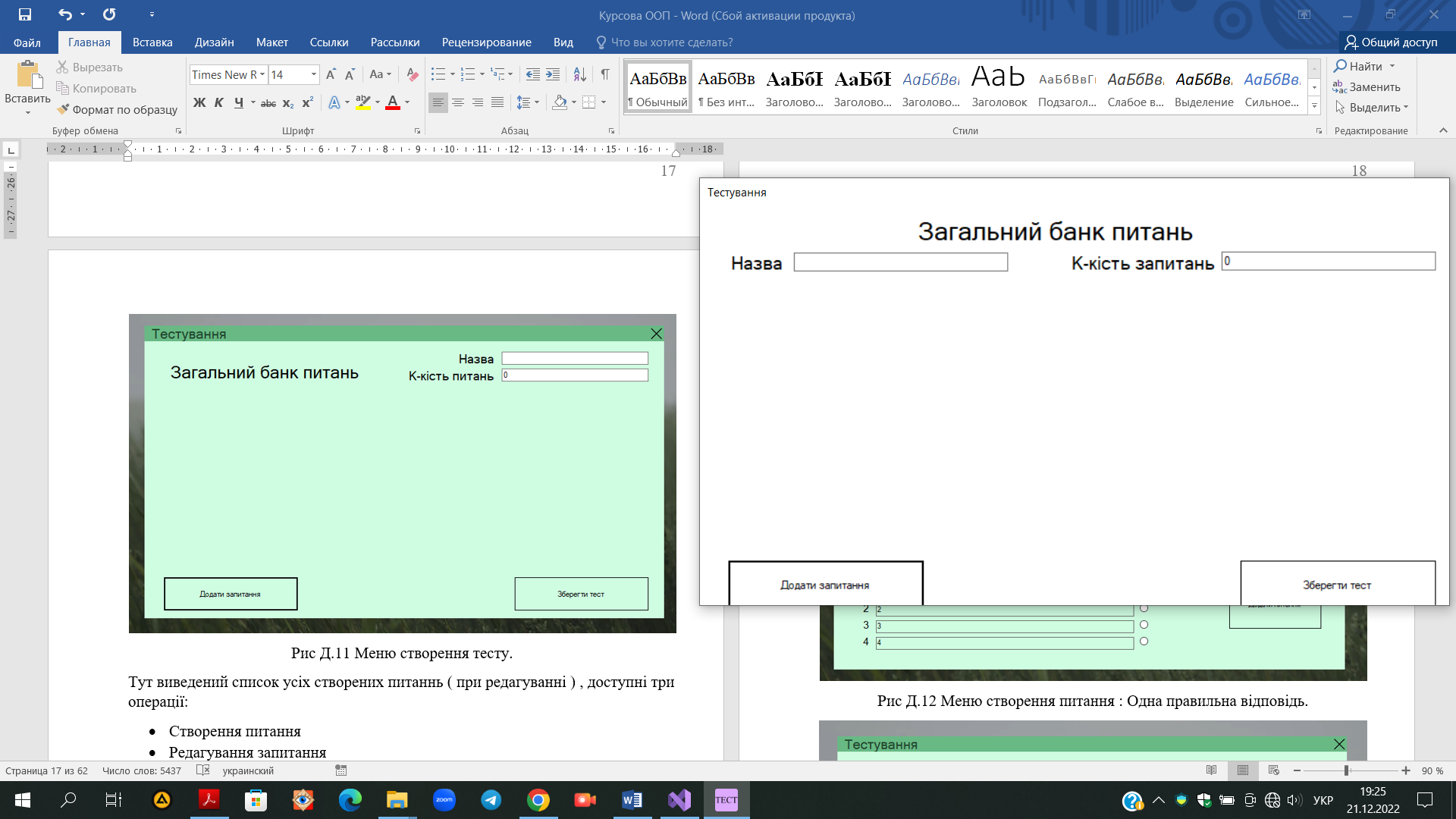


Рис Д.11 Меню створення тесту.

Тут виведений список усіх створених питань ( при редагуванні ) , доступні три операції:

* Створення питання
* Редагування запитання
* Видалення запитання

Щоб видалити запитання, потрібно натиснути кнопку “Видалити” напроти потрібного запитання.

Щоб редагувати тест , потрібно натиснути кнопку “Редагувати” напроти потрібного запитання. При редагуванні тесту , відкривається меню , аналогічне створенню запитання , із наявною кнопкою назад ( відміна ).

Щоб створити нове запитання ,потрібно натиснути кнопку “Додати запитання”.

Щоб змінити назву тесту , потрібно ввести її в поле “Назва”.

При проходженні тесту , питання обираються випадково з загального банку запитань , який може бути більшим , ніж кількість питань.

Щоб змінити кількість запитань (які обираються при проходженні з банку запитань ) потрібно ввести кількість в поле “К-кість питань”

При створені нового запитання , можливо обрати три типи питання :

* З однією правильною відповіддю.
* З кількома правильними відповідями.
* Відповідність.

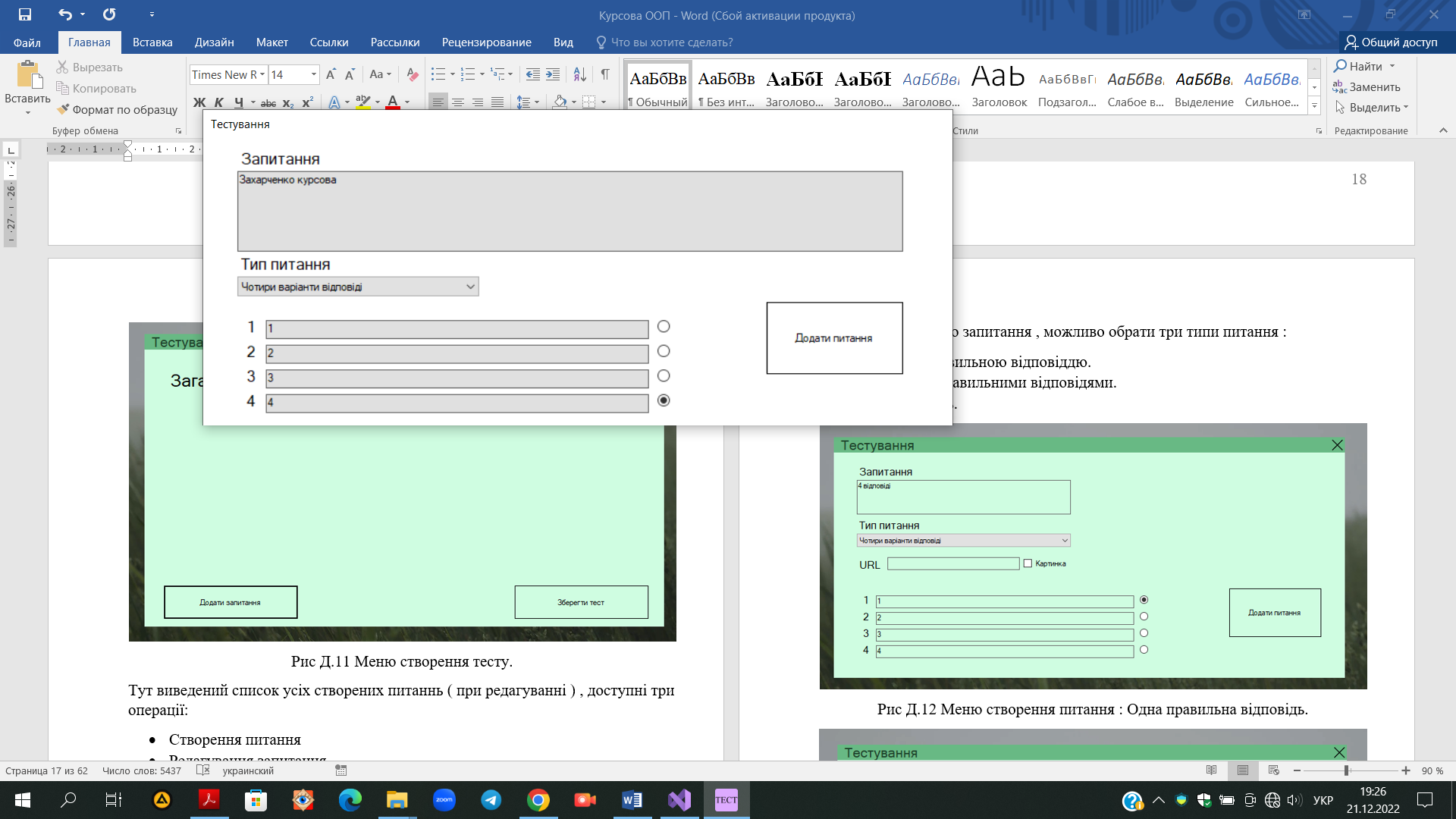


Рис Д.12 Меню створення питання : Одна правильна відповідь.

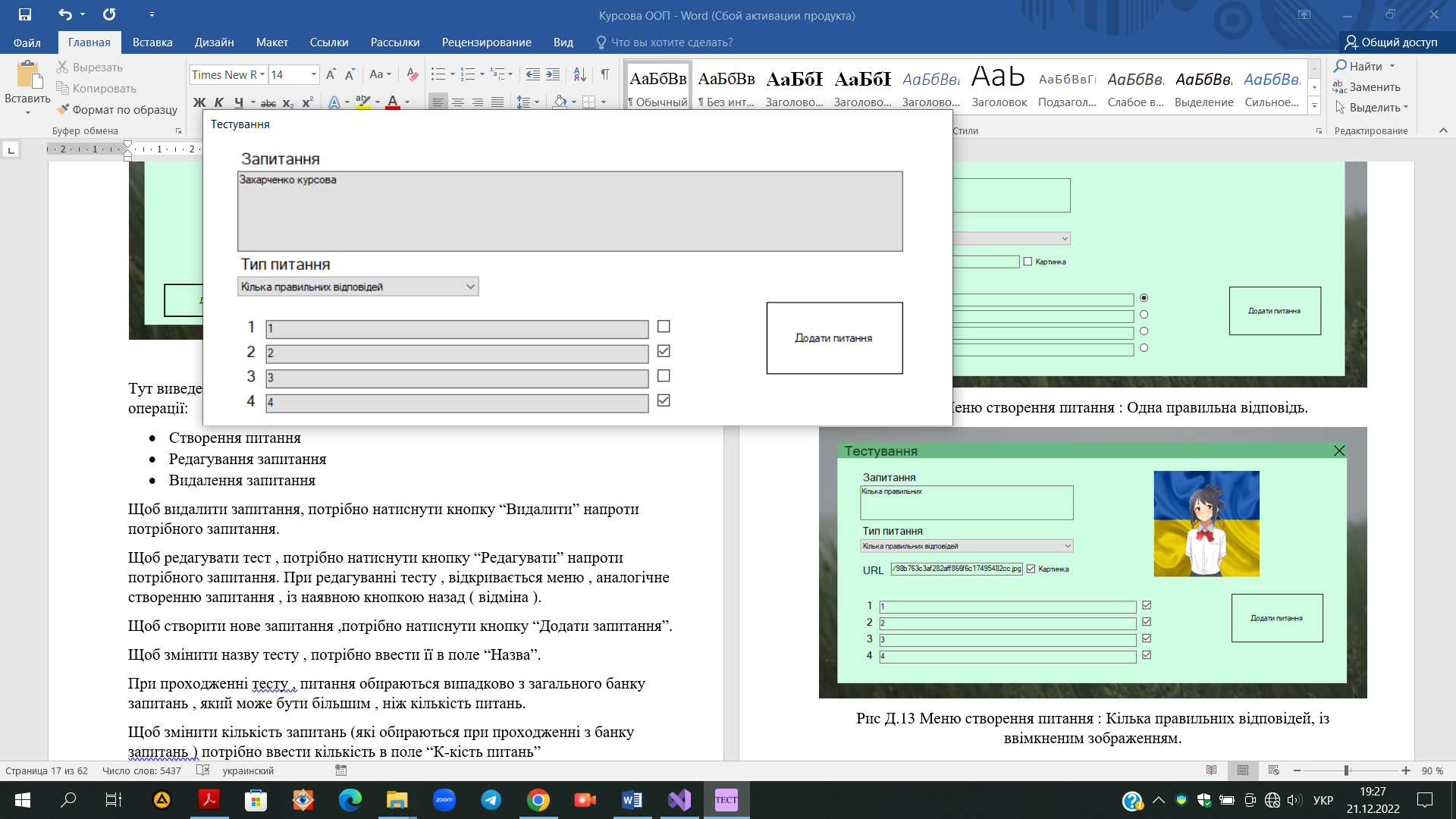


Рис Д.13 Меню створення питання : Кілька правильних відповідей.

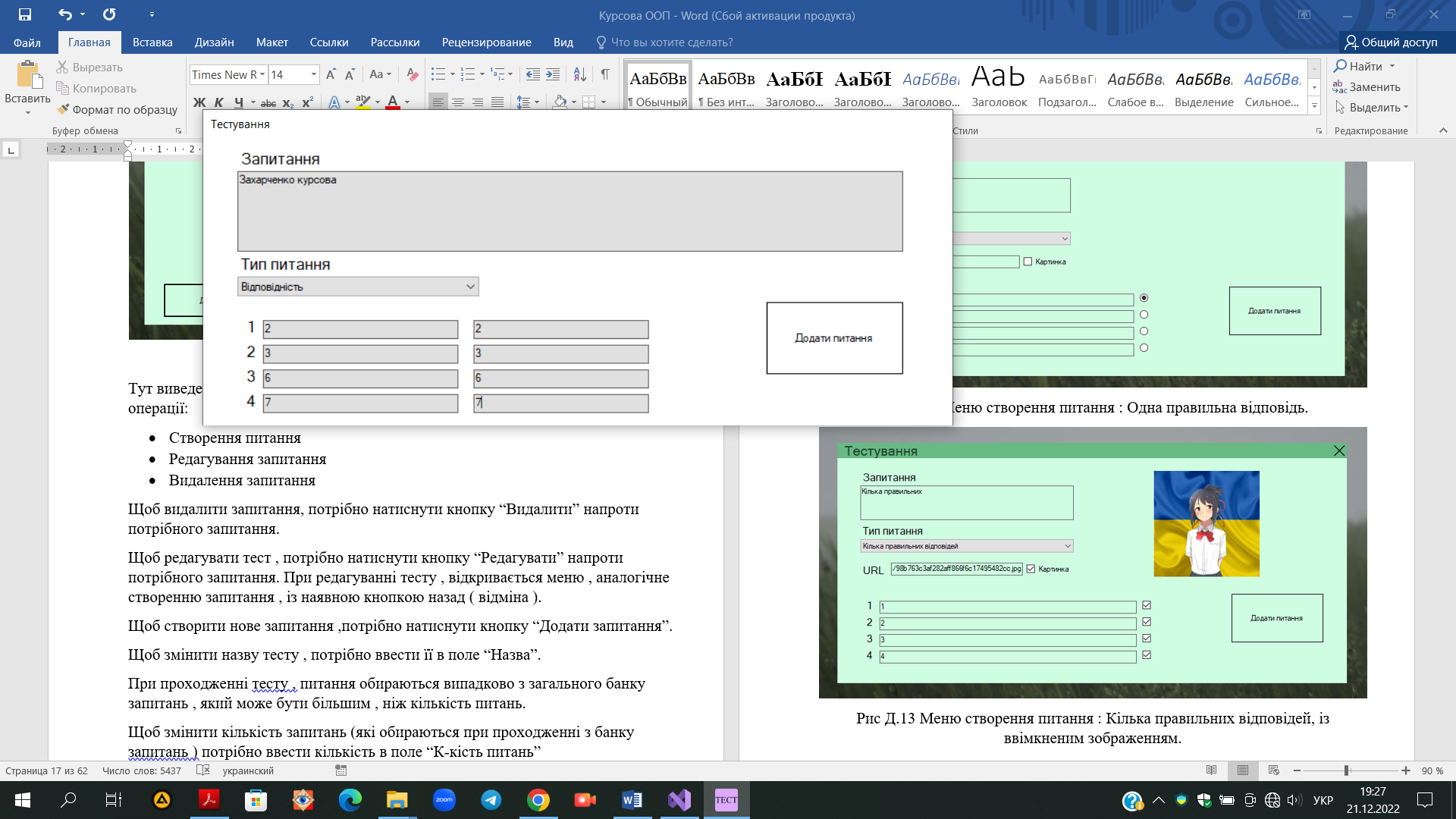


Рис Д.14 Меню створення питання : Відповідність.

При натисканні Додати запитання , запитання додається в тест , і користувач повертається в меню створення тесту.

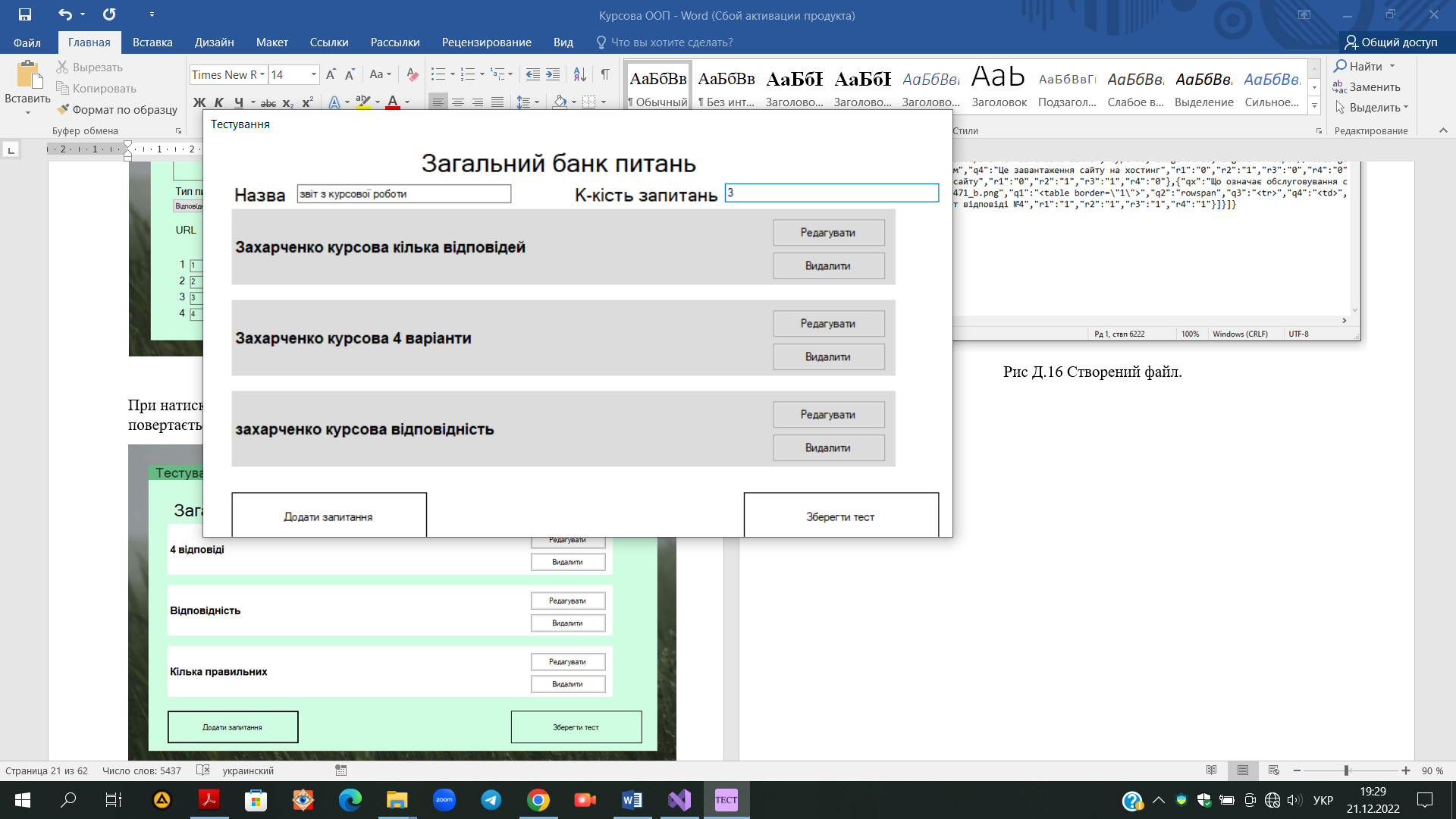


Рис Д.15 Меню створення тесту

## ВИСНОВОК

На практиці реалізована універсальна система тестування , зокрема з української літератури , яка дуже допомагає в реалізації дистанційного навчання.

З точки зору адміністрування цей тест дуже зручний для використання, адже та є декілька видів запитання тесту, також самі питання видаються користувачу в рандомному порядку, що допомагає краще перевірити знання.

Цікаво було реалізовувати адміністрування (вхід окремо для користувача та адміна), а також випадковий вибір запитань із загального банку. При натисканні проходження тесту , формується масив індексів загальної кількості запитань в банку , а згодом перемішувався створеною функцією Shuffle . Згодом , індекси запитань дістаються з перемішаного масиву індексів по черзі , рухаючись доти ,доки не пройдемо таку кількість запитань , яку ввів адміністратор при створенні тесту.

## ДОДАТОК. ТЕКСТИ ПРОГРАМНИХ ФАЙЛІВ

**Код форми Login:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.Runtime.InteropServices;

using ReadWriteBinaryFile;

namespace Kursova

{

public partial class Login : Form

{

public Login()

{

InitializeComponent();

}

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

{

TestClass.login = false;

Close();

PasswordLogin t = new PasswordLogin();

t.Show();

t.Location = this.Location;

}

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

{

TestClass.login = true;

Close();

PasswordLogin t = new PasswordLogin();

t.Show();

t.Location = this.Location;

}

private void button3\_Click(object sender, EventArgs e)

{

Application.Exit();

}

}

}

**Код форми MainForm:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Kursova

{

public partial class MainForm : Form

{

public MainForm()

{

InitializeComponent();

}

private void MainForm\_Load(object sender, EventArgs e)

{

}

private void MainForm\_Shown(object sender, EventArgs e)

{

Hide();

Login t = new Login();

t.Show();

}

}

}

**Код форми PasswordLogin:**

using Newtonsoft.Json;

using ReadWriteBinaryFile;

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Runtime.InteropServices;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using static System.Windows.Forms.VisualStyles.VisualStyleElement;

namespace Kursova

{

public partial class PasswordLogin : Form

{

public PasswordLogin()

{

InitializeComponent();

}

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

{

Close();

Login t = new Login();

t.Show();

t.Location = this.Location;

}

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

{

if (!TestClass.login)

{

//користувач

if (textBox2.Text == "user" && textBox1.Text == "88888888")

{

Close();

StudentCabinet t = new StudentCabinet();

t.Show();

t.Location = this.Location;

}

else

{

label3.Text = "ПОМИЛКА, СПРОБУЙТЕ ЩЕ РАЗ";

}

}

else

{

//адмін

if (textBox2.Text == "admin" && textBox1.Text == "11111111")

{

Close();

TeacherCabinet t = new TeacherCabinet();

t.Show();

t.Location = this.Location;

}

else

{

label3.Text = "ПОМИЛКА, СПРОБУЙТЕ ЩЕ РАЗ";

}

}

}

}

}

**Код форми TeacherCabinet:**

using Newtonsoft.Json;

using ReadWriteBinaryFile;

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Runtime.InteropServices;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.IO;

namespace Kursova

{

public partial class TeacherCabinet : Form

{

public TeacherCabinet()

{

InitializeComponent();

string data = File.ReadAllText("test.json");

TestClass.T = JsonConvert.DeserializeObject<TestClass.gt>(data);

if (TestClass.TisSelected)

{

TestClass.TisSelected = false;

}

TestClass.Tnew = true;

//----------------

for (int i = 0; i < TestClass.T.Size; i++)

{

Panel p = new Panel();

p.BackColor = Color.Gainsboro;

p.Size = new Size(700, 80);

p.Location = new Point(0, i \* 96);

// -------------

Label name = new Label();

name.Text = TestClass.T.allT[i].Name;

name.Font = new Font("Arial", 12, FontStyle.Bold);

name.AutoSize = false;

name.Size = new Size(560, 80);

name.TextAlign = ContentAlignment.MiddleLeft;

Button del = new Button();

Button red = new Button();

// -------------

del.Size = new Size(120, 30);

del.Location = new Point(570, 45);

del.Text = "Видалити";

red.Size = new Size(120, 30);

red.Location = new Point(570, 10);

red.Text = "Редагувати";

// -------------

p.Controls.Add(name);

p.Controls.Add(del);

p.Controls.Add(red);

del.Name = "d" + (i + 1);

red.Name = "r" + (i + 1);

del.Click += new EventHandler(this.d\_button\_click);

red.Click += new EventHandler(this.r\_button\_click);

panel.Controls.Add(p);

}

}

void d\_button\_click(object sender , EventArgs e)

{

//видалення

Button btn = sender as Button;

TestClass.Test[] temp;

temp = Array.Empty<TestClass.Test>();

for (int i=0;i < TestClass.T.Size; i++)

{

if (i != int.Parse(btn.Name.Remove(0, 1))-1)

{

var templist = temp.ToList();

templist.Add(TestClass.T.allT[i]);

temp = templist.ToArray();

}

}

TestClass.T.Size -= 1;

TestClass.T.allT = temp;

string output = JsonConvert.SerializeObject(TestClass.T);

File.WriteAllText("test.json", output);

Close();

TeacherCabinet t = new TeacherCabinet();

t.Show();

t.Location = this.Location;

}

void r\_button\_click(object sender, EventArgs e)

{

//редагування

Button btn = sender as Button;

TestClass.Tselected = int.Parse(btn.Name.Remove(0, 1)) - 1;

TestClass.TisSelected = true;

TestClass.Tnew = true;

TestClass.qq = TestClass.T.allT[TestClass.Tselected];

Close();

TestCreation t = new TestCreation();

t.Show();

t.Location = this.Location;

}

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

{

TestClass.qq.Name = "";

TestClass.qq.Size = 0;

TestClass.qq.questions = Array.Empty<TestClass.Question>();

Close();

TestCreation t = new TestCreation();

t.Show();

t.Location = this.Location;

}

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

{

Close();

Login t = new Login();

t.Show();

t.Location = this.Location;

}

}

}

**Код форми TestCreation:**

using Newtonsoft.Json;

using ReadWriteBinaryFile;

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Runtime.InteropServices;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.IO;

namespace Kursova

{

public partial class TestCreation : Form

{

public TestCreation()

{

InitializeComponent();

textBox1.Text = TestClass.qq.Name;

textBox2.Text = TestClass.qq.banksize.ToString();

if (TestClass.QisSelected)

{

TestClass.QisSelected = false;

}

if (TestClass.TisSelected)

{

button3.Visible = true;

}

else if(TestClass.Tnew)

{

TestClass.Tnew = false;

}

//----------------

for (int i = 0; i <= TestClass.qq.Size-1; i++)

{

Panel p = new Panel();

p.BackColor = Color.Gainsboro;

p.Size = new Size(700, 80);

p.Location = new Point(0, i \* 96);

// -------------

Label name = new Label();

name.Text = TestClass.qq.questions[i].qx;

name.Font = new Font("Arial", 12, FontStyle.Bold);

name.AutoSize = false;

name.Size = new Size(560, 80);

name.TextAlign = ContentAlignment.MiddleLeft;

Button del = new Button();

Button red = new Button();

// -------------

del.Size = new Size(120, 30);

del.Location = new Point(570, 45);

del.Text = "Видалити";

red.Size = new Size(120, 30);

red.Location = new Point(570, 10);

red.Text = "Редагувати";

// -------------

p.Controls.Add(name);

p.Controls.Add(del);

p.Controls.Add(red);

del.Name = "d" + (i + 1);

red.Name = "r" + (i + 1);

del.Click += new EventHandler(this.d\_button\_click);

red.Click += new EventHandler(this.r\_button\_click);

panel.Controls.Add(p);

}

}

void d\_button\_click(object sender , EventArgs e)

{

//видалення нажате

Button btn = sender as Button;

TestClass.Test temp = new TestClass.Test();

temp.questions = Array.Empty<TestClass.Question>();

for (int i=0;i < TestClass.qq.Size; i++)

{

if (i != int.Parse(btn.Name.Remove(0, 1))-1)

{

var templist = temp.questions.ToList();

templist.Add(TestClass.qq.questions[i]);

temp.questions = templist.ToArray();

}

}

temp.Size = TestClass.qq.Size-1;

temp.Name = TestClass.qq.Name;

TestClass.qq = temp;

//reload

Close();

TestCreation t = new TestCreation();

t.Show();

t.Location = this.Location;

}

void r\_button\_click(object sender, EventArgs e)

{

//редагування нажате

Button btn = sender as Button;

TestClass.Qselected = int.Parse(btn.Name.Remove(0, 1)) - 1;

TestClass.QisSelected = true;

Close();

NewQ t = new NewQ();

t.Show();

t.Location = this.Location;

}

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

{

Close();

NewQ t = new NewQ();

t.Show();

t.Location = this.Location;

}

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

{

if (TestClass.TisSelected)

{

TestClass.T.allT[TestClass.Tselected] = TestClass.qq;

string output = JsonConvert.SerializeObject(TestClass.T);

File.WriteAllText("test.json", output);

}

else

{

TestClass.T.Size++;

Array.Resize(ref TestClass.T.allT, TestClass.T.Size);

TestClass.T.allT[TestClass.T.Size - 1] = TestClass.qq;

string output = JsonConvert.SerializeObject(TestClass.T);

File.WriteAllText("test.json", output);

}

Close();

TeacherCabinet t = new TeacherCabinet();

t.Show();

t.Location = this.Location;

}

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

{

TestClass.qq.Name = textBox1.Text;

}

private void button3\_Click(object sender, EventArgs e)

{

Close();

TeacherCabinet t = new TeacherCabinet();

t.Show();

t.Location = this.Location;

}

private void textBox2\_TextChanged(object sender, EventArgs e)

{

TestClass.qq.banksize = Convert.ToInt32(textBox2.Text);

}

}

}

**Код форми NewQ:**

using Newtonsoft.Json;

using Newtonsoft.Json.Linq;

using ReadWriteBinaryFile;

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.IO;

using System.Linq;

using System.Runtime.InteropServices;

using System.Runtime.InteropServices.ComTypes;

using System.Security.Cryptography.X509Certificates;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using static System.Windows.Forms.VisualStyles.VisualStyleElement;

using static System.Windows.Forms.VisualStyles.VisualStyleElement.Button;

namespace Kursova

{

public partial class NewQ : Form

{

public NewQ()

{

InitializeComponent();

comboBox1.SelectedIndex = 0;

p2.Location = p1.Location;

p3.Location = p1.Location;

if (TestClass.QisSelected)

{

RLoad(TestClass.Qselected);

button1.Text = "Прийняти зміни";

button2.Visible = true;

}

}

private void RLoad(int index)

{

TestClass.Question sv = new TestClass.Question();

sv = TestClass.qq.questions[index];

switch (sv.type)

{

case 0:

p1.Show();

p2.Hide();

p3.Hide();

break;

case 1:

p1.Hide();

p2.Show();

p3.Hide();

break;

case 2:

p1.Hide();

p2.Hide();

p3.Show();

break;

}

textBox1.Text = sv.qx;

comboBox1.SelectedIndex = sv.type;

switch (sv.type)

{

case 0:

p1\_t1.Text = sv.q1;

p1\_t2.Text = sv.q2;

p1\_t3.Text = sv.q3;

p1\_t4.Text = sv.q4;

p1\_r1.Checked = stb(sv.r1);

p1\_r2.Checked = stb(sv.r2);

p1\_r3.Checked = stb(sv.r3);

p1\_r4.Checked = stb(sv.r4);

break;

case 1:

p2\_t1.Text = sv.q1;

p2\_t2.Text = sv.q2;

p2\_t3.Text = sv.q3;

p2\_t4.Text = sv.q4;

p2\_r1.Text = sv.r1;

p2\_r2.Text = sv.r2;

p2\_r3.Text = sv.r3;

p2\_r4.Text = sv.r4;

break;

case 2:

p3\_t1.Text = sv.q1;

p3\_t2.Text = sv.q2;

p3\_t3.Text = sv.q3;

p3\_t4.Text = sv.q4;

p3\_r1.Checked = stb(sv.r1);

p3\_r2.Checked = stb(sv.r2);

p3\_r3.Checked = stb(sv.r3);

p3\_r4.Checked = stb(sv.r4);

break;

}

}

private bool stb(string ss)

{

if (ss == "0")

{

return false;

}

else

{

return true;

}

}

private void comboBox1\_SelectedIndexChanged(object sender, EventArgs e)

{

switch (comboBox1.SelectedIndex)

{

case 0:

p1.Show();

p2.Hide();

p3.Hide();

break;

case 1:

p1.Hide();

p2.Show();

p3.Hide();

break;

case 2:

p1.Hide();

p2.Hide();

p3.Show();

break;

}

}

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

{

TestClass.Question sv = new TestClass.Question();

try

{

sv.qx = textBox1.Text;

sv.type = comboBox1.SelectedIndex;

switch (sv.type)

{

case 0:

sv.q1 = p1\_t1.Text;

sv.q2 = p1\_t2.Text;

sv.q3 = p1\_t3.Text;

sv.q4 = p1\_t4.Text;

/\* \*/

sv.r1 = p1\_r1.Checked ? "1" : "0";

sv.r2 = p1\_r2.Checked ? "1" : "0";

sv.r3 = p1\_r3.Checked ? "1" : "0";

sv.r4 = p1\_r4.Checked ? "1" : "0";

break;

case 1:

sv.q1 = p2\_t1.Text;

sv.q2 = p2\_t2.Text;

sv.q3 = p2\_t3.Text;

sv.q4 = p2\_t4.Text;

/\* \*/

sv.r1 = p2\_r1.Text;

sv.r2 = p2\_r2.Text;

sv.r3 = p2\_r3.Text;

sv.r4 = p2\_r4.Text;

break;

case 2:

sv.q1 = p3\_t1.Text;

sv.q2 = p3\_t2.Text;

sv.q3 = p3\_t3.Text;

sv.q4 = p3\_t4.Text;

/\* \*/

sv.r1 = p3\_r1.Checked ? "1" : "0";

sv.r2 = p3\_r2.Checked ? "1" : "0";

sv.r3 = p3\_r3.Checked ? "1" : "0";

sv.r4 = p3\_r4.Checked ? "1" : "0";

break;

}

}

catch (Exception ex)

{

MessageBox.Show(this, "Неправильний тип данних\r\n" + ex.Message);

return;

}

if (TestClass.QisSelected)

{

TestClass.qq.questions[TestClass.Qselected] = sv;

}

else

{

TestClass.qq.Size++;

Array.Resize(ref TestClass.qq.questions, TestClass.qq.Size);

TestClass.qq.questions[TestClass.qq.Size - 1] = sv;

}

try

{

Close();

TestCreation t = new TestCreation();

t.Show();

t.Location = this.Location;

}

catch (Exception ex)

{

MessageBox.Show(this, ex.Message);

}

}

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

{

Close();

TestCreation t = new TestCreation();

t.Show();

t.Location = this.Location;

}

}

}

**Код форми StudentCabinet:**

using Newtonsoft.Json;

using ReadWriteBinaryFile;

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Runtime.InteropServices;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.IO;

namespace Kursova

{

public partial class StudentCabinet : Form

{

public StudentCabinet()

{

InitializeComponent();

string data = File.ReadAllText("test.json");

TestClass.T = JsonConvert.DeserializeObject<TestClass.gt>(data);

if (TestClass.TisSelected)

{

TestClass.TisSelected = false;

}

TestClass.Tnew = true;

Array.Resize(ref TestClass.Answers, TestClass.T.Size);

//----------------

for (int i = 0; i < TestClass.T.Size; i++)

{

Panel p = new Panel();

p.BackColor = Color.Gainsboro;

p.Size = new Size(700, 80);

p.Location = new Point(0, i \* 96);

// -------------

Label name = new Label();

name.Text = TestClass.T.allT[i].Name;

name.Font = new Font("Arial", 12, FontStyle.Bold);

name.AutoSize = false;

name.Size = new Size(560, 80);

name.TextAlign = ContentAlignment.MiddleLeft;

// -------------

TestClass.Answers[i].AllM = TestClass.T.allT[i].banksize;

TestClass.Answers[i].selected = i;

Label mark = new Label();

mark.Text = TestClass.Answers[i].Mark +"/" +TestClass.Answers[i].AllM;

mark.Font = new Font("Arial", 12, FontStyle.Bold);

mark.AutoSize = false;

mark.Size = new Size(60, 80);

mark.Location = new Point(570);

mark.TextAlign = ContentAlignment.MiddleLeft;

//---------+----

Button go = new Button();

go.Size = new Size(60, 60);

go.Location = new Point(630, 10);

go.Text = "Пройти";

// -------------

p.Controls.Add(name);

p.Controls.Add(mark);

p.Controls.Add(go);

go.Name = "g" + (i + 1);

go.Click += new EventHandler(this.g\_button\_click);

panel.Controls.Add(p);

}

}

void g\_button\_click(object sender , EventArgs e)

{

//пройти

Button btn = sender as Button;

TestClass.goS = int.Parse(btn.Name.Remove(0, 1)) - 1;

TestClass.curTest = 0;

TestClass.qq = TestClass.T.allT[TestClass.goS];

Array.Resize(ref TestClass.btest, TestClass.qq.Size);

for(int i = 0; i < TestClass.qq.Size; i++ )

{

TestClass.btest[i] = i;

}

Shuffle(TestClass.btest);

TestClass.Answers[TestClass.goS].Mark = 0;

Close();

TestGo t = new TestGo();

t.Show();

t.Location = this.Location;

}

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

{

Close();

Login t = new Login();

t.Show();

t.Location = this.Location;

}

private void Shuffle(int[] arr)

{

Random rnd = new Random();

for (int i = arr.Length; i > 1; i--)

{

int pos = rnd.Next(i);

var x = arr[i - 1];

arr[i - 1] = arr[pos];

arr[pos] = x;

}

}

}

}

**Код форми TestGo:**

using ReadWriteBinaryFile;

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Collections;

using System.Linq;

using System.Runtime.InteropServices;

using System.Security.Policy;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using static System.Windows.Forms.VisualStyles.VisualStyleElement.Button;

using static System.Windows.Forms.VisualStyles.VisualStyleElement;

using ComboBox = System.Windows.Forms.ComboBox;

using System.Reflection;

namespace Kursova

{

public partial class TestGo : Form

{

TestClass.Question sv = new TestClass.Question();

public TestGo()

{

InitializeComponent();

p2.Location = new Point(p1.Location.X, p1.Location.Y-45);

p3.Location = p1.Location;

RLoad(TestClass.curTest);

label2.Text = TestClass.qq.Name;

}

private void RLoad(int index)

{

sv = TestClass.qq.questions[TestClass.btest[index]];

switch (sv.type)

{

case 0:

p1.Show();

p2.Hide();

p3.Hide();

break;

case 1:

p1.Hide();

p2.Show();

p3.Hide();

break;

case 2:

p1.Hide();

p2.Hide();

p3.Show();

break;

}

Lquestion.Text = sv.qx;

switch (sv.type)

{

case 0:

p1\_t1.Text = sv.q1;

p1\_t2.Text = sv.q2;

p1\_t3.Text = sv.q3;

p1\_t4.Text = sv.q4;

break;

case 1:

ArrayList l = new ArrayList();

p2\_t1.Text = sv.q1;

p2\_t2.Text = sv.q2;

p2\_t3.Text = sv.q3;

p2\_t4.Text = sv.q4;

l.Add(sv.r1);

l.Add(sv.r2);

l.Add(sv.r3);

l.Add(sv.r4);

RandomOrder(l, p2\_r1);

RandomOrder(l, p2\_r2);

RandomOrder(l, p2\_r3);

RandomOrder(l, p2\_r4);

break;

case 2:

p3\_t1.Text = sv.q1;

p3\_t2.Text = sv.q2;

p3\_t3.Text = sv.q3;

p3\_t4.Text = sv.q4;

break;

}

}

private bool stb(string ss)

{

if (ss == "0")

{

return false;

}

else

{

return true;

}

}

public void RandomOrder(ArrayList arrList, ComboBox box)

{

Random r = new Random();

for (int cnt = 0; cnt < arrList.Count; cnt++)

{

object tmp = arrList[cnt];

int idx = r.Next(arrList.Count - cnt) + cnt;

arrList[cnt] = arrList[idx];

arrList[idx] = tmp;

}

box.Items.Clear();

box.Items.Add(arrList[0]);

box.Items.Add(arrList[1]);

box.Items.Add(arrList[2]);

box.Items.Add(arrList[3]);

}

private void p1\_t1\_Click(object sender, EventArgs e)

{

Answer(1);

}

private void p1\_t2\_Click(object sender, EventArgs e)

{

Answer(2);

}

private void p1\_t3\_Click(object sender, EventArgs e)

{

Answer(3);

}

private void p1\_t4\_Click(object sender, EventArgs e)

{

Answer(4);

}

private void button3\_Click(object sender, EventArgs e)

{

if (p3\_t1.Checked == stb(sv.r1) && p3\_t2.Checked == stb(sv.r2) && p3\_t3.Checked == stb(sv.r3) && p3\_t4.Checked == stb(sv.r4))

{

TestClass.Answers[TestClass.goS].Mark++;

}

TestEnd();

}

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

{

if(p2\_r1.Text == sv.r1 && p2\_r2.Text == sv.r2 && p2\_r3.Text == sv.r3 && p2\_r4.Text == sv.r4)

{

TestClass.Answers[TestClass.goS].Mark++;

}

TestEnd();

}

private void Answer(int a)

{

switch(a){

case 1:

if(sv.r1 == "1")

{

TestClass.Answers[TestClass.goS].Mark++;

}

break;

case 2:

if (sv.r2 == "1")

{

TestClass.Answers[TestClass.goS].Mark++;

}

break;

case 3:

if (sv.r3 == "1")

{

TestClass.Answers[TestClass.goS].Mark++;

}

break;

case 4:

if (sv.r4 == "1")

{

TestClass.Answers[TestClass.goS].Mark++;

}

break;

}

TestEnd();

}

private void TestEnd()

{

if (TestClass.curTest < TestClass.qq.banksize - 1)

{

TestClass.curTest++;

Close();

TestGo t = new TestGo();

t.Show();

t.Location = this.Location;

}

else

{

Close();

EndTest t = new EndTest();

t.Show();

t.Location = this.Location;

}

}

}

}

**Код форми EndTest:**

using ReadWriteBinaryFile;

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Runtime.InteropServices;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Kursova

{

public partial class EndTest : Form

{

public EndTest()

{

InitializeComponent();

label2.Text = TestClass.qq.Name;

label3.Text = TestClass.Answers[TestClass.goS].Mark + "/" + TestClass.Answers[TestClass.goS].AllM;

double tw = Math.Round(((double)TestClass.Answers[TestClass.goS].Mark / (double)TestClass.Answers[TestClass.goS].AllM) \* 12);

double msize = ((double)TestClass.Answers[TestClass.goS].Mark / (double)TestClass.Answers[TestClass.goS].AllM) \* 558;

label4.Text = "Оцінка: " + tw + "/12 б.";

Panel mfront = new Panel();

mfront.BackColor = Color.Salmon;

mfront.Size = new Size((int)msize, 20);

mback.Controls.Add(mfront);

}

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

{

Close();

StudentCabinet t = new StudentCabinet();

t.Show();

t.Location = this.Location;

}

}

}