

ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД УКООПСІЛКИ
«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)

«__» _____ 202_ р.

КВАЛІФІКАЦІЙНА РОБОТА

на тему

**РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТРЕНАЖЕРУ З ТЕМИ
«ПРОТОКОЛИ КОМП'ЮТЕРНИХ МЕРЕЖ»
ДИСТАНЦІЙНОГО КУРСУ «ІНФОРМАЦІЙНІ МЕРЕЖІ»**

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

Виконавець роботи Луковець Ігор Олексійович
_____ «__» _____ 202_ р.
(підпис)

Науковий керівник к. ф.-м. н., доцент, Ольховська Олена Володимирівна
_____ «__» _____ 202_ р.
(підпис)

Рецензент к. ф.-м. н., доцент, _____
_____ «__» _____ 202_ р.
(підпис)

ПОЛТАВА 2023

ЗАВДАННЯ І КАЛЕНДАРНИЙ ГРАФІК ВИКОНАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ

на тему Розробка програмного забезпечення тренажеру з теми «протоколи комп'ютерних мереж» дистанційного курсу «інформаційні мережі»

зі спеціальності 122 Комп'ютерні науки освітня

програма «Комп'ютерні науки» ступеня бакалавр

Прізвище, ім'я, по батькові Луковець Ігор Олексійович

Затверджена наказом ректора № 141-Н від «__» _____ 2022 р.

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

Вихідні дані до кваліфікаційної роботи: публікації з теми, навчальні тренажери в дистанційних курсах із комп'ютерних наук.

Зміст пояснювальної записки (перелік питань, які потрібно розробити)

ВСТУП

1. ПОСТАНОВКА ЗАДАЧІ

2. ІНФОРМАЦІЙНИЙ ОГЛЯД

2.1. Протоколи локальних мереж

3. ТЕОРЕТИЧНА ЧАСТИНА

3.1. Протоколи, інтерфейси мереж

3.2. Алгоритм роботи тренажеру з теми «Протоколи комп'ютерних мереж» дистанційного курсу «Інформаційні мережі»

3.3. Блок-схема

3.4. Фізична структура тренажеру

3.5. Логічна структура тренажеру

4. ПРАКТИЧНА ЧАСТИНА

4.1. Опис програмної реалізації

4.2. Інструкція по використанню навчального тренажеру

4.3. Обґрунтування вибору програмних засобів

ВИСНОВОК

Консультанти розділів кваліфікаційної роботи

Розділ	ПІБ, посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв
Постанова задачі	Ольховська О. В.		
Інформаційний огляд	Ольховська О. В.		
Теоретична частина	Ольховська О. В.		
Практична реалізація	Ольховська О. В.		

Календарний графік виконання кваліфікаційної роботи

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

Дата видачі завдання «__» _____ 202_ р.

Здобувач вищої освіти Луковець Ігор Олексійович

Науковий керівник к. ф.-м. н., Ольховська О. В.

Результати захисту кваліфікаційної роботи

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

Протокол засідання ЕК № _____ від «__» _____ 202_ р.

Секретар ЕК _____
(підпис) (ініціал та прізвище)

Затверджую

Зав. кафедрою _____

к. ф.-м. н. Олена ОЛЬХОВСЬКА

«__» _____ 202_ р.

Погоджено

Науковий керівник _____

к. ф.-м. н. Олена ОЛЬХОВСЬКА

«__» _____ 202_ р.

План

кваліфікаційної роботи на тему:

«Розробка програмного забезпечення тренажеру з теми «Протоколи комп'ютерних мереж» дистанційного курсу «Інформаційні мережі»

зі спеціальності 122 Комп'ютерні науки

освітня програма 122 «Комп'ютерні науки» ступеня бакалавр

Прізвище, ім'я, по батькові Луковець Ігор Олексійович

ВСТУП

1. ПОСТАНОВКА ЗАДАЧІ

2. ІНФОРМАЦІЙНИЙ ОГЛЯД

2.1. Протоколи локальних мереж

3. ТЕОРЕТИЧНА ЧАСТИНА

3.1. Протоколи, інтерфейси мереж

3.2. Алгоритм роботи тренажеру з теми «Протоколи комп'ютерних мереж» дистанційного курсу «Інформаційні мережі»

3.3. Блок-схема

3.4. Фізична структура тренажеру

3.5. Логічна структура тренажеру

4. ПРАКТИЧНА ЧАСТИНА

4.1. Опис програмної реалізації

4.2. Інструкція по використанню навчального тренажеру

4.3. Обґрунтування вибору програмних засобів

ВИСНОВОК

Здобувач вищої освіти _____ Ігор ЛУКОВЕЦЬ

«_____» _____ 202_ р.

РЕФЕРАТ

Записка: 52 с., 33 рис., 1 таблиця, 1 додаток, 20 джерел.

ПРОТОКОЛИ КОМП'ЮТЕРНИХ МЕРЕЖ, НАВЧАЛЬНИЙ ТРЕНАЖЕР,
ІНФОРМАЦІЙНІ МЕРЕЖІ.

Об'єкт розробки – алгоритм та програмна реалізація програмного забезпечення тренажеру з теми «Протоколи комп'ютерних мереж» дистанційного курсу «Інформаційні мережі».

Мета роботи – Алгоритмізація та програмування програмного забезпечення тренажеру з теми «Протоколи комп'ютерних мереж» дистанційного курсу «Інформаційні мережі».

Методи дослідження – Інформаційні мережі, середовище візуальної розробки програм Visual Studio та мова програмування Visual Basic.

Зроблено інформаційний огляд матеріалу з теми «Протоколи комп'ютерних». Розроблено алгоритм навчального тренажера з теми «Протоколи комп'ютерних», створено блок-схему програмного забезпечення. Програмно реалізовано тренажер з дисципліни «Інформаційні мережі».

Зміст

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ, ТЕРМІНІВ	4
ВСТУП.....	5
1. ПОСТАНОВКА ЗАДАЧІ РОЗРОБКИ.....	7
2. ІНФОРМАЦІЙНИЙ ОГЛЯД	8
2.1. Протоколи локальних мереж	8
3. ТЕОРЕТИЧНА ЧАСТИНА.....	19
3.1. Протоколи, інтерфейси мереж.....	19
3.2. Алгоритм роботи тренажеру з теми «Протоколи комп'ютерних мереж» дистанційного курсу «Інформаційні мережі»	31
4. ПРАКТИЧНА ЧАСТИНА.....	38
4.1. Опис програмної реалізації.....	38
4.2. Інструкція по використанню навчального тренажеру	41
4.3. Обґрунтування вибору програмних засобів.....	48
ВИСНОВКИ.....	50
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	51
ДОДАТОК А. КОД ПРОГРАМИ.....	54

**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.
Label	Для розміщення у формі звичайного тексту (заголовків, написів до полів, іншої інформації) призначений елемент управління Label
Button	Кнопка управління, яку можна натиснути, реалізовано це за допомогою події Click.
VB	Visual Basic — засіб розробки програмного забезпечення, створений і підтримуваний корпорацією Microsoft.
CheckBox	Прапорці дозволяють користувачеві дати відповідь на поставлене запитання. У разі позитивної відповіді користувач встановлює прапорець, і він набуває вигляду квадрата, в якому розміщена галочка.

ВСТУП

Становлення інформаційного суспільства є однією з характерних особливостей сучасного періоду розвитку України. Не викликає сумнівів твердження, що підґрунтям трансформації її системи освіти, зокрема вищої освіти, становить масштабність застосування інформаційних технологій та динамічних дидактичних методик. Концептуальні зрушення у засобах отримання інформації змінюють освітню практику, віртуалізують освітній процес, трансформують традиційні методи взаємодії його суб'єктів. Сьогодні спостерігається стала залежність між ефективною діяльністю викладачів та рівнем їхньої цифрової компетентності, спроможністю використовувати новітні досягнення інформатики, педагогіки, психології, управління та інших галузей наук. Введення воєнний стан передбачає дистанційне навчання у закладах вищої освіти. Отже, за умови воєнного стану відбувається прискорена інтеграція відповідних платформ і технологій дистанційного навчання та застосування їх не лише у навчанні, а й для управління освітнім процесом.

Мета бакалаврської роботи – алгоритмізація та програмування програмного забезпечення тренажеру з теми «Протоколи комп'ютерних мереж» дистанційного курсу «Інформаційні мережі».

Об'єкт розробки – алгоритм та програмна реалізація програмного забезпечення тренажеру з теми «Протоколи комп'ютерних мереж» дистанційного курсу «Інформаційні мережі».

Предмет розробки – програмна реалізація елементів тренажера з теми «Протоколи комп'ютерних мереж» дистанційного курсу «Інформаційні мережі».

Бакалаврська робота складається з чотирьох розділів. У першому розділі описано постановку задачі для реалізації навчального тренажеру. У другому розділі описано про протоколи локальних мереж. У третьому розділі розглянуто теоретичний матеріал з теми тренажеру, детально описаний алгоритм роботи та блок-схема роботи тренажеру. У четвертому розділі описано процес програмної реалізації тренажеру та інструкція користувача. Обсяг пояснювальної записки: 52 стор., в т.ч. основна частина - 47 стор., джерела - 20 назв.

1. ПОСТАНОВКА ЗАДАЧІ РОЗРОБКИ

Важливим завданням бакалаврської роботи – програмна реалізація програмного забезпечення тренажеру з теми «Протоколи комп'ютерних мереж» дистанційного курсу «Інформаційні мережі».

Завдання бакалаврської роботи:

- Переглянути теоретичний матеріал з теми «Протоколи комп'ютерних мереж»;
- Розробити алгоритм роботи тренажеру;
- Скласти блок-схему роботи тренажеру;
- Програмна реалізація програми тренажеру;
- Описати результати програмної реалізації тренажеру.

При програмуванні та алгоритмізації тренажеру, треба врахувати наступні вікна описані нижче:

- Перше стартове вікно програми тренажеру, необхідно показати інформацію про тематику тренажеру, вивести кнопку початку тренінгу, вивести інформацію про студента та наукового керівника кваліфікаційної роботи;
- Наступні форми тренажеру - це питання з теми тренажеру у вигляді тестування, студентові необхідно обрати правильну або неправильну відповідь;
- Останнє вікно тренажеру користувачеві відкривається в кінці тренінгу, на останній формі написано повідомлення про успішне проходження: «Вітаємо! Ви успішно пройшли тренажер з теми «Протоколи комп'ютерних мереж» дистанційного курсу «Інформаційні мережі».

2. ІНФОРМАЦІЙНИЙ ОГЛЯД

2.1. Протоколи локальних мереж

Основними протоколами локальних мереж є:

- протокол TCP / IP;
- протокол NetBEUI;
- протокол IPX / SPX і NWLink.

Transmission Control Protocol / Internet Protocol (TCP/IP) - Промисловий стандартний набір протоколів, які забезпечують зв'язок в гетерогенному середовищі, тобто забезпечують сумісність між комп'ютерами різних типів. Сумісність - одна з основних переваг TCP/IP, тому більшість ЛОМ підтримує його. Крім того, TCP/IP надає доступ до ресурсів Інтернету, а також протокол маршрутизації для мереж масштабу підприємства. Оскільки TCP/IP підтримує маршрутизацію, він використовується в якості міжмережевого протоколу. TCP /IP має два головні недоліки: розмір і недостатня швидкість роботи.

NetBEUI - розширений інтерфейс NetBIOS спочатку NetBIOS і NetBEUI були тісно пов'язані і розглядалися як один протокол. Потім деякі виробники ЛОМ так відособили NetBIOS, протокол сеансового рівня, що він вже не міг використовуватися поруч з іншими транспортними протоколами. NetBIOS - це інтерфейс сеансового рівня ЛОМ, який виступає в якості прикладного інтерфейсу з мережею. Цей протокол надає програмний засіб для здійснення сеансів зв'язку з іншими мережевими програмами. Він дуже популярний, оскільки підтримується багатьма додатками. NetBEUI невеликий швидкий і ефективний протокол транспортного рівня, який поставляється з усіма мережними продуктами фірми Microsoft. До переваг NetBEUI належить невеликий розмір стека, висока швидкість передачі даних по мережі і сумісність з усіма мережами Microsoft. Основний недолік NetBEUI не підтримує маршрутизацію. Це обмеження стосується всіх мереж Microsoft.

IPX/SPX і NWLink - стек протоколів вживаний в мережах NETWARE фірми NOVELL. Як і NetBEUI, відносно невеликий і швидкий протокол, але на відміну від NetBEUI він підтримує маршрутизацію.

NWLink - реалізація IPX/SPX фірми Microsoft. Це транспортний протокол маршрутизації.

Виходячи з вище наведеного та аналізу основних тенденцій розвитку мережових протоколів вважається найбільш перспективним використання протоколу TCP/IP який найбільш повно задовольняє вимогам, що пред'являються.

Архітектуру або технології локальних мереж можна розділити на два покоління. До першого покоління відноситься архітектура, що забезпечує низьку і середню швидкість передачі інформації: Ethernet (10 Мбіт/с), Token Ring (16 Мбіт/с) і ARC net (2,5 Мбіт/с). Для передачі даних ці технології використовують кабелі з мідною жилою. До другого покоління технологій відноситься сучасна високошвидкісна архітектура: FDDI (100 Мбіт/с), ATM (155 Мбіт/с) і модернізовані версії архітектури першого покоління (Ethernet): Fast Ethernet (100 Мбіт/с) і Gigabit Ethernet (1000 Мбіт/с).

Ethernet - це найпоширеніший на сьогоднішній день стандарт локальних мереж. Загальна кількість мереж, що працюють за протоколом Ethernet у даний час, оцінюється в декілька мільйонів. Ethernet - це мережовий стандарт, заснований на експериментальній мережі Ethernet Network, яку фірма Xerox розробила і реалізувала в 1975 році. Метод доступу був випробуваний ще раніше: у другій половині 60-х років у радіомережі Гавайського університету використовувалися різні варіанти випадкового доступу до загального радіосередовища, які отримали загальну назву Aloha. В 1980 році фірми DEC, Intel і Xerox спільно розробили й опублікували стандарт Ethernet версії II для мережі, побудованої на основі коаксіального кабелю. Цю версію фірмового стандарту Ethernet називають стандартом Ethernet DIX. На основі

стандарту Ethernet DIX був розроблений стандарт IEEE 802.3, який багато в чому збігається зі своїм попередником, але деякі відмінності все ж присутні. У той час як у стандарті IEEE 802.3 функції протоколу розділені на рівні MAC і LLC, в оригінальному стандарті Ethernet вони об'єднані в єдиний каналний рівень. У Ethernet DIX визначається протокол тестування конфігурації (Ethernet Configuration Test Protocol), котрий відсутній у IEEE 802.3. Дещо відрізняється і формат кадру, хоча мінімальні і максимальні розміри кадрів у цих стандартах збігаються.

У залежності від типу фізичного середовища стандарт IEEE 802.3 має різні модифікації - 10Base-5, 10Base-2, 10Base-T, 10Base-FL, 10Base-FB.

В 1995 році був прийнятий стандарт Fast Ethernet, який багато в чому не є самостійним стандартом, про, що говорить і той факт, що його опис просто є додатковим розділом до основного стандарту 802.3 - розділом 802.3u. Аналогічно, прийнятий у 1998 році стандарт Gigabit Ethernet описаний у розділі 802.3z основного документу. Для передачі двійкової інформації по кабелю для всіх варіантів фізичного рівня технології Ethernet, що забезпечують пропускну здатність 10 Мбіт/с, використовується манчестерський код. У більш швидкісних версіях Ethernet застосовують більш ефективні щодо смуги пропускання надлишкові логічні коди.

Усі види стандартів Ethernet (у тому числі Fast Ethernet і Gigabit Ethernet) використовують один і той самий метод поділу середовища передачі даних - метод CSMA/CD.

Мережі Token Ring (стандарт 802.5) характеризується розподіленням середовищем передачі даних, яке в даному випадку складається з відрізків кабелю, що з'єднують усі станції мережі в кільце. Кільце розглядається як загальний розділений ресурс, і для доступу до нього потрібно не випадковий алгоритм, як в мережах Ethernet, а детермінований, заснований на передачі станціям права на використання кільця у визначеному порядку. Це право передається за допомогою кадру спеціального формату, званого

маркером, або токенів, (token).

Технологія Token Ring була розроблена компанією IBM в 1984 році, а потім передана в якості проекту стандарту в комітет IEEE 802, який на її основі прийняв в 1985 році стандарт 802.5. Компанія IBM протягом довгого часу використовувала технологію Token Ring в якості своєї основної мережевої технології для побудови локальних мереж на основі комп'ютерів різних класів - мейнфреймів, міні-комп'ютерів і персональних комп'ютерів.

Мережі Token Ring працюють із двома бітовими швидкостями - 4 і 16 Мбіт/с. Об'єднання станцій, що працюють на різних швидкостях, в одному кільці не допускається. Мережі Token Ring, що працюють зі швидкістю 16 Мбіт/с, мають деякі удосконалення в алгоритмі доступу в порівнянні із стандартом 4 Мбіт/с.

Технологія Token Ring є більш складною технологією, ніж Ethernet. Вона має властивості відмовостійкості. У мережі Token Ring визначені процедури контролю роботи мережі, які використовують зворотний зв'язок кільцеподібної структури - відправлений кадр завжди повертається в станцію-відправника. У деяких випадках виявлені помилки в роботі мережі усуваються автоматично, наприклад може бути відновлений загублений маркер. В інших випадках помилки тільки фіксуються, а їх усунення виконується вручну обслуговуючим персоналом.

Для контролю мережі одна зі станцій виконує роль так званого активного монітора. Активний монітор вибирається під час ініціалізації кільця як станція з максимальним значенням MAC-адреси. Якщо активний монітор виходить з ладу, процедура ініціалізації кільця повторюється і вибирається новий активний монітор. Щоб мережа могла виявити відмову активного монітора, останній в працездатному стані кожні три секунди генерує спеціальний кадр своєї присутності. Якщо цей кадр не з'являється в мережі більш 7 с, то інші станції мережі починають процедуру виборів нового активного монітора.

Технологія FDDI (Fiber Distributed Data Interface - оптоволоконний інтерфейс розподілених даних) - це перша технологія локальних мереж, в якій середовищем передачі даних є волоконно-оптичний кабель. Роботи з створенням технологій і пристроїв для використання волоконно-оптичних каналів в локальних мережах почалися в 80-ті роки, незабаром після початку промислової експлуатації подібних каналів у територіальних мережах. Група інституту ANSI розробила в період з 1986 по 1988 рік початкові версії стандарту FDDI, який забезпечує передачу кадрів зі швидкістю 100 Мбіт/с по-подвійному волоконно-оптичному кільцю довжиною до 100 км.

Мережа FDDI будується на основі двох волоконнооптичних кілець, які утворюють основний і резервний шляхи передачі даних між вузлами мережі. Наявність двох кілець - це основний спосіб підвищення відмовостійкості в мережі. Як і в Token Ring, в FDDI використовується маркерний метод доступу. відмінність лише в тому, що тут є режим раннього звільнення маркера, який передається після передачі пакета. У цій мережі не використовуються пріоритети, але визначені два види станцій для підключення:

- станції подвійного підключення (DAS) мають швидкість передачі даних 200 Мбіт/с;
- станції одиночного підключення (SAS) - швидкість передачі 100 Мбіт/с.

Максимальна кількість станцій подвійного підключення в кільці 500, максимальний діаметр подвійного кільця 100 км, а між сусідніми вузлами для оптоволокна дорівнює 2 км, для UTP категорії 5 - 100 м.

Переваги:

- хороша продуктивність;
- велика відстань;
- висока відмово стійкість;
- забезпечує відновлення логічної структури.

У 1995 році прийнятий стандарт IEEE 802.3u, що описує 100-мегабітну технологію Fast Ethernet. Новому стандарту не торкнувся каналний рівень (підрівні MAC і LLC), відповідно повністю зберігся метод доступу CSMA/CD, формати кадрів і часові співвідношення. Всі відмінності між Ethernet і Fast Ethernet зосереджені на фізичному рівні. Стек протоколів Fast Ethernet показаний на рисунку 3.1.



Рисунок 3.1 Стек протоколів Fast Ethernet (PMA – Physical Medium Attachment, підключення до фізичного середовища, PMD - Physical Medium Dependet, залежний від фізичного носія)

Оскільки однією з цілей розробки було забезпечення максимальної спадкоємності і узгодженості 10-мегабітних і 100-мегабітних мереж, було ухвалено рішення збільшити швидкість за рахунок скорочення до 10 нс бітового інтервалу (проти 100 нс в Ethernet). При цьому максимально допустимий час обертання (RTT) сигналу склав 2,6 мкс, тому максимальний діаметр сегменту Fast Ethernet складає 205 м.

Мережі Fast Ethernet можна будувати на волоконно-оптичному (багатомодовому) кабелі (два волокна), скрученій парі 5 категорії (дві пари) або скрученій парі 3 категорії (чотири пари). Коаксіальний кабель не використовується. Мережа завжди будується на концентраторах (або комутаторах) і має ієрархічну топологію (Рисунок 3.2). Без концентратора можна з'єднати (за допомогою спеціального крос-кабелю) тільки два

вузли.

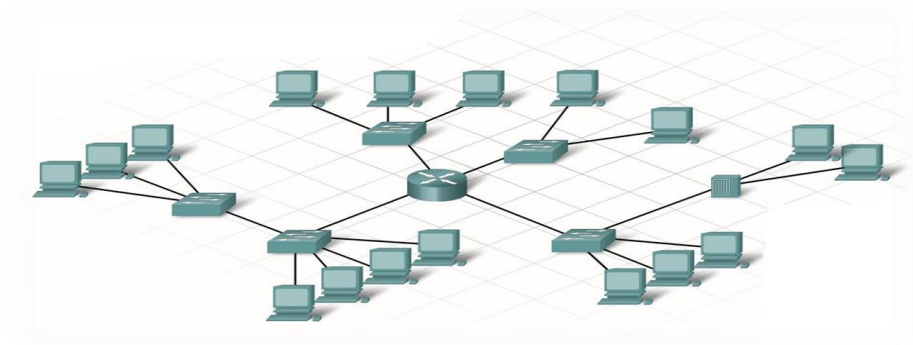


Рисунок 3.2 – Архітектура Ethernet

Фізичний рівень ділиться на підрівень узгодження (Reconciliation Sublayer, RS) і пристрій фізичного рівня (Physical Layer Device), зв'язані через інтерфейс, незалежний від середовища передачі (Media Independent Interface, МІІ). Пристрій фізичного рівня виконує логічне кодування (4В/5В або 8В/6Т), фізичне кодування (NRZI або MLT-3) і приєднання до середовища передавання, а також автоматичне узгодження режимів передачі (наприклад, дуплексний або напівдуплексний режим).

Як вже згадувалося раніше, Fast Ethernet – це назва серії, що складається з трьох специфікацій для локальних мереж із швидкістю передачі 100 Мбіт/с, описаних в додатку IEEE 802.3u до стандарту 802.3. До списку входять специфікації 100BASE-T4, 100BASE-TX і 100BASE-FX (Рисунок 3.3). Кожною з них визначається принцип використання протоколу доступу CSMA/CD, а також мережевих адаптерів, що підтримують швидкості передачі даних від 10 до 100 Мбіт/с, що дозволяє забезпечити сумісність при переході від мереж із швидкістю 10 Мбіт/с до мереж із швидкістю 100 Мбіт/с.

Версія фізичного рівня 100BaseFX визначає роботу по багатомодовому оптоволокну на довжині хвилі 1300 нм. По одному волокну передаються дані від вузла, по другому – до вузла. З технології FDDI запозичений метод логічного кодування 4В/5В і метод фізичного кодування NRZI. У напівдуплексному режимі максимальна відстань між вузлами – 412 м, при повному дуплексі – 2 км. (по одномодовому

оптоволокну – до 32 км.). 100BaseFX несумісний з 10BaseFL, оскільки використовує іншу довжину хвилі.

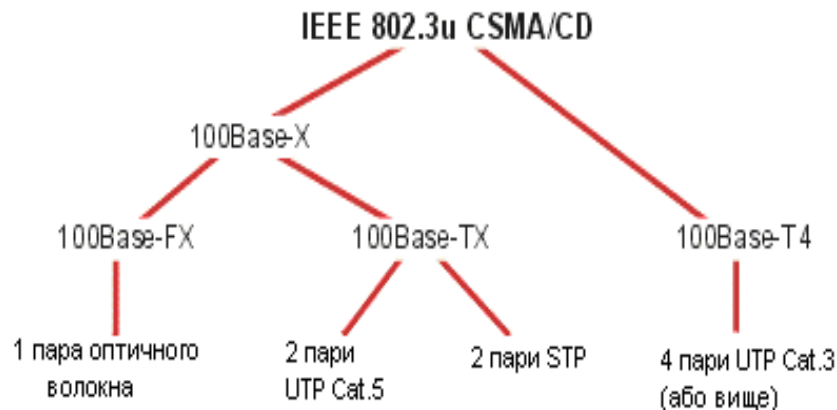


Рисунок 3.3 - Фізичні інтерфейси стандарту Fast Ethernet

Версія фізичного рівня 100BaseTX визначає роботу по скрученій парі (UTP Cat 5 або STP Type 1). Використовується дві пари, призначення провідників (розкладка по контактах роз'ємів) повністю співпадає з 10-мегабітним Ethernet. Максимальна довжина сегменту – 100 м. Логічне кодування – 4В/5В, фізичне кодування – MLT-3.

Робота мережі 100Base-T4. Це локальна мережа зіркової топології, яка використовує для передавання даних 4 скручені пари дротів третьої (UTP Cat 3), четвертої або п'ятої категорії (UTP Cat 5). PCS кодує сигнали в трійковому коді, тобто кожен байт перетворюється на шість сигналів, кожний з яких має одне з трьох значень. Тому код позначають 8В6Т, він має більш вузький спектр сигналу та при швидкості 33,3 Мбіт/с вкладається у смугу 16 МГц (скручена пара, 3 категорія). Максимальна довжина сегменту – 100 м. У кожен момент часу передавання або приймання забезпечують три пари дротів.

Четверта пара призначена для прослуховування частоти-носія та виявлення конфліктів.

Швидкість передавання даних у мережі 100Base-T4 становить 100 Мбіт/с. Збільшення швидкості за рахунок:

- використані три пари дротів, що дало змогу збільшити швидкість утричі;

- кодування 8В6Т ще збільшило її у 2.67 рази;
- збільшено частоту з 20 до 25 МГц.

Усе це разом ($3 \cdot 2.67 \cdot 1.25 = 10$) і дало змогу досягти такої швидкості передавання. Порівняльні характеристики фізичних інтерфейсів Fast Ethernet приведені в таблиці 2.1.

Таблиця 3.1. Фізичні інтерфейси стандарту Fast Ethernet (IEEE 802.3u) і їх основні характеристики

Фізичний інтерфейс	100Base-FX	100Base-TX	100Base-T4
Порт пристрою	Duplex SC	RJ-45	RJ-45
Середовище передачі	Оптичне волокно	Скручена пара UTP Cat. 5	Скручена пара UTP Cat.3,4,5
Сигнальна схема	4В/5В	4В/5В	>8В/6Т
Бітове кодування	NRZI	MLT-3	NRZI
Число кручених пар/ волокон	2 волокна	2 скручені пари	4 скручені пари
Протяжність сегмента	до 412 м(mm) до 2 км (mm)* до 100 км (sm)*	до 100 м	до 100 м
Позначення: mm - багатомодове волокно, sm – одномодове волокно, * - вказані відстані можуть бути досягнуті лише при дуплексному зв'язку.			

Пристрої, підтримуючі 100BaseTX або 100BaseT4, повинні мати функцію узгодження режимів (auto-negotiation), що дозволяє вибрати найефективніший режим, доступний обом учасникам обміну. Всього визначено 5 режимів (перераховані в порядку зростання пріоритету): 10Base, дуплексний 10BaseT, 100BaseTX, 100BaseT4, дуплексний

100BaseTX. Для узгодження вузли передають пакети з 17-ти імпульсів FLP (Fast Link Pulse), в яких міститься слово, що кодує найбільш пріоритетний з доступних режимів роботи. Якщо один з вузлів посилає імпульси NLP (використовувані в 10BaseT для контролю цілісності лінії), другий вузол розуміє, що єдиний можливий режим – це 10BaseT.

Залежно від того, який набір методів логічного кодування підтримує концентратор, він відноситься до класу I (підтримує і 4B/5B, і 8B/6T) або до класу II (підтримує тільки один з методів). Концентратори класу I мають порти всіх трьох типів – 100BaseFX, 100BaseTX, 100BaseT4, але за рахунок необхідності перетворення схем кодування, вносять велику затримку – до 140 bt. Тому в одному домені колізій може бути не більше одного повторювача класу I. Концентратори класу II мають або порти 100BaseFX і 100BaseTX, або тільки порти 100BaseT4. Затримка, що вноситься, може досягати 92 bt (TX/FX) або 67 bt (T4). [1].

Максимальна кількість концентраторів класу II в одному домені колізій – 2. Оскільки максимальний діаметр домену колізій Fast Ethernet для скрученої пари – 205 м, а кожна станція може бути віддалена від повторювача на відстань до 100 м, у гіршому разі концентратори (тільки класу II) повинні з'єднуватися кабелем завдовжки не більше 5]

В таблиці 3.2 наведена порівняльна характеристика технологій побудови локальних мереж.

Таблиця 3.2 Порівняльні характеристики найбільш поширених технологій ЛОМ

Характеристики	FDDI	Ethernet	Token Ring
Швидкість передачі	100 Мбіт / с	10 (100) Мбіт / с	16 Мбіт / с
Топологія	кільце	шина/зірка	кільце / зірка
Середовище передачі	оптоволокну, вита пара	коаксіальний кабель, вита пара, оптоволокну	вита пара, оптоволокну
Метод доступу	маркер	CSMA/CD	маркер

Максимальна довжина мережі	100 км	2500 м	4000 м
Максимальна кількість вузлів	500	1024	260
Максимальна відстань між вузлами	2 км	2500 м	100 м

3. ТЕОРЕТИЧНА ЧАСТИНА

3.1. Протоколи, інтерфейси мереж

До адреси вузла мережі та схеми його призначення можна висунути кілька вимог:

1. Адреса повинна унікально ідентифікувати комп'ютер у мережі будь-якого масштабу;
1. Схема призначення адрес має зводити до мінімуму ручну працю адміністратора й імовірність дублювання адрес;
1. Адреса повинна мати ієрархічну структуру, зручну для побудови великих мереж. Цю проблему добре ілюструють міжнародні поштові адреси, які дають змогу поштовій службі, що організує доставку листів між країнами, скористатися лише назвою країни адресата і не враховувати назву його міста, а тим паче вулиці;
1. Адреса має бути зручна для користувачів мережі, а це означає, що вона повинна мати символічне подання наприклад, `www.gu.net`;
1. Адреса повинна мати якомога компактніше подання, щоб не перевантажувати пам'ять комунікаційної апаратури — мережних адаптерів, маршрутизаторів і т. ін.

На практиці звичайно використовується відразу кілька схем, тож комп'ютер одночасно має кілька адрес-імен. Кожна адреса використовується в тій ситуації, коли відповідний вид адресації є найзручнішим. А щоб не виникало плутанини і комп'ютер завжди однозначно вирізнявся своєю адресою, використовуються спеціальні допоміжні протоколи, що за адресою одного типу можуть визначити адреси інших типів.

Найбільше поширення одержали три схеми адресації вузлів:

1. *Апаратні адреси.* Ці адреси призначені для мережі невеликого чи середнього розміру, тому вони не мають ієрархічної структури. Типовим представником адреси такого типу є адреса мережного адаптера локальної мережі. Така адреса звичайно використовується тільки апаратурою, тому її намагаються зробити якомога компактнішою і записують у вигляді двійкового чи шістнадцяткового значення, наприклад 0081005e24a8. У процесі завдання апаратних адрес звичайно не потрібне виконання ручної роботи, адже вони або вбудовуються в апаратуру, або генеруються автоматично під час кожного нового запуску устаткування, причому унікальність адреси в межах мережі забезпечує устаткування. Крім відсутності ієрархії, використання апаратних адрес пов'язане ще з одним недоліком — у випадку заміни апаратури, наприклад мережного адаптера, змінюється й адреса комп'ютера. Більше того, якщо встановити кілька мережних адаптерів, у комп'ютера з'являється кілька адрес, що не дуже зручно для користувачів.

1. *Символьні адреси чи імена.* Ці адреси призначені для запам'ятовування людьми і тому зазвичай несуть значення навантаження. Символьні адреси легко використовувати як у невеликих, так і у великих мережах. Для роботи у великих мережах символічне ім'я може мати складну ієрархічну структуру, наприклад ftp-archl.ucl.ac.uk. Ця адреса говорить про те, що даний комп'ютер підтримує ftp-архів у мережі одного з коледжів Лондонського університету (University College London — ucl) і ця мережа належить до академічної галузі (ac) Internet Великобританії (United Kingdom — uk). Під час роботи в межах мережі Лондонського університету таке довге символічне ім'я явно надмірне, і замість нього зручно користуватися коротким символічним ім'ям, на роль якого добре підходить наймолодша частина складового повного імені, тобто ім'я ftp-archl.

1. *Числові складені адреси.* Символьні імена зручні для людей, але через перемінний формат і потенційно велику довжину їх передавання мережею не є надто економним. Тож у багатьох випадках для роботи у великих мережах як адреси вузлів використовують числові складені адреси фіксованого та компактного форматів. Типовими представниками адрес цього типу є IP- та IPX-адреси. У них підтримується дворівнева ієрархія, адреса поділяється на старшу частину — номер мережі — і молодшу — номер вузла. Такий розподіл дає можливість передавати повідомлення між мережами лише на підставі номера мережі, а номер вузла використовується тільки після доставки повідомлення в потрібну мережу; так само, як назва вулиці використовується листоношею лише після того, як лист доставлений у потрібне місто. Віднедавна, щоб зробити маршрутизацію у великих мережах ефективнішою, пропонуються складніші варіанти числової адресації, відповідно до яких адреса має три і більше складових. Такий підхід, зокрема, реалізований у новій версії протоколу IPv6, призначеного для роботи в мережі Internet.

У сучасних мережах для адресації вузлів застосовуються, як правило, одночасно всі три наведені вище схеми. Користувачі адресують комп'ютери символьними іменами, що автоматично замінюються в повідомленнях, переданих по мережі, на числові номери. За допомогою цих числових номерів повідомлення передаються з однієї мережі в іншу, а після доставки повідомлення в мережу призначення замість числового номера використовується апаратна адреса комп'ютера. Нині така схема характерна навіть для невеликих автономних мереж, де, здавалося б, вона явно надлишкова, — це робиться для того, щоб під час включення цієї мережі у велику мережу не потрібно було змінювати склад операційної системи.

Проблема встановлення відповідності між адресами різних типів, якою опікується служба дозволу імен, може розв'язувати-

ся як цілком централізованими, так і розподіленими засобами. У випадку централізованого підходу в мережі виділяється один комп'ютер (сервер імен), у якому зберігається таблиця відповідності одне одному імен різних типів, наприклад, символьних імен і числових номерів. Усі інші комп'ютери звертаються до сервера імен, щоб за символьним ім'ям знайти числовий номер комп'ютера, з яким необхідно обмінятися даними.

За іншого, розподіленого підходу, кожен комп'ютер сам встановлює відповідність між іменами. Наприклад, якщо користувач указав для вузла призначення числовий номер, то перед початком передавання даних комп'ютер-відправник посилає всім комп'ютерам мережі повідомлення (таке повідомлення називається ширококомовним) із проханням упізнати це числове ім'я. Усі комп'ютери після одержання цього повідомлення порівнюють заданий номер зі своїм власним. Той комп'ютер, у якого вони збіглися, посилає відповідь, що містить його апаратну адресу, після чого стає можливим відправлення повідомлень локальною мережею.

Для взаємодії між об'єктами, що працюють у мережі, застосовуються протоколи. На практиці в процесі реалізації мереж прагнуть використовувати стандартні протоколи. Це можуть бути фірмові, національні чи міжнародні стандарти.

На початку 80-х років низка міжнародних організацій зі стандартизації — ISO, ITU-T та деякі інші — розробили модель, що відіграла значну роль у розвитку мереж. Ця модель називається *моделлю взаємодії відкритих систем (Open System Interconnection, OSI)* чи моделлю OSI. Модель OSI визначає різні рівні взаємодії систем, дає їм стандартні імена та вказує, які функції має виконувати кожен рівень. Модель OSI була розроблена на підставі великого досвіду, отриманого в процесі створення комп'ютерних мереж, переважно глобальних, у 70-ті роки. У моделі OSI засоби взаємодії поділяються на сім рівнів: прикладний, представницький, сеансовий, транспортний, мережний, каналний і фізичний. Кожен рівень має справу з одним певним аспектом взаємодії мережних пристроїв.

Дана модель описує тільки системні засоби взаємодії, реалізовані операційною системою, системними утилітами, системними апаратними засобами. Вона не включає засоби взаємодії додатків кінцевих користувачів.

Фізичний рівень призначений для передавання бітів фізичними каналами зв'язку, такими, наприклад, як коаксіальний кабель, кручена пара, оптичноволоконний кабель або цифровий територіальний канал. До цього рівня мають відношення характеристики фізичних середовищ передавання даних такі, як смуга пропускання, перешкодозахищеність, хвильовий опір тощо. На цьому ж рівні визначаються характеристики електричних сигналів, що передають дискретну інформацію, наприклад, крутість фронтів імпульсів, рівні чи напруги струму переданого сигналу, тип кодування, швидкість передавання сигналів. Крім цього, тут стандартизуються типи рознімів і призначення кожного контакту.

Функції фізичного рівня реалізуються в усіх пристроях, підімкнених до мережі. З боку комп'ютера функції фізичного рівня виконуються мережним адаптером чи послідовним портом.

Прикладом протоколу фізичного рівня може служити специфікація 10Base-T технології Ethernet, що визначає як використовуваний кабель неекрановану кручену пару категорії 3 із хвильовим опором 100 Ом, роз'ємом RJ-45, максимальну довжину фізичного сегмента 100 метрів, манчестерський код для подання даних у кабелі, а також деякі інші характеристики середовища й електричних сигналів.

На фізичному рівні просто пересилаються біти. При цьому не враховується, що в деяких мережах, у яких лінії зв'язку використовуються (розділяються) поперемінно кількома парами взаємодіючих комп'ютерів, фізичне середовище передання може бути зайнято. Тому одним із завдань *канального рівня* є перевірка доступності середовища передавання. Іншим завданням канального рівня є реалізація механізмів виявлення та корекції помилок. Для цього на канальному рівні біти групуються в набори, які

називаються *кадрами*. Канальний рівень забезпечує коректність передавання кожного кадру, поміщаючи спеціальну послідовність бітів у початок і кінець кожного кадру, для його виділення, а також обчислює контрольну суму, обробляючи всі байти певним способом і додаючи контрольну суму до кадру. Коли кадр надходить по мережі, одержувач знову обчислює контрольну суму отриманих даних і порівнює результат з контрольною сумою з кадру. Якщо вони збігаються, кадр вважається правильним і приймається. Якщо ж контрольні суми не збігаються, то фіксується помилка. Канальний рівень може не лише виявляти помилки, а й виправляти їх за рахунок повторного передавання ушкоджених кадрів. Необхідно зазначити, що функція виправлення помилок не є обов'язковою для канального рівня, тому в деяких протоколах цього рівня вона відсутня, наприклад, у Ethernet.

У протоколах канального рівня, використовуваних у локальних мережах, закладена певна структура зв'язків між комп'ютерами і способи їх адресації. Хоча канальний рівень і забезпечує доставку кадру між будь-якими двома вузлами локальної мережі, він це робить тільки в мережі з цілком певною топологією зв'язків — саме тією топологією, для якої він був розроблений. До таких типових топологій, підтримуваних протоколами канального рівня локальних мереж, належать спільна шина, кільце і зірка, а також структури, отримані з них за допомогою мостів і комутаторів. Прикладами протоколів канального рівня є протоколи Ethernet, Token Ring, FDDI.

У локальних мережах протоколи канального рівня використовуються комп'ютерами, мостами, комутаторами та маршрутизаторами. У комп'ютерах функції канального рівня реалізуються спільними зусиллями мережних адаптерів і їх драйверів.

У глобальних мережах, що рідко коли мають регулярну топологію, канальний рівень часто забезпечує обмін повідомленнями лише між двома сусідніми комп'ютерами, з'єднаними індиві-

дуальною лінією зв'язку. Прикладами таких протоколів можуть служити широко розповсюджені протоколи PPP і LAR-B. У таких випадках для доставки повідомлень між кінцевими вузлами через усю мережу використовуються засоби мережного рівня. Іноді в глобальних мережах функції канального рівня в чистому вигляді виділити важко, тому що в тому самому протоколі вони поєднуються з функціями мережного рівня. Прикладами такого підходу можуть бути протоколи технологій АТМ.

У цілому канальний рівень являє собою вельми могутній і викінчений набір функцій з пересилання повідомлень між вузлами мережі. У деяких випадках протоколи канального рівня виявляються самодостатніми транспортними засобами і можуть допускати роботу поверх них безпосередньо протоколів прикладного рівня чи додатків, без залучення засобів мережного й транспортного рівнів. Наприклад, існує реалізація протоколу керування мережею SNMP безпосередньо поверх Ethernet, хоча стандартно цей протокол працює поверх мережного протоколу IP і транспортного протоколу UDP. Природно, що застосування такої реалізації буде обмеженим — вона не підходить для складених мереж різних технологій, наприклад, Ethernet і X.25, і навіть для такої мережі, в якій у всіх сегментах застосовується Ethernet. А для двосегментної мережі Ethernet, об'єднаної мостом, реалізація SNMP над канальним рівнем буде цілком можлива.

Проте для забезпечення якісного транспортування повідомлень у мережах будь-яких топологій і технологій функцій канального рівня виявляється недостатньо, тому в моделі OSI виконання цього завдання покладається на два наступні рівні — мережний і транспортний.

Мережний рівень служить для утворення єдиної транспортної системи, що поєднує кілька мереж, причому ці мережі можуть використовувати різні принципи передавання повідомлень між кінцевими вузлами та мати довільну структуру зв'язків. Функції мережного рівня

досить різноманітні. Почнемо їх розгляд на прикладі об'єднання локальних мереж.

Протоколи канального рівня локальних мереж забезпечують доставку даних між будь-якими вузлами лише в мережі з відповідною типовою топологією, наприклад топологією ієрархічної зірки. Це суворе обмеження, що не дає змоги будувати мережі з розвинутою структурою, наприклад, мережі, що поєднують кілька мереж підприємства в єдину мережу, чи високонадійні мережі, в яких існують надлишкові зв'язки між вузлами. Можна було б ускладнювати протоколи канального рівня для підтримки петлеподібних надлишкових зв'язків, але принцип поділу обов'язків між рівнями приводить до іншого рішення. Щоб, з одного боку, зберегти простоту процедури передавання даних для типових топологій, а з іншого допустити використання довільних топологій, вводиться додатковий мережний рівень.

На мережному рівні сам термін *мережа* наділяється специфічним значенням. У даному разі під мережею розуміють сукупність комп'ютерів, з'єднаних між собою відповідно до однієї зі стандартних типових топологій і таких, що використовують для передавання даних один із протоколів канального рівня, визначений для цієї топології.

Усередині мережі доставка даних забезпечується відповідним канальним рівнем, а ось доставкою даних між мережами займається мережний рівень, що й підтримує можливість правильного вибору маршруту передавання повідомлення навіть у тому разі, коли структура зв'язків між складовими мережами має характер, відмінний від прийнятого в протоколах канального рівня.

Мережі з'єднуються між собою спеціальними пристроями, що їх називають маршрутизаторами. *Маршрутизатор* — це пристрій, що збирає інформацію про топологію міжмережєвих з'єднань і на її підставі пересилає пакети мережного рівня в мережу призначення. Щоб передати повідомлення від відправника, який перебуває в одній мережі, одержувачу

в іншій мережі, потрібно зробити кілька транзитних передач між мережами. Таким чином, маршрут являє собою послідовність маршрутизаторів, через які проходить пакет.

Проблема вибору найкращого шляху називається *маршрутизацією*, і її розв'язання є одним з головних завдань мережного рівня. Ця проблема ускладнюється тим, що найкоротший шлях не завжди найкращий. Часто критерієм під час вибору маршруту є час передавання даних цим маршрутом; він залежить від пропускної здатності каналів зв'язку й інтенсивності графіка, що може змінюватися з часом. Деякі алгоритми маршрутизації намагаються пристосуватися до зміни навантаження, тоді як інші приймають рішення на основі середніх показників за тривалий час. Вибір маршруту може здійснюватися і за іншими критеріями, наприклад, надійністю передавання.

В усякому разі функції мережного рівня ширші, ніж функції передавання повідомлень по зв'язках з нестандартною структурою. Мережний рівень вирішує також питання узгодження різних технологій, спрощення адресації у великих мережах і створення надійних та гнучких бар'єрів на шляху небажаного графіка між мережами.

Повідомлення мережного рівня називають *пакетами*. В процесі організації доставки пакетів на мережному рівні використовується поняття «номер мережі». У цьому разі адреса одержувача складається зі старшої частини — номера мережі та молодшої — номера вузла в цій мережі. Усі вузли однієї мережі повинні мати ту саму старшу частину адреси, тому терміну «мережа» на мережному рівні можна дати й інше, формальніше визначення: мережа — це сукупність вузлів, мережна адреса яких містить один і той самий номер мережі.

На мережному рівні визначаються два види протоколів. Перший вид — *мережні протоколи* — реалізують проходження пакетів через мережу. Саме ці протоколи звичайно мають на увазі, коли говорять про протоколи

мережного рівня. Проте часто до мережного рівня відносять і інший вид протоколів, названих протоколами обміну маршрутною інформацією чи просто *протоколами маршрутизації*. За допомогою цих протоколів маршрутизатори збирають інформацію про топологію міжмережевих з'єднань. Протоколи мережного рівня реалізуються програмними модулями операційної системи, а також програмними й апаратними засобами маршрутизаторів.

На мережному рівні працюють протоколи ще одного типу, що відповідають за відображення адреси вузла, використовуваного на мережному рівні, у локальній адресі мережі. Такі протоколи часто називають *протоколами дозволу адрес* — Address Resolution Protocol, ARP. Іноді їх відносять не до мережного рівня, а до каналного, хоча тонкощі класифікації не змінюють їх суті.

Прикладами протоколів мережного рівня є протокол міжмережевої взаємодії IP стека TCP/IP і протокол міжмережевого обміну пакетами IPX стека Novell.

На шляху від відправника до одержувача пакети можуть бути перекручені чи загублені. Хоча деякі додатки мають власні засоби обробки помилок, існують і такі, котрі відразу можуть мати справу з надійним з'єднанням.

Транспортний рівень забезпечує додаткам — прикладному та сеансовому — передання даних з тим ступенем надійності, що їм потрібно. Модель OSI визначає п'ять класів сервісу, наданих транспортним рівнем. Ці види сервісу вирізняються якістю наданих послуг: терміновістю, можливістю відновлення перерваного зв'язку, наявністю засобів мультиплексування кількох з'єднань між різними прикладними протоколами через загальний транспортний протокол, а головне — здатністю до виявлення та виправлення помилок передавання — таких, як перекручування, втрата і дублювання пакетів.

Вибір класу сервісу транспортного рівня визначається, з одного боку, тим, якою мірою завдання забезпечення надійності виконується самими додатками і протоколами вищих, ніж транспортний, рівнів, а з іншого боку, цей вибір залежить від того, наскільки надійною є система транспортування даних у мережі, забезпечувана рівнями, розташованими нижче транспортного — мережним, каналним і фізичним. Так, наприклад, якщо якість каналів передавання дуже висока й імовірність виникнення помилок, не виявлених протоколами нижчих рівнів, невелика, то розумно скористатися одним з полегшених сервісів транспортного рівня, не обтяжених численними перевірками та іншими прийомами підвищення надійності. Якщо ж транспортні засоби нижніх рівнів справді дуже ненадійні, то доцільно звернутися до найрозвинутішого сервісу транспортного рівня, що працює, використовуючи максимум засобів для виявлення й усунення помилок, — за допомогою попереднього встановлення логічного з'єднання, контролю доставки повідомлень щодо контрольних сум і циклічної нумерації пакетів тощо.

Як правило, усі протоколи, починаючи з транспортного рівня і вище, реалізуються програмними засобами кінцевих вузлів мережі — компонентами їх мережних операційних систем. Як приклад транспортних протоколів можна навести протоколи TCP і UDP стека TCP/IP і протокол SPX стека Novell.

Протоколи нижніх чотирьох рівнів узагальнено називають мережним транспортом або транспортною підсистемою, тому що вони повністю виконують завдання транспортування повідомлень із заданим рівнем якості в складених мережах з довільною топологією і різними технологіями. Інші три верхні рівні виконують завдання надання прикладних сервісів на підставі наявної транспортної підсистеми.

Сеансовий рівень забезпечує керування діалогом: фіксує, яка зі сторін є активною на даний момент, надає засоби синхронізації. Останні дають змогу встановлювати контрольні точки в довгі передачі, щоб у разі

відмови можна було повернутися назад до останньої контрольної точки, а не починати все від початку. На практиці деякі додатки використовують сеансовий рівень, і він рідко коли реалізується у вигляді окремих протоколів, хоча функції цього рівня часто поєднують з функціями прикладного рівня та реалізують в одному протоколі.

Представницький рівень має справу з формою подання інформації, що передається по мережі, не змінюючи при цьому її змісту. За рахунок рівня подання інформація, що передається прикладним рівнем однієї системи, завжди зрозуміла прикладному рівню іншої системи. За допомогою засобів даного рівня протоколи прикладних рівнів можуть подолати синтаксичні відмінності в поданні даних або ж відмінності в кодах символів, наприклад кодів ASCII та EBCDIC. На цьому рівні може виконуватися шифрування та дешифрування даних, завдяки якому секретність обміну даними забезпечується відразу для всіх прикладних служб. Прикладом такого протоколу є протокол Secure Socket Layer (SSL), який забезпечує секретний обмін повідомленнями для протоколів прикладного рівня стека TCP/IP.

Прикладний рівень — це насправді просто набір різноманітних протоколів, за допомогою яких користувачі мережі отримують доступ до ресурсів, що розділяються, — таких, як файли, принтери чи гіпертекстові Web-сторінки, а також організують свою спільну роботу, наприклад, за допомогою протоколу електронної пошти. Одиниця даних, якою оперує прикладний рівень, звичайно називається повідомленням (*message*).

Існує чимало різноманітних служб прикладного рівня: NCP в операційній системі Novell NetWare, SMB в Microsoft Windows NT, NFS, FTP і TFTP, що входять в стек TCP/IP.

3.2. Алгоритм роботи тренажеру з теми «Протоколи комп'ютерних мереж» дистанційного курсу «Інформаційні мережі»

Після запуску створеного програмного забезпечення, на стартовій формі користувачу виведеться інформація:

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

Якщо натиснути на кнопку початку тренінгу, користувачу відкриється перший крок з алгоритмізації тренажеру, в якому висвітиться перше питання з матеріалу лекції.

Перший крок. Користувачу відкриється питання з лекції, можна вибрати відповідь натиснувши лише на один варіант відповіді: «Група комп'ютерів, що обслуговуються спільним сервером, який керує розподілом прав доступу користувачів до ресурсів мережі, називається: »

1. Домен (вірна відповідь)
2. Вузол
3. Група

Якщо користувач вибрав правильний варіант відповіді, він зможе відкрити наступний крок. Якщо користувач вибрав неправильний варіант відповіді, то йому виведеться помилка: «Неправильно! Спробуйте ще раз!»

Другий крок. Користувачу відкриється питання з лекції, можна вибрати відповідь натиснувши лише на один варіант відповіді: «Набір правил, за якими здійснюється обмін даними між пристроями комп'ютерних мереж це?»

1. Список
2. Протокол (вірна відповідь)
3. Домен

Якщо користувач вибрав правильний варіант відповіді, він зможе відкрити наступний крок. Якщо користувач вибрав неправильний варіант відповіді, то йому виведеться помилка: «Неправильно! Спробуйте ще раз!»

Третій крок. Користувачу відкриється питання з лекції, можна вибрати відповідь натиснувши лише на один варіант відповіді: «У переважній більшості комп'ютерних мереж обмін даними здійснюється на основі сімейства (стеку) протоколів?»

1. POP3
2. HTTP
3. TCP/IP (вірна відповідь)

Якщо користувач вибрав правильний варіант відповіді, він зможе відкрити наступний крок. Якщо користувач вибрав неправильний варіант відповіді, то йому виведеться помилка: «Неправильно! Спробуйте ще раз!»

Четвертий крок. Користувачу відкриється питання з лекції, можна вибрати відповідь натиснувши лише на один варіант відповіді: «Стандартний протокол поштового з'єднання?»

1. POP3 (вірна відповідь)
2. UDP
3. ARP

Якщо користувач вибрав правильний варіант відповіді, він зможе відкрити наступний крок. Якщо користувач вибрав неправильний варіант відповіді, то йому виведеться помилка: «Неправильно! Спробуйте ще раз!»

П'ятий крок. Користувачу відкриється питання з лекції, можна вибрати відповідь натиснувши лише на один варіант відповіді: «Протокол для пересилки інформації, що зберігається у файлах?»

1. HTTP
2. FTP (вірна відповідь)
3. TCP

Якщо користувач вибрав правильний варіант відповіді, він зможе відкрити наступний крок. Якщо користувач вибрав неправильний варіант відповіді, то йому виведеться помилка: «Неправильно! Спробуйте ще раз!»

Шостий крок. Користувачу відкриється питання з лекції, можна вписати з клавіатури відповідь на питання та натиснути кнопку «наступне

питання»: «Протокол доставки web -документів, основний протокол служби WWW?» Якщо користувач вписав правильний варіант відповіді, він зможе відкрити наступний крок. Якщо користувач вибрав неправильний варіант відповіді, то йому виведеться помилка: «Неправильно! Підказка: НТТ*»

Сьомий крок. Користувачу відкриється питання з лекції, можна вписати з клавіатури відповідь на питання та натиснути кнопку «наступне питання»: «Протокол емуляції термінала, що дозволяє користувачеві підключатися до інших віддалених станцій і працювати з ними зі своєї машини, начебто вона була їх віддаленим терміналом?» Якщо користувач вписав правильний варіант відповіді, він зможе відкрити наступний крок. Якщо користувач вибрав неправильний варіант відповіді, то йому виведеться помилка: «Неправильно! Підказка: Те*net»

Восьмий крок. Користувачу відкриється питання з лекції, можна вибрати відповідь натиснувши лише на один варіант відповіді: «IEEE 802.3(Ethernet) відноситься до протоколів?» Якщо користувач вписав правильний варіант відповіді, він зможе відкрити наступний крок. Якщо користувач вибрав неправильний варіант відповіді, то йому виведеться помилка: «Неправильно! Підказка: Н*зк*го р*вня»

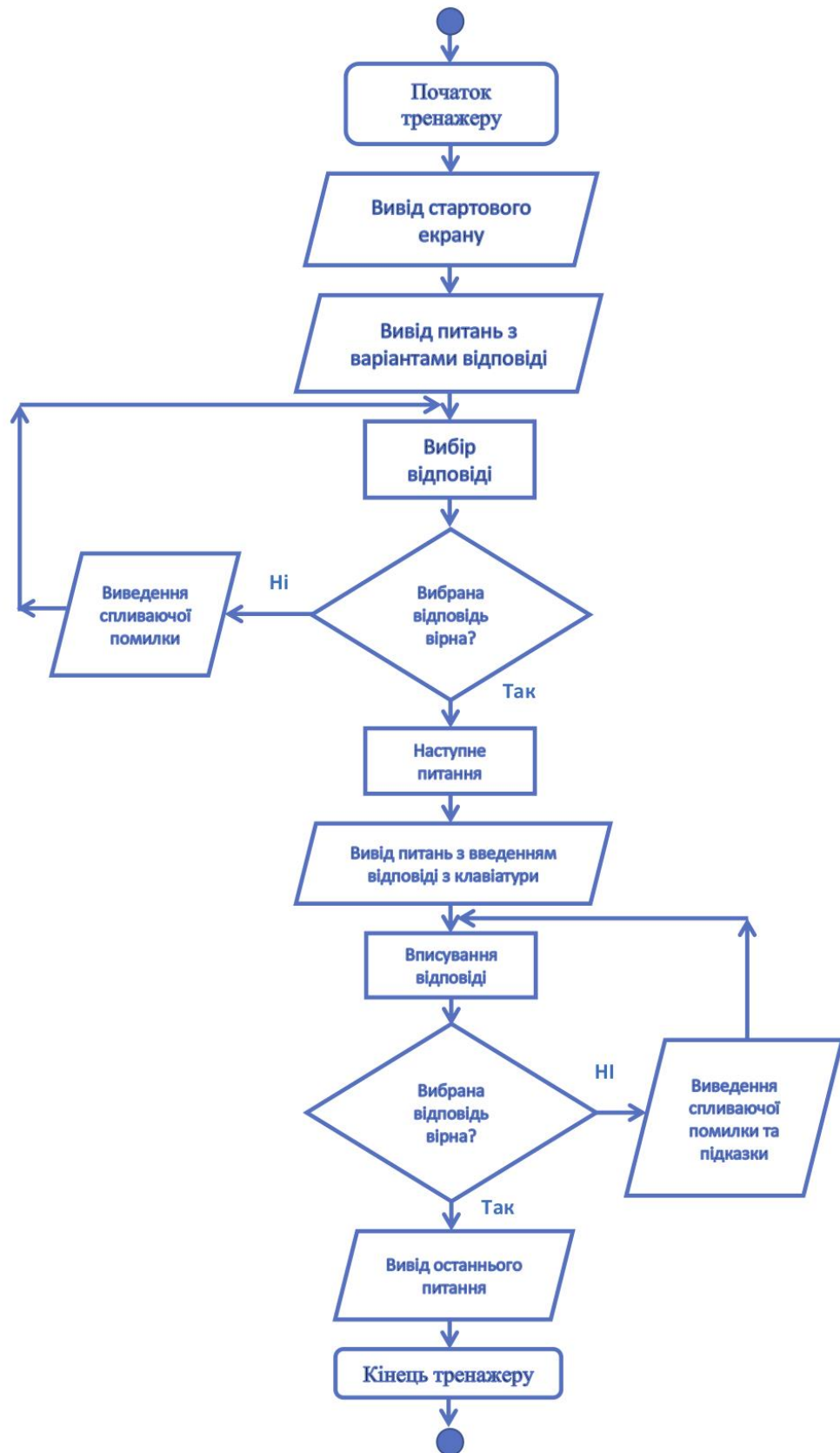
Дев'ятий крок. Користувачу відкриється питання з лекції, можна вибрати відповідь натиснувши лише на один варіант відповіді: «Служба доменних імен, які перетворюють доменне ім'я в IP-адресу?» Якщо користувач вписав правильний варіант відповіді, він зможе відкрити наступний крок. Якщо користувач вибрав неправильний варіант відповіді, то йому виведеться помилка: «Неправильно! Підказка: D*S»

Десятий крок. Користувачу відкриється питання з лекції, можна вибрати відповідь натиснувши лише на один варіант відповіді: «Яка мережа відноситься до мереж з передачею маркера?» Якщо користувач вписав правильний варіант відповіді, він зможе відкрити наступний крок.

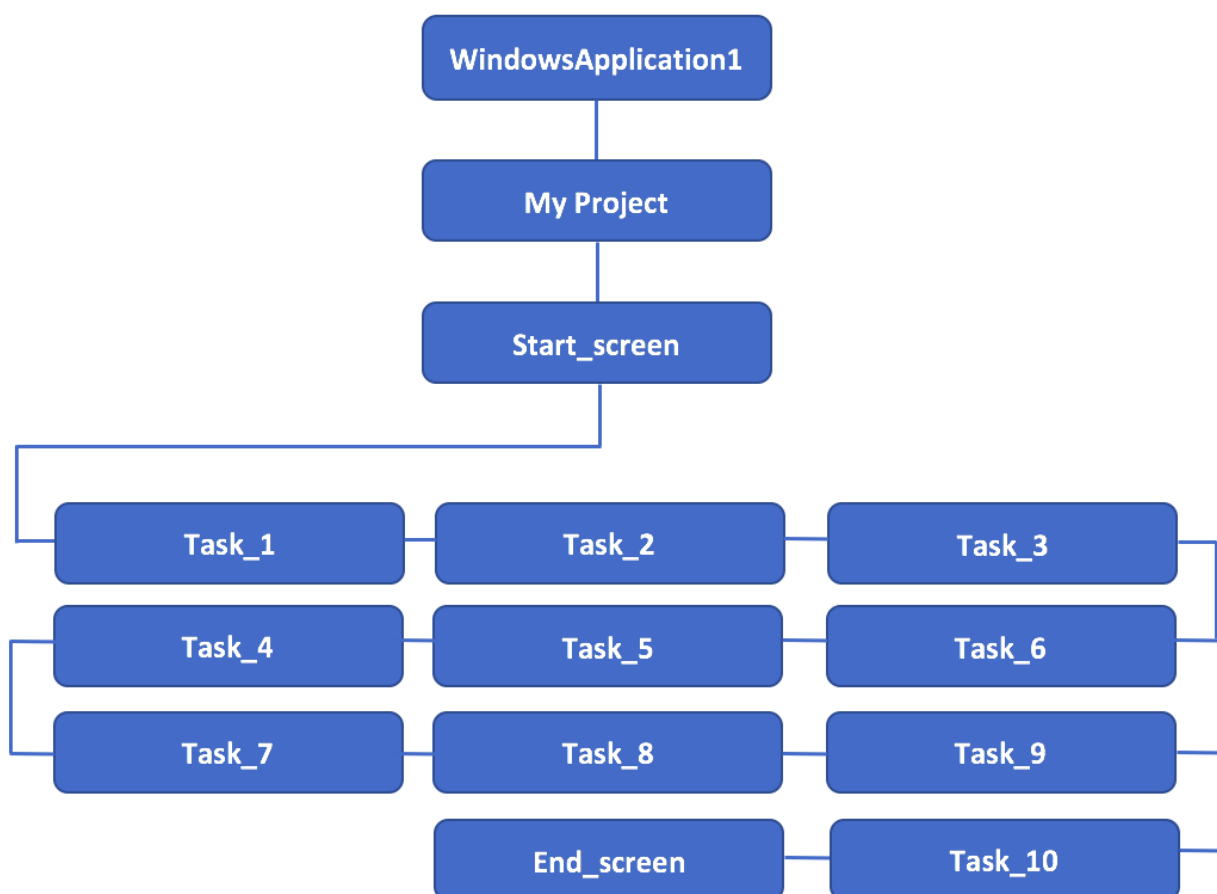
Якщо користувач вибрав неправильний варіант відповіді, то йому виведеться помилка: «Неправильно! Підказка: T*k*n R*ng»

Після десятого кроку відкривається підсумкова сторінка тренажеру, на якій написано: «Вітаємо! Ви успішно пройшли тренажер з теми «Протоколи комп'ютерних мереж» дистанційного курсу «Інформаційні мережі».

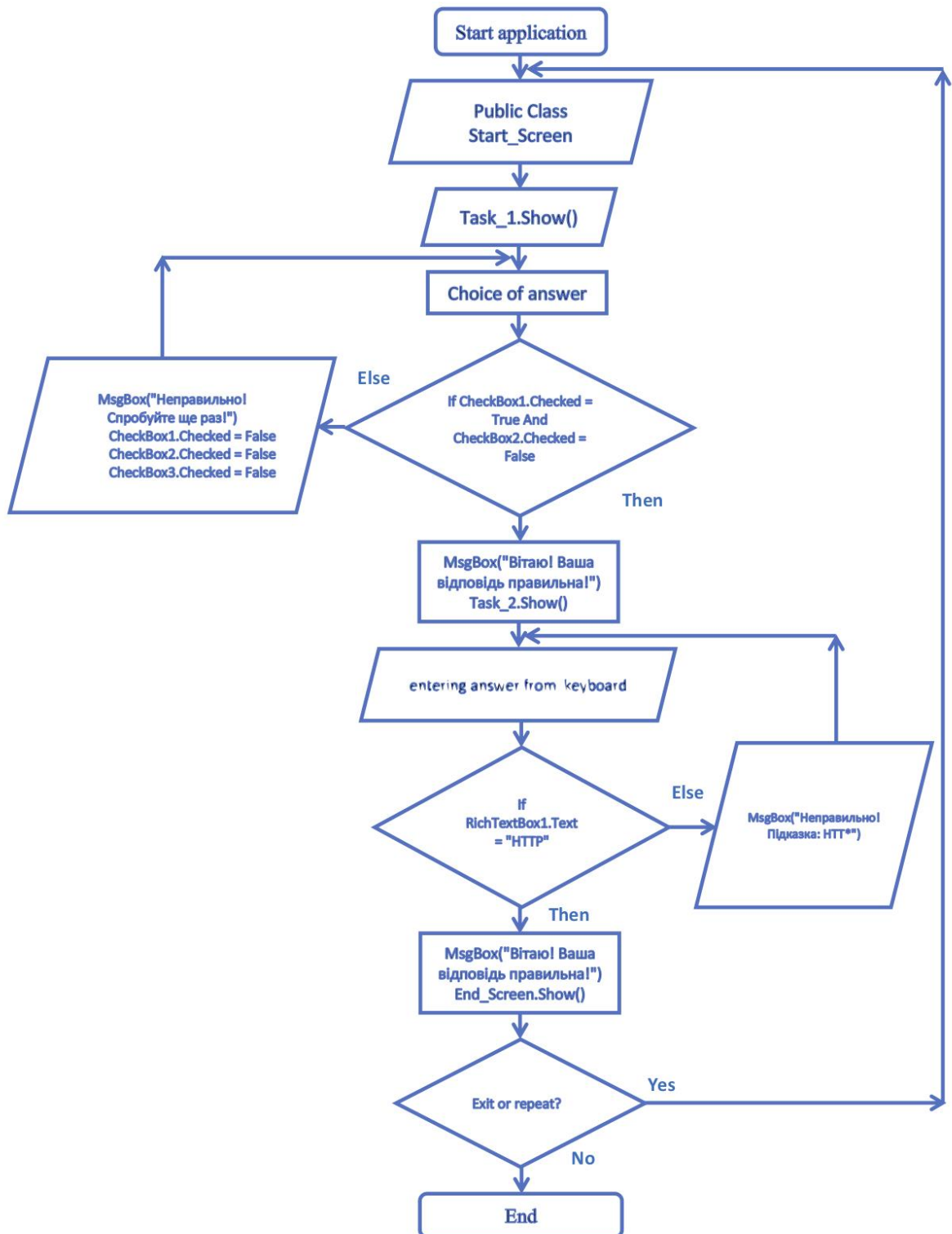
3.3. Блок-схема тренажеру



3.4 Фізична структура тренажеру



3.5 Логічна структура тренажера



4. ПРАКТИЧНА ЧАСТИНА

4.1 Опис програмної реалізації

Програму тренажер було реалізовано на мові програмування Visual Basic, в середовищі розробки Visual Studio. Створено новий проект Windows Forms (рис. 4.1)

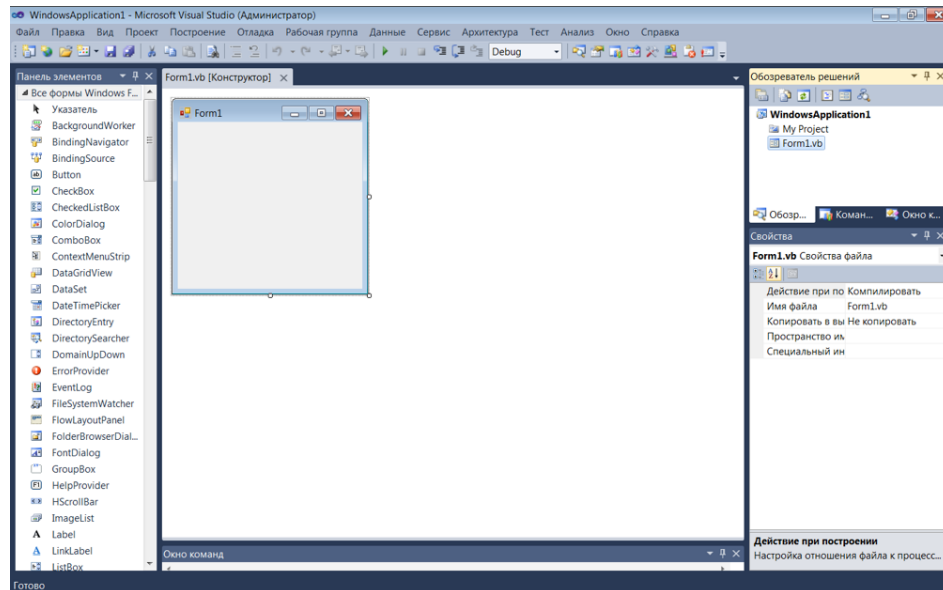


Рисунок 4.1 – процес створення нового проекту.

Після створення нового проекту в середовищі розробки, розроблено титульну стартову форму тренажеру (рис. 4.2)

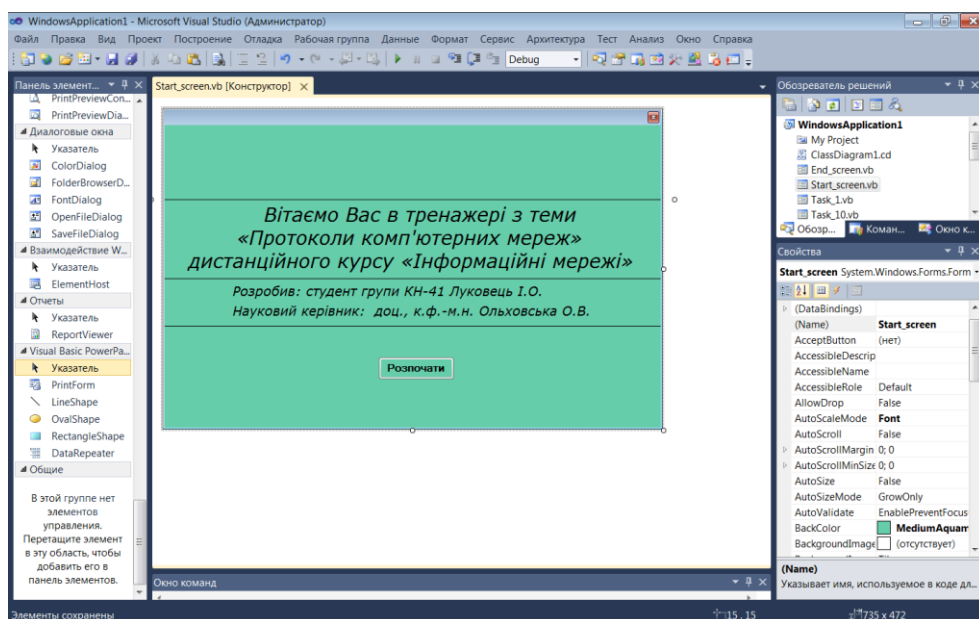


Рисунок 4.2 – Стартова форма тренажеру.

Після титульної форми, реалізовано перше питання з тематики тренажеру. Запрограмовано наступну форму (рис. 4.3), що є перший крок алгоритму

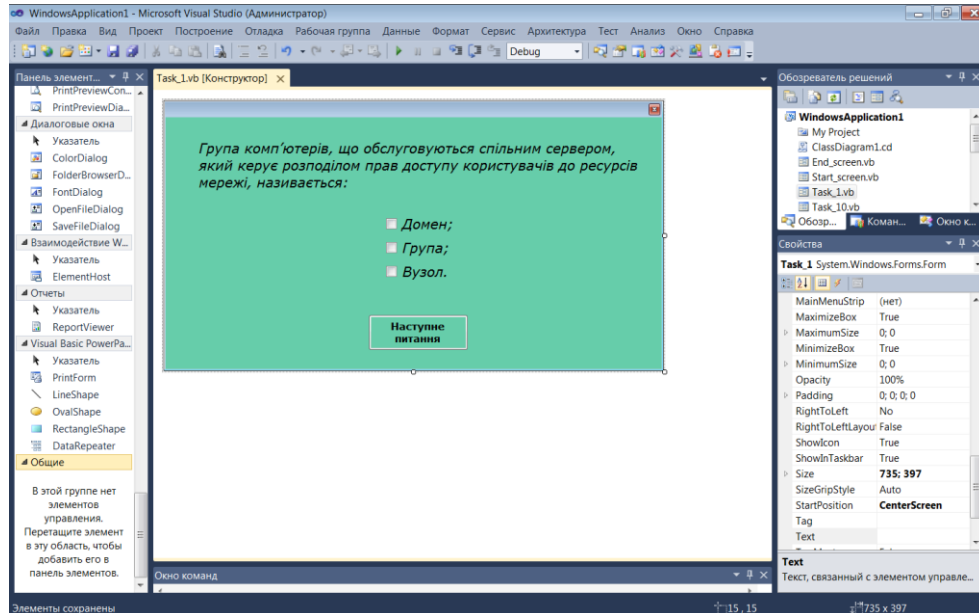


Рисунок 4.3 – перший крок алгоритмізації тренажеру.

Потім, після першого завдання, потрібно реалізувати друге питання з тематики тренажеру (рис. 4.4)

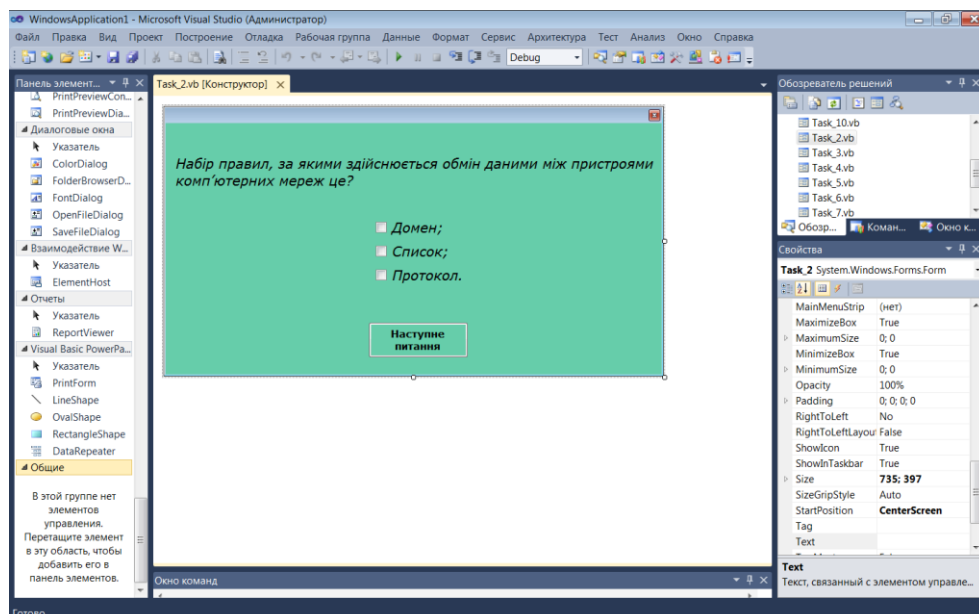


Рисунок 4.4 – другий крок алгоритмізації тренажеру.

За програмну реалізацію тестування в тренажері відповідає наступний програмний код:

```
1 Public Class Task_1
2     Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
3         If CheckBox1.Checked = True And CheckBox2.Checked = False Then
4             MsgBox("Вітаю! Ваша відповідь правильна!")
5             Task_2.Show()
6         Else
7             MsgBox("Неправильно! Спробуйте ще раз!")
8             CheckBox1.Checked = False
9             CheckBox2.Checked = False
10            CheckBox3.Checked = False
11        End If
12    End Sub
13
14    Private Sub Step1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
15        Start_screen.Hide()
16    End Sub
17
18    Private Sub CheckBox1_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles CheckBox1.CheckedChanged
19        If CheckBox1.Checked = True Then
20            CheckBox2.Enabled = False
21            CheckBox3.Enabled = False
22        End If
23        If CheckBox1.Checked = False Then
24            CheckBox2.Enabled = True
25            CheckBox3.Enabled = True
26        End If
27    End Sub
28
29    Private Sub CheckBox2_CheckedChanged(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles CheckBox2.CheckedChanged
30        If CheckBox2.Checked = True Then
31            CheckBox1.Enabled = False
32            CheckBox3.Enabled = False
33        End If
34        If CheckBox2.Checked = False Then
35            CheckBox1.Enabled = True
36            CheckBox3.Enabled = True
37        End If
38    End Sub
39 End Class
```

4.2. Інструкція по використанню навчального тренажеру

Після запуску навчального тренажеру, студенту відображається стартова форма тренажеру з кнопкою «Розпочати» (рис. 4.5)

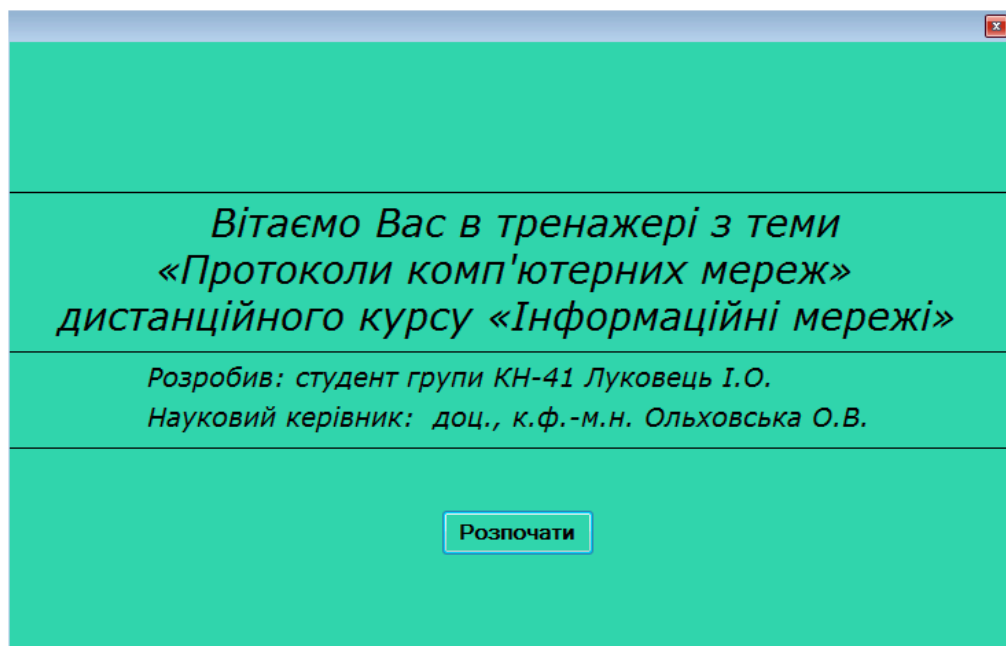


Рисунок 4.5 – стартова форма.

Після натискання кнопки «Розпочати», студенту відкривається перше питання, він може відповісти натиснувши на якесь питання (рис. 4.6)

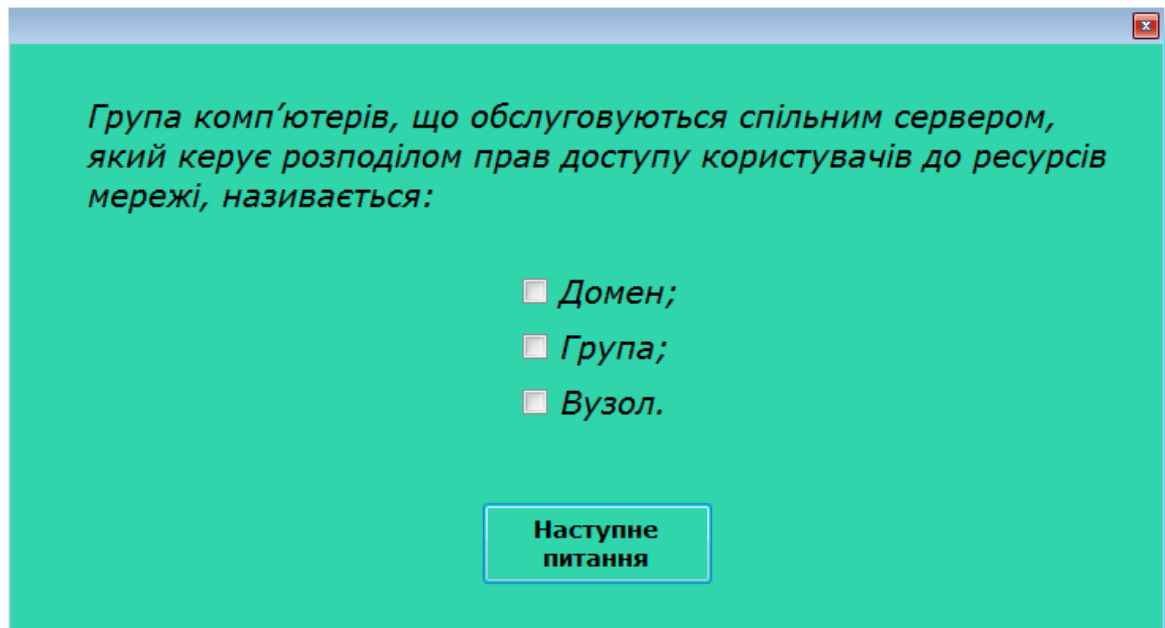


Рисунок 4.6 – перше питання.

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

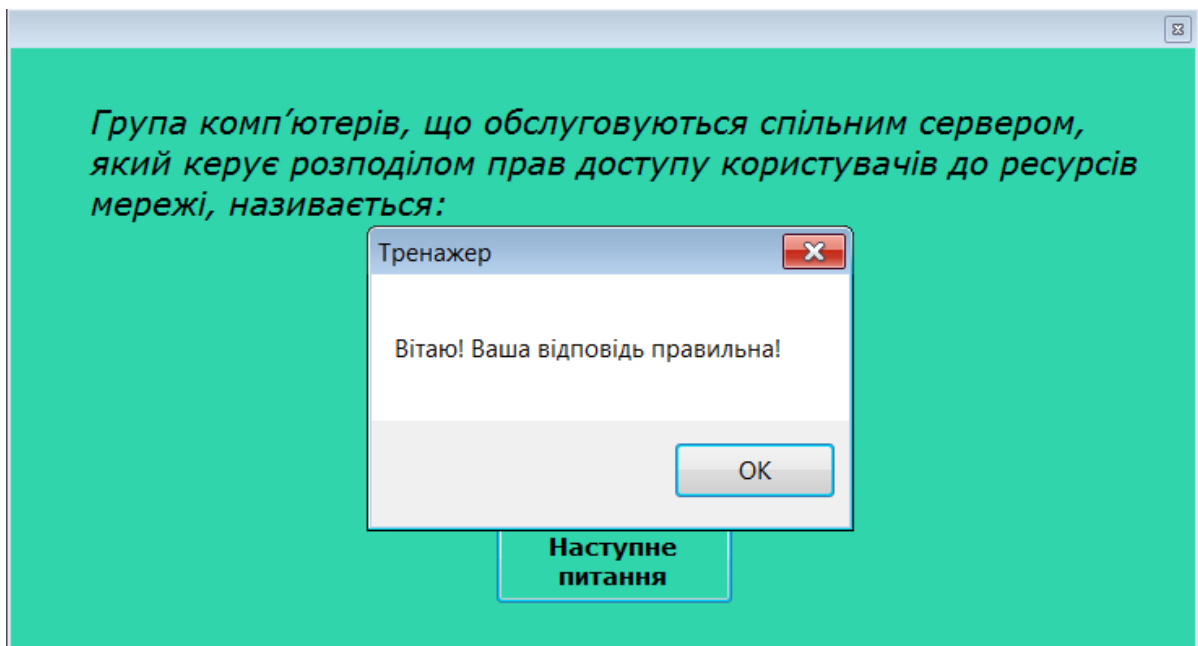


Рисунок 4.7 – вікно вірної відповіді.

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

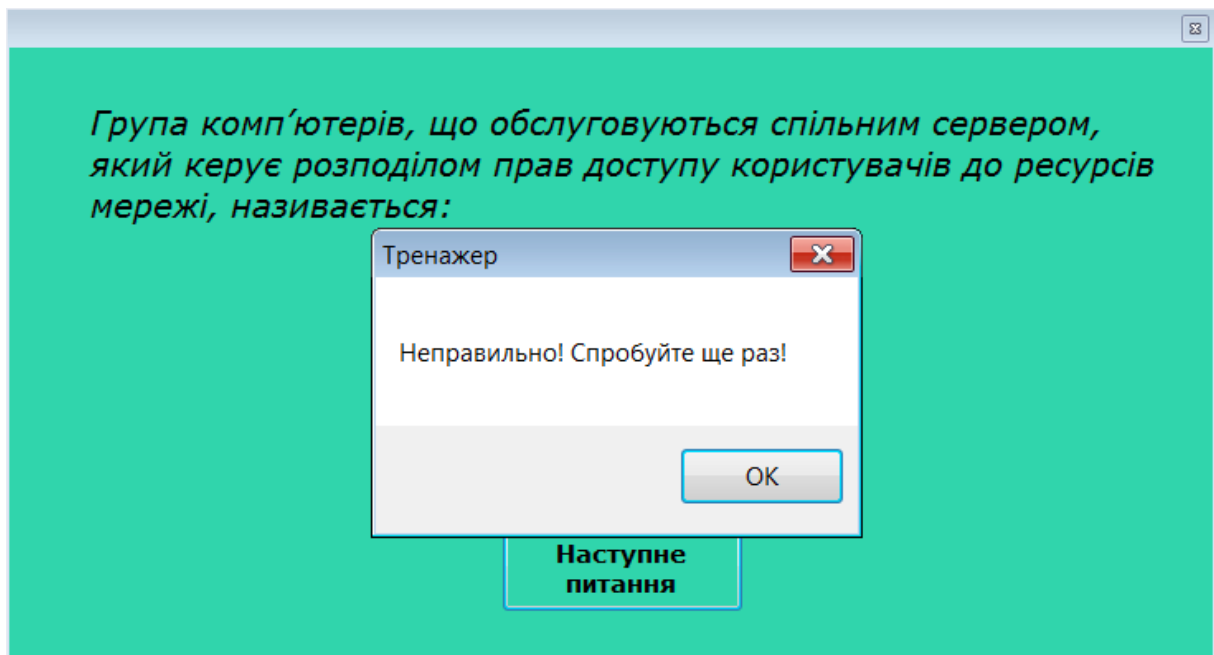


Рисунок 4.8 – вікно невірної відповіді.

Після правильної відповіді, користувач натискає на наступне питання з теми (рис. 4.9)

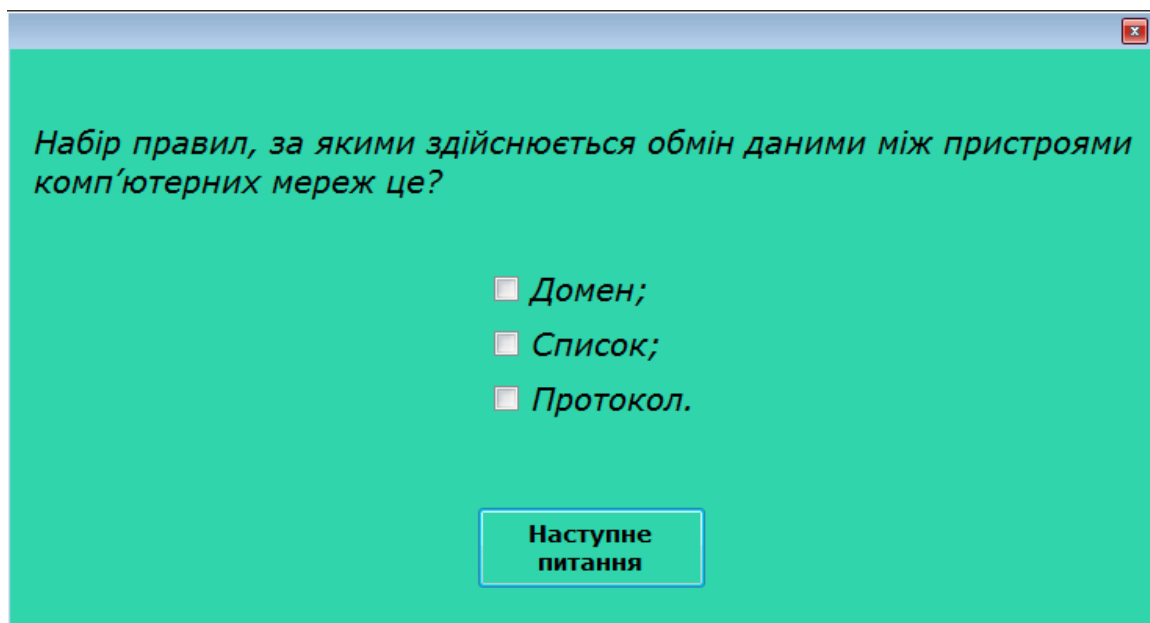


Рисунок 4.9 – вікно питання номер 2.

Потім після правильної відповіді користувачу відкриється питання номер 3 (рис. 5.0)

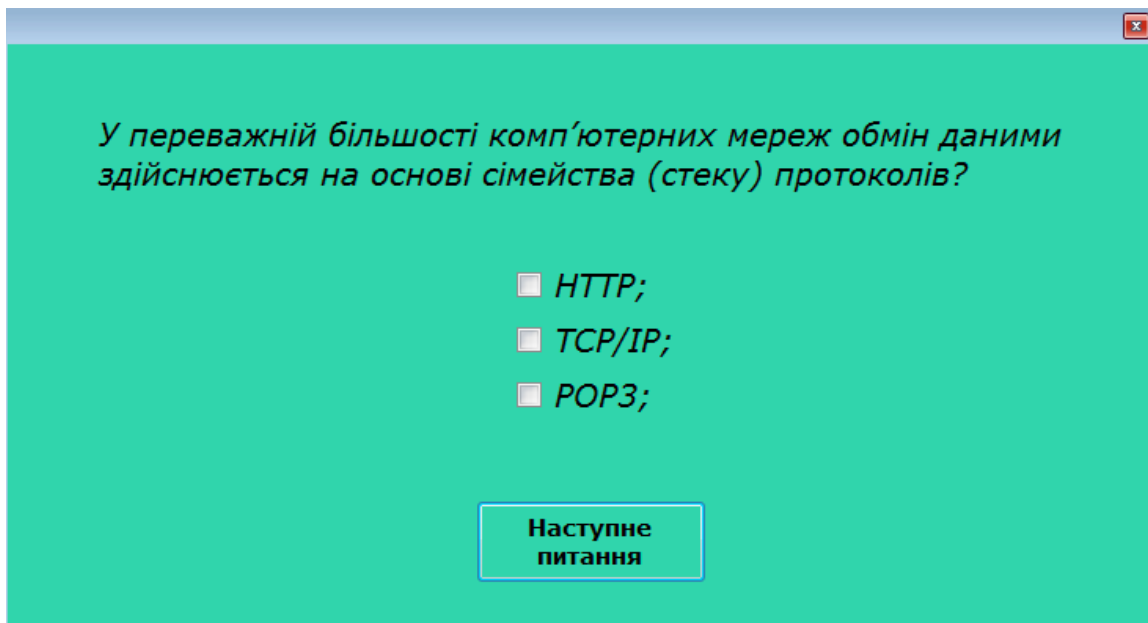


Рисунок 5.0 – вікно питання номер 3.

Після 3 питання користувачу по порядку відкривається 4 питання з теми (рис. 5.1)

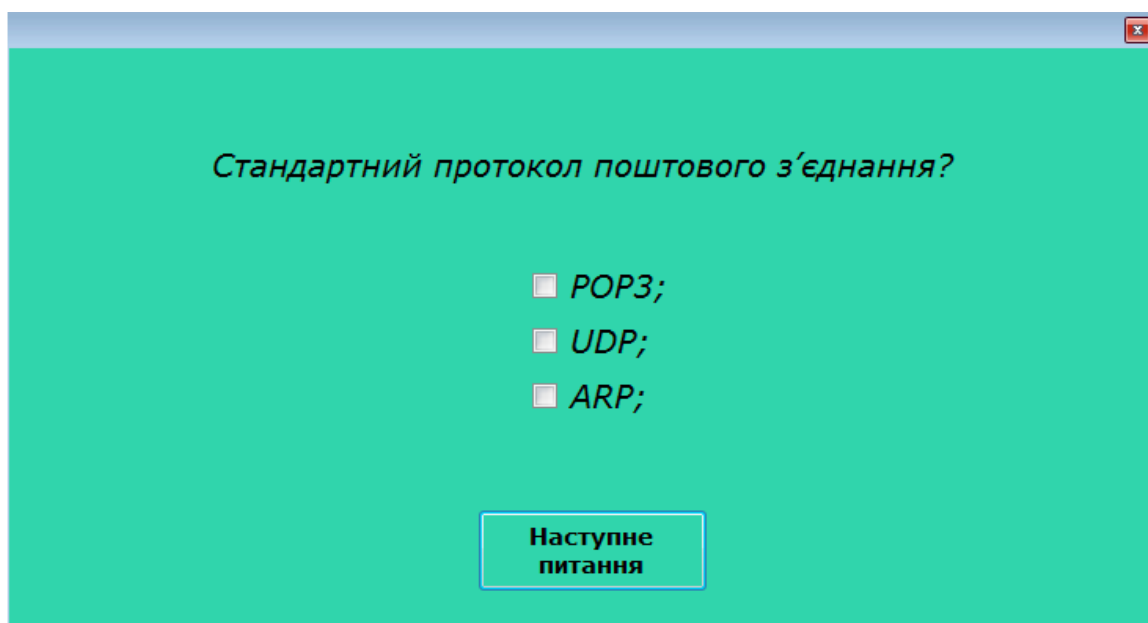
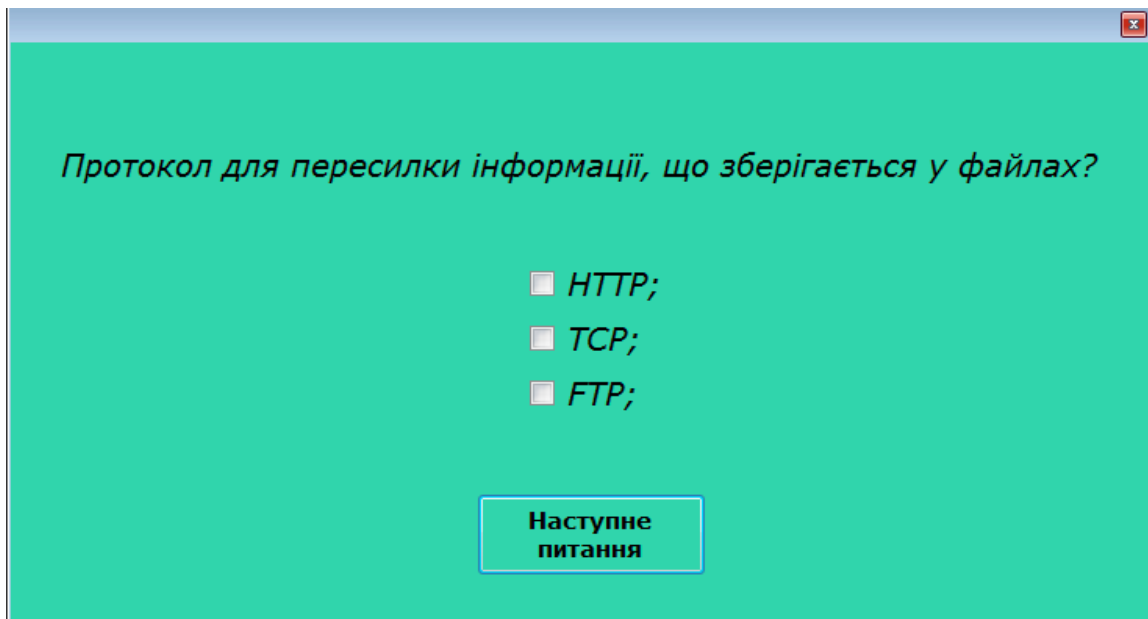


Рисунок 5.1 – вікно питання номер 4.

Якщо користувач вибрав правильний варіант відповіді, йому відкриється питання номер 5 (рис. 5.2)



Протокол для пересилки інформації, що зберігається у файлах?

☐ *HTTP;*

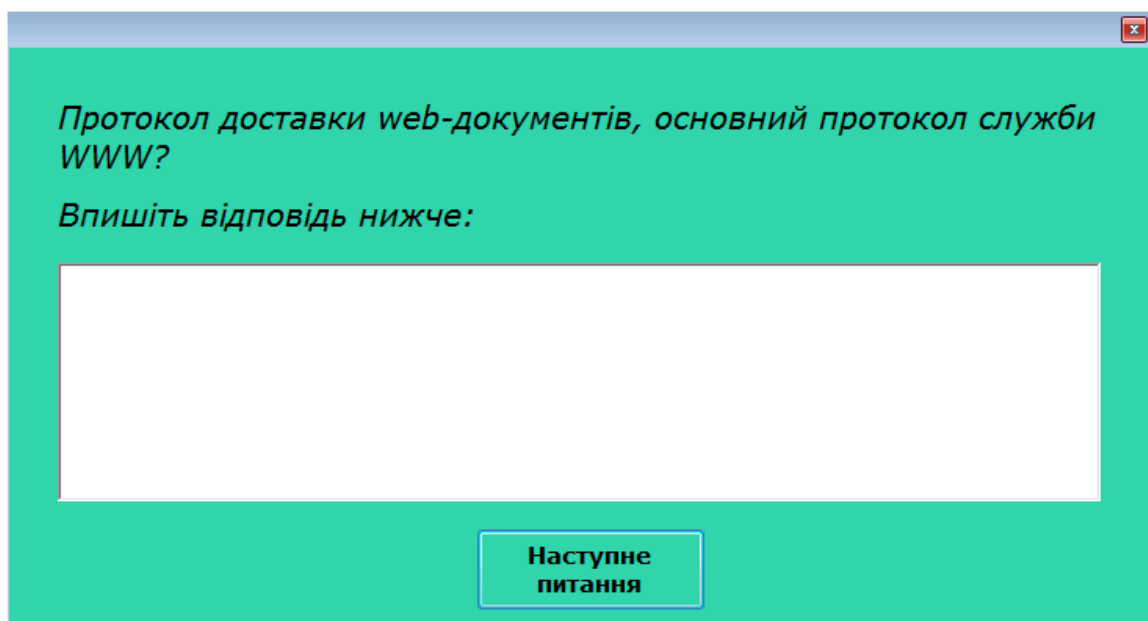
☐ *TCP;*

☐ *FTP;*

Наступне питання

Рисунок 5.2 – вікно питання номер 5.

Після 5 питання, по порядку відкривається питання номер 6 (рис. 5.3)



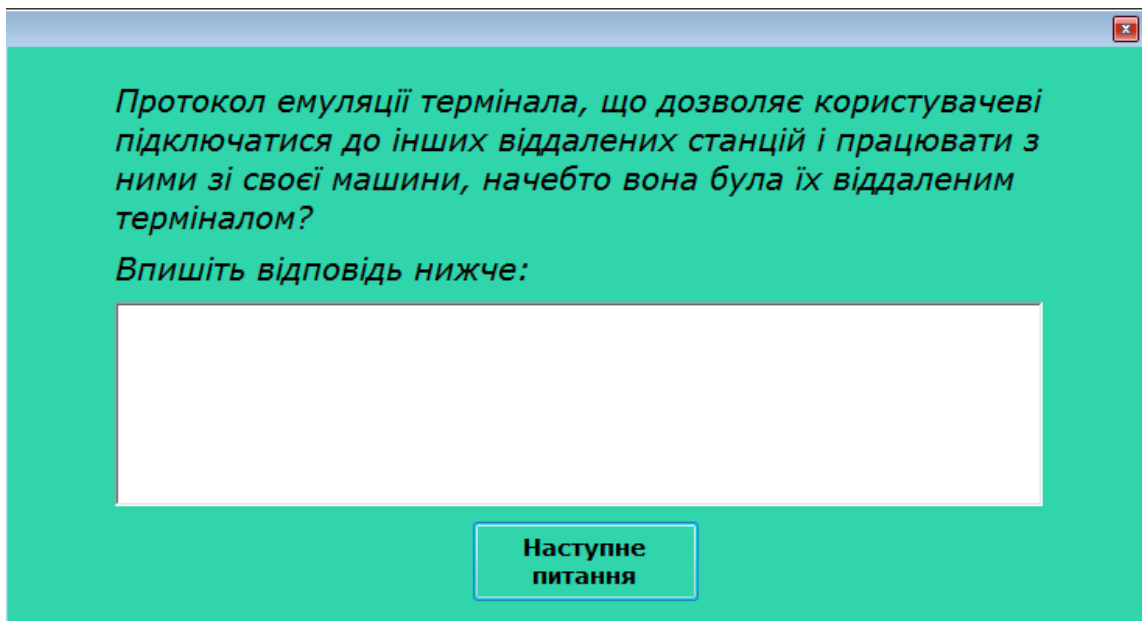
Протокол доставки web-документів, основний протокол служби WWW?

Впишіть відповідь нижче:

Наступне питання

Рисунок 5.3 – вікно питання номер 6.

Потім користувачеві відкривається питання номер 7 (рис. 5.4)



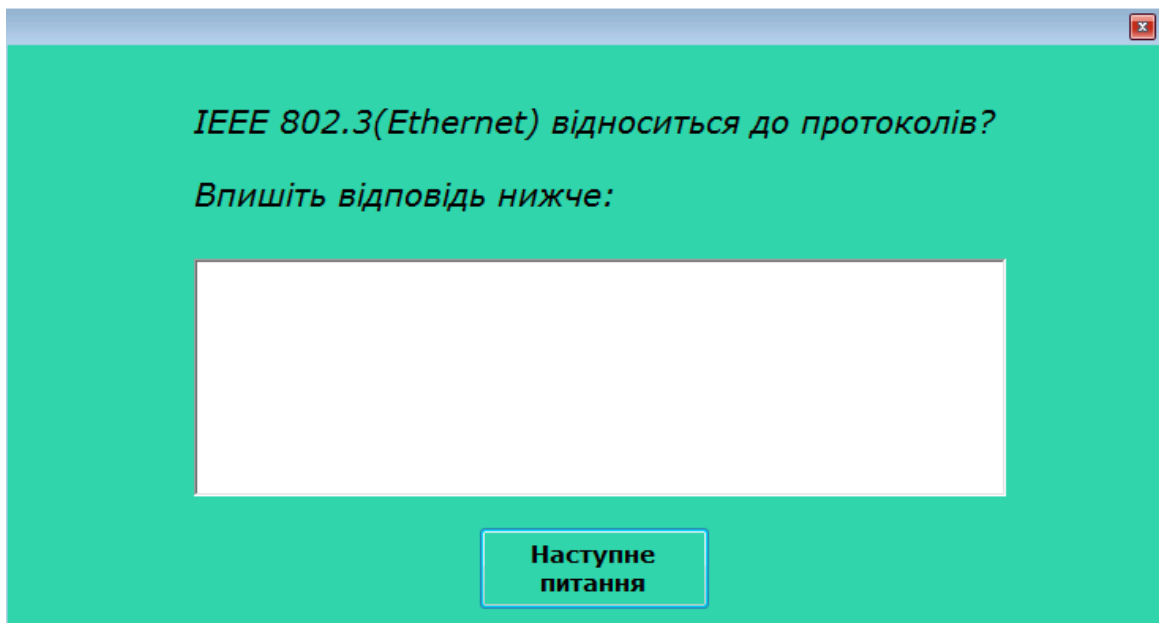
Протокол емуляції термінала, що дозволяє користувачеві підключатися до інших віддалених станцій і працювати з ними зі своєї машини, начебто вона була їх віддаленим терміналом?

Впишіть відповідь нижче:

Наступне питання

Рисунок 5.4 – вікно питання номер 7.

Потім користувачеві відкривається питання номер 8 (рис. 5.5)



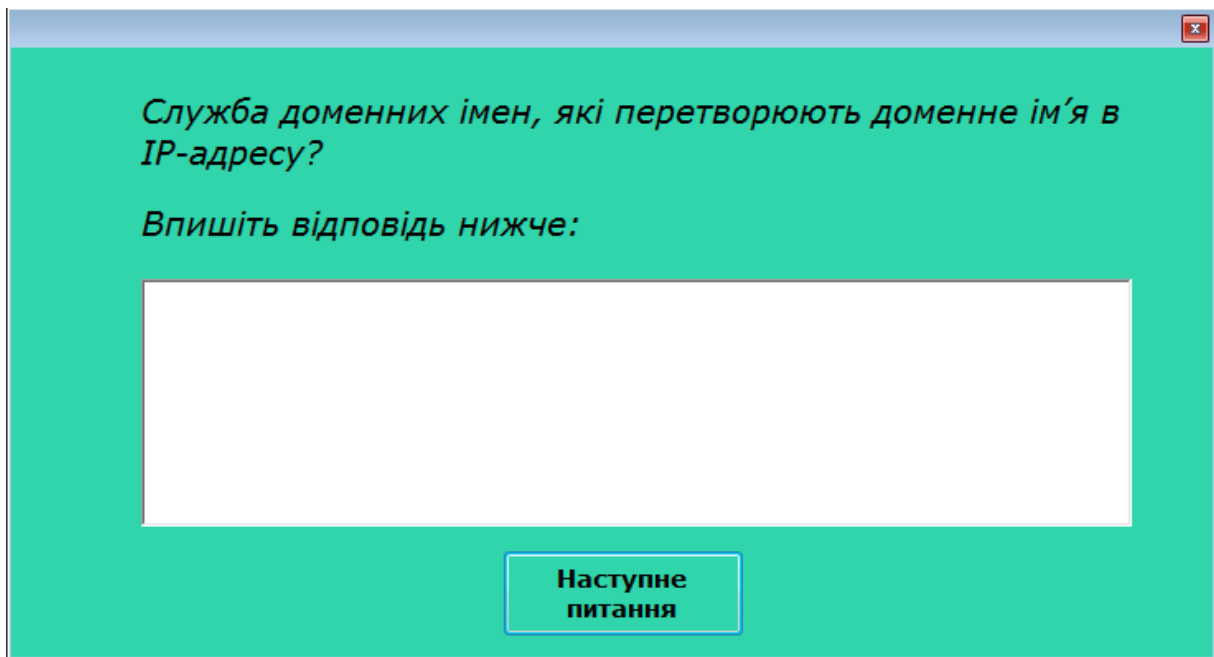
IEEE 802.3(Ethernet) відноситься до протоколів?

Впишіть відповідь нижче:

Наступне питання

Рисунок 5.5 – вікно питання номер 8.

Потім користувачеві відкриється питання номер 9 (рис. 5.6)



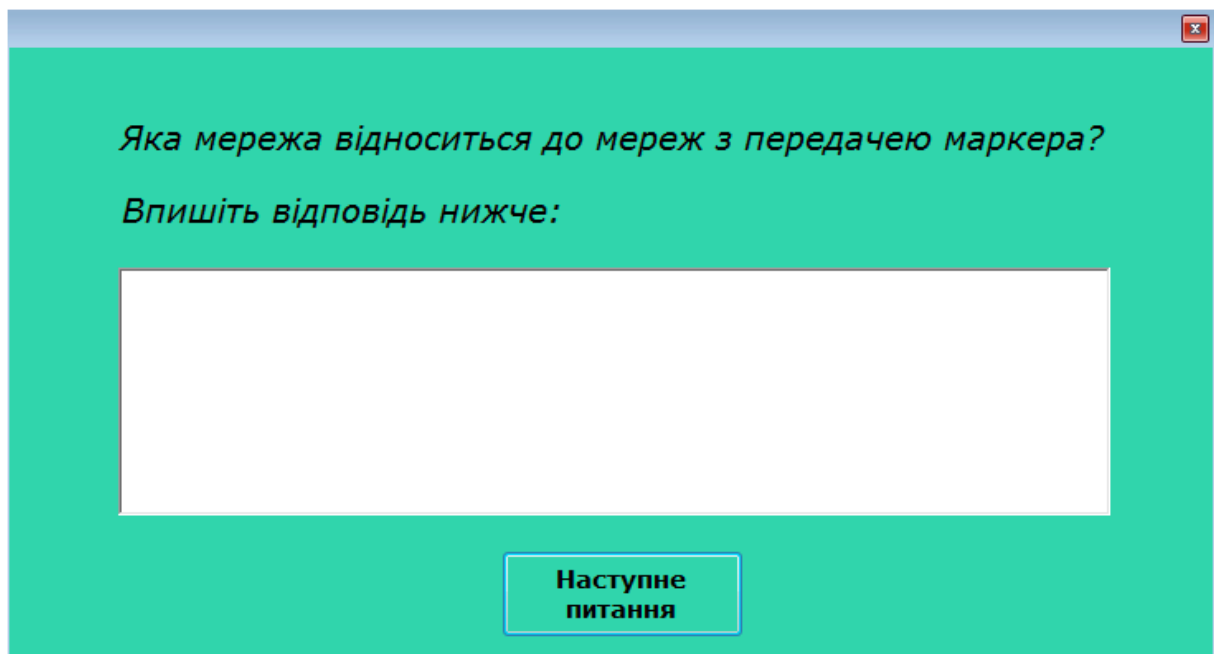
Служба доменних імен, які перетворюють доменне ім'я в IP-адресу?

Впишіть відповідь нижче:

Наступне питання

Рисунок 5.6 – вікно питання номер 9.

Потім користувачеві відкриється питання номер 10 (рис. 5.7)



Яка мережа відноситься до мереж з передачею маркера?

Впишіть відповідь нижче:

Наступне питання

Рисунок 5.7 – вікно питання номер 10.

Потім на десятому питанні користувач натискає кнопку «ОК», відкриється підсумкове вікно тренажера, де буде привітання та кнопка завершення тренінгу (рис. 5.8)

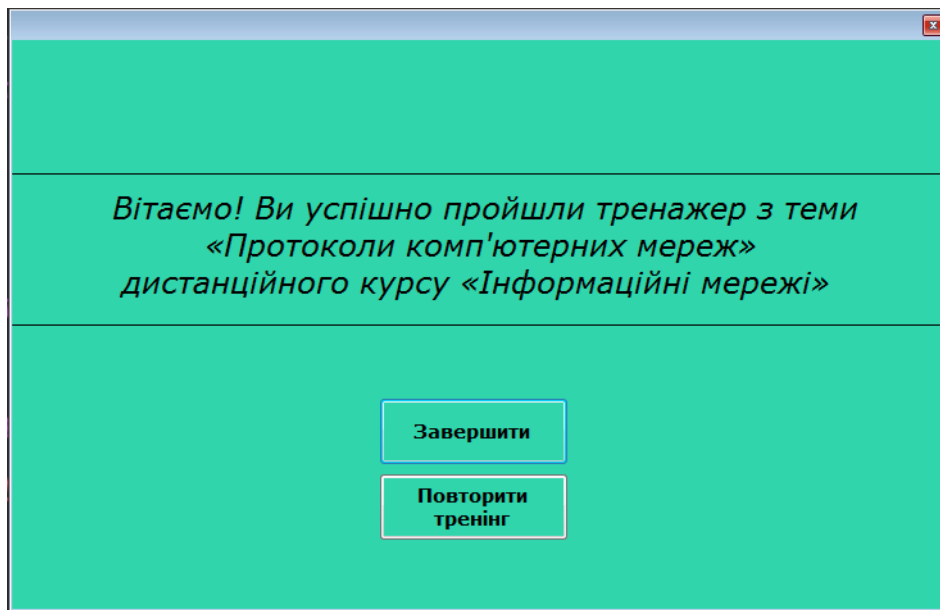


Рисунок 5.8 – підсумкове вікно тренажеру.

4.3. Обґрунтування вибору програмних засобів

Microsoft Visual Basic — засіб розроблення програмного забезпечення, створений і підтримуваний корпорацією Microsoft, який складається з мови програмування і середовища розроблення. Мова Visual Basic успадкувала дух, стиль і частково синтаксис свого предка - мови Бейсік, в якій є чимало діалектів. У той же час Visual Basic поєднує в собі процедури та елементи об'єктно-орієнтованих та компонентно-орієнтованих мов програмування. Середовище розробки VB містить інструменти для візуального конструювання користувацького інтерфейсу.

Visual Basic одночасно і улюблений, і зневажений багатьма програмістами. Visual Basic вважається ідеальним засобом швидкої розробки прототипів програми, розробки додатків баз даних і взагалі для компонентного способу створення програм, що працюють під управлінням операційних систем родини Microsoft Windows.

Перше визнання серйозними розробниками Visual Basic отримав після виходу версії 3 - VB3. Остаточне визнання як повноцінного засобу програмування для Windows - при виході версії 5 - VB5. Версія VB6, що

входить до складу Microsoft Visual Studio 6.0, стала по-справжньому зрілим і функціонально багатим продуктом. Після цього розробники з Microsoft суттєво змінили напрямок розвитку даної технології.

Visual Basic.NET не дозволяє програмувати по-старому, бо по суті є зовсім іншою мовою, такою самою, як і будь-яка інша мова програмування для платформи .NET. Індивідуальність мови і її переваги (простота, скромність створення програм, легкість використання готових компонент) при використанні в середовищі .NET не мають такого значення, як раніше - усе зосереджено на можливостях самої системи .NET, на її бібліотеці класів. Тому сьогодні (лютий 2008) треба говорити про класичний Visual Basic, його діалекти Visual Basic for Applications (VBA) і Visual Basic Scripting Edition (VBScript) і про мову для платформи .NET - Visual Basic.NET. Ця мова дуже сильно прив'язана до свого середовища розроблення й до операційної системи Windows, оскільки вона є виключно інструментом написання Windows-додатків. Прив'язаність до середовища полягає в тому, що існує велика кількість засобів, призначених для допомоги й зручності програмування: вбудований відладчик, перегляд змінних і структур даних на льоту, вікно налагодження, спливаюча підказка при наборі тексту програми (Intellisense). Всі ці переваги роблять марним і навіть неможливим використання Visual Basic поза середовищем для розроблення, наприклад, у звичайному текстовому редакторі. (VBA) це засіб програмування, практично нічим не відрізняється від класичного Visual Basic, що призначене для написання макросів та інших прикладних програм для конкретних програм. Найбільшу популярність здобув завдяки своєму використанню в пакеті Microsoft Office. Широке розповсюдження Visual Basic for Applications в поєднанні з споконвічно недостатньою увагою до питань безпеки призвело до значного поширення макровірусів. (VBScript) Це скриптова мова, що є дещо урізаною версією звичайного Visual Basic. Використовується в основному для автоматизації

адміністрування систем Windows і для створення сторінок ASP та сценаріїв для Internet Explorer.

ВИСНОВКИ

Отже, розробка навчальних тренажерів дисциплін та навчання за дистанційними технологіями є пріоритетною формою навчання в сучасних навчальних умовах. Така форма навчання надає незаперечні переваги як для викладачів, так і для студентів. Дистанційне навчання та використання програм тренажерів дає змогу оптимізувати часові витрати викладача, а також підвищити ефективність процесу навчання в цілому. Студент при цьому стає активним учасником освітнього процесу, спроможним будувати індивідуальну навчальну траєкторію, виходячи з власних потреб. Це сприяє формуванню компетентного фахівця, конкурентоспроможного у сучасних умовах.

Результатом виконання бакалаврської роботи є розроблений навчальний тренажер з теми «Протоколи комп'ютерних мереж» дистанційного курсу «Інформаційні мережі».

Завдання бакалаврської роботи виконано. Створено зручний навчальний тренажер для підготовки та навчання студентів з теми «Протоколи комп'ютерних мереж» дистанційного курсу «Інформаційні мережі».

Основні результати роботи:

- Розроблено технічне завдання, щодо програмування тренажеру з теми «Протоколи комп'ютерних мереж» дистанційного курсу «Інформаційні мережі».
- Розроблено алгоритм роботи тренажера.
- Програмно реалізовано тренажер.
- Проведено тестування та налаштування програмного продукту.
- Проведено опис процесу розробки та тренажеру з теми «Протоколи комп'ютерних мереж» дистанційного курсу «Інформаційні мережі».

- Отримані результати опубліковані в XLVI Міжнародній науковій студентській конференції «Актуальні питання розвитку науки та забезпечення якості освіти у XXI столітті».

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

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 та змішане навчання [Електронний ресурс]. – Режим доступу: <https://www.knuba.edu.ua/wp-content/uploads/2023/01/Proceedings-of-the-V-International-Conference-11-17-2022.pdf#page=306>
3. Дистанційна освіта [Електронний ресурс]. – Режим доступу: http://osvita.ch.ua/distanciyna_osvita/
4. Протокол TCP/IP [Електронний ресурс]. – Режим доступу: <https://studfile.net/preview/7349561/page:34/>
5. Протоколи, інтерфейси мереж [Електронний ресурс]. – Режим доступу: <https://studfile.net/preview/7007236/page:6/>
6. Типи мережевих протоколів і їх призначення [Електронний ресурс]. – Режим доступу: <https://deltahost.ua/ua/tipi-merezhevix-protokoliv-i-ih-priznachennya-http-ip-ssh-ftp-pop3-mac.html>
7. Пояснення протоколів комп'ютерної мережі, основні типи [Електронний ресурс]. – Режим доступу: <https://techukraine.net/%D0%BF%D0%BE%D1%8F%D1%81%D0%BD%D0%B5%D0%BD%D0%BD%D1%8F-%D0%BF%D1%80%D0%BE%D1%82%D0%BE%D0%BA%D0%BE%D0%BB%D1%96%D0%B2-%D0%BA%D0%BE%D0%BC%D0%BF%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%BE%D1%97-%D0%BC/>
8. Мережеві протоколи [Електронний ресурс]. – Режим доступу: <https://mozok.click/741-merezhev-protokoli.html>

9. Основні протоколи Мережі [Електронний ресурс]. – Режим доступу:
<https://hyperhost.ua/info/uk/osnovni-protokoli-merezhi-navishcho-voni-vikoristovuyutsya>
10. ПРОТОКОЛИ, МЕТОДИ І ТЕХНОЛОГІЇ ЗАХИСТУ [Електронний ресурс]. – Режим доступу:
<https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwjGsoqBlcr9AhUy-yoKHd8WATMQFnoECBoQAQ&url=https%3A%2F%2Funivd.edu.ua%2Fscience-issue%2Fissue%2F3328&usg=AOvVaw375X5TtcGsnvnKJnxUZuff>
11. Бібліографічний запис. Бібліографічний опис. Загальні вимоги та правила складання: ДСТУ 7.1-2006. – [Чинний від 2007-07-01]. – К. : Держспоживстандарт України, 2007. – 47 с.
12. Microsoft Visual Studio [Електронний ресурс]. – Режим доступу:
https://uk.wikipedia.org/wiki/Microsoft_Visual_Studio
13. Microsoft Visual Basic [Електронний ресурс]. – Режим доступу:
https://uk.wikipedia.org/wiki/Visual_Basic
14. 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
15. 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
16. Комп'ютерні мережі – Львів, «Магнолія 2006», 2013. – 256 с.
17. Опис редактору Visual Basic [Електронний ресурс]. – Режим доступу:
<https://studfile.net/preview/7785340/>
18. Randolph Visual Studio 2010 for professionals // Randolph, Nick, Gardner, David, Minutillo, Michael, Anderson, Chris.: Trans. with English - М.: LLC "I.D. Williams", 2011. - 1184 p.

19. Appleman D. E72 Win32 API and Visual Basic. For professionals (+SD) / D. Appleman. - Peter, 2002. - 1120 p.
20. Halvorson M. Microsoft Visual Basic 2005. Series "Step by step" / M. Halvorson. - ECOM Publishers. - 640 p.

ДОДАТОК А. КОД ПРОГРАМИ