

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
ПРОГРАМНА РЕАЛІЗАЦІЯ НАВЧАЛЬНОГО ОСВІТНЬОГО ЗАСОБУ З
ТЕМИ «ІТЕРАЦІЙНІ АЛГОРИТМИ»

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

Виконавець роботи Коляка Денис Вікторович
_____ «___» _____ 202_ р.
(підпис)

Науковий керівник к.пед.н., доцент, Кошова Оксана Петрівна
_____ «___» _____ 202_ р.
(підпис)

ПОЛТАВА 2025 р.

ЗАТВЕРДЖУЮЗавідувач кафедри _____ Олена ОЛЬХОВСЬКА
(підпис)

«___» _____ 202__ р.

ЗАВДАННЯ І КАЛЕНДАРНИЙ ГРАФІК ВИКОНАННЯ КВАЛІФІКАЦІЙНОЇ РОБОТИ

на тему Програмна реалізація навчального освітнього засобу з теми «Ітераційні алгоритми».

зі спеціальності 122 Комп'ютерні науки

освітня програма «Комп'ютерні науки»

ступеня магістр

Прізвище, ім'я, по батькові Коляка Денис Вікторович

Затверджена наказом ректора № ____-Н від «___» _____ 202__ р.

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

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

Зміст пояснювальної записки (перелік питань, які потрібно розробити)

ВСТУП

1. ПОСТАНОВКА ЗАДАЧІ РОЗРОБКИ

2. ІНФОРМАЦІЙНИЙ ОГЛЯД

2.1. Що таке інтерактивний навчальний засіб

2.2. Огляд схожого навчального засобу

3. ТЕОРЕТИЧНА ЧАСТИНА

3.1. Точні й ітераційні методи

3.2. Алгоритм роботи програмного забезпечення навчального освітнього засобу з теми «Ітераційні алгоритми»

3.3. Блок-схема освітнього засобу

4. ПРАКТИЧНА ЧАСТИНА

4.1. Обґрунтування вибору програмних засобів

4.2. Опис програмної реалізації освітнього засобу

4.3. Опис роботи освітнього засобу

ВИСНОВКИ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

ДОДАТОК А. КОД ПРОГРАМИ

Перелік графічного матеріалу: 33 ілюстрації.

Консультанти розділів кваліфікаційної роботи

Розділ	Прізвище, ініціали, посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Постановка задачі	Кошова О.П.		
Інформаційний огляд	Кошова О.П.		
Теоретична частина	Кошова О.П.		
Практична частина	Кошова О.П.		

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 _____ Коляка Денис Вікторович
(підпис)Науковий керівник _____ к.пед.н., доц. Кошова О.П.
(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 «__» _____ 202__ р.

Секретар ЕК _____
(підпис) (ініціали та прізвище)

Затверджую

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

Погоджено

Науковий керівник _____
 к.пед.н., Оксана КОШОВА
 «____» _____ 202__ р.

План

кваліфікаційної роботи на тему
 «Програмна реалізація навчального освітнього засобу з теми «Ітераційні алгоритми»»
 зі спеціальності 122 Комп'ютерні науки
 освітня програма 122 «Комп'ютерні науки»
 ступеня магістр
 Прізвище, ім'я, по батькові Коляка Денис Вікторович

ВСТУП**1. ПОСТАНОВКА ЗАДАЧІ РОЗРОБКИ****2. ІНФОРМАЦІЙНИЙ ОГЛЯД**

2.1. Що таке інтерактивний навчальний засіб

2.2. Огляд схожого навчального засобу

3. ТЕОРЕТИЧНА ЧАСТИНА

3.1. Точні й ітераційні методи

3.2. Алгоритм роботи програмного забезпечення навчального освітнього засобу з теми «Ітераційні алгоритми»

3.3. Блок-схема освітнього засобу

4. ПРАКТИЧНА ЧАСТИНА

4.1. Обґрунтування вибору програмних засобів

4.2. Опис програмної реалізації освітнього засобу

4.3. Опис роботи освітнього засобу

ВИСНОВКИ**СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ****ДОДАТОК А. КОД ПРОГРАМИ**

Здобувач вищої освіти _____ Денис КОЛЯКА
 «____» _____ 202__ р.

РЕФЕРАТ

Записка: 43 сторінок, 33 рисунки, 1 додаток, 20 літературних джерел.

Мета роботи – створення програмного забезпечення навчального освітнього засобу з теми «Ітераційні алгоритми».

Об'єкт розробки – розроблений освітній засіб.

Методи дослідження – ітераційні алгоритми, розробка навчальних освітніх засобів.

Проведено інформаційний аналіз ітераційних алгоритмів. Розроблено алгоритм освітнього засобу, намальовано блок-схему програми. Написано програмний продукт освітнього засобу з теми «Ітераційні алгоритми».

Ключові слова: ІТЕРАЦІЙНІ АЛГОРИТМИ, ОСВІТНІЙ ЗАСІБ, ЗАСТОСУНОК.

ЗМІСТ

ВСТУП	7
1. ПОСТАНОВКА ЗАДАЧІ.....	9
2. ІНФОРМАЦІЙНИЙ ОГЛЯД.....	10
2.1. Що таке інтерактивний навчальний засіб	10
2.2. Огляд схожого навчального засобу	14
3. ТЕОРЕТИЧНА ЧАСТИНА	19
3.1. Точні й ітераційні методи	19
3.2. Алгоритм роботи програмного забезпечення навчального освітнього засобу з теми «Ітераційні алгоритми»	22
3.3. Блок-схема освітнього засобу.....	25
4. ПРАКТИЧНА ЧАСТИНА	26
4.1. Обґрунтування вибору програмних засобів.....	26
4.2. Опис програмної реалізації освітнього засобу	28
4.3. Опис роботи освітнього засобу	33
ВИСНОВКИ.....	37
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	38
ДОДАТОК А. ВИХІДНІ КОДИ.....	41

ВСТУП

Актуальність. Актуальність програмного забезпечення зумовлена зростаючою роллю інформаційних технологій у сучасному освітньому процесі та потребою в інтерактивних інструментах, які сприяють глибокому засвоєнню складних тем. Ітераційні алгоритми є однією з головних тем у вивченні алгоритмів і програмування, оскільки вони широко застосовуються у багатьох сферах: від наукових розрахунків до створення програмного забезпечення. Однак ця тема часто викликає труднощі у студентів через абстрактність і багатоступеневість виконання обчислень. Тому впровадження сучасного програмного забезпечення, яке візуалізує процес виконання ітерацій, дозволяє значно підвищити якість навчання.

Використання такого засобу забезпечує можливість створення динамічних навчальних сценаріїв, у яких учні можуть наочно спостерігати за виконанням алгоритмів, змінювати початкові параметри задачі та аналізувати результати кожного кроку. Це не лише полегшує розуміння концепцій ітераційних процесів, а й розвиває алгоритмічне мислення та навички аналізу. Крім того, програмний засіб може включати інтерактивні завдання, автоматичну перевірку рішень і систему підказок, що робить навчальний процес більш ефективним та індивідуалізованим.

Розробка такого програмного забезпечення відповідає сучасним вимогам до цифровізації освіти та підвищення її доступності. Воно дозволяє забезпечити якісний навчальний контент для широкої аудиторії, включаючи учнів загальноосвітніх шкіл, студентів закладів вищої освіти, а також слухачів курсів підвищення кваліфікації.

У світі, де технології швидко розвиваються, навчальні засоби стають невід'ємною частиною освіти. Сучасні методи навчання вимагають інноваційних підходів, а розробка програмного забезпечення для навчальних цілей — це не лише вимога часу, але й ключовий компонент успішної освітньої стратегії.

Ітераційні алгоритми, зокрема в сферах криптографії та кібербезпеки, стають невід'ємною складовою боротьби зі злочинністю та забезпечення національної безпеки. Для здобувачів освіти у вищому навчальному закладі, ітераційні алгоритми, що становлять важливу складову сучасної обчислювальної науки, виявляються особливо необхідними та важливими. Їх розуміння та використання, надає не лише успішність у навчанні, але й можливість майбутнього створення та вдосконалення програмних продуктів в різних сферах. Тому, розробка такого навчального програмного забезпечення є дуже важливим завданням для забезпечення якісної освіти та підготовки конкурентоспроможних фахівців.[2]

Метою роботи є створення програмного забезпечення навчального освітнього засобу з теми «Ітераційні алгоритми».

Об'єктом розробки є розроблений освітній засіб.

Предметом розробки є програмне забезпечення навчального освітнього засобу з теми «Ітераційні алгоритми».

Ця робота поділена на чотири розділи. У першому розділі розкрито постановку завдання роботи. В другому розділі розглянуто ітераційні методи. У третьому розділі наведено теоретичний матеріал з теми ітераційних методів та складено алгоритм функціонування програмного забезпечення. У четвертому розділі описано процес створення застосунку та описано детально про можливості розробленого додатку.

Результатом виконання кваліфікаційної роботи є створення програмного забезпечення навчального освітнього засобу з теми «Ітераційні алгоритми».

Обсяг пояснювальної записки: 45 сторінок, 33 рисунки, 1 додаток, 20 літературних джерел.

1. ПОСТАНОВКА ЗАДАЧІ

Першочерговим завданням кваліфікаційної роботи є створення повноцінного програмного забезпечення навчального освітнього засобу з теми «Ітераційні алгоритми», програмування засобу, розробка блок-схеми та створення детального алгоритму програмного засобу.

Завдання роботи:

- Ознайомлення з матеріалом на тему «Ітераційні алгоритми»;
- Алгоритмізація компонентів освітнього засобу;
- Малювання детальної блок-схеми алгоритму роботи освітнього засобу;
- Створення програмного забезпечення освітнього засобу в зручному середовищі програмування Visual Studio;
- Опис результатів створення освітнього засоби та опис загального підсумку;
- Створення інструкції користувача для програмного забезпечення освітнього засобу.

На етапі програмування, плануються наступні вікна освітнього засобу:

- Перше вікно з привітанням користувача, на якому написано детально тематику освітнього засобу, кнопка початку роботи з програмним забезпеченням, прописано детально інформацію про розробника програми та наукового керівника даного курсового проекту з фаху;
- Наступні вікна програмного засобу будуть містити різні завдання, на тему «Ітераційні алгоритми», для якісної перевірки рівня підготовки користувача.
- Останнє підсумкове вікно містить результати проходження освітнього засобу.

2. ІНФОРМАЦІЙНИЙ ОГЛЯД

2.1. Що таке інтерактивний навчальний засіб

Інтерактивний навчальний засіб — це програмне забезпечення або платформа, що дозволяє користувачеві активно взаємодіяти з навчальним матеріалом. Такі інструменти використовують сучасні технології для створення динамічного навчального середовища, де учні можуть не лише отримувати інформацію, але й практикуватися, тестувати свої знання та отримувати зворотний зв'язок. Інтерактивні навчальні інструменти можуть бути представлені у вигляді мобільних додатків, онлайн-платформ, ігрових симуляцій, відеоуроків або віртуальних лабораторій.

Інтерактивні навчальні засоби стали важливим елементом сучасної освітньої практики. Вони є потужними засобами, що сприяють більш ефективному засвоєнню знань, розвитку навичок та стимулюють мотивацію учнів. Інтерактивні технології змінюють традиційний підхід до навчання, надаючи можливість учням активно взаємодіяти з матеріалом, що дозволяє краще усвідомлювати та застосовувати знання в реальних умовах. В цій статті ми розглянемо основні аспекти інтерактивних навчальних інструментів, їх переваги та перспективи розвитку в контексті сучасної освіти.

Завдяки інтерактивному характеру, такі інструменти можуть включати різні види навчальних активностей: вікторини, тестування, ігри, завдання на програмування, віртуальні екскурсії, інтерактивні діаграми та багато інших форм. Унікальність таких інструментів полягає в тому, що вони не тільки дають можливість отримати знання, а й забезпечують реальну взаємодію з навчальним процесом, що покращує його ефективність.

Інтерактивні тести для перевірки знань користувачів з теми ітераційних алгоритмів є ефективним інструментом навчання та оцінювання рівня розуміння цієї ключової теми інформатики. Ітераційні алгоритми, які реалізують повторювані дії до досягнення певної умови, є основою багатьох

задач, пов'язаних із циклічними обчисленнями. Створення якісних тестів з цієї теми передбачає використання різних форматів питань, що сприяє не лише перевірці знань, але й глибшому розумінню принципів роботи алгоритмів.

Переваги інтерактивних навчальних інструментів

Активне навчання Інтерактивні інструменти дозволяють студентам активно залучатися до навчального процесу, що значно підвищує їхню мотивацію та інтерес до теми. Завдяки активному залученню до вирішення задач та обговорення питань, учні краще запам'ятовують матеріал і розвивають аналітичні та критичні навички.

Індивідуалізація навчання Інтерактивні інструменти можуть бути адаптовані під індивідуальні потреби учнів. Вони дозволяють налаштовувати рівень складності завдань, враховуючи поточні знання та навички користувача. Це створює можливість для персоналізованого навчання, що є особливо важливим у разі різних рівнів підготовки учнів.

Миттєвий зворотний зв'язок Однією з основних переваг інтерактивних інструментів є можливість миттєвої перевірки результатів. Користувач отримує зворотний зв'язок відразу після виконання завдання, що дозволяє йому швидко усвідомити помилки, виправити їх і покращити свої знання. Це допомагає значно пришвидшити процес навчання.

Залучення до навчання через гейміфікацію Багато інтерактивних інструментів використовують елементи гейміфікації, що робить процес навчання більш захоплюючим і веселим. Наприклад, учні можуть заробляти бали або досягнення за успішне виконання завдань, що стимулює їх до подальшого навчання.

Доступність і мобільність Інтерактивні інструменти зазвичай доступні онлайн, що дозволяє учням вивчати матеріал в будь-якому місці та в будь-який час. Це зручно для тих, хто має обмежений доступ до традиційних класних занять або хоче самостійно контролювати свій прогрес.

Залучення візуальних та аудіо ресурсів Інтерактивні платформи зазвичай інтегрують відео, графіку, анімацію та звукові ефекти, що робить навчання

більш наочним і різноманітним. Це дозволяє краще пояснити складні концепції, особливо в таких областях, як фізика, хімія або математика.

Виділення категорій інтерактивних навчальних інструментів

Існує багато різних типів інтерактивних інструментів, кожен з яких має свої переваги залежно від конкретних навчальних цілей. Ось кілька категорій:

Інтерактивні вікторини та тести Такі інструменти допомагають оцінити рівень знань учнів, дозволяючи їм виконувати завдання на вибір правильної відповіді, заповнення пропусків чи рішення логічних задач. Вони можуть бути адаптовані під конкретні теми або курси.

Онлайн-курси з інтерактивними елементами Ці курси можуть включати відеоуроки, завдання для самостійної роботи, інтерактивні дискусії, а також інші ресурси для навчання. Вони є дуже ефективними для навчання в умовах онлайн-освіти.

Ігрові симуляції Ігри та симуляції — це потужний інструмент для навчання, який дозволяє учням практично застосовувати свої знання та отримувати досвід у віртуальних умовах. Ігри можуть бути як розважальними, так і навчальними, наприклад, для вивчення іноземних мов або програмування.

Віртуальні лабораторії Віртуальні лабораторії дають можливість учням проводити експерименти в різних галузях науки без необхідності фізично перебувати в лабораторії. Це дозволяє проводити складні експерименти, які можуть бути небезпечними або дуже дорогими в реальному житті.

Інтерактивні платформи для програмування Платформи, які дозволяють учням писати код, отримувати зворотний зв'язок і покращувати свої навички програмування в реальному часі, стають все більш популярними. Це включає платформи для вивчення різних мов програмування, алгоритмів і структур даних.

Одним із головних ідей інтерактивних тестів є використання питань, які стимулюють активну участь користувачів. Це можуть бути завдання на вибір правильної відповіді, заповнення пропусків у коді, створення псевдокоду, виявлення помилок у готових ітераційних алгоритмах або навіть виконання

міні-проектів у реальному часі. Наприклад, тест може запропонувати учаснику оптимізувати алгоритм обчислення факторіала, використовуючи цикл "while" замість "for". Такі завдання дозволяють оцінити не лише базові знання, але й здатність застосовувати їх у практичних ситуаціях.

Сучасні інтерактивні платформи, такі як Kahoot, Quizizz або Codility, пропонують інтеграцію тестових завдань із візуальними інструментами, що допомагають краще зрозуміти складні концепції. Наприклад, користувач може працювати з візуалізацією роботи циклів у реальному часі, відстежуючи, як змінюються змінні при кожній ітерації. Такий підхід дозволяє ефективніше пояснювати концепцію ітераційних алгоритмів, таких як сортування, пошук мінімального або максимального значення у масиві, обчислення середнього арифметичного тощо.

Окремо варто виділити роль завдань із відкритою відповіддю. Вони змушують користувачів самостійно формулювати розв'язки, що стимулює критичне мислення. Наприклад, учасникам можуть запропонувати написати ітераційний алгоритм, який визначає, чи є число простим, або реалізувати симуляцію роботи банкомата, що видає мінімальну кількість банкнот. Такі задачі дозволяють виявити глибину розуміння принципів циклів та умов у програмуванні.

Не менш важливою є можливість миттєвого зворотного зв'язку, що надається інтерактивними тестами. Автоматична перевірка правильності розв'язку, пояснення помилок та рекомендації щодо їх виправлення дозволяють користувачам одразу отримувати корисну інформацію для вдосконалення своїх знань. Наприклад, якщо користувач неправильно реалізує умову виходу з циклу, тест може показати, як це впливає на кінцевий результат, та запропонувати оптимальне рішення.

Ще одним важливим елементом інтерактивних тестів є адаптивність. Завдяки алгоритмам, які аналізують успішність виконання завдань, система може пропонувати наступні питання відповідно до рівня підготовки користувача. Наприклад, якщо учасник успішно справляється із базовими

завданнями, йому пропонують більш складні задачі, як-от оптимізація ітераційного алгоритму або аналіз його часової складності.

Інтерактивні тести з ітераційних алгоритмів є потужним інструментом для викладачів, які прагнуть зробити процес навчання цікавим і результативним. Вони сприяють формуванню практичних навичок, розвитку логічного мислення та підготовці до реальних завдань, що виникають у професійній діяльності. Таким чином, інтерактивні тести забезпечують всебічний підхід до навчання та перевірки знань з теми ітераційних алгоритмів, створюючи міцну основу для подальшого вивчення інформатики.

2.2. Огляд схожого навчального засобу

TestDome — це онлайн-платформа для оцінки навичок кандидатів, яка призначена для проведення технічних та нетехнічних тестів різного рівня складності. Платформа використовується компаніями для перевірки кваліфікації кандидатів при прийомі на роботу, а також для оцінки їхніх професійних навичок.

Типи тестів:

Технічні тести: платформа пропонує тести для програмістів, математичні задачі, перевірку алгоритмів, розробку програмного забезпечення та інші IT-навички.

Нетехнічні тести: тести для оцінки м'яких навичок (soft skills), таких як комунікація, лідерство, продажі, управління часом тощо.

Можливість налаштування тестів: компанії можуть створювати власні тести, адаптуючи їх до специфіки вакансії або вимог до кандидатів.

Оцінка результатів:

Тести включають автоматичне оцінювання відповідей, що дозволяє швидко отримувати об'єктивні результати.

Платформа також пропонує можливість тестувати кандидатів на практичні завдання, де вони можуть продемонструвати свої навички в реальному контексті.

Простий у використанні інтерфейс для створення та проведення тестів.

Платформа дозволяє керувати кандидатами, моніторити їхні успіхи та порівнювати результати.

Безпека та конфіденційність:

TestDome має вбудовані функції безпеки, які запобігають шахрайству, наприклад, блокування можливості копіювання або обміну відповідями між кандидатами.

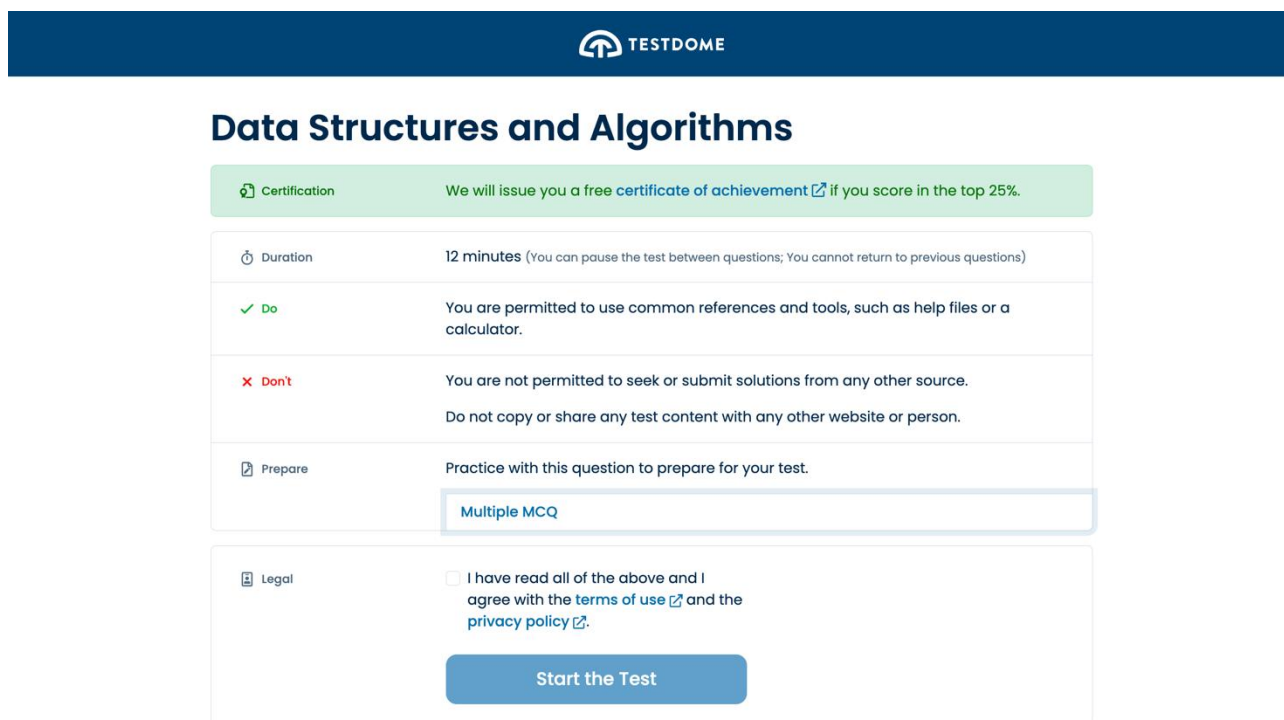
Підтримка різних форматів тестів:

Платформа дозволяє включати різноманітні формати тестів: питання з вибором однієї або кількох відповідей, програмування, реальні завдання на вирішення проблем тощо.

Інтеграція з іншими інструментами:

TestDome може бути інтегрований з іншими системами управління кандидатами (ATS) та HR-платформами.

Розглянемо основний інтерфейс цього інструменту на прикладі перевірки знань користувачів з теми алгоритмів та структур даних. (рис 2.1)



TESTDOME

Data Structures and Algorithms

 **Certification** We will issue you a free [certificate of achievement](#) if you score in the top 25%.

 Duration	12 minutes (You can pause the test between questions; You cannot return to previous questions)
 Do	You are permitted to use common references and tools, such as help files or a calculator.
 Don't	You are not permitted to seek or submit solutions from any other source. Do not copy or share any test content with any other website or person.
 Prepare	Practice with this question to prepare for your test. Multiple MCQ
 Legal	☐ I have read all of the above and I agree with the terms of use and the privacy policy .

Start the Test

Рисунок 2.1 – Головна сторінка застосунку

Після початку перевірки знань, користувачу відображається основний список завдань на тему алгоритмів та структур даних. На відповіді дається 4 хвилини. (рис 2.2)

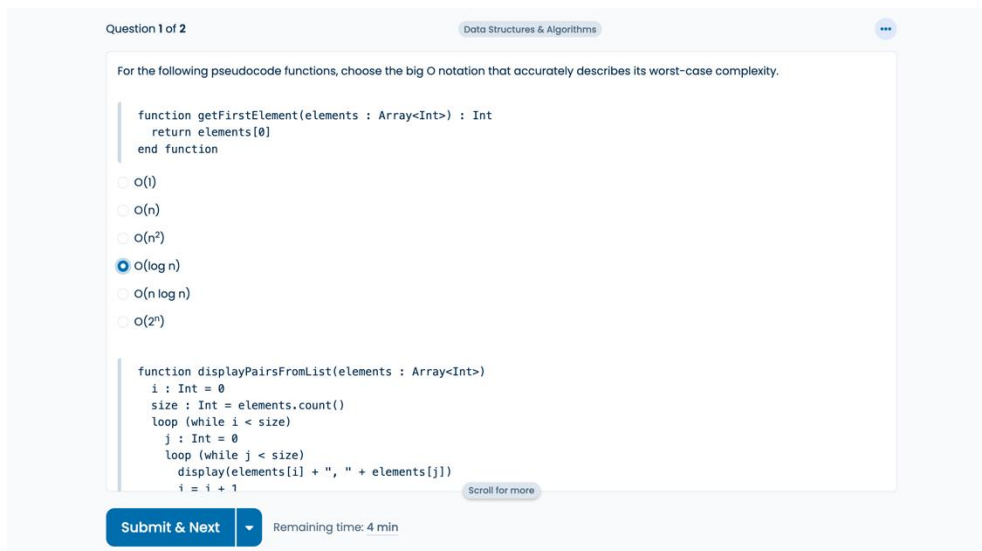


Рисунок 2.2 – Список завдань додатку

Також у користувача є можливість поставити таймер на паузу перед наступним питанням, якщо користувач заморився. (рис 2.3)

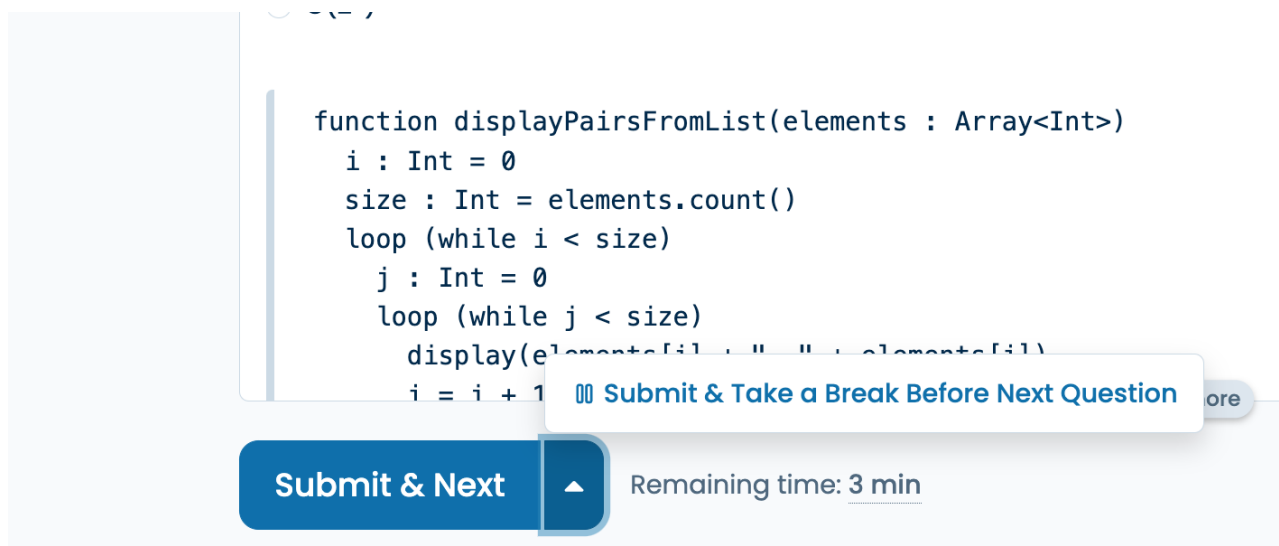


Рисунок 2.3 – Можливість поставити паузу таймеру

В кінці проходження завдань користувачу система додатково виведе сповіщення і додатково попросить користувача підтвердити завершення завдань. (рис 2.4)

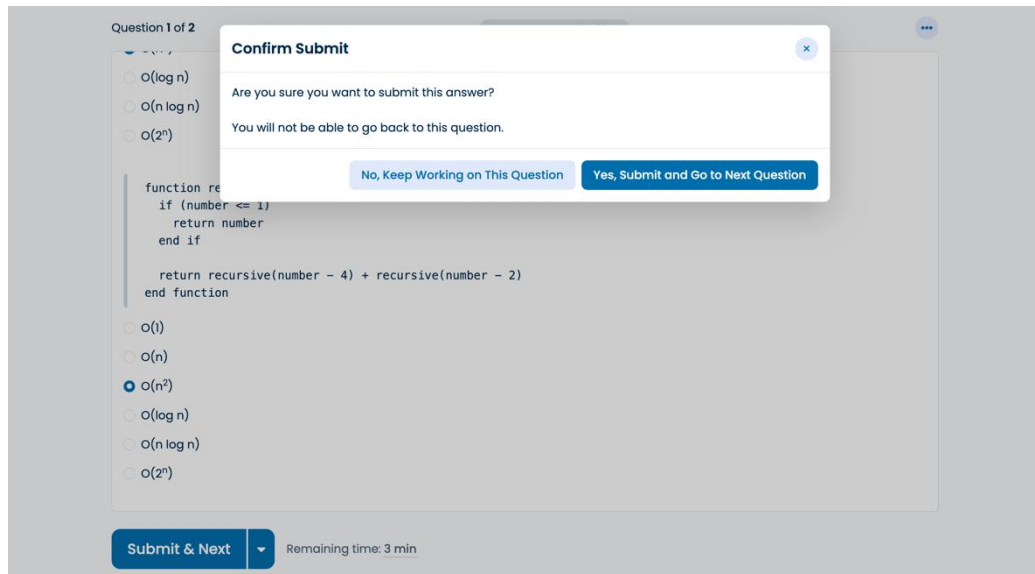


Рисунок 2.4 – Підтвердження перед закінченням завдань

В кінці система виведе користувачу його результати у відсотках, також є кнопка для повернення до інших завдань (рис 2.5)

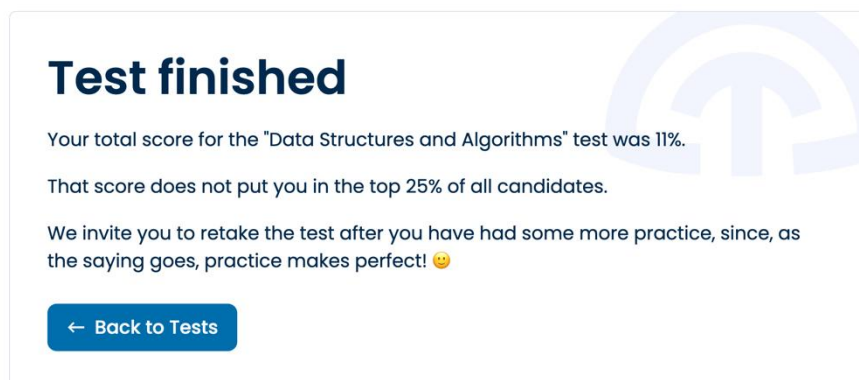


Рисунок 2.5 – Вивід результатів

Плюси - TestDome включають зручний і простий інтерфейс для створення та проведення тестів, що дозволяє швидко оцінювати навички кандидатів. Платформа пропонує різноманітні типи тестів, від технічних до нетехнічних, що дозволяє компаніям оцінювати як професійні, так і м'які навички. Тести автоматично оцінюються, що значно пришвидшує процес відбору, а також дає об'єктивні результати без людського впливу. Крім того, TestDome дозволяє створювати кастомізовані тести, що підходять під конкретні вимоги вакансії або компанії. Інтеграція з іншими HR-системами та системами управління кандидатами (ATS) робить процес найму ще зручнішим. Також платформа забезпечує високий рівень безпеки, що запобігає шахрайству, наприклад, через блокування можливості копіювати або обмінюватися відповідями між кандидатами.

Мінуси - TestDome можуть включати обмежену кількість тестів для деяких спеціалізацій або галузей, що може вимагати додаткової кастомізації тестів. Хоча платформа надає багато можливостей для налаштування, для компаній з нестандартними вимогами можуть виникнути труднощі при створенні тестів, які точно відповідають їхнім потребам. Крім того, автоматичне оцінювання може не завжди відображати весь спектр можливостей кандидата, особливо у випадку складних завдань, де потрібен більш глибокий аналіз відповідей. Для деяких користувачів ціна підписки може бути високою, особливо для малих або середніх підприємств, що обмежує доступність платформи для деяких компаній.

3. ТЕОРЕТИЧНА ЧАСТИНА

3.1. Точні й ітераційні методи

Методи розв'язування систем лінійних алгебраїчних рівнянь поділяються на дві групи:

1. Точні методи.
2. Ітераційні методи.

Точні методи розв'язування – це методи, що дозволяють одержати точне значення всіх невідомих у результаті скінченного числа арифметичних операцій.

Ітераційні методи – це методи, що дозволяють одержати розв'язок у вигляді границі послідовності векторів, побудову яких проводять однаковим процесом, названим процесом ітерації.

За цих умов для знаходження кореня системи використовують *наближені* чисельні методи.

Нехай потрібно розв'язати систему $Ax = b$; $b \in R^m$, $A \in R^{m \times m}$.

Уведемо матрицю $Q \in R^{m \times m}$ як матрицю, що апроксимує матрицю

А. Матриця Q повинна мати вигляд, що дозволяє спростити розв'язування наближеної системи: $Ax = b$; $\Rightarrow Qu = d$.

На практиці така матриця найчастіше є нижньотрикутною, верхньотрикутною, діагональною або добутком скінченного числа таких простих матриць.

Отже, розв'язування системи $Qu = d$, де Q – матриця розщеплення, не повинно представляти особливих труднощів, оскільки розв'язок може бути отриманий з використанням алгоритму, подібного до методу Гауса.

Перепишемо систему $Ax = b$ у вигляді: $(Q - Q + A)x = b$.

Використовуючи еквівалентні перетворення, запишемо:

$$Qx = (Q - A)x + b$$

Введемо верхній індекс, як номер ітерації

$$Qx^{(n+1)} = (Q - A)x^{(n)} + b \quad n; = 0, 1, 2, \dots$$

Поділивши обидві частини виразу на Q , одержимо:

$$x^{(n+1)} = Bx^{(n)} + d \quad n; = 0, 1, 2, \dots, \quad (3.60)$$

$$\text{де} \quad B = Q^{-1}A = I - Q^{-1}A; \quad d = Q^{-1}b, \quad I - \text{одинична матриця.}$$

Q

Це лінійна схема першого порядку. Оскільки ітераційна матриця B на ітераціях є постійною, то таку схему називають стаціонарною.

Метод простих ітерацій

$$x^{(n+1)} = Bx^{(n)} + d \quad n; = 0, 1, 2, \dots$$

сходиться до єдиного розв'язку початкової системи

$$Ax = b;$$

при будь-якому початковому наближенні $x^{(0)}$ зі швидкістю, не повільнішою за геометричну прогресію, якщо будь-яка норма матриці B є меншою за одиницю.

$$\|B\| < 1, \text{ де}$$

$$\|B\|_l = \max_{1 \leq j \leq n} \sum_{i=1}^n |b_{ij}| - l\text{-норма}, \quad \|B\|_m = \max_{1 \leq i \leq n} \sum_{j=1}^n |b_{ij}| - m\text{-норма},$$

$$\|B\|_2 = \sqrt{\sum_{i,j} |b_{ij}|^2} - \text{норма Фробеніуса}$$

Обчислювальну схему визначають способом вибору матриці розщеплення Q . Представимо

$$A = A^- + D + A^+,$$

де D – діагональна матриця з діагональними членами матриці A ;

A^- – нижньотрикутна матриця, що містить частину матриці A , що лежить нижче центральної діагоналі;

A^+ – верхньотрикутна матриця, що містить частину матриці A , що лежить вище центральної діагоналі.

$$A = A^- + D + A^+,$$

$$\text{де } A^- = \begin{pmatrix} 0 & 0 & \dots & 0 \\ a_{21} & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & 0 \end{pmatrix}, A^+ = \begin{pmatrix} 0 & a_{12} & \dots & a_{1m} \\ 0 & 0 & \dots & a_{2m} \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & 0 \end{pmatrix},$$

$$D = \begin{pmatrix} a_{11} & 0 & \dots & 0 \\ 0 & a_{22} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & a_{mm} \end{pmatrix}. \text{ Виберемо як матрицю розщеплення } Q = D.$$

Підставимо в загальну ітераційну формулу

$$x^{(n+1)} = Bx^{(n)} + d; \quad n = 0, 1, 2, \dots$$

значення B й d з виразів $B = -I - Q^{-1}A$; $d = Q^{-1}b$. Тоді

одержимо:

$$x^{(n+1)} = (I - Q^{-1}A)x^{(n)} + Q^{-1}b; \quad n = 0, 1, 2, \dots$$

Якщо $Q = D$ то $Q^{-1} = D^{-1}$. Звідси

$$x^{(n+1)} = (I - D^{-1}A)x^{(n)} + D^{-1}b; \quad n = 0, 1, 2, \dots$$

Підставимо замість A її представлення у вигляді суми матриць:

$$A = A^- + D + A^+$$

виконаємо елементарні перетворення:

або в неявній формі

Із цих виразів видно, що якщо ітераційний метод сходиться, то він сходиться до розв'язку початкової системи, оскільки при цьому $Ax^{(n)} = b$.

Ітераційний параметр – це параметр, який залежить від номера ітерації і введений в ітераційну формулу з метою прискорення збіжності ітераційного процесу.

Вибір ітераційних параметрів з'ясовується при дослідженні збіжності.

Однокроковим ітераційним методом називають такий метод, у якому для знаходження $x^{(n+1)}$ потрібно пам'ятати тільки одну попередню ітерацію $x^{(n)}$. Ітераційний метод може бути записаний у різний спосіб.

Тому вводиться стандартна (канонічна) форма запису ітераційних методів у матричній формі.

3.2. Алгоритм роботи програмного забезпечення навчального освітнього засобу з теми «Ітераційні алгоритми»

Детальні кроки програмного забезпечення:

Крок перший: запуск навчального освітнього засобу;

Крок другий: запуск та завантаження головного стартового вікна додатку;

Крок третій: запуск та завантаження головного стартового вікна додатку;

Крок четвертий: відкриття секції з питаннями навчального освітнього засобу;

Крок п'ятий: Що таке ітераційний алгоритм?

- a) Алгоритм, що використовує рекурсію для вирішення завдання
- b) Алгоритм, що використовує цикли для повторення послідовності дій (+)
- c) Алгоритм, що використовує функціональне програмування для обробки даних

Крок шостий: Який з наведених циклів не є ітераційним?

- a) for
- b) while
- c) if (+)

Крок сьомий: Яка основна різниця між ітераційним і рекурсивним алгоритмом?

- a) Рекурсивний алгоритм завжди ефективніший
- b) Ітераційний алгоритм не викликає функції рекурсії (+)

с) Рекурсивний алгоритм не використовує цикли

Крок восьмий: Яка складність часу (time complexity) ітераційного алгоритму з використанням циклу for для обробки масиву довжини n?

a) $O(n)$ (+)

b) $O(\log n)$

c) $O(n^2)$

Крок дев'ятий: Які види завдань найкраще вирішувати за допомогою ітераційних алгоритмів?

a) Завдання, що потребують розгалуження

b) Завдання, що потребують пошуку

c) Завдання, що потребують повторних обчислень (+)

Крок десятий: Який цикл краще використовувати для ітераційних алгоритмів, якщо потрібно забезпечити, щоб код виконувався хоча б один раз?

a) for

b) while

c) do-while (+)

Крок одинадцятий: Яка перевага використання ітераційних алгоритмів над рекурсивними?

a) Вища швидкість виконання (+)

b) Більша простота у виразі

c) Легша реалізація стеку викликів

Крок дванадцятий: Якість якої структури даних найбільше підходить для ітераційних алгоритмів?

a) Стек (+)

b) Черга

c) Дерево

Крок тринадцятий: У чому полягає основна перевага використання ітераційних алгоритмів для обробки великих наборів даних?

a) Менше використання пам'яті

b) Менше шансів на переповнення стеку викликів

с) Швидше виконання завдання (+)

Крок чотирнадцятий: Який з наступних прикладів найкраще демонструє використання ітераційного алгоритму?

а) Обчислення факторіалу числа

б) Пошук максимального елемента у впорядкованому масиві (+)

с) Розв'язання задачі про Ханойські вежі за допомогою рекурсії

3.3. Блок-схема освітнього засобу

Блок-схема освітнього засобу (рис. 3.1)



Рисунок 3.1 - Блок-схема освітнього засобу.

4. ПРАКТИЧНА ЧАСТИНА

4.1. Обґрунтування вибору програмних засобів

VB.NET — це мова програмування, розроблена корпорацією Майкрософт як частина платформи .NET. Це еволюція класичної мови Visual Basic (VB), розробленої як сучасна об'єктно-орієнтована мова програмування, яка є частиною ширшої екосистеми .NET. VB.NET має багато спільних функцій із C# (C-sharp), іншою мовою сімейства .NET, але має синтаксис, більш знайомий розробникам, які використовували попередні версії Visual Basic.

Найпопулярніші випадки використання VB.NET:

Програми для робочого столу Windows:

Опис: VB.NET зазвичай використовується для розробки настільних програм Windows із графічним інтерфейсом користувача (GUI). Він надає функції для створення інтуїтивно зрозумілих і зручних програм за допомогою Windows Forms.

Програми корпоративного рівня:

Опис: VB.NET підходить для створення великих корпоративних програм. Його інтеграція з .NET Framework дозволяє розробникам використовувати широкий спектр бібліотек і фреймворків для різних функцій.

Веб-додатки:

Опис. Хоча ASP.NET із використанням C# є більш поширеним для веб-розробки, VB.NET також можна використовувати для створення веб-додатків. Веб-форми ASP.NET і ASP.NET MVC підтримують VB.NET як мову програмування.

Програми бази даних:

Опис: VB.NET часто використовується для розробки програм баз даних, включаючи програми, які взаємодіють із Microsoft SQL Server або іншими базами даних. Його інтеграція з ADO.NET і LINQ забезпечує надійні можливості доступу до даних.

Служби Windows:

Опис: VB.NET підходить для розробки служб Windows, які є фоновими процесами, що виконуються незалежно від інтерфейсу користувача. Служби Windows зазвичай використовуються для таких завдань, як заплановані операції, обробка даних або обслуговування системи.

Аналіз даних і звітність:

Опис: VB.NET можна використовувати для розробки додатків, які передбачають аналіз даних, звітування та візуалізацію. Він може взаємодіяти з базами даних, обробляти дані та створювати звіти за допомогою таких інструментів, як Crystal Reports.

Скрипти автоматизації:

Опис: VB.NET можна використовувати для створення сценаріїв автоматизації для різних завдань. Наприклад, розробники можуть використовувати VB.NET для створення сценаріїв повторюваних завдань у Excel, Word або інших програмах Microsoft Office.

Освітні цілі:

Опис: VB.NET часто використовується як мова навчання в навчальних закладах через її простоту та легкість розуміння. Це хороший вступ до концепцій програмування для початківців.

Обслуговування застарілої системи:

Опис: багато організацій мають існуючі програми, написані на VB.NET або Visual Basic 6. VB.NET використовується для підтримки та оновлення цих застарілих систем.

Швидка розробка додатків (RAD):

Опис: VB.NET підтримує швидку розробку програм, дозволяючи розробникам швидко створювати прототипи або прості програми. Функції перетягування Visual Studio спрощують розробку інтерфейсів.

4.2. Опис програмної реалізації освітнього засобу

Додаток навчального освітнього засобу написано за допомогою мови програмування Visual Basic у середовищі Visual Studio. Наступним етапом є створення нового проекту типу Windows Forms. (рис. 4.1)

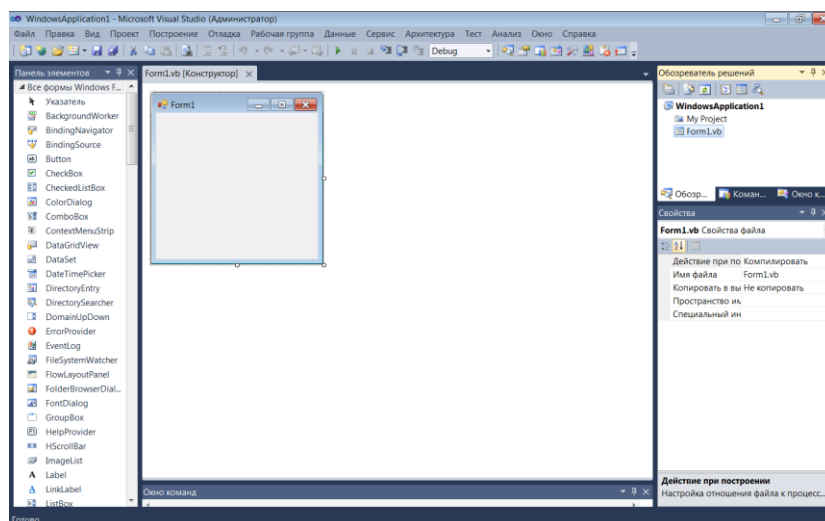


Рисунок 4.1 – створення нового проекту Windows Forms.

Після створення нового проекту в середовищі розробки розробляється основне вікно навчального освітнього засобу. (рис. 4.2)

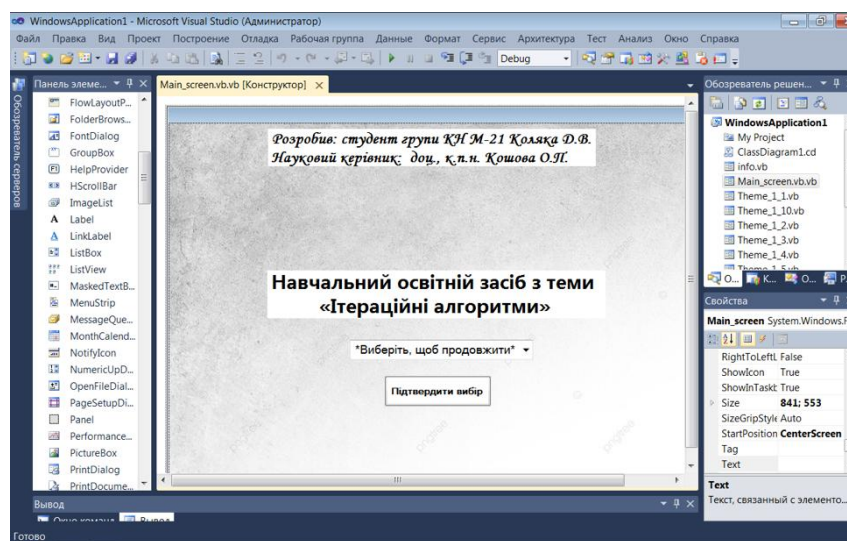


Рисунок 4.2 – головне вікно навчального освітнього засобу.

Після головного екрану, розпочинається процес програмної реалізації першого завдання з теми освітнього засобу (рис. 4.3)

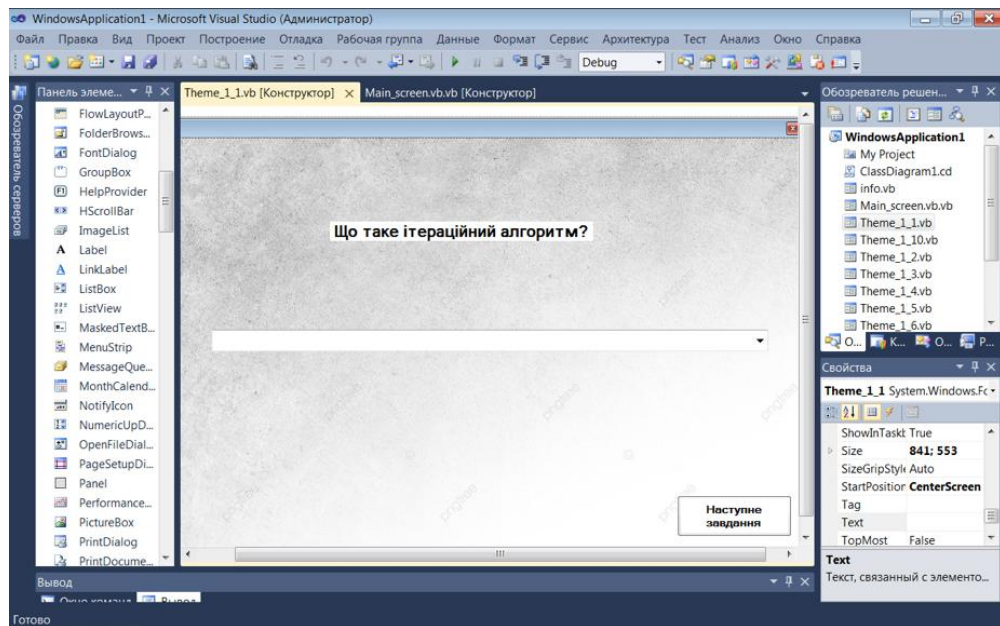


Рисунок 4.3 – перше завдання освітнього засобу.

Після першого завдання, проходить процес реалізації другого питання з теми освітнього засобу (рис. 4.4)

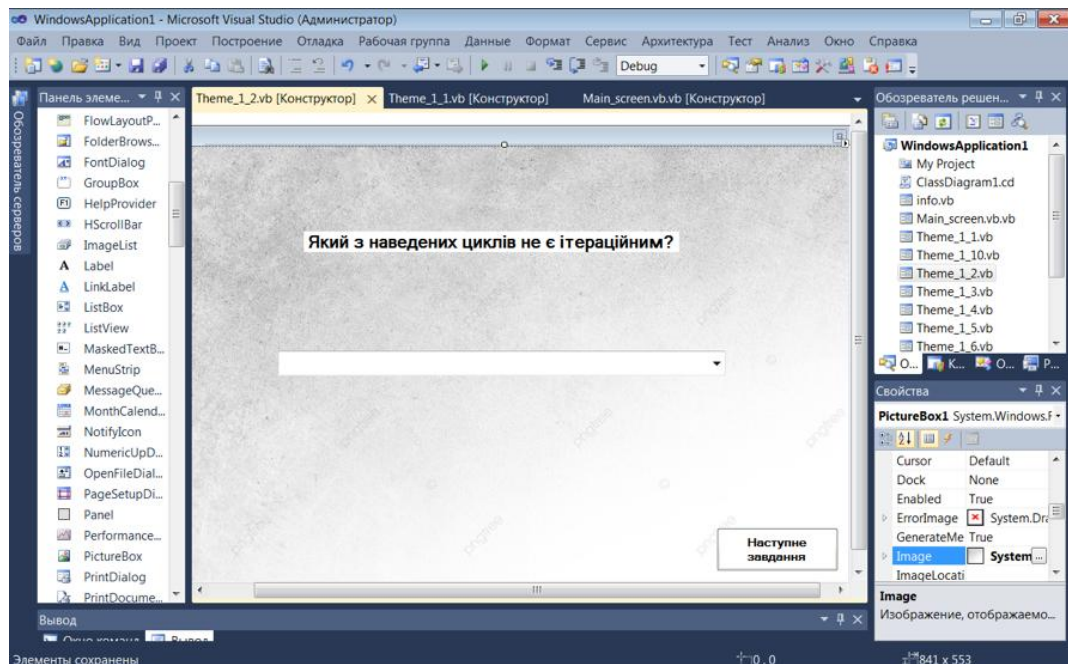


Рисунок 4.4 – друге завдання освітнього засобу.

Для кращого інформування користувача, додатково створено інформаційне вікно. (рис. 4.5)

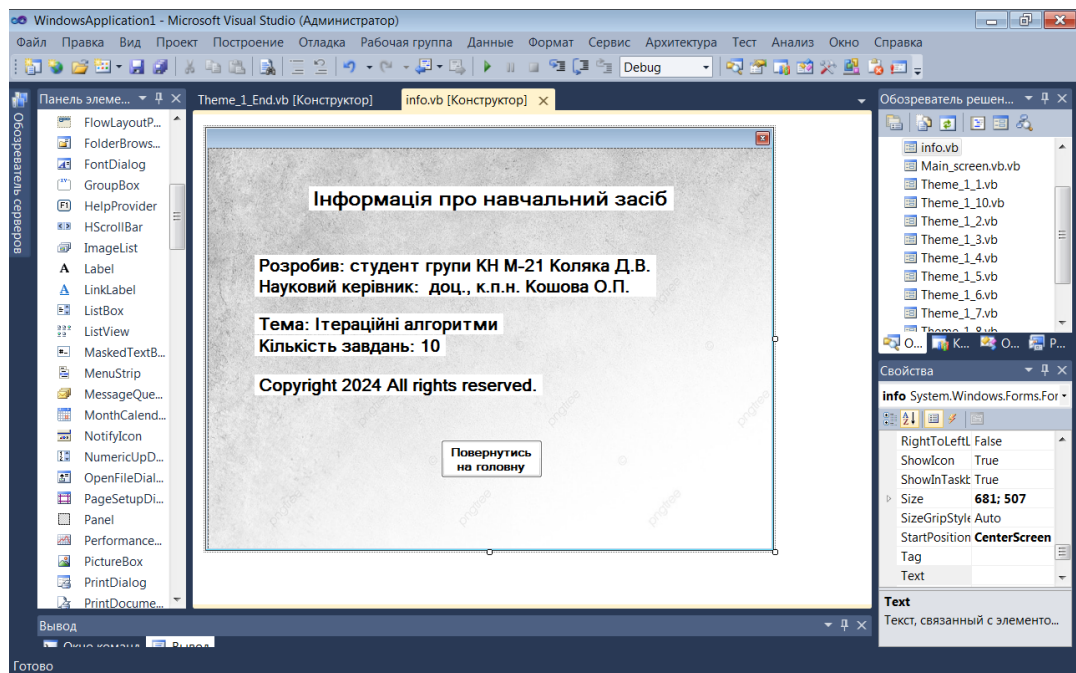


Рисунок 4.5 – інформаційне вікно.

В кінці освітнього засобу, користувача зустрічає вікно з результатами. (рис. 4.6)

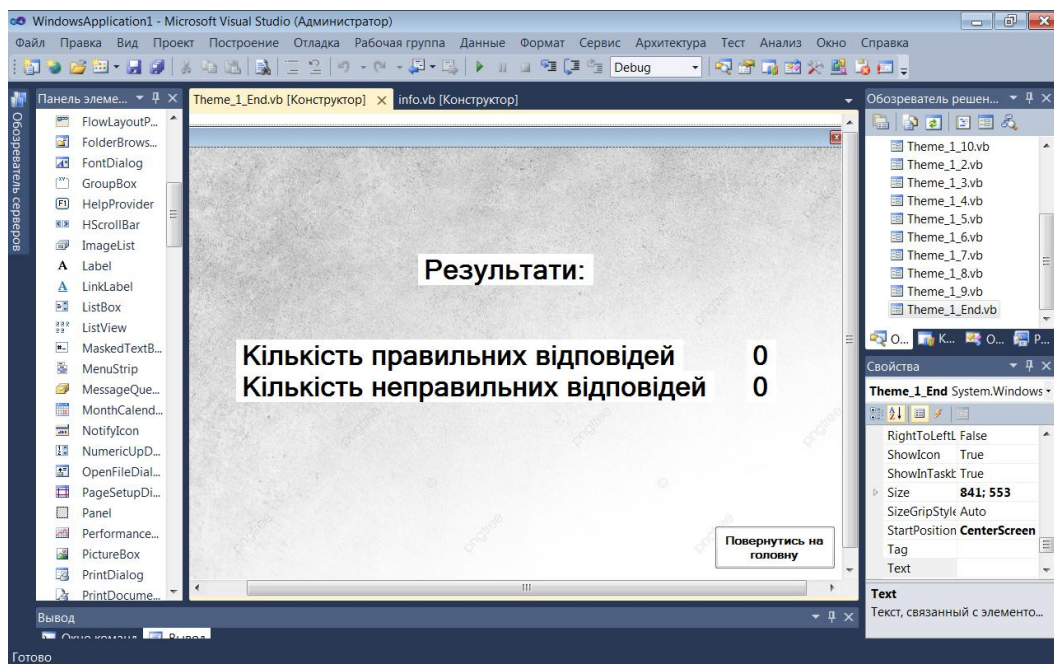


Рисунок 4.6 – вікно з результатами.

За логіку роботи освітнього засобу, відповідає запрограмований на мові програмування VB.NET код. (рис. 4.7)

```

2 Partial Class Theme_1_1
23 <System.Diagnostics.DebuggerStepThrough()> _
24 Private Sub InitializeComponent()
25 Dim resources As System.ComponentModel.ComponentResourceManager = New System.ComponentModel.ComponentResourceManager(GetType(Theme_1_1))
26 Me.Label1 = New System.Windows.Forms.Label()
27 Me.Button1 = New System.Windows.Forms.Button()
28 Me.ComboBox1 = New System.Windows.Forms.ComboBox()
29 Me.PictureBox1 = New System.Windows.Forms.PictureBox()
30 CType(Me.PictureBox1, System.ComponentModel.ISupportInitialize).BeginInit()
31 Me.SuspendLayout()
32 '
33 'Label1
34 '
35 Me.Label1.AutoSize = True
36 Me.Label1.Font = New System.Drawing.Font("Microsoft Sans Serif", 12.0!, System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, CType(284, Byte))
37 Me.Label1.ForeColor = System.Drawing.SystemColors.ActiveCaptionText
38 Me.Label1.Location = New System.Drawing.Point(238, 188)
39 Me.Label1.Name = "Label1"
40 Me.Label1.Size = New System.Drawing.Size(338, 25)
41 Me.Label1.TabIndex = 3
42 Me.Label1.Text = "Чи таке ітераційний алгоритм?"
43 '
44 'Button1
45 '
46 Me.Button1.BackColor = System.Drawing.Color.White
47 Me.Button1.Cursor = System.Windows.Forms.Cursors.Hand
48 Me.Button1.Font = New System.Drawing.Font("Microsoft Sans Serif", 7.8!, System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, CType(284, Byte))
49 Me.Button1.Location = New System.Drawing.Point(676, 460)
50 Me.Button1.Name = "Button1"
51 Me.Button1.Size = New System.Drawing.Size(147, 53)
52 Me.Button1.TabIndex = 4
53 Me.Button1.Text = "Наступне завдання"
54 Me.Button1.UseVisualStyleBackColor = False
55 '
56 'ComboBox1
57 '
58 Me.ComboBox1.Font = New System.Drawing.Font("Microsoft Sans Serif", 10.2!, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(284, Byte))
59 Me.ComboBox1.FormattingEnabled = True
60 Me.ComboBox1.Items.AddRange(New Object() { "а) Алгоритм, що використовує рекурсію для вирішення завдання", "б) Алгоритм, що використовує цикли для повторення послідовності дій",
61 "в) Алгоритм, що використовує масиви для зберігання даних", "г) Алгоритм, що використовує графіки для візуалізації даних" })
62 Me.ComboBox1.Location = New System.Drawing.Point(78, 247)
63 Me.ComboBox1.Name = "ComboBox1"
64 Me.ComboBox1.Size = New System.Drawing.Size(715, 28)
65 Me.ComboBox1.TabIndex = 9
66 '
67 'PictureBox1
68 '

```

Рисунок 4.7 – код додатку.

Розглянемо детально що відбувається в програмному коді застосунку:

Обробник події Theme_1_1_Load:

Коли форма Theme_1_1 завантажується (подія Load), відбувається кілька дій.

Спочатку приховується форма Main_screen за допомогою Main_screen.Hide().

Це означає, що основний екран або вікно програми більше не буде видно під час роботи з цією формою. Потім встановлюється розташування елементів управління (мітки та комбобоксу): Label1 розташовується по центру екрану горизонтально (відстань від лівого краю форми до мітки обчислюється як різниця між шириною форми та шириною мітки, поділена на два). Вертикально мітка розташовується на висоті 100 пікселів.

ComboBox1 також розташовується по центру екрану горизонтально, а вертикально — на висоті 240 пікселів.

Обробник події Button1_Click_1:

Це обробник події для натискання кнопки Button1. Коли користувач натискає цю кнопку, перевіряється вибір у комбобоксі (ComboBox1.Text). Якщо вибрана відповідь "b) Алгоритм, що використовує цикли для повторення послідовності дій", то виводиться повідомлення "Відповідь правильна!". Далі відбувається перевірка значення на етикетці Label4 на іншій формі Theme_1_End. Якщо його значення менше за 255, воно збільшується на 1. Незалежно від того, чи була відповідь правильною чи ні, відкривається наступна форма Theme_1_2 за допомогою Theme_1_2.Show(). Якщо відповідь неправильна, виводиться повідомлення з підказкою, що правильний варіант відповіді — "b) Алгоритм, що використовує цикли для повторення послідовності дій". Також збільшується значення в Label5 на формі Theme_1_End (якщо його значення менше за 255), і форма Theme_1_2 все одно відкривається.

Обробник події PictureBox1_Click:

Цей обробник не має реалізації, оскільки він порожній. Тобто, коли користувач натискає на картинку в елементі PictureBox1, нічого не відбувається.

Загалом, програмний код реалізує просту форму для проходження завдань, де необхідно вибрати правильну відповідь із комбобоксу, після чого користувач отримує підказку або підтвердження правильності відповіді. Також, відбувається оновлення певних значень на інших формах та відкриття нових форм для продовження процесу навчання.

4.3. Опис роботи освітнього засобу

Після запуску освітнього засобу, користувачу відкривається головне вікно програмного забезпечення (рис. 4.8)

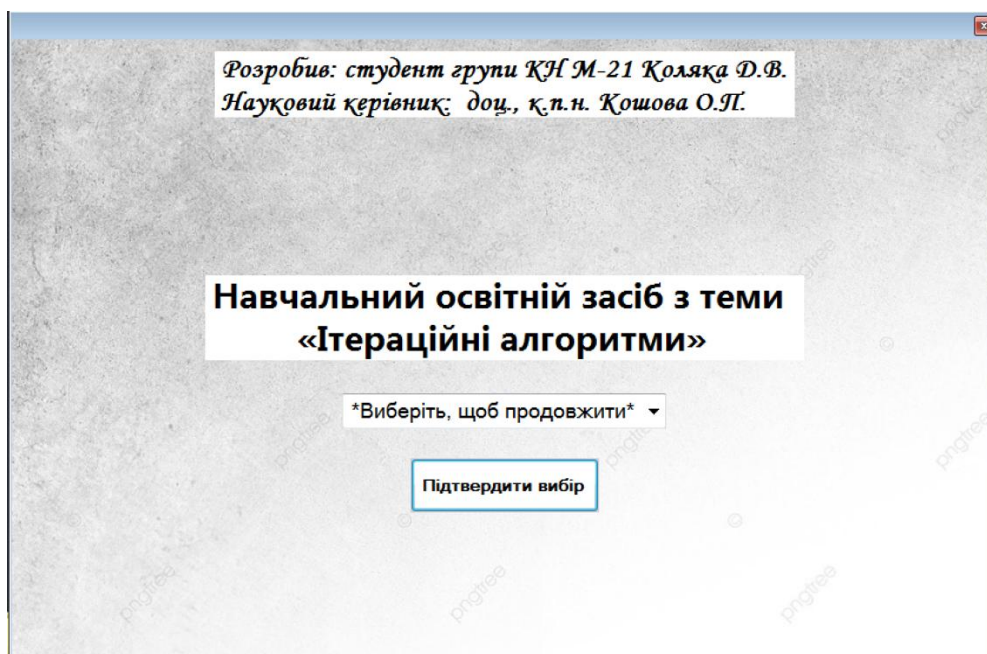


Рисунок 4.8 – головне вікно.

Потім, користувач може обрати один з трьох варіантів опції засобу з випадального списку. (рис. 4.9)

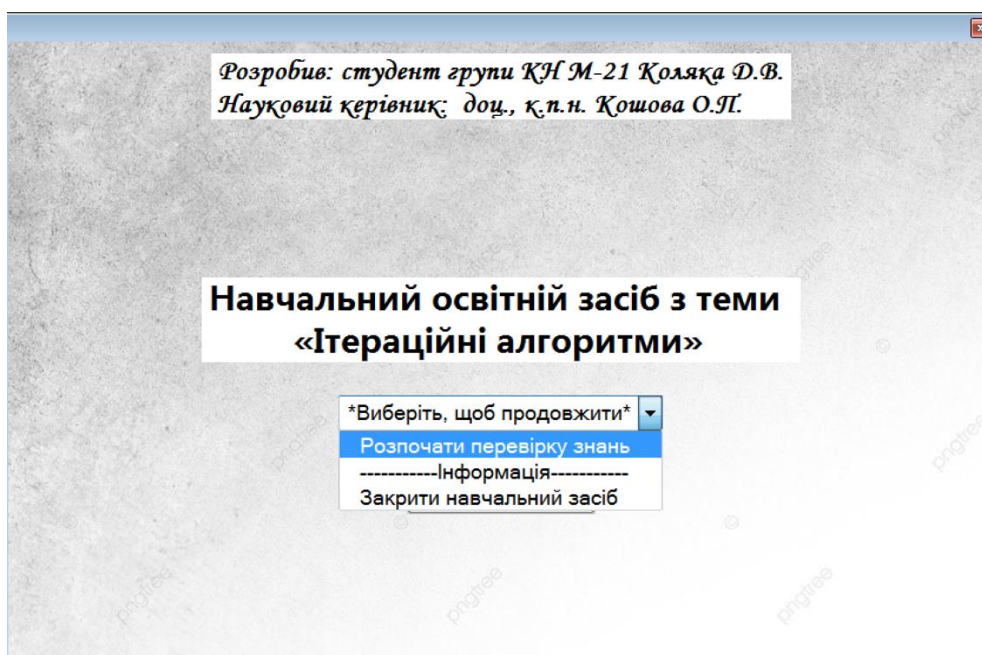


Рисунок 4.9 – випадючий список.

Якщо користувач вибрав кнопку «Розпочати перевірку знань», користувачу відкривається перше питання, він може відповісти вибравши з випадючого списку будь який варіант відповіді (рис. 5.0)

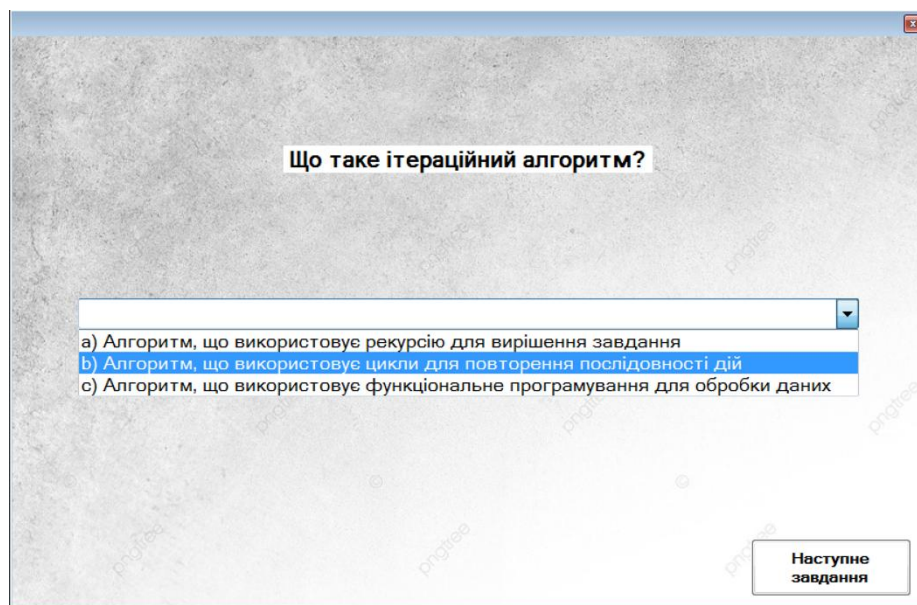


Рисунок 5.0 – перше завдання.

Після вибору відповіді, користувач натискає кнопку «Наступне завдання», якщо відповідь вірна, висвітиться наступне вікно (рис. 5.1)

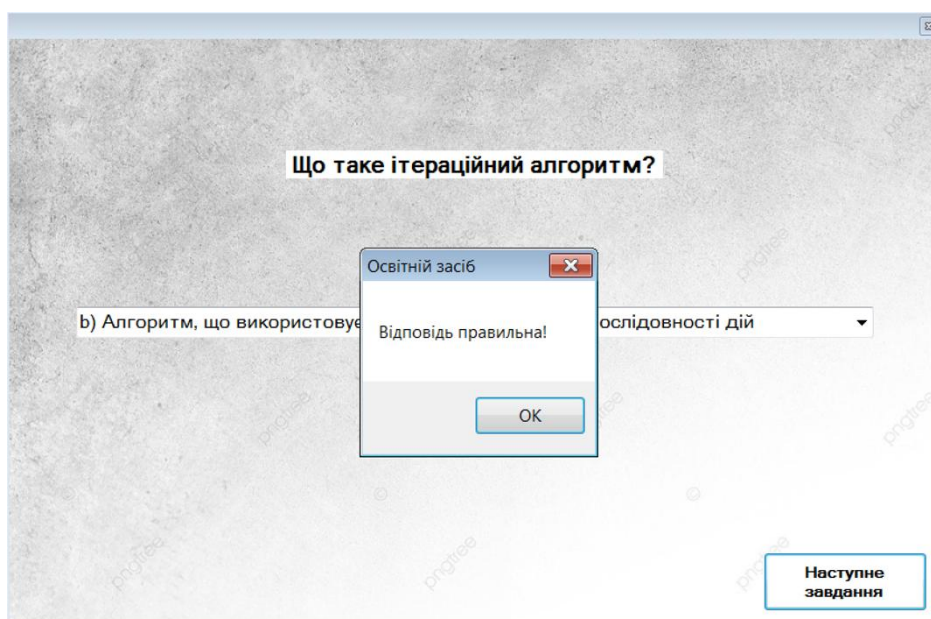


Рисунок 5.1 – вікно успішного вибору.

Якщо допущено помилку, висвітиться наступне вікно (рис. 5.2)

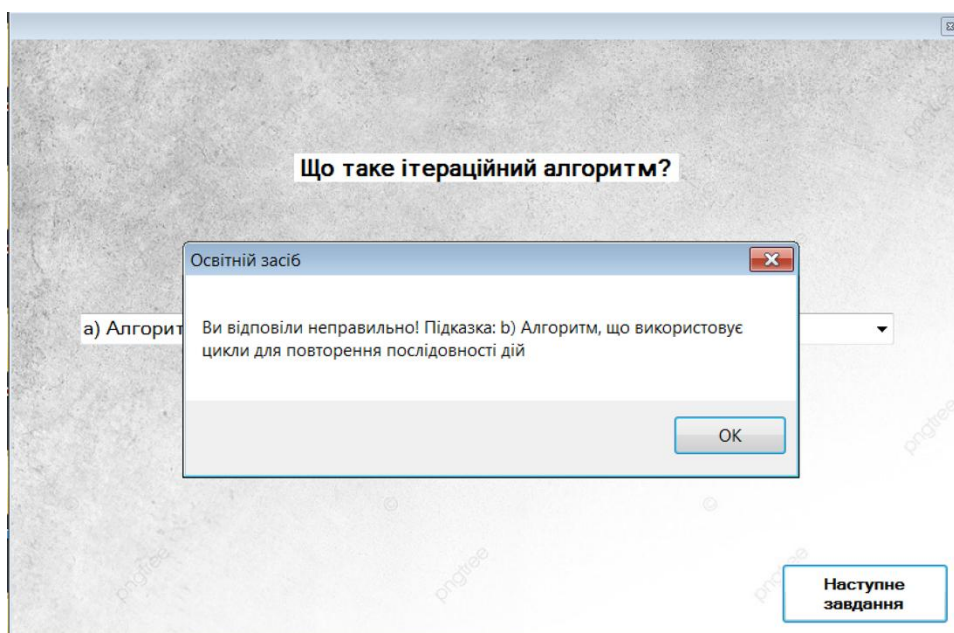


Рисунок 5.2 – вікно неправильної відповіді.

В кінці перевірки знань, користувачу відкриється вікно з результатами.
(рис. 5.3)

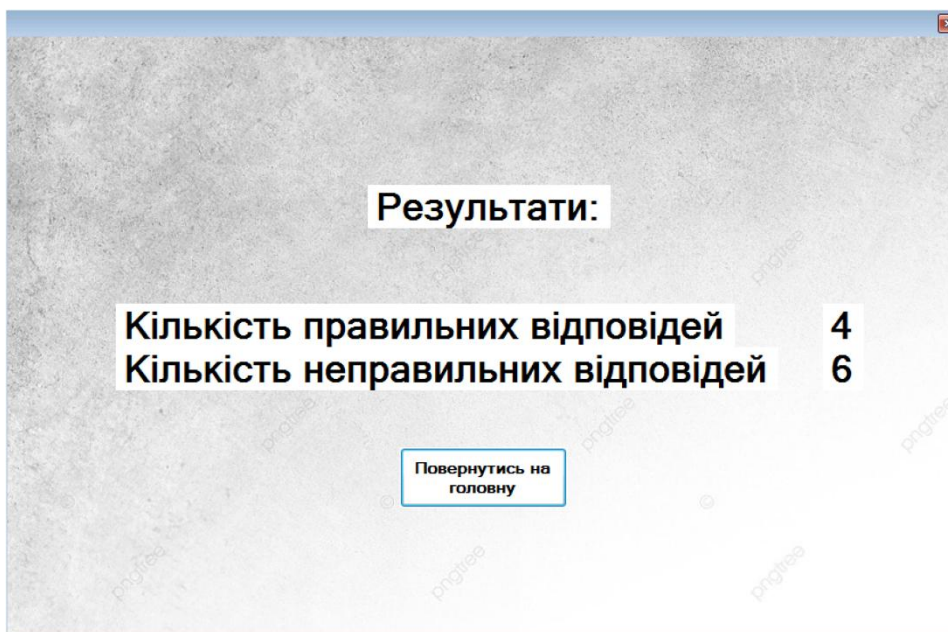


Рисунок 5.3 – вікно результатів.

Якщо користувач вибере з випадаючого списку інформацію, то йому розкриється вікно з інформацією про освітній засіб. (рис. 5.4)

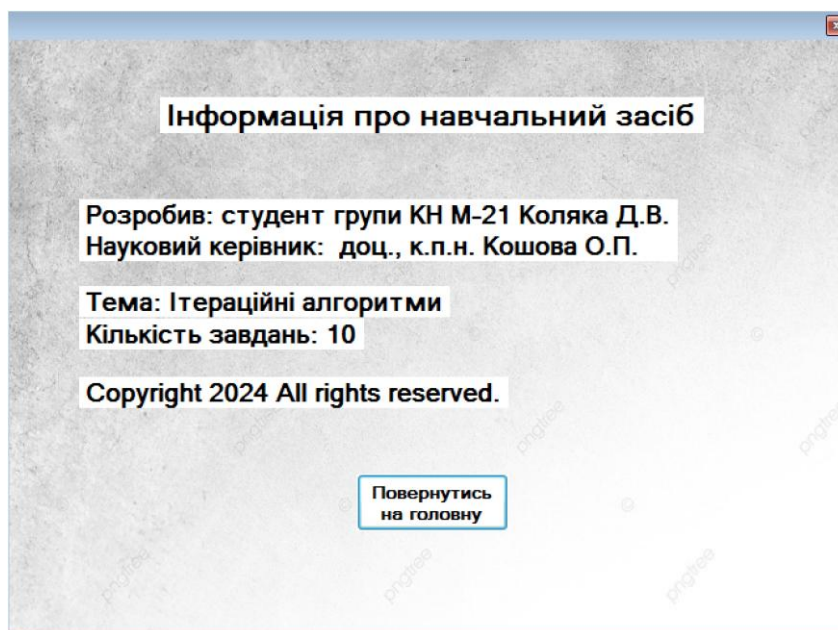


Рисунок 5.4 – вікно інформації про засіб.

ВИСНОВКИ

Отже, розробка навчального програмного забезпечення з ітераційними алгоритмами є дуже важливим завданням для підвищення якості освіти та забезпечення національної безпеки в умовах сучасного світу. Розробка програмного забезпечення на тему ітераційних алгоритмів є перспективною та корисною в сучасному освітньому процесі. Розроблене програмне забезпечення покращує якість освіти та формування висококваліфікованих спеціалістів. Зокрема, такі алгоритми розвивають навички користувачів у сферах де потрібні розширені обчислення та аналіз даних.

В цілому, розвиток таких програмних технологій стає ще важливішим у сучасному світі, де інформаційна безпека відіграє ключову роль у забезпеченні стабільності та захищеності. На фоні зростаючих загроз кібербезпеці, знання ітераційних алгоритмів дозволяє більш ефективно захищати інформацію та протидіяти кібератакам.

Кваліфікаційну роботу успішно завершено, створено програмне забезпечення навчального освітнього засобу з теми «Ітераційні алгоритми».

СПИСОК ВИКОРИСТАНОЇ 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. Важливість використання ПЗ у сучасному бізнес-світі [Електронний ресурс]. – Режим доступу: <https://armedsoft.com/ua/blog/vazhlyvist-vykorystannya-pz-u-suchasnomu-biznes-sviti#:~:text=%D0%9F%D1%80%D0%BE%D1%84%D0%B5%D1%81%D1%96%D0%B9%D0%BD%D0%B0%20%D1%80%D0%BE%D0%B7%D1%80%D0%BE%D0%B1%D0%BA%D0%B0%20%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BD%D0%BE%D0%B3%D0%BE%20%D0%B7%D0%B0%D0%B1%D0%B5%D0%B7%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%BD%D1%8F%20%D0%B4%D0%BE%D0%B7%D0%B2%D0%BE%D0%BB%D0%B8%D1%82%D1%8C,%D0%9F%D0%9A%2C%20%D0%B7%D0%B0%D0%B1%D0%B5%D0%B7%D0%BF%D0%B5%D1%87%D1%83%D1%8E%D1%82%D1%8C%20%D1%80%D0%B5%D0%B0%D0%BB%D1%96%D0%B7%D0%B0%D1%86%D1%96%D1%8E%20%D1%80%D1%96%D0%B7%D0%BD%D0%B8%D1%85%20%D0%B7%D0%B0%D0%B4%D0%B0%D1%87>.
3. Основи програмування [Електронний ресурс]. – Режим доступу: <https://studfile.net/preview/9325359/>
4. Microsoft Visual Studio [Електронний ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/Microsoft_Visual_Studio
5. What is VB.NET and use cases of VB.NET? [Електронний ресурс]. – Режим доступу: <https://www.devopsschool.com/blog/what-is-vb-net-and-use-cases-of-vb-net/>

6. Ітераційні алгоритми [Електронний ресурс]. – Режим доступу: <http://lib.kart.edu.ua/bitstream/123456789/2195/1/Miroshnyk.pdf>
7. ІТЕРАЦІЙНІ АЛГОРИТМИ АПРОКСИМАЦІЇ ФУНКЦІЙ [Електронний ресурс]. – Режим доступу: <http://mcm-tech.kpnu.edu.ua/article/view/23996>
8. Ітераційні алгоритми розв'язання плоскої задачі [Електронний ресурс]. – Режим доступу: <https://elar.khmnu.edu.ua/items/4eba098c-0d76-4f30-bd41-bf1bbec14a44>
9. Елементи теорії об'єктно-орієнтованого програмування [Електронний ресурс]. – Режим доступу: <https://naurok.com.ua/test/testi-z-temi-elementi-teori-ob-ektno-orientovanogo-programuvannya-7788.html>
10. Об'єктно-орієнтоване програмування (Advanced Encryption Standard instructions) [Електронний ресурс]. – Режим доступу: <https://vseosvita.ua/test/elementy-teorii-objektno-oriientovanoho-prohramuvannia-stvorennia-klasiv-i-objektiv-3907011.html>
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. Опис редактору Visual Basic [Електронний ресурс]. – Режим доступу: <https://studfile.net/preview/7785340/>
13. 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.
14. Appleman D. E72 Win32 API and Visual Basic. For professionals (+SD) / D. Appleman. - Peter, 2002. - 1120 p.
15. Halvorson M. Microsoft Visual Basic 2005. Series "Step by step" / M. Halvorson. - ECOM Publishers. - 640 p.
16. Диференціально-різницеві ітераційні алгоритми [Електронний ресурс]. – Режим доступу: <http://www.fmmit.lviv.ua/index.php/fmmit/article/view/31>
17. Ітераційні цикли [Електронний ресурс]. – Режим доступу: <https://studfile.net/preview/5470346/page:5/>

18. Ітераційні методи [Електронний ресурс]. – Режим доступу: [https://ukrayinska.libretexts.org/%D0%9C%D0%B0%D1%82%D0%B5%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0/%D0%9D%D0%B0%D1%83%D0%BA%D0%BE%D0%B2%D1%96_%D0%BE%D0%B1%D1%87%D0%B8%D1%81%D0%BB%D0%B5%D0%BD%D0%BD%D1%8F_%D1%82%D0%B0_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%8E%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F/%D0%9D%D0%B0%D1%83%D0%BA%D0%BE%D0%B2%D0%B0_%D0%BE%D0%B1%D1%87%D0%B8%D1%81%D0%BB%D1%8E%D0%B2%D0%B0%D0%BB%D1%8C%D0%BD%D0%B0_%D1%82%D0%B5%D1%85%D0%BD%D1%96%D0%BA%D0%B0_\(Chasnov\)/I%3A_%D0%A7%D0%B8%D1%81%D0%B5%D0%BB%D1%8C%D0%BD%D1%96_%D0%BC%D0%B5%D1%82%D0%BE%D0%B4%D0%B8/7%3A_%D0%86%D1%82%D0%B5%D1%80%D0%B0%D1%86%D1%96%D0%B9%D0%BD%D1%96_%D0%BC%D0%B5%D1%82%D0%BE%D0%B4%D0%B8](https://ukrayinska.libretexts.org/%D0%9C%D0%B0%D1%82%D0%B5%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0/%D0%9D%D0%B0%D1%83%D0%BA%D0%BE%D0%B2%D1%96_%D0%BE%D0%B1%D1%87%D0%B8%D1%81%D0%BB%D0%B5%D0%BD%D0%BD%D1%8F_%D1%82%D0%B0_%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%8E%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F/%D0%9D%D0%B0%D1%83%D0%BA%D0%BE%D0%B2%D0%B0_%D0%BE%D0%B1%D1%87%D0%B8%D1%81%D0%BB%D1%8E%D0%B2%D0%B0%D0%BB%D1%8C%D0%BD%D0%B0_%D1%82%D0%B5%D1%85%D0%BD%D1%96%D0%BA%D0%B0_(Chasnov)/I%3A_%D0%A7%D0%B8%D1%81%D0%B5%D0%BB%D1%8C%D0%BD%D1%96_%D0%BC%D0%B5%D1%82%D0%BE%D0%B4%D0%B8/7%3A_%D0%86%D1%82%D0%B5%D1%80%D0%B0%D1%86%D1%96%D0%B9%D0%BD%D1%96_%D0%BC%D0%B5%D1%82%D0%BE%D0%B4%D0%B8)
19. Ітераційні алгоритми знаходження чисел [Електронний ресурс]. – Режим доступу: https://www.irbis-nbuv.gov.ua/cgi-bin/irbis_nbuv/cgiirbis_64.exe?I21DBN=LINK&P21DBN=UJRN&Z21ID=&S21REF=10&S21CNR=20&S21STN=1&S21FMT=ASP_meta&C21COM=S&2_S21P03=FILA=&2_S21STR=itvo_2015_23_7
20. Алгоритми обчислювальних процесів [Електронний ресурс]. – Режим доступу: https://ami.lnu.edu.ua/wp-content/uploads/2013/11/1.2.07_Algorithms_of-computational_processes_pm.pdf

ДОДАТОК А. ВИХІДНІ КОДИ

```

<Global.Microsoft.VisualBasic.CompilerServices.DesignerGenerated()> _
Partial Class Main_screen
    Inherits System.Windows.Forms.Form

    <System.Diagnostics.DebuggerNonUserCode()> _
    Protected Overrides Sub Dispose(ByVal disposing As Boolean)
        Try
            If disposing AndAlso components IsNot Nothing Then
                components.Dispose()
            End If
        Finally
            MyBase.Dispose(disposing)
        End Try
    End Sub

    <System.Diagnostics.DebuggerStepThrough()> _
    Private Sub InitializeComponent()
        Dim resources As
System.ComponentModel.ComponentResourceManager = New
System.ComponentModel.ComponentResourceManager(GetType(Main_scre
n))

        Me.Label2 = New System.Windows.Forms.Label()
        Me.Label1 = New System.Windows.Forms.Label()
        Me.Button1 = New System.Windows.Forms.Button()
        Me.ComboBox1 = New System.Windows.Forms.ComboBox()
        Me.PictureBox1 = New System.Windows.Forms.PictureBox()
    End Sub

```

```

        CType(Me.PictureBox1,
System.ComponentModel.ISupportInitialize).BeginInit()
        Me.SuspendLayout()
        '
        'Label2
        '
        Me.Label2.AutoSize = True
        Me.Label2.BackColor = System.Drawing.Color.White
        Me.Label2.Font = New System.Drawing.Font("Monotype Corsiva",
13.8!, CType((System.Drawing.FontStyle.Bold Or
System.Drawing.FontStyle.Italic), System.Drawing.FontStyle),
System.Drawing.GraphicsUnit.Point, CType(204, Byte))
        Me.Label2.ForeColor =
System.Drawing.SystemColors.ActiveCaptionText
        Me.Label2.Location = New System.Drawing.Point(149, 9)
        Me.Label2.Name = "Label2"
        Me.Label2.Size = New System.Drawing.Size(492, 56)
        Me.Label2.TabIndex = 8
        Me.Label2.Text = "Розробив: студент групи КН М-11 Коляка Д.В." &
Global.Microsoft.VisualBasic.ChrW(13) &
Global.Microsoft.VisualBasic.ChrW(10) & "Науковий керівник: доц., к.п.н.
Кош" & _
        "ова О.П."
        '
        'Label1
        '
        Me.Label1.AutoSize = True
        Me.Label1.BackColor = System.Drawing.Color.White

```

```

Me.Label1.Font = New System.Drawing.Font("Microsoft YaHei UI",
16.2!, System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point,
CType(204, Byte))
Me.Label1.ForeColor =
System.Drawing.SystemColors.ActiveCaptionText
Me.Label1.Location = New System.Drawing.Point(148, 221)
Me.Label1.Name = "Label1"
Me.Label1.Size = New System.Drawing.Size(508, 72)
Me.Label1.TabIndex = 7
Me.Label1.Text = "Навчальний освітній засіб з теми " &
Global.Microsoft.VisualBasic.ChrW(13) &
Global.Microsoft.VisualBasic.ChrW(10) & "      «Ітераційні алгоритми»"
'
'Button1
'
Me.Button1.BackColor = System.Drawing.Color.White
Me.Button1.Cursor = System.Windows.Forms.Cursors.Hand
Me.Button1.Font = New System.Drawing.Font("Microsoft Sans Serif",
7.8!, System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point,
CType(204, Byte))
Me.Button1.Location = New System.Drawing.Point(324, 379)
Me.Button1.Name = "Button1"
Me.Button1.Size = New System.Drawing.Size(160, 46)
Me.Button1.TabIndex = 6
Me.Button1.Text = "Підтвердити вибір"
Me.Button1.UseVisualStyleBackColor = False
'
'ComboBox1
'
```

```

Me.ComboBox1.Font = New System.Drawing.Font("Microsoft Sans
Serif", 10.8!, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, CType(204, Byte))
Me.ComboBox1.FormattingEnabled = True
Me.ComboBox1.Items.AddRange(New Object() {" Розпочати
перевірку знань", " -----Інформація-----", " Закрити навчальний
засіб"})
Me.ComboBox1.Location = New System.Drawing.Point(272, 325)
Me.ComboBox1.Name = "ComboBox1"
Me.ComboBox1.Size = New System.Drawing.Size(275, 30)
Me.ComboBox1.TabIndex = 12
Me.ComboBox1.Text = " *Виберіть, щоб продовжити*"
'
'PictureBox1
'
Me.PictureBox1.Image =
CType(resources.GetObject("PictureBox1.Image"), System.Drawing.Image)
Me.PictureBox1.Location = New System.Drawing.Point(0, 0)
Me.PictureBox1.Name = "PictureBox1"
Me.PictureBox1.Size = New System.Drawing.Size(841, 553)
Me.PictureBox1.SizeMode =
System.Windows.Forms.PictureBoxSizeMode.StretchImage
Me.PictureBox1.TabIndex = 13
Me.PictureBox1.TabStop = False
'
'Main_screen
'
Me.AutoScaleDimensions = New System.Drawing.SizeF(8.0!, 16.0!)
Me.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font
Me.BackColor = System.Drawing.Color.LightGray

```

```

        Me.ClientSize = New System.Drawing.Size(835, 525)
        Me.Controls.Add(Me.ComboBox1)
        Me.Controls.Add(Me.Label2)
        Me.Controls.Add(Me.Label1)
        Me.Controls.Add(Me.Button1)
        Me.Controls.Add(Me.PictureBox1)
        Me.FormBorderStyle =
System.Windows.Forms.FormBorderStyle.FixedToolWindow
        Me.Name = "Main_screen"
        Me.StartPosition =
System.Windows.Forms.FormStartPosition.CenterScreen
        CType(Me.PictureBox1,
System.ComponentModel.ISupportInitialize).EndInit()
        Me.ResumeLayout(False)
        Me.PerformLayout()

    End Sub

    Friend WithEvents Label2 As System.Windows.Forms.Label
    Friend WithEvents Label1 As System.Windows.Forms.Label
    Friend WithEvents Button1 As System.Windows.Forms.Button
    Friend WithEvents ComboBox1 As System.Windows.Forms.ComboBox
    Friend WithEvents PictureBox1 As System.Windows.Forms.PictureBox

End Class

<Global.Microsoft.VisualBasic.CompilerServices.DesignerGenerated()> _
Partial Class Theme_1_1
    Inherits System.Windows.Forms.Form

    <System.Diagnostics.DebuggerNonUserCode()> _

```

```

Protected Overrides Sub Dispose(ByVal disposing As Boolean)
    Try
        If disposing AndAlso components IsNot Nothing Then
            components.Dispose()
        End If
    Finally
        MyBase.Dispose(disposing)
    End Try
End Sub

<System.Diagnostics.DebuggerStepThrough()> _
Private Sub InitializeComponent()
    Dim resources As
System.ComponentModel.ComponentResourceManager = New
System.ComponentModel.ComponentResourceManager(GetType(Theme_1_1
))

    Me.Label1 = New System.Windows.Forms.Label()
    Me.Button1 = New System.Windows.Forms.Button()
    Me.ComboBox1 = New System.Windows.Forms.ComboBox()
    Me.PictureBox1 = New System.Windows.Forms.PictureBox()
    CType(Me.PictureBox1,
System.ComponentModel.ISupportInitialize).BeginInit()
    Me.SuspendLayout()
    '
    'Label1
    '

    Me.Label1.AutoSize = True
    Me.Label1.Font = New System.Drawing.Font("Microsoft Sans Serif",
12.0!, System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point,
CType(204, Byte))

```

```

Me.Label1.ForeColor =
System.Drawing.SystemColors.ActiveCaptionText
Me.Label1.Location = New System.Drawing.Point(230, 108)
Me.Label1.Name = "Label1"
Me.Label1.Size = New System.Drawing.Size(338, 25)
Me.Label1.TabIndex = 3
Me.Label1.Text = "Що таке ітераційний алгоритм?"
'
'Button1
'

Me.Button1.BackColor = System.Drawing.Color.White
Me.Button1.Cursor = System.Windows.Forms.Cursors.Hand
Me.Button1.Font = New System.Drawing.Font("Microsoft Sans Serif",
7.8!, System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point,
CType(204, Byte))
Me.Button1.Location = New System.Drawing.Point(676, 460)
Me.Button1.Name = "Button1"
Me.Button1.Size = New System.Drawing.Size(147, 53)
Me.Button1.TabIndex = 4
Me.Button1.Text = "Наступне завдання"
Me.Button1.UseVisualStyleBackColor = False
'
'ComboBox1
'

Me.ComboBox1.Font = New System.Drawing.Font("Microsoft Sans
Serif", 10.2!, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, CType(204, Byte))
Me.ComboBox1.FormattingEnabled = True
Me.ComboBox1.Items.AddRange(New Object() {"a) Алгоритм, що
використовує рекурсію для вирішення завдання", "b) Алгоритм, що

```

використовує цикли для повторення послідовності дій", "с) Алгоритм, що використовує функціональне програмування для обробки даних"})

```
Me.ComboBox1.Location = New System.Drawing.Point(78, 247)
```

```
Me.ComboBox1.Name = "ComboBox1"
```

```
Me.ComboBox1.Size = New System.Drawing.Size(715, 28)
```

```
Me.ComboBox1.TabIndex = 9
```

```
,
```

```
'PictureBox1
```

```
,
```

```
Me.PictureBox1.Image =
```

```
CType(resources.GetObject("PictureBox1.Image"), System.Drawing.Image)
```

```
Me.PictureBox1.Location = New System.Drawing.Point(0, 0)
```

```
Me.PictureBox1.Name = "PictureBox1"
```

```
Me.PictureBox1.Size = New System.Drawing.Size(841, 553)
```

```
Me.PictureBox1.SizeMode =
```

```
System.Windows.Forms.PictureBoxSizeMode.StretchImage
```

```
Me.PictureBox1.TabIndex = 14
```

```
Me.PictureBox1.TabStop = False
```

```
,
```

```
'Theme_1_1
```

```
,
```

```
Me.AutoScaleDimensions = New System.Drawing.SizeF(8.0!, 16.0!)
```

```
Me.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font
```

```
Me.BackColor = System.Drawing.Color.White
```

```
Me.ClientSize = New System.Drawing.Size(835, 525)
```

```
Me.Controls.Add(Me.ComboBox1)
```

```
Me.Controls.Add(Me.Button1)
```

```
Me.Controls.Add(Me.Label1)
```

```
Me.Controls.Add(Me.PictureBox1)
```



```
        Me.FormBorderStyle =  
System.Windows.Forms.FormBorderStyle.FixedToolWindow  
        Me.Name = "Theme_1_1"  
        Me.StartPosition =  
System.Windows.Forms.FormStartPosition.CenterScreen  
        CType(Me.PictureBox1,  
System.ComponentModel.ISupportInitialize).EndInit()  
        Me.ResumeLayout(False)  
        Me.PerformLayout()  
  
    End Sub  
  
    Friend WithEvents Label1 As System.Windows.Forms.Label  
    Friend WithEvents Button1 As System.Windows.Forms.Button  
    Friend WithEvents ComboBox1 As System.Windows.Forms.ComboBox  
    Friend WithEvents PictureBox1 As System.Windows.Forms.PictureBox  
  
End Class
```