

ПОЛТАВСЬКИЙ УНІВЕРСИТЕТ ЕКОНОМІКИ І ТОРГІВЛІ
Навчально-науковий інститут денної освіти
Форма навчання денна
Кафедра комп'ютерних наук та інформаційних технологій

Допускається до захисту
Завідувач кафедри
_____ Олена ОЛЬХОВСЬКА
(підпис)

« ____ » _____ 2023 р.

КВАЛІФІКАЦІЙНА РОБОТА
на тему

**АЛГОРИТМІЗАЦІЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ ТРЕНАЖЕРА З
ТЕМИ «ПОДІЇ EXCEL VBA» ДИСТАНЦІЙНОГО НАВЧАЛЬНОГО
КУРСУ «ОРГАНІЗАЦІЯ ТА ОБРОБКА ЕЛЕКТРОННОЇ ІНФОРМАЦІЇ»**

зі спеціальності 122 Комп'ютерні науки
освітня програма «Комп'ютерні науки»
ступеня бакалавра

Виконавець роботи Покидько Максим Богданович
_____ « ____ » _____ 2023 р.
(підпис)

Науковий керівник к. ф.-м. н., доцент, Чілікіна Тетяна Василівна
_____ « ____ » _____ 2023 р.
(підпис)

Рецензент

ПОЛТАВА 2023 р.

ЗАТВЕРДЖУЮЗавідувач кафедри _____ Олена ОЛЬХОВСЬКА
(підпис)

«___» _____ 202__ р.

**ЗАВДАННЯ І КАЛЕНДАРНИЙ ГРАФІК
ВИКОНАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ**

на тему «Алгоритмізація та програмна реалізація тренажера з теми «Події Excel VBA» дистанційного навчального курсу «Організація та обробка електронної інформації»

зі спеціальності 122 Комп'ютерні науки

освітня програма «Комп'ютерні науки»

ступеня бакалавр

Прізвище, ім'я, по батькові Покидько Максим БогдановичЗатверджена наказом ректора № 144-Н від «01» вересня 2022 р.

Термін подання студентом роботи «___» _____ 202__ р.

Вихідні дані до кваліфікаційної роботи: публікації з теми, системи дистанційного навчання.

Зміст пояснювальної записки (перелік питань, які потрібно розробити)

**ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ,
ТЕРМІНІВ****ВСТУП****РОЗДІЛ 1. ПОСТАНОВКА ЗАДАЧІ****РОЗДІЛ 2. ІНФОРМАЦІЙНИЙ ОГЛЯД**

2.1. Введення до VBA

РОЗДІЛ 3. ТЕОРЕТИЧНА ЧАСТИНА

3.1. Об'єктно-орієнтована мова програмування високого рівня VBA

3.2. Алгоритм роботи тренажера з теми «Події Excel VBA» дистанційного навчального курсу «Організація та обробка електронної інформації»

3.3. Блок-схема тренажера

РОЗДІЛ 4. ПРАКТИЧНА ЧАСТИНА

4.1. Обґрунтування вибору програмних засобів

4.2. Опис програмної реалізації

4.3. Опис роботи тренажера

ВИСНОВКИ**СПИСОК ЛІТЕРАТУРИ****ДОДАТОК А. КОД ПРОГРАМИ**

Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали, посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Постановка задачі	Чілікіна Т.В.		
Інформаційний огляд	Чілікіна Т.В.		
Теоретична частина	Чілікіна Т.В.		
Практична частина	Чілікіна Т.В.		

Календарний графік виконання кваліфікаційної роботи

Зміст роботи	Термін виконання	Фактичне виконання
1. Вступ		
2. Вивчення методичних рекомендацій та стандартів та звіт керівнику		
3. Постановка задачі		
4. Інформаційний огляд джерел бібліотек та інтернету		
5. Теоретична частина		
6. Практична частина		
7. Закінчення оформлення		
8. Доповідь студента на кафедрі		
9. Доробка (за необхідністю), рецензування		

Дата видачі завдання «__» _____ 202__ р.

Здобувач вищої освіти _____ Покидько Максим Богданович
(підпис)Науковий керівник _____ к. ф.-м. н., доц. Чілікіна Тетяна Василівна
(підпис) (науковий ступінь, вчене звання, ініціали та прізвище)

Результати захисту кваліфікаційної роботи

Кваліфікаційна робота оцінена на _____
(балів, оцінка за національною шкалою, оцінка за ECTS)

Протокол засідання ЕК № _____ від «____» _____ 2023 р.

Секретар ЕК _____
(підпис) (ініціали та прізвище)

Затверджую

Зав. кафедрою _____
к.ф.-м.н. Олена ОЛЬХОВСЬКА
« ____ » _____ 202__ р.

Погоджено

Науковий керівник _____
к. ф.-м. н., Тетяна ЧІЛКІНА
« ____ » _____ 202__ р.

План

кваліфікаційної роботи на тему

«Алгоритмізація та програмна реалізація тренажера з теми «Події Excel VBA»
дистанційного навчального курсу «Організація та обробка електронної
інформації»

зі спеціальності 122 Комп'ютерні науки
освітня програма 122 «Комп'ютерні науки»
ступеня бакалавр

Прізвище, ім'я, по батькові Покидько Максим Богданович

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ,
ТЕРМІНІВ

ВСТУП

РОЗДІЛ 1. ПОСТАНОВКА ЗАДАЧІ

РОЗДІЛ 2. ІНФОРМАЦІЙНИЙ ОГЛЯД

2.1. Введення до VBA

РОЗДІЛ 3. ТЕОРЕТИЧНА ЧАСТИНА

3.1. Об'єктно-орієнтована мова програмування високого рівня VBA

3.2. Алгоритм роботи тренажера з теми «Події Excel VBA» дистанційного
навчального курсу «Організація та обробка електронної інформації»

3.3. Блок-схема тренажера

РОЗДІЛ 4. ПРАКТИЧНА ЧАСТИНА

4.1. Обґрунтування вибору програмних засобів

4.2. Опис програмної реалізації

4.3. Опис роботи тренажера

ВИСНОВКИ

СПИСОК ЛІТЕРАТУРИ

ДОДАТОК А. КОД ПРОГРАМИ

Здобувач вищої освіти _____ Максим ПОКИДЬКО
« ____ » _____ 202__ р.

РЕФЕРАТ

Записка: 41 с., 20 рис., 1 таблиця, 1 додаток, 15 джерел.

ПОДІЇ, EXCEL VBA, НАВЧАЛЬНИЙ ТРЕНАЖЕР, ОРГАНІЗАЦІЯ ТА ОБРОБКА ЕЛЕКТРОННОЇ ІНФОРМАЦІЇ.

Об'єкт розробки – алгоритм, блок-схема, розробка програмного забезпечення тренажера.

Мета роботи – створення тренажера з теми «Події Excel VBA» дистанційного навчального курсу «Організація та обробка електронної інформації».

Предмет розробки – створення тренажера з теми «Події Excel VBA» дистанційного навчального курсу «Організація та обробка електронної інформації».

Методи дослідження – події Excel VBA, середовище візуальної розробки програм Visual Studio та мова програмування Visual Basic.

Зроблено інформаційний огляд матеріалу по темі тренажеру. Розроблено алгоритм навчального тренажера з теми події Excel VBA, створено блок-схему програмного забезпечення. Програмно реалізовано тренажер з дисципліни «Організація та обробка електронної інформації».

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ, ТЕРМІНІВ	7
ВСТУП	8
РОЗДІЛ 1. ПОСТАНОВКА ЗАДАЧІ.....	10
РОЗДІЛ 2. ІНФОРМАЦІЙНИЙ ОГЛЯД.....	11
2.1. Введення до VBA.....	11
РОЗДІЛ 3. ТЕОРЕТИЧНА ЧАСТИНА	19
3.1. Об'єктно-орієнтована мова програмування високого рівня VBA	19
3.2. Алгоритм роботи тренажеру з теми «Події Excel VBA» дистанційного навчального курсу «Організація та обробка електронної інформації».....	21
3.3. Блок-схема тренажеру.....	26
РОЗДІЛ 4. ПРАКТИЧНА ЧАСТИНА	27
4.1. Обґрунтування вибору програмних засобів.....	27
4.2. Опис програмної реалізації.....	28
4.3. Опис роботи тренажеру	32
ВИСНОВКИ.....	38
СПИСОК ЛІТЕРАТУРИ.....	39
ДОДАТОК А. КОД ПРОГРАМИ	41

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ, ТЕРМІНІВ

Умовні позначення, символи, скорочення, терміни	Пояснення умовних позначень, скорочень, символів
Visual Basic	Це швидкий засіб створення програми для Windows з графічним інтерфейсом, характеризується простим синтаксисом, надає можливість компіляції в машинний код або інтерпретації в момент налагодження.
Visual Studio	Інтегроване середовище розробки програмного забезпечення.
Windows Forms	Інтерфейс програмування додатків (API), відповідальний за графічний інтерфейс користувача і є частиною Microsoft .NET Framework.
if..esle..then	Конструкція, яка дозволяє виконувати деякі дії, не тільки якщо вихідна умова вірна, але і якщо вона неправильна.

ВСТУП

24 лютого 2022 року відбулося повномасштабне воєнне вторгнення Російської федерації на територію України. За таких умов довелося змінити освітній процес у країні. 25 лютого 2022 року Міністерством освіти і науки України було направлено листа керівникам департаментів (управлінь) освіти і науки обласних, Київської міської державних адміністрацій і керівникам закладів професійної, фахової передвищої та вищої освіти з рекомендацією щодо припинення освітнього процесу в усіх закладах освіти та оголошення канікул на два тижні (Міністерство освіти і науки України, 2022). Дистанційна форма навчання та введення навчальних тренажерів дисциплін набуло значного поширення за останні роки (у зв'язку з уведенням карантинних обмежень, спричинених розповсюдженням COVID-19) як у ЗВО, так і у ЗЗСО, однак наразі ситуація ускладнюється тим, що фізично вчителі чи викладачі подекуди розміщені в різних містах чи закордоном. Окрім того, варто враховувати специфіку кожного ЗЗСО, стиль роботи вчителів предметників та інтенсивність воєнних дій у окремих регіонах України. [2]

Мета кваліфікаційної роботи – програмна реалізація тренажера з теми «Події Excel VBA» дистанційного навчального курсу «Організація та обробка електронної інформації».

Об'єкт розробки – алгоритм, блок-схема, створення тренажера.

Предмет розробки – програмна реалізація тренажера з теми «Події Excel VBA» дистанційного навчального курсу «Організація та обробка електронної інформації».

При програмній реалізації тренажеру було використано наступні інструменти: редактор коду «Visual Studio Code» мову програмування Visual Basic, середовище розробки додатків Visual Studio.

Пояснювальна записка складається з чотирьох розділів, а саме: ВСТУП, РОЗДІЛ 1. ПОСТАНОВКА ЗАДАЧІ, РОЗДІЛ 2. ІНФОРМАЦІЙНИЙ ОГЛЯД, 2.1. Введення до VBA, РОЗДІЛ 3. ТЕОРЕТИЧНА ЧАСТИНА, 3.1. Об'єктно-орієнтована мова програмування високого рівня VBA, 3.2. Алгоритм роботи тренажеру з теми «Події Excel VBA» дистанційного навчального курсу «Організація та обробка електронної інформації», 3.3. Блок-схема тренажеру, РОЗДІЛ 4. ПРАКТИЧНА ЧАСТИНА, 4.1. Обґрунтування вибору програмних засобів, 4.2. Опис програмної реалізації, 4.3. Опис роботи тренажеру, ВИСНОВКИ, СПИСОК ЛІТЕРАТУРИ.

РОЗДІЛ 1. ПОСТАНОВКА ЗАДАЧІ

Найголовнішим завданням кваліфікаційної роботи – є програмна реалізація тренажеру з теми «Події Excel VBA» дистанційного навчального курсу «Організація та обробка електронної інформації».

Завдання кваліфікаційної роботи:

Розглянути матеріал з теми «Події Excel VBA»;

Створити алгоритм роботи тренажеру;

Створити блок-схему алгоритму роботи тренажеру;

Програмно реалізувати тренажер;

Описати результати готового тренажеру.

При створенні програмного забезпечення тренажеру, слід врахувати наступні вікна:

Стартове вікно тренажеру, на цьому вікні формі буде інформація про тематику тренажеру, кнопка для яка починає тренінг, інформація про розробника та наукового керівника кваліфікаційної роботи;

Інші вікна тренажеру - це питання по темі тренажеру, питання у вигляді тестування, на які потрібно користувачеві правильно або неправильно дати відповідь.

РОЗДІЛ 2. ІНФОРМАЦІЙНИЙ ОГЛЯД

2.1. Введення до VBA

У мові VBA застосована технологія візуального програмування, тобто конструювання робочої поверхні додатка і його елементів керування безпосередньо на екрані. Головною особливістю даної мови є його укрій висока гнучкість. Він відноситься до мов об'єктно-орієнтованого програмування, яке можна описати як методику аналізу, проектування і написання додатків за допомогою об'єктів.

Суть об'єктно-орієнтованого програмування можна сформулювати так: “Не програми керують даними, а дані, зв'язані один з одним і описують якийсь реальний або уявлюваний об'єкт, на базі якого реалізована модель якого-небудь явища, керують програмами”.

Модель об'єкта базується на трьох основних поняттях: інкапсуляції, спадкуванні і поліморфізмі.

Інкапсуляція – комбінування даних із процедурами і функціями, що оперують цими даними.

Спадкування – можливість використання уже визначених об'єктів для побудови нових об'єктів, тобто можливість створення ієрархії об'єктів. Кожний з “спадкоємців” успадковує опис даних «прабатька» і одержує доступ до його властивостей і методів.

Поліморфізм – можливість визначення єдиного імені методу, застосовного одночасно до всіх об'єктів ієрархії, причому кожний з об'єктів ієрархії може мати свою особливість реалізації цього методу. Однак Visual Basic for Applications не підтримує механізму поліморфізму.

В основі моделі об'єкта лежить ідея об'єднання в одній структурі даних і дій, що виробляються над цими даними, тобто об'єднання властивостей і методів об'єкта.

Об'єкт – це абстрактний тип даних, що складає з взаємозалежних (звичайно добре структурованих) даних різних типів і різних процедур і функцій, керуючих цими даними. При цьому об'єкт – це щось неподільне, сутність, що зберігає свої властивості, тільки залишаючись єдиним цілим.

Сутність технології об'єктно-орієнтованого програмування полягає в тому, що разом з даними, що описують якесь явище, зберігається і програмний код, що описує поведінку цього явища. Усі разом і називається об'єктом, а сам принцип об'єднання в єдине ціле даних і програмного коду, що описує поведінкові властивості об'єкта, відомий під терміном інкапсуляція.

З об'єктом нерозривно зв'язане поняття класу. Клас (class) задає властивості і поведінку (методи) об'єктів класу. Кожен об'єкт є екземпляром якого-небудь класу. Причому між об'єктом і класом існують співвідношення, аналогічні співвідношенням між перемінною і її типом. Наприклад, кожен робочий лист книги MS Excel є об'єктом класу WorkSheets. Таким чином, клас – це логічна конструкція, а об'єкт – це його фізичне втілення. Клас містять дані і функції, що обробляють ці дані. Дані називаються властивостями класу, а функції – методами класу.

Перевагою Visual Basic for Applications (VBA) як убудованої мови MS Office є наявність бібліотек об'єктів, що дозволяють працювати з такими різними об'єктами, як, наприклад, робоча книга MS Excel, лист MS Excel і навіть окремий осередок MS Excel.

Сімейство (об'єкт **Collection**) являє собою об'єкт, що містить кілька інших об'єктів, як правило, того самого типу. Наприклад, об'єкт **Workbooks**

(робітники книги) містить усі відкриті об'єкти **Workbook** (робоча книга). Кожен елемент сімейства нумерується і може бути ідентифікований або по номері, або по імені. Наприклад, **Worksheets(1)** – перший робітник лист активної книги, а **Worksheets(лист1)** – робочий лист з ім'ям **Лист1**.

Властивостями називаються дані, що утримуються в об'єкті і звичайно описують параметри об'єкта.

Наприклад, одним із властивостей об'єкта **Worksheet** є його ім'я (властивість **Name**). Значенням даної властивості може бути, наприклад, “Лист1”. Конкретне значення властивості об'єкта визначає стан об'єкта.

Методами називаються процедури і функції, за допомогою яких можна зробити над даними (властивостями) об'єкта які-небудь дії.

Наприклад, конкретний лист **MS Excel** можна зробити активним. Для цього варто викликати для нього метод **Activate**.

При роботі з об'єктами виникають події. Наприклад, натискання на клавішу або щиглика мишею по одній із кнопок панелі інструментів. З появою події система посилає об'єктові повідомлення, що може бути оброблено методом, створеним при розробці класу. Такі методи називають оброблювачами подій або процедурами обробки подій. Події можуть виникати як у результаті дій користувача, так і в результаті роботи системи.

Перевагою **VBA** є те, що користувачеві немає необхідності створювати свої класи (хоча це і можливо), оскільки можна використовувати вже готові. З цього випливає ще одне достоїнство – легкість у вивченні й освоєнні мови, а саме не потрібно вникати в технологію створення класів – досить тільки вміти ними користуватися.

Програма **VBA** конструюється (складається) з модулів.

Модуль – це набір описів і процедур мовою VBA, зібраних в одну програмну одиницю.

Існує два основних типи модулів: модулі класу і стандартні модулі.

Модуль класу – це модуль, зв'язаний з яким-небудь об'єктом. Наприклад, модулі форм. Ці модулі зв'язані з визначеною формою. У процесі створення форми створюється клас, що є спадкоємцем класу UserForm. Модулі форм і звітів звичайно містять процедури обробки подій, а також виклики процедур зі стандартних модулів. Однак модулі класу можуть існувати незалежно від форм Середовище розробки VB.

Перед розглядом питання проектування інтерфейсу додатка на Visual Basic, необхідно представляти, з чого взагалі складається цей проект?

У Visual Basic будь-який проект складається з наступних файлів:

файл кожної форми (розширення `frm`). Це звичайний ASCII текстовий файл, у якому записаний весь код, поміщений у форму, а також властивості всіх поміщених на форму елементів керування і самої форми теж.

файл кожної форми, що містить бінарну інформацію (наприклад картинку в PictureBox) (розширення `frx`)

файл проекту, що містить інформацію про проект (розширення `vbp`)

інформація про робочу область проекту (workspace) (розширення `vbw`)

Це необхідний мінімум. (Хоча, бувають і виключення, наприклад, коли в проекті не використовуються форми. Тоді замість `frm` файлу, буде `bas` файл.)

Далі перелічимо додаткові файли, що можуть бути підключені до проекту:

файл кожного модуля (розширення `bas`) Це текстовий файл.

файл кожного модуля класів (розширення `cls`). Це текстовий файл.

файл кожного додаткового елемента керування (розширення `ctl`) Це теж текстовий файл.

файл ресурсів (розширення `res`)

інші файли (osx, tlb, і т.д...)

Запам'ятовувати призначення всіх цих файлів не обов'язково, досить запам'ятати 2 файли: frm-файл, у якому зберігаються код форми і властивості всіх поміщених на дану форму елементів керування. І bas-файл - модуль. У ньому можуть бути оголошені глобальні перемінні, константи, функції і т.д.

Редактор VBA активізується з робочої книги MS Excel через виконання команд <Сервіс Макрос Редактор Visual Basic> або комбінацією клавіш <Alt> + <F11>. Вийти з редактора в робочу книгу можна через виконання комбінації клавіш <Alt> + <F11> або іншим способом.

Інтерфейс редактора VBA включає наступні основні компоненти: вікно проекту (Project Explorer), вікно властивостей (Properties Window), вікно редагування коду (Code), вікно редагування форм (UserForm), вікно перегляду об'єктів (Object Browser), меню і панелі інструментів.

Вікно Code (Програма) представляє текстовий редактор, призначений для написання і редагування тексту програми. Це вікно з'являється на екрані, наприклад, при створенні нового модуля. Для його виклику досить виконати подвійний щиглик мишею на формі або модулі у вікні проекту. Виклик вікна Code можна також здійснити виконанням команд: <View Code> або <View Project Explorer View Code>.

У верхній частині вікна програми присутні два випадаючі списки: Object (Об'єкт) і Procedure (Процедура). Їхній уміст залежить від того, з яким модулем у даний момент йде робота. У випадку модуля форми список Object дозволяє вибрати об'єкт, для якого у вікні програми будуть відображатися процедури. Список Procedure дозволяє вказати процедуру об'єкта, обраного в списку Object. При цьому у вікні програми відобразиться текст процедури.

Деякі властивості списку Object:

- якщо робота ведеться зі стандартним модулем, то в списку

Object вибрати об'єкти не можна;

- якщо робота ведеться з модулем класу, то в цьому списку можна

вибрати елементи General і Class. При цьому якщо в списку Object обраний елемент Class, те в списку Procedure автоматично стають доступними процедури Initialize і Terminate;

- якщо в списку Object обраний елемент General, а в списку Procedure – Declarations, те в даному розділі встановлюються деякі режими роботи редактори Visual Basic і з'являються перемінні рівня модуля.

Вікно редагування коду служить також як редактор для введення і зміни коду процедур додатка. Код усередині модуля організований у виді окремих розділів для кожного об'єкта, програмувального в модулі. У вікні редагування доступні два режими представлення коду: перегляд окремої процедури і всього модуля. Переключення режимів роботи вікна редагування коду здійснюється вибором однієї з двох кнопок у нижньому лівому куті вікна редагування коду <Procedure view> і <Full Module view>, або установкою або зняттям прапорця <Default to Full Module View> (Перегляд усього модуля) вкладки < Editor > (Редактор) діалогового вікна < Options > (Параметри), відображуваного на екрані командою <Tools Options > (Сервіс, Параметри).

Для створення засобів спілкування з користувачем у проект включаються форми, що являють собою типові вікна Windows.

Вони мають рядок заголовка з кнопками керування і системним меню, а також можливості керування мишею. Усе це не треба програмувати. Форма, включена в проект, має ці властивості. Порожня поверхня форми (сіре тло, покритий сіткою) заповнюється в процесі проектування [1].

Редактор форм є одним з основних інструментів візуального програмування. Форма в проект додається за допомогою команди <Insert  UserForm> (Вставка, Форма) або натисканням кнопки <Insert UserForm> (Уставити UserForm). У результаті на екран виводяться незаповнена форма і панель інструментів <Toolbox> (Панель елементів).

Використовуючи панель інструментів <Toolbox> (Панель елементів), з незаповненої форми можна сконструювати будь-яке необхідне для додатка діалогове вікно. Розміщення нового керуючого елемента у формі здійснюється в наступному порядку:

- клацнути мишею на панелі інструментів по значку того елемента, що потрібно розмістити на поверхні форми;
- помістити покажчик миші на те місце у формі, де буде розташовуватися керуючий елемент;
- натиснути ліву кнопку миші і, не відпускаючи неї, розтягти появившийся прямокутник до необхідних розмірів;
- відпустити кнопку миші. Елемент керування створений на потрібному місці.

Розміри форми і розташованих на ній елементів керування можна змінювати. Технологія зміни розмірів стандартна для Windows:

- виділити змінюваний елемент;
- помістити покажчик миші на одному з розмірних маркерів і протягти його при натиснутій лівій кнопці миші так, щоб об'єкт прийняв необхідні розміри;
- відпустити кнопку миші і скасувати виділення елемента.

Вікно редагування форм підтримує операції буфера обміну. Таким чином, можна копіювати, вирізувати і вставляти елементи керування, розташовані на поверхні форми. Для полегшення розміщення і вирівнювання елементів керування використовується сітка форми. Активізувати її можна за допомогою вкладки <General> (Загальні) діалогового вікна <Tools □ Options> (Сервіс, параметри), там же встановлюється крок сітки. Крім того, команди меню

<Format> (Формат) автоматизують і полегшують процес вирівнювання елементів керування, як по їхньому взаємному місцю розташування, так і по розмірах.

Вікно перегляду об'єктів призначено для перегляду структури відповідного класу об'єктів, доступних при створенні програми.

Вікно <Object Browser> (Перегляд об'єктів) викликається командою <View Object Browser > (Вид, Перегляд об'єктів) або натисканням кнопки <Object Browser> (Перегляд об'єктів) на панелі інструментів. У цьому вікні приведений список всіх об'єктів, що мають у системі і які можна використовувати при створенні проекту. Вікно перегляду об'єктів складається з трьох основних частин:

- списку <Project/Library> (Проект/Бібліотека) у лівому верхньому куті вікна. У цьому списку, що розкривається, можна вибрати раз
- особисті проекти і бібліотеки об'єктів. Зокрема, бібліотеки об'єктів Excel, VBA, Office і VBA Project (об'єкти користувальницького проекту). Вибір у списку рядка <All Libraries > (Усі бібліотеки) відображає список об'єктів усіх бібліотек;
- списку < Classes > (Класи). Після вибору зі списку, що розкривається, <Project/Library> (Проект/Бібліотека) бібліотеки, що переглядається, наприклад VBA, усі класи об'єктів обраної библи
- набряки виводяться в списку < Classes > (Класи)
- ; - списку <Members> (Компоненти). Після вибору класу зі списку <Classes> (Класи) бібліотеки, що переглядається, наприклад FileSystem, усі компоненти обраного класу виводяться в списку < Members > (Компоненти). При виділенні рядка в цьому списку
- в нижній частині вікна < Object Browser > (Перегляд об'єктів) приводиться додаткова інформація про обраний компонент.

Це вікно можна використовувати також для пошуку методу або властивості об'єкта.

РОЗДІЛ 3. ТЕОРЕТИЧНА ЧАСТИНА

3.1. Об'єктно-орієнтована мова програмування високого рівня VBA

В даний час при розробці програм використовується технологія об'єктно-орієнтованого програмування (ООП). Об'єктно-орієнтована програма складається з об'єктів. VBA – це об'єктно-орієнтована мова програмування високого рівня. У кожному VBA-додатку є свій унікальний набір об'єктів з їх власними властивостями (характеристиками стану) і методами (командами обробки цих властивостей). Об'єкти, що мають загальні властивості і поведінку (події й методи), поєднуються в класи.

Клас – це опис або визначення об'єкту, а об'єкт – реалізація такого визначення, що зветься екземпляром. Клас визначає призначення об'єкту, його властивості і ті дії, які можуть бути виконані над об'єктом. Ім'я класу вказує на тип об'єктів, що він містить. Класи можна розглядати як основу для створення інших об'єктів цього ж типу, тому всі об'єкти одного класу будуть діяти однаково. У системі зберігається програмний опис кожного класу, на підставі якого, при необхідності, створюються екземпляри класу.

Властивість – це характеристика об'єкту. Методом називається дія, яку можна виконати з об'єктом. Подія – це деяка дія, яку можна виконувати з об'єктом і на яку треба програмувати відгук (у відповідь на реакцію системи). Розглянемо процес створення процедури обробки події, яка відбувається при клацанні по кнопці, що розміщена на робочому листі Excel. Спочатку треба визвати Excel і на поточному робочому листі за допомогою панелі інструментів Элементы управления розмістити кнопку, потім задати їй властивості. Для цього слід, знаходячись на кнопці, викликати контекстне меню, вибрати Свойства. На екрані з'явиться Окно свойств Properties, у якому перераховані всі властивості для вибраного елемента. У цьому вікні можна змінити потрібну властивість. Для кнопки основними властивостями є: Name – ім'я кнопки, що використовується при зверненні до кнопки в програмі на VBA (наприклад,

CmdПример1); Caption – напис на кнопці. Процедура обробки події описує ті дії, які необхідно виконати програмним шляхом у відповідь на реакцію користувача (у нашому прикладі – клацання по кнопці). Для завдання процедури необхідно, знаходячись в режимі Конструктор, встановити покажчик на кнопку, виконати подвійне клацання. Автоматично відкриється вікно програми, в якому будуть задані перший і останній оператори процедури. Текст програми набирається між ними.

Програмний код VBA складається з наступних блоків. Оператор – це найменша одиниця VBA-коду. Він призначений для визначення змінної, установки параметрів або виконання якої-небудь дії в програмі. Допустимо два або декілька операторів записувати в одному рядку. Тоді вони розділяються двокрапкою. Процедура – це окрема одиниця програмного коду VBA, яка має ім'я і може виконуватися самостійно. Будь-яка процедура містить один або декілька операторів. Модуль – це іменована одиниця, що складається з однієї або декількох процедур і розділу оголошень, в якому оголошуються змінні, константи і типи даних користувача, а також встановлюються параметри компілятора. Проект – включає всі модулі, форми і пов'язані з додатком об'єкти, що відносяться до конкретного документа, причому проект зберігається разом з самим цим документом. Слід відмітити, що програма не є самостійним структурним елементом мови VBA, і тому редактор VBA розпізнає не програми, а процедури, модулі і проекти, які мають ім'я. Будь-яка програма на VBA обов'язково містить не менше однієї процедури, тому що компілятор мови VBA може виконувати тільки оператори, що знаходяться в процедурі. Процедура може, у свою чергу, звертатися до однієї або декількох інших процедур, що знаходяться в одному або декількох модулях, які входять до складу одного або декількох проектів. Іншими словами, в VBA рядки програмного коду організовані в процедури, що розміщуються в модулях, а модулі розміщуються в проектах.

3.2. Алгоритм роботи тренажеру з теми «Події Excel VBA» дистанційного навчального курсу «Організація та обробка електронної інформації»

Якщо запустити створене програмне забезпечення, на головній формі користувач побаче інформацію:

- Про розробника тренажеру;
- Про наукового керівника студента;
- Тема тренажеру.

Якщо натиснути на кнопку яка розпочинає тренінг, користувачу висвітиться перший крок алгоритму тренажеру, в якому висвітиться перше питання з лекційного матеріалу теми.

Перший крок. Користувачу відобразиться питання з теми, на нього можна дати відповідь натиснувши лише на один варіант відповіді: «Подія – це...»

1. Дія, яка може викликати виконання зазначеного макросу; (вірна відповідь)
2. Рядок символів, який відрізняє його від змінних та об'єктів програми;
3. Це комірка пам'яті, що зберігає якесь одне значення.

Якщо користувач вибрав правильний варіант відповіді, то він може перейти на наступний крок. Якщо користувач вибрав неправильний варіант відповіді, то виводиться помилка: «Ваша відповідь невірна! Спробуйте ще раз!»

Другий крок. Користувачу відобразиться питання з теми, на нього можна дати відповідь натиснувши лише на один варіант відповіді: «Що таке подія в Excel VBA?»

1. Це місце, де ви можете зберігати дані;
2. Це процес, який автоматично відбувається при певному дії користувача або програми; (вірна відповідь)

3. Це функція, яка обчислює значення.

Якщо користувач вибрав правильний варіант відповіді, то він може перейти на наступний крок. Якщо користувач вибрав неправильний варіант відповіді, то виводиться помилка: «Ваша відповідь невірна! Спробуйте ще раз!»

Третій крок. Користувачу відобразиться питання з теми, на нього можна дати відповідь натиснувши лише на один варіант відповіді: «Як створити подію у Excel VBA? »

1. Події створюються автоматично;
2. Використовуючи команду Insert/Procedure;
3. Використовуючи ключове слово "Event" у коді VBA. (вірна відповідь)

Якщо користувач вибрав правильний варіант відповіді, то він може перейти на наступний крок. Якщо користувач вибрав неправильний варіант відповіді, то виводиться помилка: «Ваша відповідь невірна! Спробуйте ще раз!»

Четвертий крок. Користувачу відобразиться питання з теми, на нього можна дати відповідь натиснувши лише на один варіант відповіді: «Яка подія спрацьовує при відкритті книги Excel?»

1. Worksheet_Activate;
2. Worksheet_Change;
3. Workbook_Open (вірна відповідь)

Якщо користувач вибрав правильний варіант відповіді, то він може перейти на наступний крок. Якщо користувач вибрав неправильний варіант відповіді, то виводиться помилка: «Ваша відповідь невірна! Спробуйте ще раз!»

П'ятий крок. Користувачу відобразиться питання з теми, на нього можна дати відповідь натиснувши лише на один варіант відповіді: «Як назвати підпрограму, що викликається під час спрацьовування події?»

1. Event_Procedure; (вірна відповідь)
2. Event_Handler;
3. Event_Sub.

Якщо користувач вибрав правильний варіант відповіді, то він може перейти на наступний крок. Якщо користувач вибрав неправильний варіант відповіді, то виводиться помилка: «Ваша відповідь невірна! Спробуйте ще раз!»

Шостий крок. Користувачу відобразиться питання з теми, на нього можна дати відповідь натиснувши лише на один варіант відповіді: «Яка подія спрацьовує при видаленні аркуша?»

1. Worksheet_Change;
2. Workbook_SheetDelete; (вірна відповідь)
3. Workbook_Open.

Якщо користувач вибрав правильний варіант відповіді, то він може перейти на наступний крок. Якщо користувач вибрав неправильний варіант відповіді, то виводиться помилка: «Ваша відповідь невірна! Спробуйте ще раз!»

Сьомий крок. Користувачу відобразиться питання з теми, на нього можна дати відповідь натиснувши лише на один варіант відповіді: «Яка подія спрацьовує при закритті книги Excel?»

1. Worksheet_Change;

2. Workbook_Open;

3. Workbook_BeforeClose (вірна відповідь)

Якщо користувач вибрав правильний варіант відповіді, то він може перейти на наступний крок. Якщо користувач вибрав неправильний варіант відповіді, то виводиться помилка: «Ваша відповідь невірна! Спробуйте ще раз!»

Восьмий крок. Користувачу відобразиться питання з теми, на нього можна дати відповідь натиснувши лише на один варіант відповіді: «Яка подія спрацьовує при переміщенні аркуша в іншу позицію?»

1. Worksheet_Move; (вірна відповідь)

2. Workbook_Open;

3. Workbook_SheetActivate.

Якщо користувач вибрав правильний варіант відповіді, то він може перейти на наступний крок. Якщо користувач вибрав неправильний варіант відповіді, то виводиться помилка: «Ваша відповідь невірна! Спробуйте ще раз!»

Дев'ятий крок. Користувачу відобразиться питання з теми, на нього можна дати відповідь натиснувши лише на один варіант відповіді: «Яка подія спрацьовує при вставці нового рядка у діапазон даних?»

1. Workbook_Open;

2. Worksheet_BeforeInsert; (вірна відповідь)

3. Worksheet_Change.

Якщо користувач вибрав правильний варіант відповіді, то він може перейти на наступний крок. Якщо користувач вибрав неправильний варіант відповіді, то виводиться помилка: «Ваша відповідь невірна! Спробуйте ще раз!»

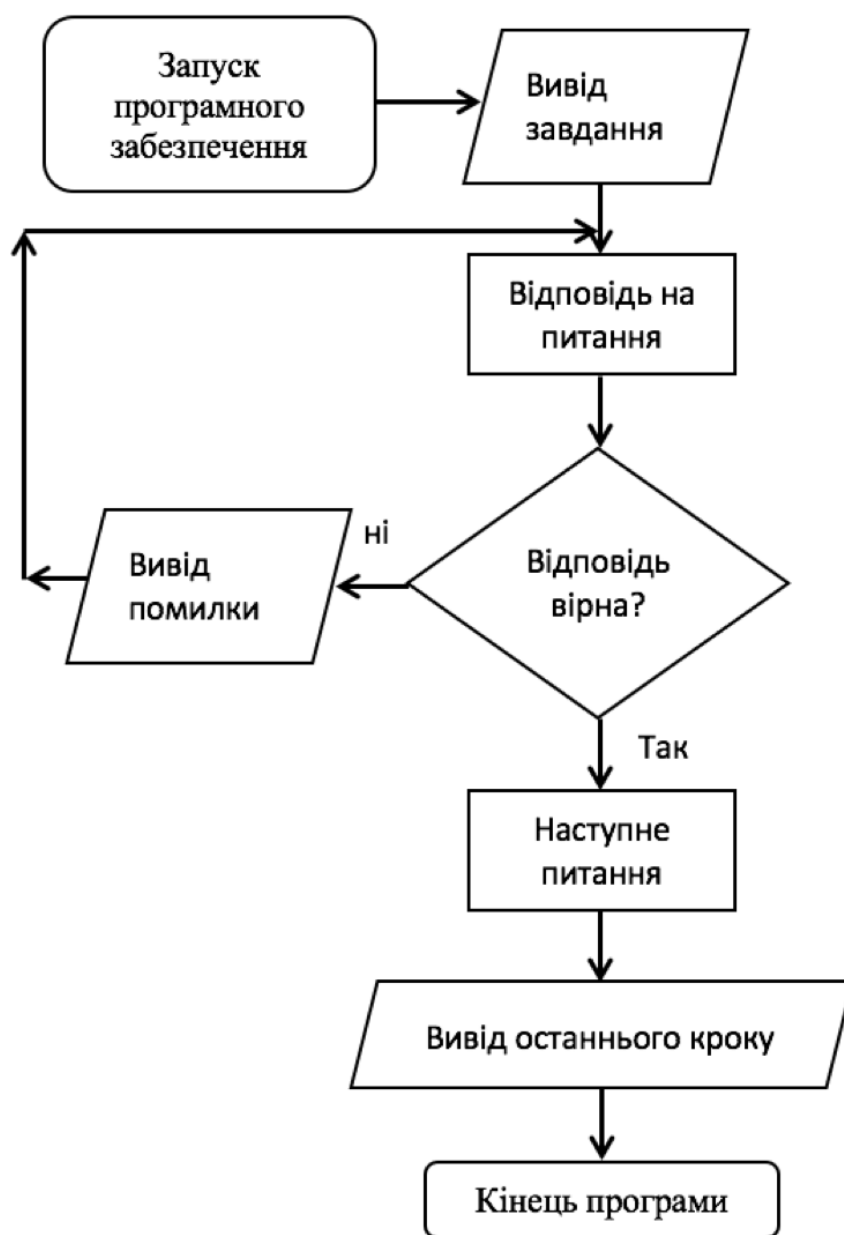
Десятий крок. Користувачу відобразиться питання з теми, на нього можна дати відповідь натиснувши лише на один варіант відповіді: «Щоб відкрити вікно програмного коду потрібно...»

1. Один раз клікнути правою клав'яшею мишки по об'єкту;
2. вибрати Insert - User Form;
3. Двічі клікнути лівою клав'яшею мишки по об'єкту. (вірна відповідь)

Якщо користувач вибрав правильний варіант відповіді, то він може перейти на наступний крок. Якщо користувач вибрав неправильний варіант відповіді, то виводиться помилка: «Ваша відповідь невірна! Спробуйте ще раз!»

Після десятого кроку, відкривається підсумкова сторінка тренажеру, на якій написано: «Вітаємо! Ви успішно пройшли тренажер з теми «Події Excel VBA» дистанційного навчального курсу «Організація та обробка електронної інформації».

3.3. Блок-схема тренажеру



РОЗДІЛ 4. ПРАКТИЧНА ЧАСТИНА

4.1. Обґрунтування вибору програмних засобів

Робота з Visual Basic передбачає використання візуальних компонентів та контролів, визначення атрибутів та дій для компонентів, створення додаткового коду для розширення функціональних можливостей.

Завдяки певним значенням і діям для компонентів з'являється можливість розробити просту програму без написання коду фахівцем. У попередніх версіях були деякі проблеми пов'язані з продуктивністю програм, проте нинішні комп'ютери і компіляція власного коду дозволяють знизити негативний вплив цього фактора.

Створення форм здійснюється способом drag and drop, що в перекладі означає «перетягни та кинь». Елементи управління розміщуються на формі (наприклад, поля введення, кнопки та ін.). У цих елементів управління є власні атрибути та обробники подій. Багато атрибутів передбачають можливість зміни в процесі виконання програми, тому можна розробляти програми, які динамічно реагуватимуть на дії користувача.

VB може створити програми, що виконуються (EXE файли), елементи керування ActiveX, бібліотеки DLL. Але все ж таки призначенням даної мови є створення програм під OS Windows.

4.2. Опис програмної реалізації

Елементи тренажеру було реалізовано на мові програмування Visual Basic, в середовищі розробки Visual Studio. На самому початку розробки створюємо новий проект Windows Forms (рис. 4.1)

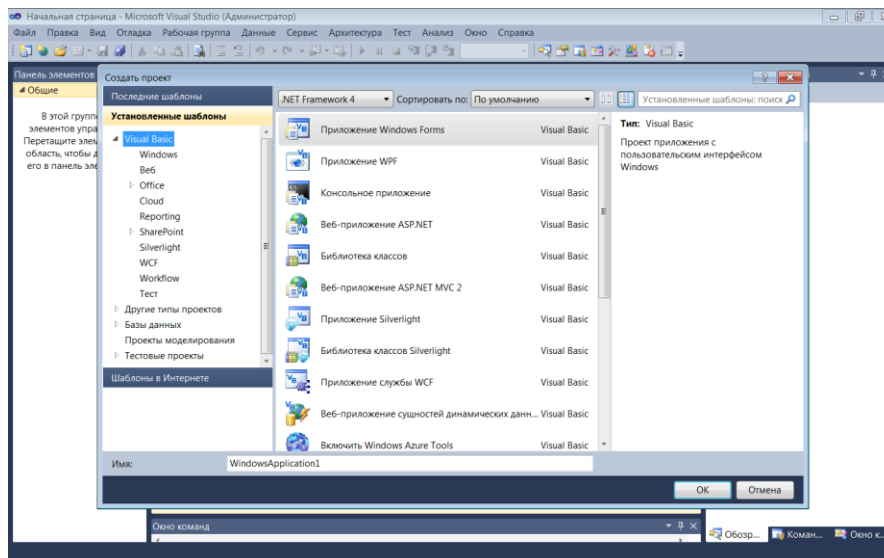


Рисунок 4.1 – створення проекту Windows Forms.

Після створення нового проекту створюється початкова форма тренажеру (рис. 4.2)

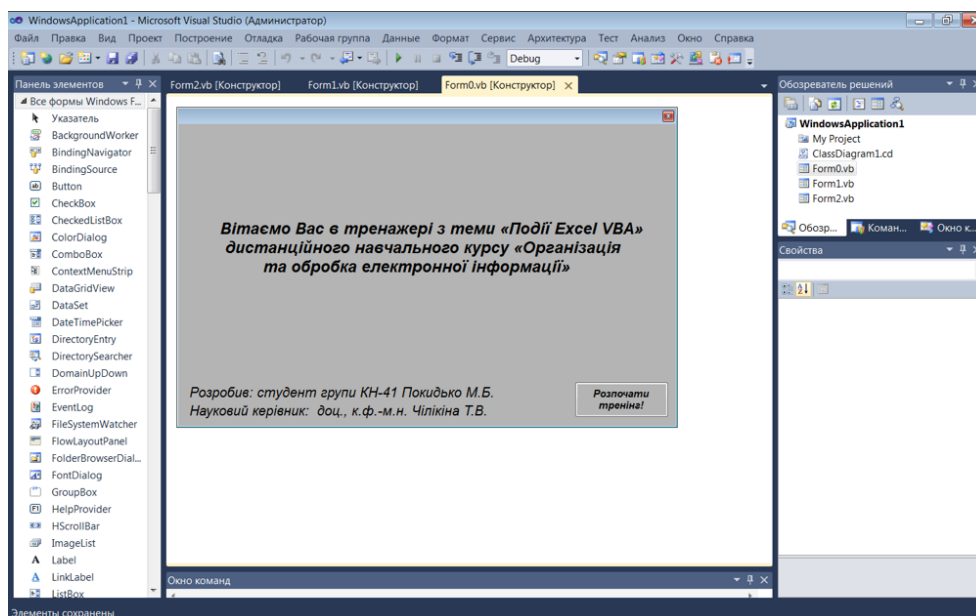


Рисунок 4.2 – початкова форма тренажеру.

Після створення початкової форми тренажеру, потрібно запрограмувати перше питання з теми тренажеру, тобто це буде перший крок з алгоритму (рис. 4.3)

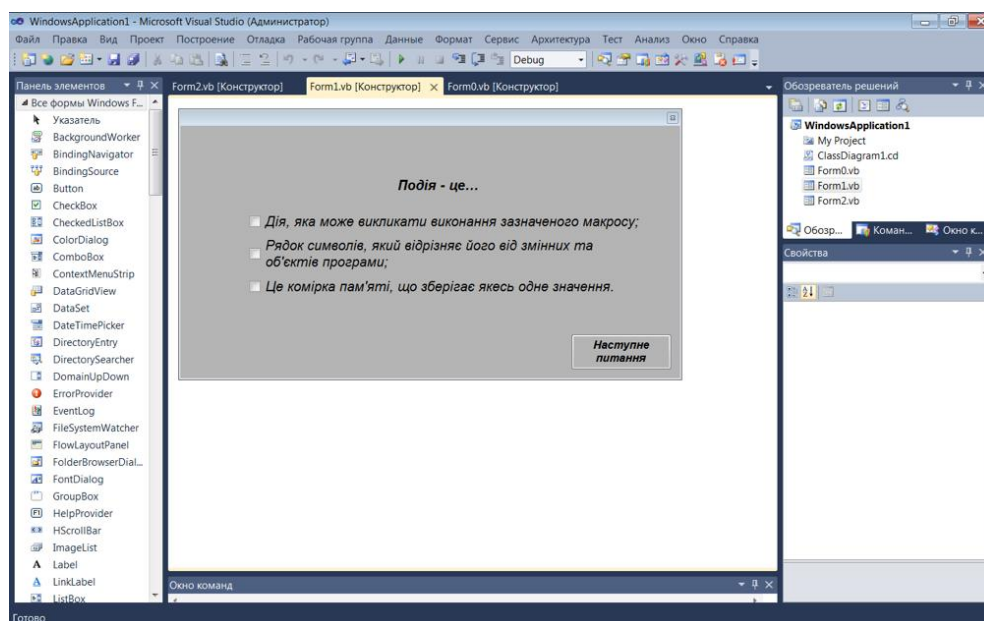


Рисунок 4.3 – перший крок з алгоритму тренажеру.

Потім, після першого завдання, потрібно програмно розробити наступне питання з теми тренажеру. (рис. 4.4)

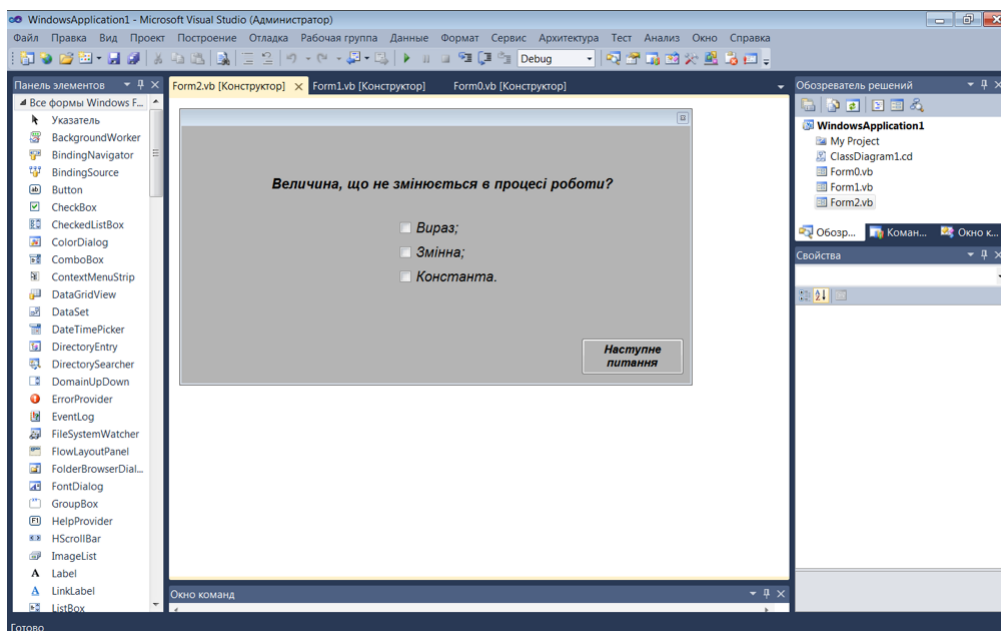


Рисунок 4.4 – наступний крок алгоритму тренажеру.

За всю логіку роботи тренажеру та програмну реалізацію, відповідає наступний фрагмент програмного коду:

Public Class Form0

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    Form1.Show()
End Sub
```

```
Private Sub Start_screen_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
```

```
End Sub
End Class
```

Public Class Form1

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
    If CheckBox1.Checked = True And CheckBox2.Checked = False Then
        MsgBox("Вітаю! Ваша відповідь правильна!")
        Form2.Show()
    Else
        MsgBox("Неправильно! Спробуйте ще раз!")
        CheckBox1.Checked = False
        CheckBox2.Checked = False
        CheckBox3.Checked = False
    End If
End Sub
```

```
Private Sub Step1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
    Form0.Hide()
End Sub
```

```
Private Sub CheckBox1_CheckedChanged(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles CheckBox1.CheckedChanged
    If CheckBox1.Checked = True Then
        CheckBox2.Enabled = False
        CheckBox3.Enabled = False
    End If
    If CheckBox1.Checked = False Then
        CheckBox2.Enabled = True
```

```
        CheckBox3.Enabled = True
    End If
End Sub
```

```
Private Sub CheckBox2_CheckedChanged(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles CheckBox2.CheckedChanged
    If CheckBox2.Checked = True Then
        CheckBox1.Enabled = False
        CheckBox3.Enabled = False
    End If
    If CheckBox2.Checked = False Then
        CheckBox1.Enabled = True
        CheckBox3.Enabled = True
    End If
End Sub
```

```
Private Sub CheckBox3_CheckedChanged(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles CheckBox3.CheckedChanged
    If CheckBox3.Checked = True Then
        CheckBox1.Enabled = False
        CheckBox2.Enabled = False
    End If
    If CheckBox3.Checked = False Then
        CheckBox1.Enabled = True
        CheckBox2.Enabled = True
    End If
End Sub
End Class
```

4.3. Опис роботи тренажеру

Після запуску тренажеру користувачу відкривається початкова сторінка (рис. 4.5)

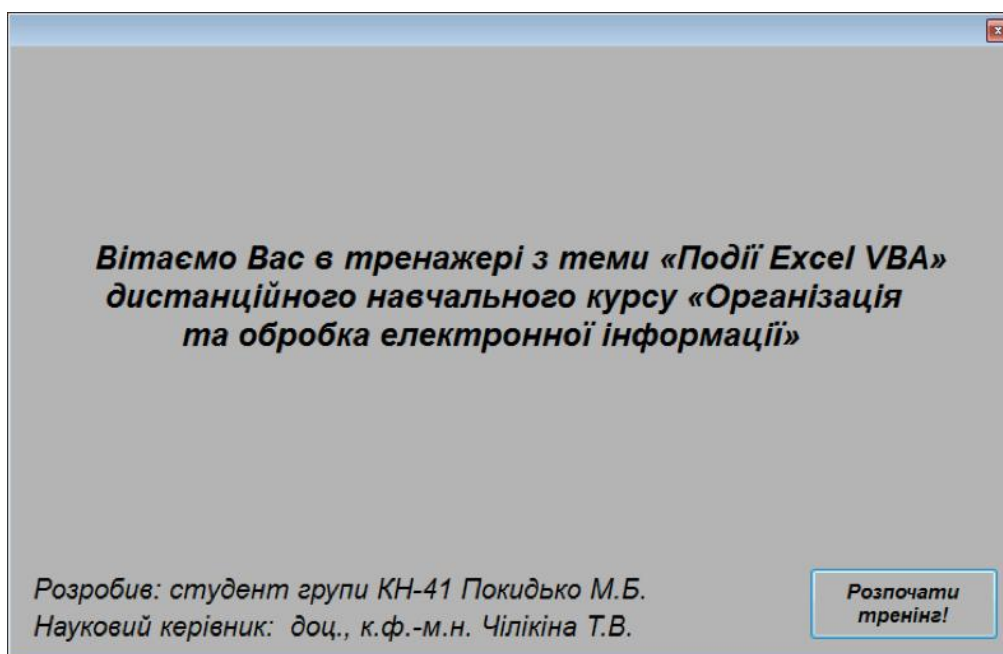


Рисунок 4.5 – початкова сторінка.

Після натискання кнопки на вікні «Розпочати тренінг!», користувачу відкривається перше питання, він може відповісти на це питання натиснувши на будь який варіант відповіді (рис. 4.6)

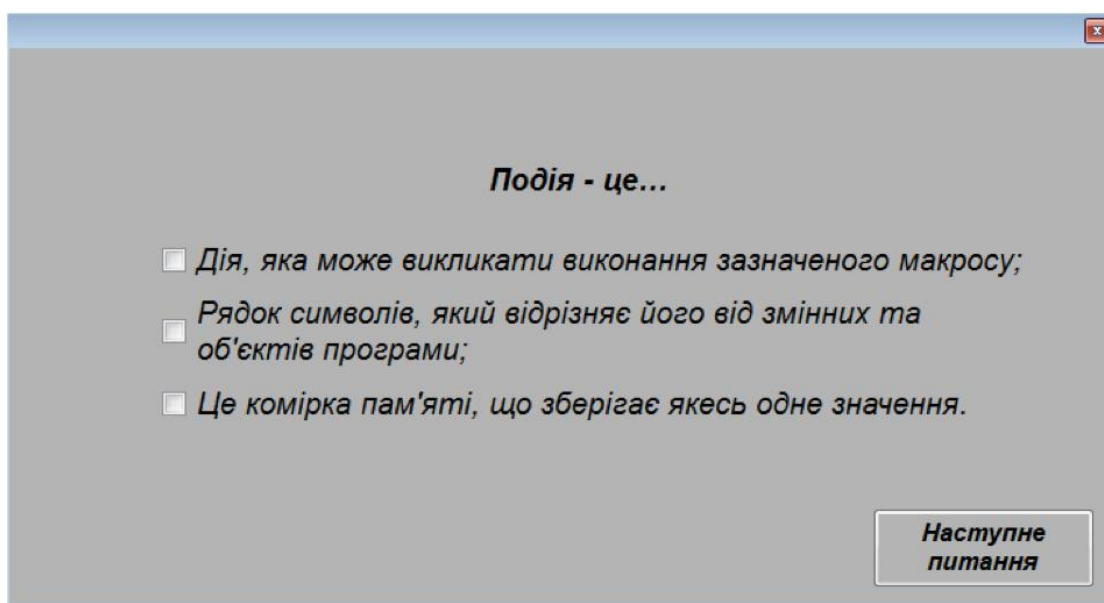


Рисунок 4.6 – перше питання.

Після вибору відповіді, користувач натискає кнопку «Наступне питання», якщо вибрана відповідь вірна, висвітиться вікно «Ваша відповідь вірна» (рис. 4.7)

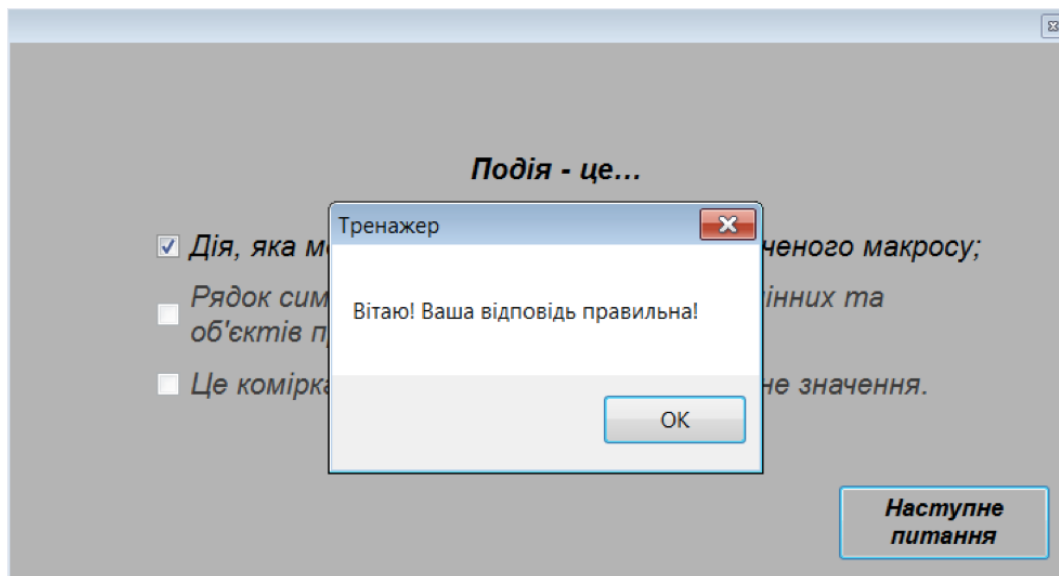


Рисунок 4.7 – вікно вірної відповіді.

Якщо користувач відповів невірно, висвітиться помилка (рис. 4.8)

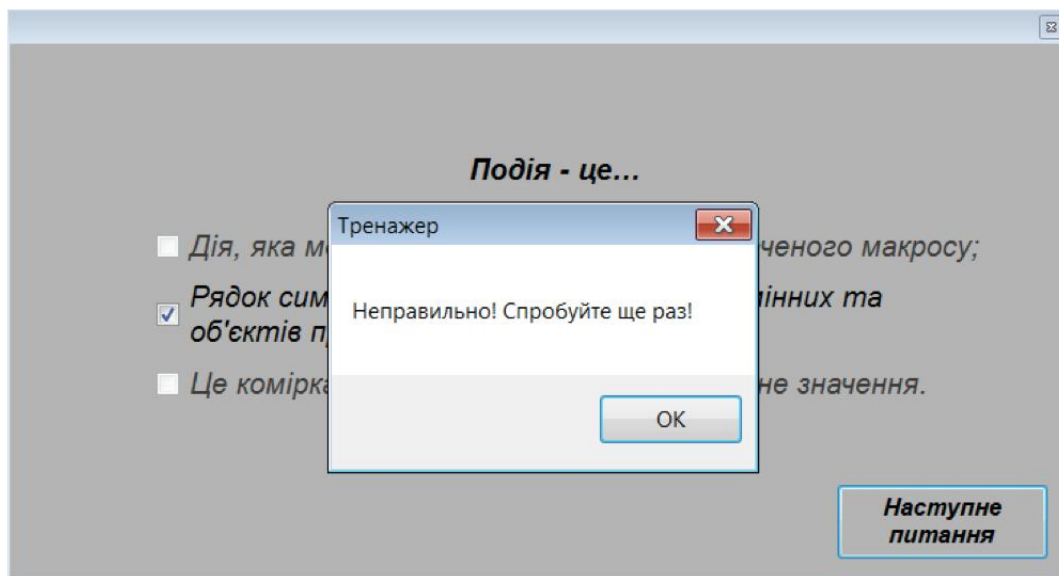
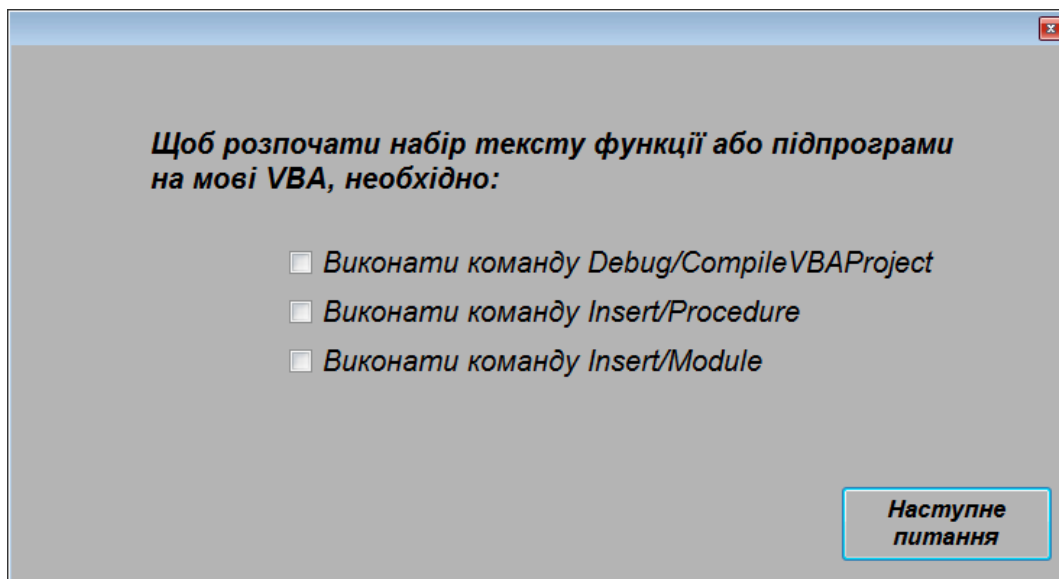


Рисунок 4.8 – вікно помилки.

Питання тренажеру 3 (рис. 4.9)



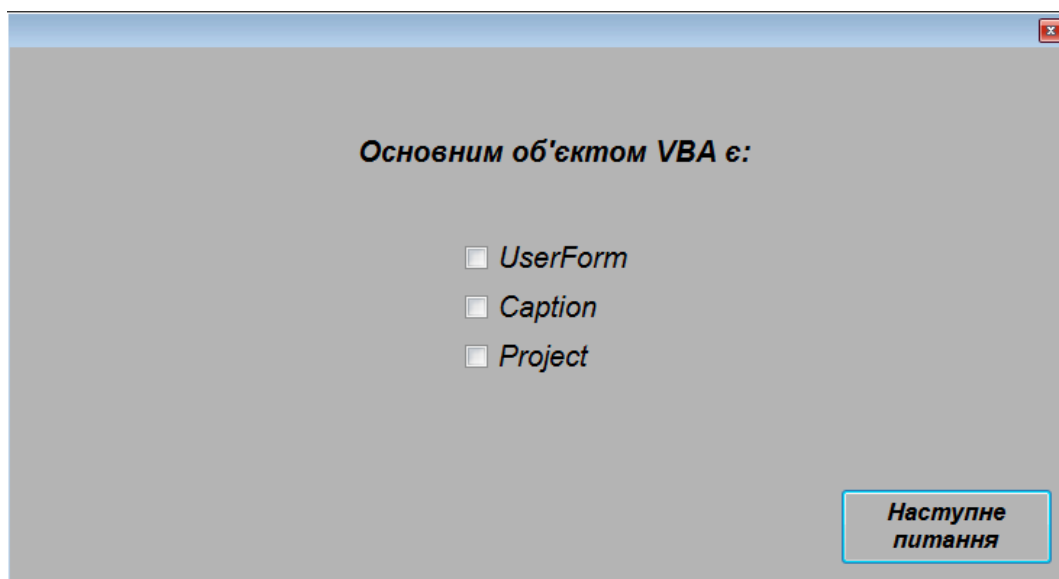
Щоб розпочати набір тексту функції або підпрограми на мові VBA, необхідно:

- ☐ Виконати команду *Debug/CompileVBAProject*
- ☐ Виконати команду *Insert/Procedure*
- ☐ Виконати команду *Insert/Module*

Наступне питання

Рисунок 4.9 – питання 3.

Питання тренажеру 4 (рис. 5.0)



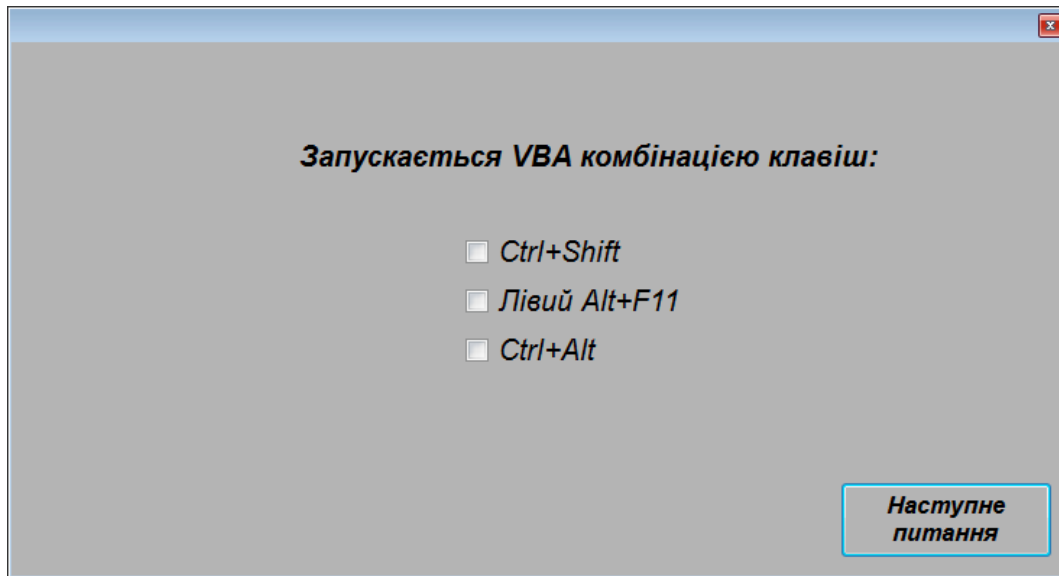
Основним об'єктом VBA є:

- ☐ *UserForm*
- ☐ *Caption*
- ☐ *Project*

Наступне питання

Рисунок 5.0 – питання 4.

Питання тренажеру 5 (рис. 5.1)



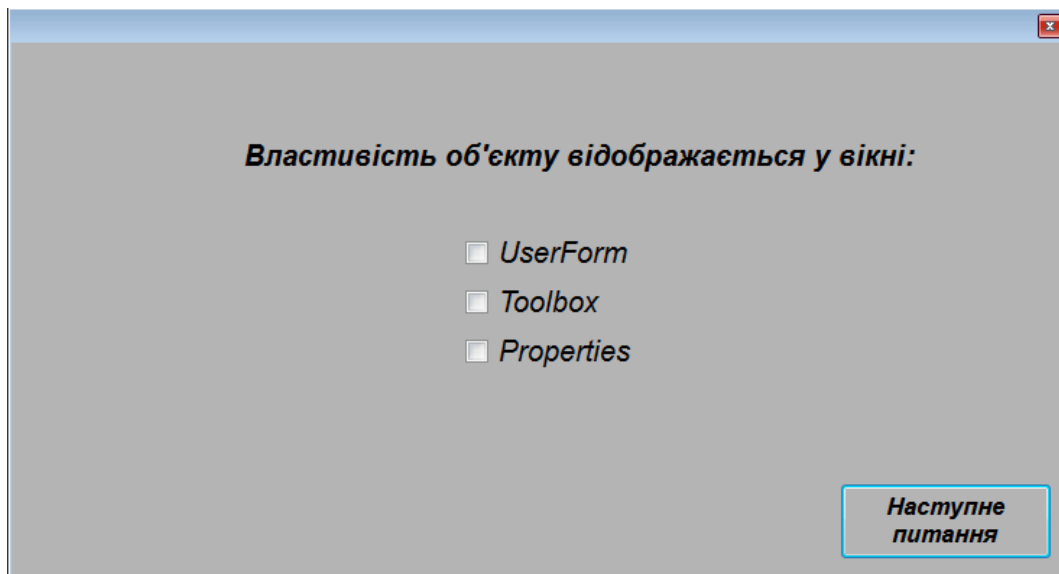
Запускається VBA комбінацією клавіш:

- ☐ Ctrl+Shift
- ☐ Лівий Alt+F11
- ☐ Ctrl+Alt

Наступне питання

Рисунок 5.1 – питання 5.

Питання тренажеру 6 (рис. 5.2)



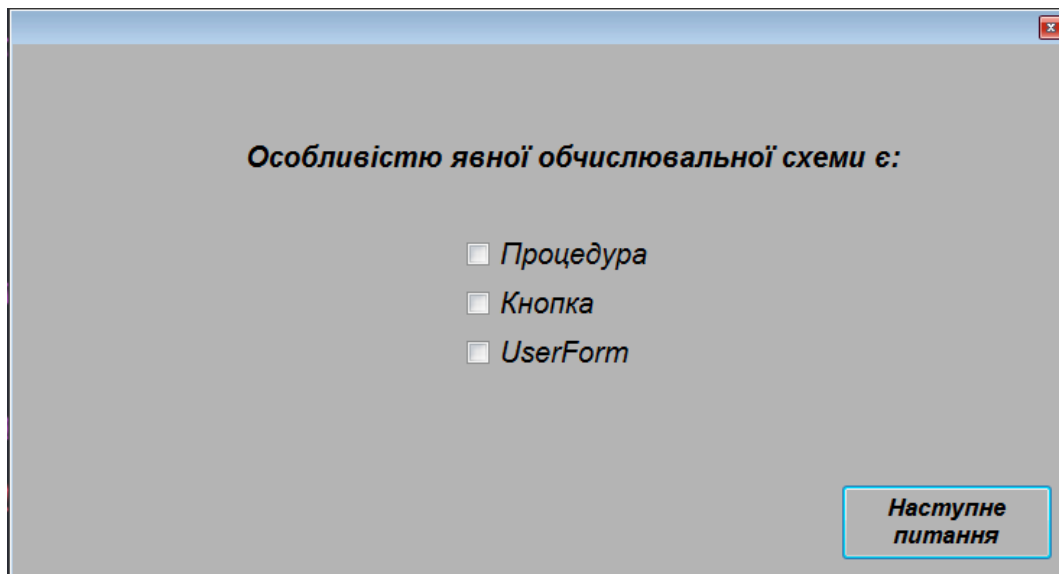
Властивість об'єкту відображається у вікні:

- ☐ UserForm
- ☐ Toolbox
- ☐ Properties

Наступне питання

Рисунок 5.2 – питання 6.

Питання тренажеру 7 (рис. 5.3)



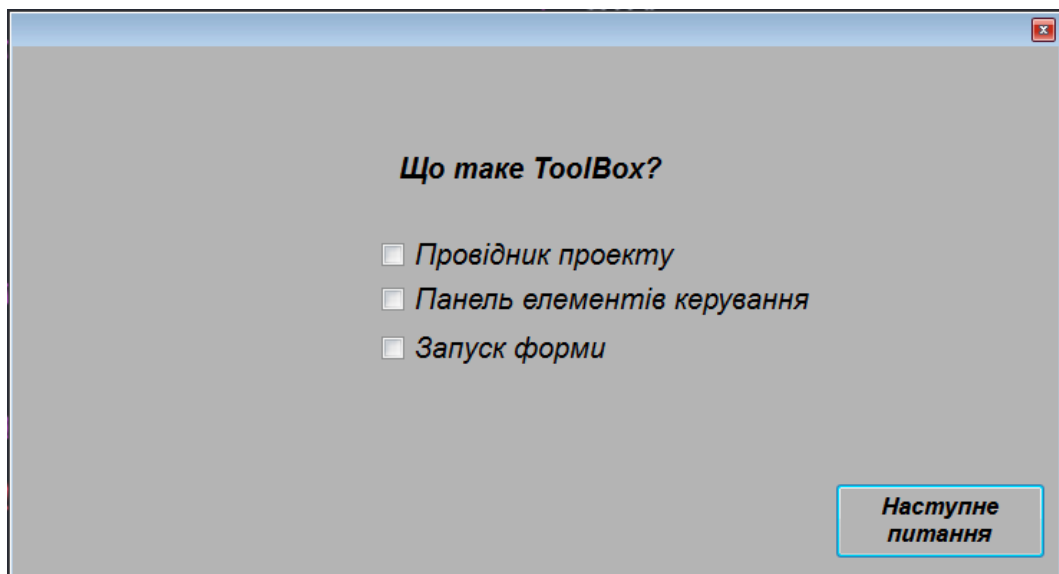
Особливістю явної обчислювальної схеми є:

- ☐ Процедура
- ☐ Кнопка
- ☐ UserForm

Наступне питання

Рисунок 5.3 – питання 7.

Питання тренажеру 8 (рис. 5.4)



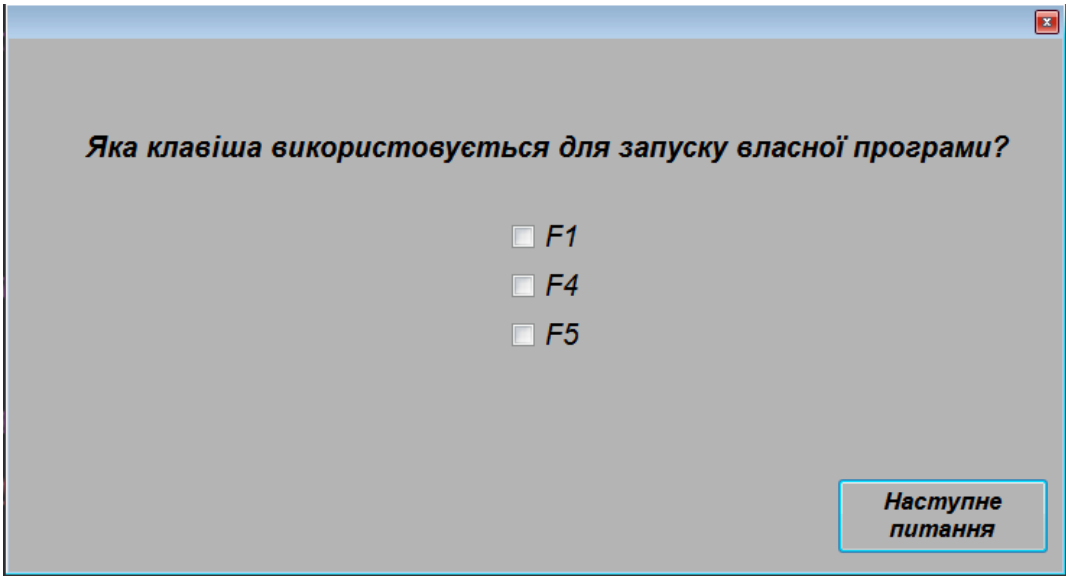
Що таке ToolBox?

- ☐ Провідник проекту
- ☐ Панель елементів керування
- ☐ Запуск форми

Наступне питання

Рисунок 5.4 – питання 8.

Питання тренажеру 9 (рис. 5.5)



Яка клавіша використовується для запуску власної програми?

☐ F1

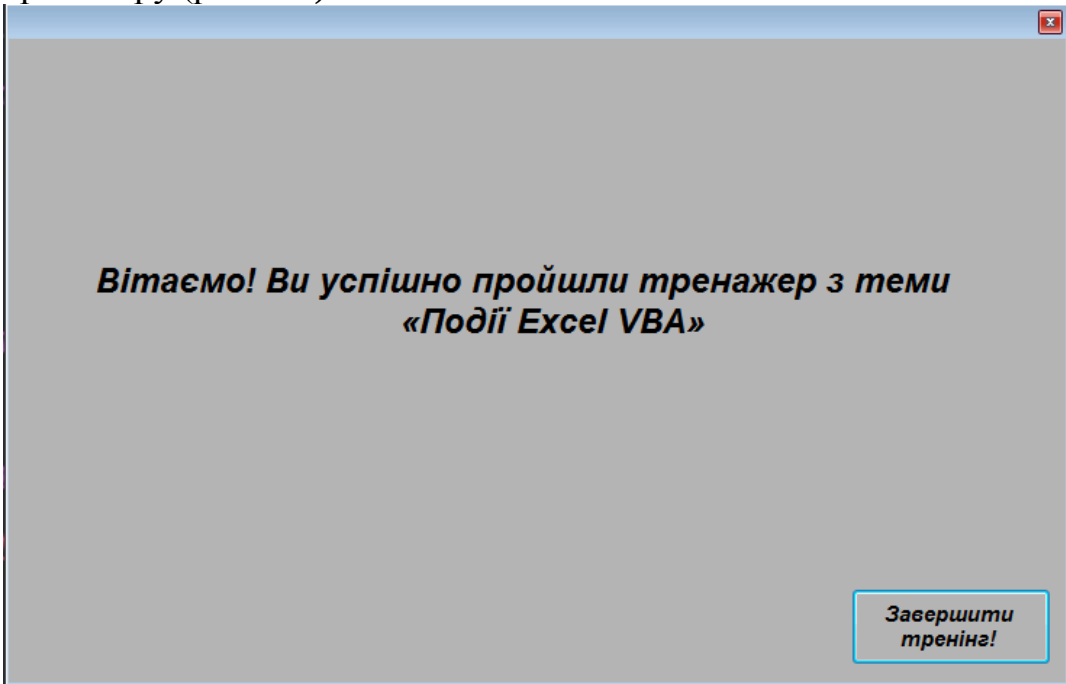
☐ F4

☐ F5

Наступне питання

Рисунок 5.5 – питання 9.

Після проходження тренінгу, користувачу відкривається кінцевий крок тренажеру (рис. 5.6)



**Вітаємо! Ви успішно пройшли тренажер з теми
«Події Excel VBA»**

Завершити тренінг!

Рисунок 5.6 – Кінцевий крок.

ВИСНОВКИ

Отже, організація навчального процесу у вищих навчальних закладах під час воєнного стану виявляється найбільш оптимальною за рахунок дистанційного навчання. При цьому досліджуючи специфіку режимів дистанційного навчання (синхронний, асинхронний чи біхронний) краще надати перевагу біхронному як режиму організації навчального процесу, який допоможе викладачеві врахувати ряд факторів та умов, що виникають внаслідок окупації частини територій України. Під час організації навчального процесу викладачеві треба враховувати категорії учнів в умовах війни: непереміщені, внутрішньо переміщені, зовнішньо переміщені. Проблеми технічного та ергономічного характеру також впливають на планування навчального процесу та окремих предметів. Частини цих проблем учителі можуть уникнути плануючи навчання в асинхронному режимі. Класичний синхронний режим навчання має низку вад, особливо під час оголошення повітряних тривог, територіальну віддаленість окремих студентів та викладачів (наприклад, різні часові пояси).

Результатом виконання кваліфікаційної роботи є розроблений навчальний тренажер з теми «Події Excel VBA» дистанційного навчального курсу «Організація та обробка електронної інформації».

Під час створення додатку було використано наступні програмні засоби: середовище візуальної розробки програм Visual Studio та мова програмування Visual Basic, редактор коду Visual Studio Code, графічний редактор Adobe Photoshop.

Завдання кваліфікаційної роботи виконано. Створено навчальний тренажер для підготовки студентів з теми «Події Excel VBA» дистанційного навчального курсу «Організація та обробка електронної інформації».

СПИСОК ЛІТЕРАТУРИ

1. Черненко О. О. Курсовий проєкт із фаху: методичні рекомендації щодо оформлення пояснювальних записок до курсового проєкту для студентів спеціальності 122 Комп'ютерні науки освітня програма «Комп'ютерні науки» ступеня бакалавра, магістра / О. О. Черненко. – Полтава : ПУЕТ, 2022. – 58 с. – 1 електрон. опт. диск (CVD-ROM).
2. Дистанційне навчання [Електронний ресурс]. – Режим доступу: <https://mon.gov.ua/ua/osvita/pozashkilna-osvita/distancijne-navchannya>
3. Дистанційна освіта [Електронний ресурс]. – Режим доступу: http://osvita.ch.ua/distanciyna_osvita/
4. Microsoft Visual Studio [Електронний ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/Microsoft_Visual_Studio
5. Microsoft Visual Basic [Електронний ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/Visual_Basic
6. Бібліографічний запис. Бібліографічний опис. Загальні вимоги та правила складання: ДСТУ 7.1-2006. – [Чинний від 2007-07-01]. – К. : Держспоживстандарт України, 2007. – 47 с.
7. Програмування на Visual BASIC [Електронний ресурс]. – Режим доступу: https://kursoviks.com.ua/bd_kompyuterni/article_post/65-lektsiya-osnovi-programuvannya-movoyu-visual-basic-6-0
8. Microsoft Visual Basic 2010 Step by Step [Електронний ресурс]. – Режим доступу: https://books.google.com.ua/books?hl=uk&lr=&id=Ap9CAwAAQBAJ&oi=fnd&pg=PT18&dq=visual+basic&ots=n5NfMNuX8q&sig=8OFek2tHiXjaKynsUtD6SgQKabY&redir_esc=y#v=onepage&q=visual%20basic&f=false
9. Randolph Visual Studio 2010 for professionals // Randolph, Nick, Gardner, David, Minutillo, Michael, Anderson, Chris.: Trans. with English - M.: LLC "I.D. Williams", 2011. - 1184 p.

10. Halvorson M. Microsoft Visual Basic 2005. Series "Step by step" / M. Halvorson. - ECOM Publishers. - 640 p.
11. Основи VBA [Електронний ресурс]. – Режим доступу: <https://onlinetestpad.com/ru/testresult/326660-osnovy-vba?res=lm6u4c5zrgzps>
12. Введення до VBA [Електронний ресурс]. – Режим доступу: <https://studfile.net/preview/5726056/>
13. Події VBA [Електронний ресурс]. – Режим доступу: <https://uk.excel-lib.net/11705335-excel-vba-events-an-easy-and-complete-guide>
14. Процедури та форми VBA [Електронний ресурс]. – Режим доступу: <https://ela.kpi.ua/bitstream/123456789/42865/1/IT-VBA2019.pdf>
15. Програмування на VBA [Електронний ресурс]. – Режим доступу: <https://iea.gov.ua/uchast-u-navchalnomu-treningu-nt-vba-for-excel-programuvannya-na-vba-dlya-microsoft-excel-kiyiv-9-11-grudnya-2019/>

ДОДАТОК А. КОД ПРОГРАМИ

```
Public Class Form0
```

```
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
        Form1.Show()
    End Sub
```

```
    Private Sub Start_screen_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
```

```
    End Sub
End Class
```

```
Public Class Form2
```

```
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
        If CheckBox3.Checked = True And CheckBox2.Checked = False Then
            MsgBox("Вітаю! Ваша відповідь правильна!")
            Form3.Show()
        Else
            MsgBox("Неправильно! Спробуйте ще раз!")
            CheckBox1.Checked = False
            CheckBox2.Checked = False
            CheckBox3.Checked = False
        End If
    End Sub
```

```
    Private Sub Step2_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
        Form1.Hide()
    End Sub
```

```
    Private Sub CheckBox1_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles CheckBox1.CheckedChanged
        If CheckBox1.Checked = True Then
            CheckBox2.Enabled = False
            CheckBox3.Enabled = False
        End If
        If CheckBox1.Checked = False Then
            CheckBox2.Enabled = True
            CheckBox3.Enabled = True
        End If
    End Sub
```

```
    Private Sub CheckBox2_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles CheckBox2.CheckedChanged
        If CheckBox2.Checked = True Then
            CheckBox1.Enabled = False
            CheckBox3.Enabled = False
        End If
    End Sub
```

```

End If
If CheckBox2.Checked = False Then
    CheckBox1.Enabled = True
    CheckBox3.Enabled = True
End If
End Sub

```

```

Private Sub CheckBox3_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
CheckBox3.CheckedChanged
    If CheckBox3.Checked = True Then
        CheckBox1.Enabled = False
        CheckBox2.Enabled = False
    End If
    If CheckBox3.Checked = False Then
        CheckBox1.Enabled = True
        CheckBox2.Enabled = True
    End If
End Sub
End Class

```

```

Public Class Form3

```

```

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
    If CheckBox3.Checked = True And CheckBox2.Checked = False Then
        MsgBox("Вітаю! Ваша відповідь правильна!")
        Form4.Show()
    Else
        MsgBox("Неправильно! Спробуйте ще раз!")
        CheckBox1.Checked = False
        CheckBox2.Checked = False
        CheckBox3.Checked = False
    End If
End Sub

```

```

Private Sub Step2_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
    Form2.Hide()
End Sub

```

```

Private Sub CheckBox1_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
CheckBox1.CheckedChanged
    If CheckBox1.Checked = True Then
        CheckBox2.Enabled = False
        CheckBox3.Enabled = False
    End If
    If CheckBox1.Checked = False Then
        CheckBox2.Enabled = True
        CheckBox3.Enabled = True
    End If
End Sub

```

```

Private Sub CheckBox2_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
CheckBox2.CheckedChanged

```

```
If CheckBox2.Checked = True Then
    CheckBox1.Enabled = False
    CheckBox3.Enabled = False
End If
If CheckBox2.Checked = False Then
    CheckBox1.Enabled = True
    CheckBox3.Enabled = True
End If
End Sub

Private Sub CheckBox3_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
CheckBox3.CheckedChanged
    If CheckBox3.Checked = True Then
        CheckBox1.Enabled = False
        CheckBox2.Enabled = False
    End If
    If CheckBox3.Checked = False Then
        CheckBox1.Enabled = True
        CheckBox2.Enabled = True
    End If
End Sub
End Class
```