

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 СИСТЕМИ ТЕСТУВАННЯ ЗНАНЬ З КУРСУ "ОСНОВИ
НАУКОВИХ ДОСЛІДЖЕНЬ В ІНФОРМАТИЦІ"**

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

Виконавець роботи Задворкін Максим Олександрович
_____ «___» _____ 202_ р.
(підпис)

Науковий керівник к. ф.-м. н., доцент, Черненко Оксана Олексіївна
_____ «___» _____ 202_ р.
(підпис)

Рецензент _____

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ, ТЕРМІНІВ	3
ВСТУП.....	4
1. ПОСТАНОВКА ЗАДАЧІ РОЗРОБКИ	6
2. ІНФОРМАЦІЙНИЙ ОГЛЯД	7
2.1. Огляд існуючих комп’ютерних систем тестування знань	7
3. ТЕОРЕТИЧНА ЧАСТИНА	10
3.1. Методологія розробки.....	10
3.2. Алгоритм роботи системи тестування знань курсу «Основи наукових досліджень в інформатиці».....	18
3.3 Блок-схема системи тестування знань.....	22
4. ПРАКТИЧНА ЧАСТИНА.....	23
4.1. Обґрунтування вибору програмних засобів.....	23
4.2. Опис програмної реалізації.....	24
4.3. Опис роботи системи тестування знань	27
ВИСНОВКИ	41
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	42
ДОДАТОК А. КОД ПРОГРАМИ	44

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ, ТЕРМІНІВ

Умовні позначення, символи, скорочення, терміни	Пояснення умовних позначень, скорочень, символів
System	System містить базові класи та типи даних, необхідні для роботи з основними системними функціями, такими як введення-виведення, обробка винятків, робота з потоками і т.д.
Dim	Це ключове слово, яке використовується в різних мовах програмування для визначення змінних (variables). Наприклад, в мові програмування Visual Basic (VB), ключове слово "Dim" використовується для створення змінних.
Sub	У мові програмування, такій як Visual Basic (VB) чи VBA (Visual Basic for Applications), "Sub" може також використовуватися для визначення підпрограм, що виконуються незалежно від повернення значень.
ReadLine	Це метод, який часто використовується в програмуванні для зчитування рядка тексту з консолі або іншого введення.

ВСТУП

Дистанційне навчання стало необхідною реальністю в сучасному світі, перетворюючи традиційні підходи до освіти. За останні роки цей метод отримав особливий розквіт, допомагаючи освітнім установам та студентам подолати численні виклики, які виникають у зв'язку зі швидкими змінами у суспільстві та технологічним прогресом. Вся робота має бути добре оформлена згідно методичних рекомендацій щодо оформлення кваліфікаційної роботи для студентів спеціальності 122 Комп'ютерні науки. [1]

Важливість дистанційного навчання полягає не лише в забезпеченні доступу до знань для широкого кола людей, але й у його здатності адаптуватися до різних потреб та обставин. Цей метод дозволяє студентам вивчати матеріал в комфортний для них спосіб, розвиваючи в них навички самостійності, саморегуляції та використання сучасних інформаційних технологій.

Розробка та впровадження навчальних систем тестування знань має великий потенціал стати важливим інструментом для розвитку сучасної освіти, враховуючи потреби та можливості сучасного суспільства. [2]

Мета кваліфікаційної роботи – розробка програмного забезпечення системи тестування знань, дистанційного навчального курсу «Основи наукових досліджень в інформатиці».

Об'єкт розробки – алгоритм, блок-схема, розробка програмного забезпечення системи тестування знань.

Предмет розробки – система тестування знань дистанційного навчального курсу «Основи наукових досліджень в інформатиці» та розробка її програмного забезпечення.

Кваліфікаційна робота складається з чотирьох розділів. У першому розділі описано постановку задачі для реалізації системи тестування знань. У другому розділі розглянуто існуючі комп'ютерні системи тестування знань. У третьому розділі описано методологію розробки, детально описаний алгоритм роботи та блок-схема роботи системи тестування знань. У четвертому розділі описано процес програмної реалізації системи тестування знань та опис роботи системи. Обсяг пояснювальної записки: 44 стор., в т.ч. основна частина – 39 стор., джерела – 20 назв.

1. ПОСТАНОВКА ЗАДАЧІ РОЗРОБКИ

Основним завданням кваліфікаційної роботи є розробка програмного забезпечення для реалізації системи тестування знань, що належить дистанційному курсу "Основи наукових досліджень в інформатиці".

Завдання в рамках роботи включають наступне:

1. Активне вивчення матеріалу за курсом "Основи наукових досліджень в інформатиці".
2. Розробка алгоритму для ефективної працездатності системи тестування знань.
3. Створення блок-схеми для чіткого відображення алгоритму системи тестування знань.
4. Реалізація програмних компонентів, необхідних для функціонування системи тестування знань.
5. Детальний опис результатів, отриманих під час розробки програмних елементів системи тестування знань.

У процесі створення програмного забезпечення для системи тестування знань необхідно враховувати такі важливі програмні елементи:

1. Стартове вікно системи тестування знань, яке містить інформацію про тематику тестування, кнопку для початку тесту та дані про розробника та наукового керівника.
2. Інші вікна системи, що включають тестові завдання та питання з теми тестування, на які користувач повинен надавати вірні або невірні відповіді.
3. Вікно, в якому описана детальна інформація про систему тестування знань.
4. Меню, за допомогою якого в користувача є можливість вибрати тему.
5. Кінцеві фінальні вікна після кожної теми, які містять інформацію про результати проходження поточної теми.

2. ІНФОРМАЦІЙНИЙ ОГЛЯД

2.1. Огляд існуючих комп'ютерних систем тестування знань

Комп'ютерні системи тестування знань дозволяють перевіряти знання учнів в обсязі всього предметного курсу, до того ж їх результати можна легко порівнювати між собою. Однак деякі сторони знань і умінь предметні тести діагностувати все ж таки не можуть, наприклад, уміння виражати свої думки в усній формі, а також глибину знань з певної теми, творчі уміння.

Розглянемо детальніше комп'ютерну систему тестування знань, що може використовуватися для контролю успішності знань студентів з математики в університеті.

Майстер-Тест – це безкоштовний Інтернет сервіс, що дозволяє створювати тести. Тести можуть бути створені як онлайн так і завантажені і проходитися тест без підключення до Інтернету. [3]

Для роботи на даному сайті потрібно зареєструватися та авторизуватися в системі. Головна сторінка користувача має вигляд як на рисунку (рис. 2.1). Зліва вверху є дві вкладки - «Викладач», «Студент». Від вибраної вкладки залежить список пунктів меню нижче. Нижче зліва також є блок в якому вказавши е-мейл до процесу тестування можна запросити учня чи вчителя для створення тесту.

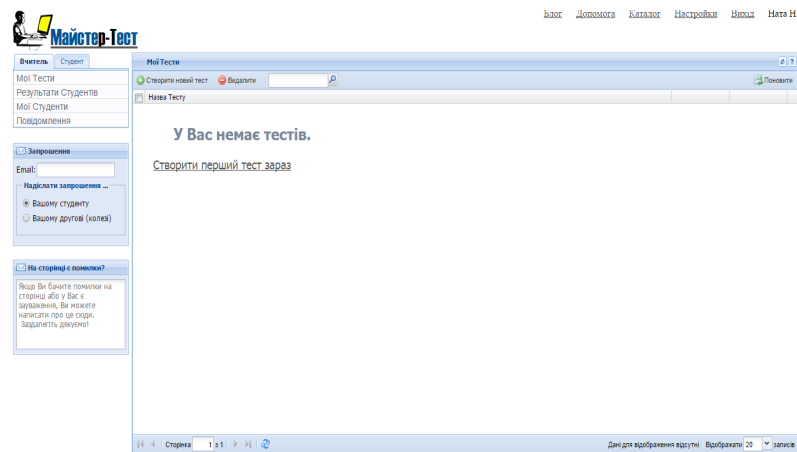


Рисунок 2.1 - Головна сторінка Майстер-тест.

Для створення тесту потрібно вибрати відповідну кнопку «Створити новий тест», розміщено у верхній частині вікна. Після цього буде відкрито нове діалогове вікно для створення тесту (рис. 2.2). У ньому потрібно вказати назву запитання, його вагу, тип правильної відповіді та варіанти відповіді. Так наприклад для закритого тесту існує два варіанти: 1 правильна відповідь або декілька (рис. 2.2), для відкритої – слово, номер, співвідношення (рис. 2.3). Якщо потрібно додати графічні матеріали то потрібно вибрати опцію додатково (рис. 2.3).

The screenshot shows a dialog box titled 'Тест без назви' (Test without name). It has a sidebar on the left with 'Тест' (Test) selected. The main area contains the following fields and options:

- Заголовок:** Яким називають відрізок, що сполучає дві точки кола та проходить через його центр?
- Тип запитання:** Одна Відповідь (dropdown menu).
- Відповідь:** Three radio buttons for 'діаметр' (diameter), 'радіус' (radius), and 'нонус' (arc). Each has a 'Видалити' (Delete) button to its right. There is also a 'Додати Відповідь' (Add Answer) button.
- Вага Питання:** 1 (dropdown menu).
- Buttons:** 'Готово' (Done) at the bottom right.

Рисунок 2.2 - Створення запитання закритого типу

The screenshot shows a dialog box titled 'тест'. It has a sidebar on the left with 'Тест' (Test) selected. The main area contains the following fields and options:

- Заголовок:** Скільки градусів становить розгорнутий кут?
- Додатковий текст:** A text area containing a diagram of a straight line with points A, O, and B. A semi-circle is drawn above the line with its center at O. The text 'Розгорнутий кут дорівнює 180°' and '∠AOB=180°' is written above the diagram.
- Тип запитання:** Слово (dropdown menu).
- Відповідь:** Three checkboxes for 'Враховувати регістр' (Case sensitive), 'Використання більше однієї відповіді' (Use more than one answer), and 'Ігнорувати порядок' (Ignore order). There is also a text input field containing '180'.
- Вага Питання:** 6 (dropdown menu).
- Buttons:** 'Готово' (Done) at the bottom right.

Рисунок 2.3 - Створення запитання відкритого типу із додаванням графічних матеріалів

Після того як було заповнено всі дані, на цьому вікні потрібно натиснути кнопку «Готово» та «Зберегти», тоді тест потрапить у базу запитань. Для додавання нового запитання в тест потрібно натиснути кнопку «Додати питання» у верхньому правому кутку вікна. У верхній панелі меню є інші елементи налаштування проходження тестів.

Сторінка проходження тесту має вигляд зображений (рис. 2.4.) Після вибору відповіді потрібно натиснути «Продовжити», а для того щоб закінчити тест – «Завершити». Результати тесту відображаються на сторінці (рис. 2.5.)

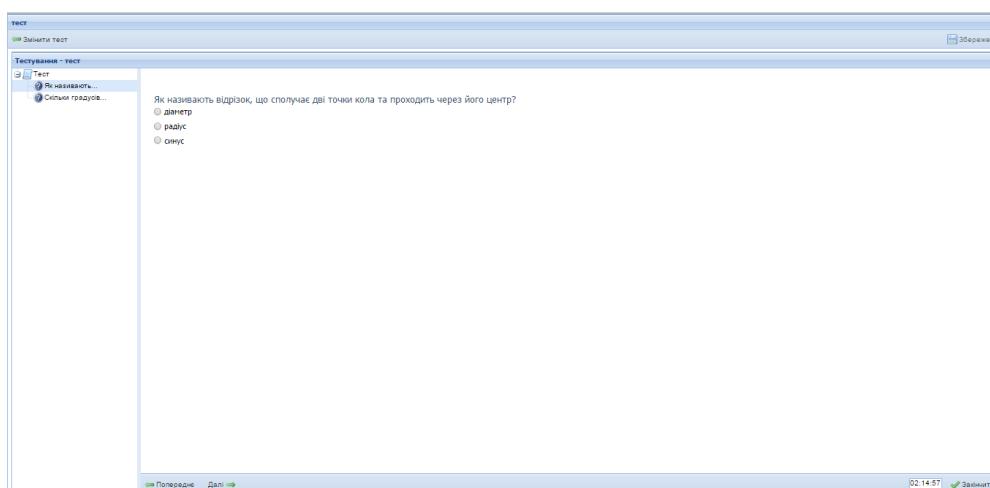


Рисунок 2.4 - Проходження тесту

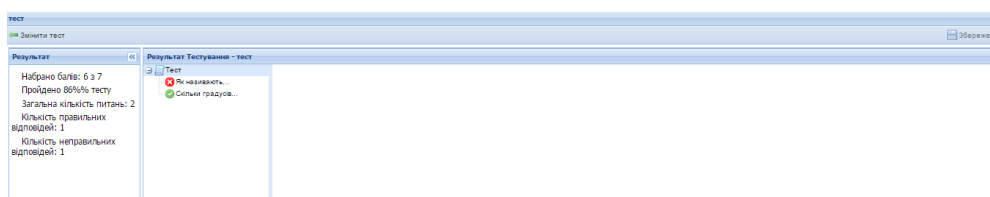


Рисунок 2.5 - Відображення результатів тестів

Також на даному сайті вже є існуюча база тестів, яку можна використовувати у вільному доступі. Для цього потрібно перейти на сторінку «Каталог» та обрати відповідну категорію та відповідний тест.

Загалом сайт має інтуїтивно зрозумілий інтерфейс користувача, для створення тестів не потрібно особливих навичок користування комп'ютером та веб-технологіями.

3. ТЕОРЕТИЧНА ЧАСТИНА

3.1. Методологія розробки

Розробка якісного продукту починається з визначення його життєвого циклу. Це чіткий план дій, що дозволяє зрозуміти, що має вийти в розробників, як досягти результату та які методи для цього використати. Методологія розробки програмного забезпечення — це перевірені способи та практики, що дозволяють створити діджитал-продукт правильно та якісно.

Найбільш помітними є такі методології розроблення ПЗ:

Rational Unified Process (RUP), що пропонує ітеративну модель розроблення, що включає чотири фази: початок, дослідження, побудову та впровадження. Проходження через чотири основні фази називається циклом розроблення, кожен цикл завершується генерацією працездатної версії системи. У ході супроводу продукт продовжує розвиватися і знову проходить ті самі фази. Розроблення ПЗ на базі RUP базується на створенні та супроводі моделей на базі UML.

Microsoft Solution Framework (MSF) – методологія, що розроблялася для підвищення керованості процесів розроблення окремого підприємства (Microsoft), а потім стала застосовуватися розробниками, які використовують продукти Microsoft. Методологія, подібна до RUP, також включає чотири фази: аналіз, проектування, розроблення, стабілізацію, є ітераційною, припускає використання об'єктно-орієнтованого моделювання. MSF порівняно з RUP здебільшого орієнтована на розроблення бізнес додатків.

eXtreme Programming (XP) – методологія, що приділяє основну увагу ефективній комунікації між замовником і виконавцем упродовж усього проекту з розроблення ІС для відслідковування зміни вимог. В основу підходу покладена командна робота та постійне тестування прототипів.

Гнучке розроблення ПЗ (Agile) – клас методологій розроблення програмного забезпечення на основі ітеративної моделі ЖЦ, в якій вимоги та рішення еволюціонують через співпрацю між самоорганізовуваними командами.

Об'єктно-орієнтована методологія розробки (ООМР) є підходом до програмування, що базується на концепції об'єктів та класів. Вона дозволяє розглядати програмне забезпечення як колекцію об'єктів, які взаємодіють між собою для виконання певних завдань. ООМР спрямована на полегшення розробки, управління та розуміння складних систем, шляхом використання об'єктно-орієнтованих концепцій.

Основні принципи об'єктно-орієнтованої методології розробки включають:

1. Інкапсуляція: Ідея полягає в тому, що об'єкти мають внутрішній стан та поведінку, які вони приховують від зовнішнього світу. Всі зміни стану об'єкта повинні відбуватися за допомогою відповідних методів (або функцій), які контролюють доступ до даних.

2. Наслідування: Об'єкти можуть успадковувати властивості та методи від інших об'єктів, що спрощує повторне використання коду та забезпечує ієрархічну структуру класів.

3. Поліморфізм: Це означає, що об'єкти можуть виконувати однакові дії, але різними способами. Наприклад, одна і та ж операція може мати різні реалізації в залежності від типу об'єкта.

4. Абстракція: Вона дозволяє моделювати реальний світ у вигляді об'єктів з властивостями та методами, що відображають важливі аспекти предметної області, і приховувати від програміста непотрібні деталі реалізації.

ООМР включає в себе також використання класів, об'єктів, спадкування, поліморфізму, інтерфейсів та інших конструкцій, щоб створити структуровані, легко змінювані та розширювані програмні системи. Кожен клас може мати свої властивості (дані) та методи (функції), які визначають його поведінку.

Взаємодія між об'єктами відбувається через виклик методів одного об'єкта з іншого.

ООМР не лише дозволяє розробникам легко створювати програми, але й сприяє підвищенню їхньої якості та обслуговуваності, зменшуючи залежність між різними частинами коду та сприяючи повторному використанню коду.

Процедурне програмування полягає в розбитті великої програми на сукупність процедур підпрограм. Одна з підпрограм повина бути головною і з неї починається виконання програми.

Процедурний підхід потребував структуризації майбутньої програми, розділення її на окремі процедури. При розробці окремої процедури про інші процедури потрібно було знати тільки їх призначення і спосіб виклику.

Структурне програмування - це програмування, при якому програма в цілому і окремі процедури розглядалися як послідовності канонічних структур: лінійних ділянок, циклів і розгалужень.

Дані - це представлення фактів і ідей у формалізованому вигляді, придатному для передачі і переробці в якомусь процесі, а інформація - це сенс, який додається даним при їх уявленні.

Обробка даних - це виконання систематичної послідовності дій з даними. Дані зберігаються на т.з. носіях даних. Сукупність носіїв даних,

використовуваних при якій-небудь обробці даних, називатимемо інформаційним середовищем. Набір даних, що містяться в який-небудь момент в інформаційному середовищі, називатимемо станом цього інформаційного середовища.

Процес можна визначити як послідовність станів деякого інформаційного середовища, що змінюють один одного.

Описати процес - означає визначити послідовність станів заданого інформаційного середовища. Для того, щоб процес породжувався автоматично на ПКі, необхідно, щоб це й опис був формалізованим. Такий опис називається програмою.

Програми розробляються з розрахунку на те, щоб ними могли користуватися люди, що не беруть участь в їх розробці (їх називають користувачами). Для освоєння програми користувачем крім її тексту потрібна певна додаткова документація. Програма або логічно зв'язана сукупність програм на носіях даних, забезпечена програмною документацією, називається програмним засобом (ПЗ). Програма дозволяє здійснювати деяку автоматичну обробку даних на комп'ютері. Програмна документація дозволяє зрозуміти, які функції виконує та або інша програма ПЗ, як підготувати початкові дані і запустити необхідну програму в процес її виконання, а також: що означають отримувані результати (або який ефект виконання цієї програми). Крім того, програмна документація допомагає розібратися в самій програмі, що необхідне, наприклад, при її модифікації.

Продуктом технології програмування є ПЗ, що містить програми, які виконують необхідні функції. Тут під "програмою" часто розуміють правильну програму, тобто програму, що не містить помилок.

Надійність ПЗ - це його здатність безвідмовно виконувати певні функції за заданих умов протягом заданого періоду часу з достатньо

великою вірогідністю. При цьому під відмовою в ПЗ розуміють прояв в ній помилки.

Технологія програмування це сукупність виробничих процесів, що приводить до створення необхідного ПЗ, а також опис цієї сукупності процесів.

Не слід також плутати технологію програмування з методологією програмування. В обох випадках вивчаються методи, але в технології програмування методи розглядаються "зверху" - з погляду організації технологічних процесів, а в методології програмування методи розглядаються "знизу" - з погляду основ їх побудови

- Неформальний характер вимог до ПЗ (постановки завдання), але формалізований основний об'єкт розробки - програми ПЗ.
- Творчий характер розробки ПЗ (на кожному кроці доводиться робити який-небудь вибір, ухвалювати яке-небудь рішення).
- Програмний продукт є деякою сукупністю текстів (тобто статичних об'єктів), сенс же (семантика) цих текстів виражається процесами обробки даних і діями користувачів, що запускають ці процеси (тобто є динамічним).
- Продукт розробки має і іншу специфічну особливість: ПЗ при своєму використанні (експлуатації) не витрачається і не витрачає використовуваних ресурсів.

Під життєвим циклом ПЗ розуміють весь період його розробки і експлуатації (використання), починаючи від моменту виникнення задуму ПЗ і кінчаючи припиненням всіх видів його використання. Життєвий цикл включає всі процеси створення і використання ПЗ (software process).

Розрізняють наступні стадії життєвого циклу ПЗ: розробку ПЗ, виробництво програмних виробів (ПВ) і експлуатацію ПЗ.

Стадія розробки (development) ПЗ складається з етапу його зовнішнього опису, етапу конструювання ПЗ, етапу кодування (програмування у вузькому сенсі) ПЗ і етапу атестації ПЗ

Зовнішній опис (Requirements document) ПЗ є описом його поведінки з погляду зовнішнього по відношенню до нього спостерігача з фіксацією вимог щодо його якості.

Конструювання (design) ПЗ охоплює процеси: розробку архітектури ПЗ, розробку структур програм ПЗ і їх детальну специфікацію.

Кодування (coding) створення текстів програм на мовах програмування, їх відладку з тестуванням ПЗ.

На етапі атестації ПЗ проводиться оцінка якості ПЗ, після успішного завершення якого розробка ПЗ вважається закінченою.

Програмний виріб (ПВ) - екземпляр або копія, знята з розробленого ПЗ. Стадія експлуатації ПЗ охоплює процеси зберігання, впровадження і супроводу ПЗ, а також транспортування і застосування (operation) ПЗ по своєму призначенню. Вона складається з двох фаз, що паралельно проходять: фази застосування ПЗ і фази супроводу ПЗ.

Застосування (operation) ПЗ - це використання ПЗ для вирішення практичних завдань на комп'ютері шляхом виконання її програм.

Супровід (maintenance) ПЗ - це процес збору інформації про його якість в експлуатації, усунення виявлених в нім помилок, його доопрацювання і модифікації, а також сповіщення користувачів про внесені до нього зміни.

Кожен ПЗ повинне виконувати задуманими функціями. Якість ПЗ - це сукупність його рис і характеристик, які впливають на його здатність задовольняти задані потреби користувачів

В даний час критеріями якості ПЗ прийнято вважати:

- функціональність
- надійність
- легкість застосування
- ефективність
- супровід
- мобільність.

Функціональність - це здатність ПЗ виконувати набір функцій, що задовольняють заданим або таким, що маються на увазі потребам користувачів. Набір вказаних функцій визначається в зовнішньому описі ПЗ.

Надійність обговорювалась раніше.

Легкість застосування - це характеристики ПЗ, які дозволяють мінімізувати зусилля користувача по підготовці початкових даних, застосуванню ПЗ і оцінці отриманих результатів, а також викликати позитивні емоції певного або такого, що мається на увазі користувача.

Ефективність - це відношення рівня послуг, ПЗ, що надаються, користувачеві за заданих умов, до об'єму використовуваних ресурсів.

Супровід - це характеристики ПЗ, які дозволяють мінімізувати зусилля по внесенню змін для усунення в нім помилок і по його модифікації відповідно до потреб користувачів, що змінюються.

Мобільність - це здатність ПЗ бути перенесеним з одного середовища (оточення) в іншу, зокрема, з однією ЕОМ на іншу.

Функціональність і надійність є обов'язковими критеріями якості ПЗ, причому забезпечення надійності буде червоною ниткою проходити по всіх етапах і процесах розробки ПЗ. Решта критеріїв використовується залежно від потреб користувачів відповідно до вимог до ПЗ - їх забезпечення обговорюватиметься у відповідних розділах курсу.

Загальні принципи забезпечення надійності ПЗ:

- попередження помилок;
- самовиявлення помилок;
- самовиправлення помилок;
- забезпечення стійкості до помилок.

Аналіз природи помилок при розробці ПЗ потребує розглянути наступні питання:

- боротьба з складністю
- забезпечення точності перекладу

- подолання бар'єру між користувачем і розробником
- забезпечення контролю ухвалюваних рішень.

Самовиявлення помилки в програмі означає, що програма містить засоби виявлення відмови в процесі її виконання. Самовиправлення помилки в програмі означає виправлення наслідків цієї відмови, для чого в програмі повинні бути відповідні засоби. Забезпечення стійкості програми до помилок означає, що в програмі містяться засоби, що дозволяють локалізувати область впливу відмови програми, або зменшити його неприємні наслідки, а іноді запобігти катастрофічним наслідкам відмови.

- забезпечення незалежності компонент системи;
- використання в системах ієрархічних структур.

Забезпечення незалежності компонент означає розбиття системи на такі частини, між якими повинні залишитися по можливості менше зв'язків. Одним з втілень цього методу є модульне програмування.

Використання ієрархічних структур дозволяє локалізувати зв'язки між компонентами, допускаючи їх лише між компонентами, що належать суміжним рівням ієрархії. Цей метод, означає розбиття великої системи на підсистеми.

Забезпечення точності перекладу направлене на досягнення однозначності інтерпретації документів різними розробниками, а також користувачами ПЗ. Весь процес перекладу можна розбити на наступні етапи:

- Зрозумійте завдання;
- Складіть план (включаючи цілі і методи рішення);
- Виконайте план (перевіряючи правильність кожного кроку);
- Проаналізуйте отримане рішення.

Для цього необхідно правильно зрозуміти, по-перше, чого хоче користувач, і, по-друге, його рівень підготовки і навколишнє його оточення.

Обов'язковим кроком в кожному процесі (етапі) розробки ПЗ повинна бути перевірка правильності ухвалених рішень. Це дозволить виявляти і виправляти помилки на найранішій стадії після її виникнення, що, по-перше, істотно знижує вартість її виправлення і, по-друге, підвищує вірогідність правильного її усунення.

3.2 . Алгоритм роботи системи тестування знань курсу «Основи наукових досліджень в інформатиці»

Загальний алгоритм роботи додатку:

- Крок 1: Запуск системи;
- Крок 2: Вивід стартового екрану;
- Крок 3: По бажанню перегляд інформації про систему;
- Крок 4: Запуск меню вибору теми, користувач обирає одну з 8-ми тем;
- Крок 5: Після вибору теми, користувачеві відкривається вибрана тема з блоком питань;
- Крок 6: Відображення питань та відповідь на них;
- Крок 7: Вивід вікна тема завершена та результати тестування;
- Крок 8: Завершення додатку або повторне тестування та повторний вибір теми.

Алгоритм роботи коду додатку:

- Крок 1. Користувач запускає систему тестування, йому виводиться головний екран "Main_screen" з трьома кнопками управління.
- Крок 2. Якщо користувач натисне кнопку "Про систему", виконається подія Button3_Click. Це реалізує відображення інформаційного екрану (info), який містить інформацію про додаток та розробника.
- Крок 3. Якщо користувач натисне на головному вікні кнопку "Закрити систему", виконається подія Button2_Click. Ця кнопка припиняє виконання системи тестування знань, оскільки викликається функція "End", яка припиняє роботу додатку.

Крок 4. Користувач натискає кнопку "Розпочати тестування", викликається подія `Button1_Click`. Подія реалізує відображення вікна меню "Menu_screen", який буде показувати 8 тем для користувача.

Крок 5. При відкритті вікна меню (Menu_screen), викликається подія "Menu_screen_Load".

Крок 6. Головний екран (Main_screen) закривається, щоб вікна не накладалися одне на інше.

Крок 7. Розташування та розміри кожної кнопки (від `Button1` до `Button8`) і заголовку (`Label1`) налаштовуються так, щоб вони були відцентровані по горизонталі та розташовані вертикально одна під одною з відступом 20 пікселів.

Крок 8. Для того щоб розпочати підготовку з теми 1, користувач натискає "Button1", після цього викликається подія "Button1_Click", це реалізує відображення теми 1 (`Theme_1_1`).

Крок 9. Для того щоб розпочати підготовку з теми 2, користувач натискає "Button2", після цього викликається подія "Button2_Click", це реалізує відображення теми 2 (`Theme_2_1`).

Крок 10. Для того щоб розпочати підготовку з теми 3, користувач натискає "Button3", після цього викликається подія "Button3_Click", це реалізує відображення теми 3 (`Theme_3_1`).

Крок 11. Для того щоб розпочати підготовку з теми 4, користувач натискає "Button4", після цього викликається подія "Button4_Click", це реалізує відображення теми 4 (`Theme_4_1`).

Крок 12. Для того щоб розпочати підготовку з теми 5, користувач натискає "Button5", після цього викликається подія "Button5_Click", це реалізує відображення теми 5 (`Theme_5_1`).

Крок 13. Для того щоб розпочати підготовку з теми 6, користувач натискає "Button6", після цього викликається подія "Button6_Click", це реалізує відображення теми 6 (`Theme_6_1`).

Крок 14. Для того щоб розпочати підготовку з теми 7, користувач натискає “Button7”, після цього викликається подія “Button7_Click”, це реалізує відображення теми 7 (Theme_7_1).

Крок 15. Для того щоб розпочати підготовку з теми 8, користувач натискає “Button8”, після цього викликається подія “Button8_Click”, це реалізує відображення теми 8 (Theme_8_1).

Крок 16. Якщо користувач обирає тему 1 (Theme_1_1), в цей момент викликається подія “Theme_1_1_Load” і запускається вікно з завданням.

Крок 17. Вікно меню (Menu_screen) приховується, щоб вікна не накладувались одне на інше.

Крок 18. Позиція та розміри елементів інтерфейсу, при завантаженні завдання, такі елементи як мітка (Label1), кнопка (Button1) та комбінований список (ComboBox1), налаштовуються так, щоб вони були відцентровані по горизонталі та розташовані відповідно до відступів від верхнього краю вікна.

Крок 19. При натисканні кнопки “Button1” викликається подія “Button1_Click_1”. Подія детально перевіряє вміст комбінованого списку “ComboBox1”. Якщо вибране користувачем значення дорівнює "а) механістичної;", виводиться повідомлення "Правильно!" та відбувається збільшення лічильника правильних відповідей на +1. Після цього користувачу виводиться наступне питання (Theme_1_2).

Крок 20. Якщо вибране користувачем значення не дорівнює "а) механістичної;", виводиться повідомлення "Неправильно! Підказка: механістичної" та відбувається збільшення лічильника неправильних відповідей теж на +1. Після цього також відбувається перехід на наступне завдання (Theme_1_2).

Крок 21. Коли користувач виконав всі завдання, йому відкривається останнє підсумкове вікно теми 1 (Theme_1_End), для цього викликається подія “Theme_1_End_Load”.

Крок 22. При переході на останнє підсумкове вікно, питання номер 30 (Theme_1_30) приховується, реалізовано для того, щоб вікна не накладалися одне на одне.

Крок 23. Позиція та розміри всіх елементів інтерфейсу підсумкового вікна, такі як мітка (Label1) та кнопка (Button1), налаштовані кодом так, щоб вони були відцентровані по горизонталі та розташовані відповідно до відступів від верхнього краю вікна.

Крок 24. Після завершення теми 1, користувач знаходиться на останньому вікні (Theme_1_End), на цьому вікні можна переглянути правильні та неправильні відповіді.

Крок 25. На останньому вікні, після ознайомлення з відповідями, користувач натискає кнопку “Button1”, після цього викликається подія “Button1_Click”. У цій події викликається функція “Application.Restart()”, яка перезапускає додаток і відкриває користувачу головний екран додатку, для подальшого проходження тем, або завершення роботи з системою тестування знань.

3.3 Блок-схема системи тестування знань

Блок-схема системи (рис. 3.1)

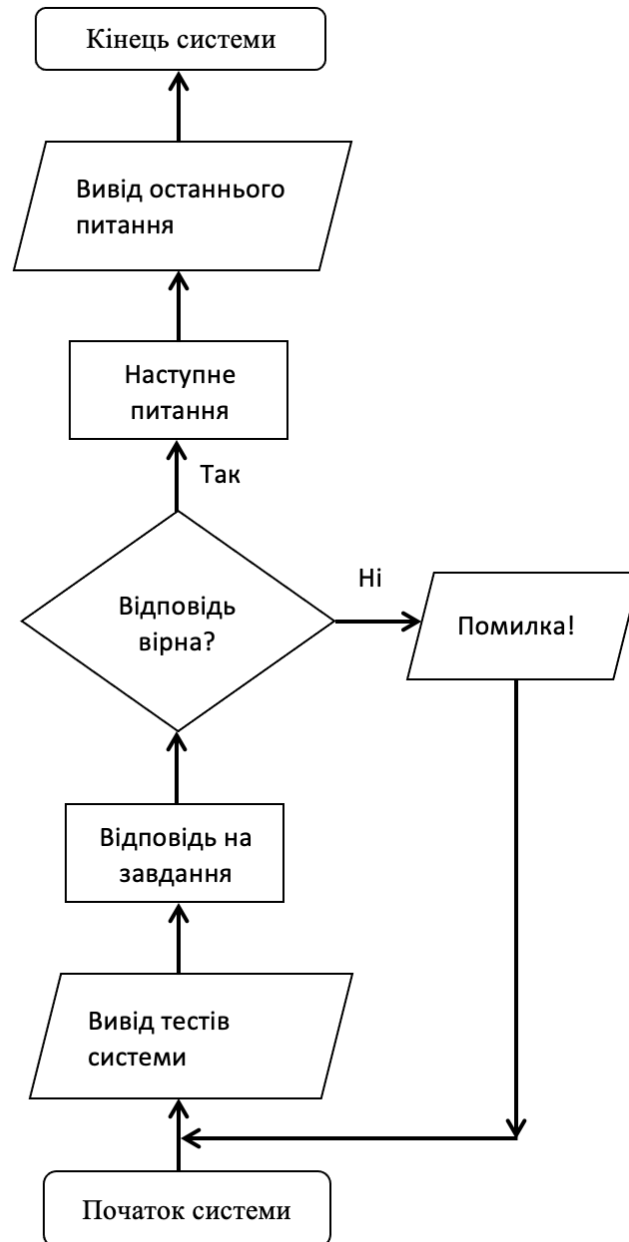


Рисунок 3.1 – Блок-схема системи.

4. ПРАКТИЧНА ЧАСТИНА

4.1. Обґрунтування вибору програмних засобів

Visual Basic є мовою програмування третього покоління, випущеною в 1987 році Microsoft. Це перший інструмент візуальної розробки від компанії, визнаний однією з найпотужніших мов програмування. У порівнянні з іншими мовами, такими як C або C++, VB вигідний своєю простотою вивчення і зрозумінням при наявності відданості та рішучості. Мова програмування Visual Basic дозволяє розробникам створювати програмний інтерфейс та код в зручному графічному середовищі. VB представляє собою комбінацію компонентів, що використовуються у формах із визначеними атрибутами та діями. Це сприяє швидкій розробці додатків на основі візуального підходу та полегшує доступ до баз даних через ADO, дозволяючи використовувати елементи керування ActiveX та різноманітні об'єкти.

Хоча VB спрямований на розробку програм, він також корисний для створення ігор для обмежених цілей. В порівнянні з іншими мовами, він може бути менш швидким, але його гнучкість і легкість використання роблять його ефективним для завдань, які можуть бути складними в інших мовах.

Однією з ключових переваг VB є можливість генерації виконуваних файлів Exe, що дозволяє розробникам створювати програми для використання як інтерфейси до баз даних. Наявність різноманітних ресурсів, таких як книги та онлайн матеріали, робить VB однією з найпопулярніших мов програмування та сприяє зручному розвитку навичок програмування на цій мові.

4.2. Опис програмної реалізації

Програмні компоненти системи тестування знань програмно реалізовані на основі мови програмування Visual Basic.(рис. 4.1)

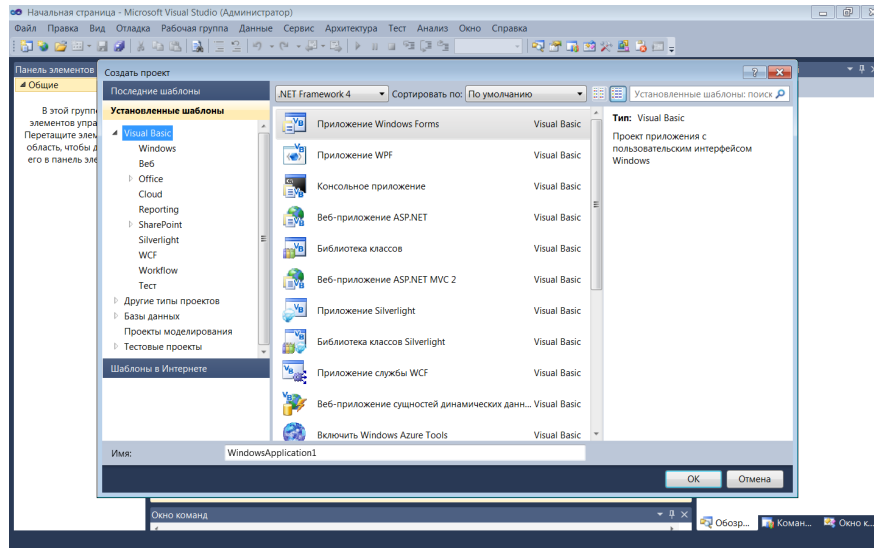


Рисунок 4.1 – Середовище розробки додатків Visual Studio

Після створення нового проекту в середовищі розробки, наступним етапом є створення початкової форми системи тестування знань, яка функціонує як головний екран для користувача. (рис. 4.2)

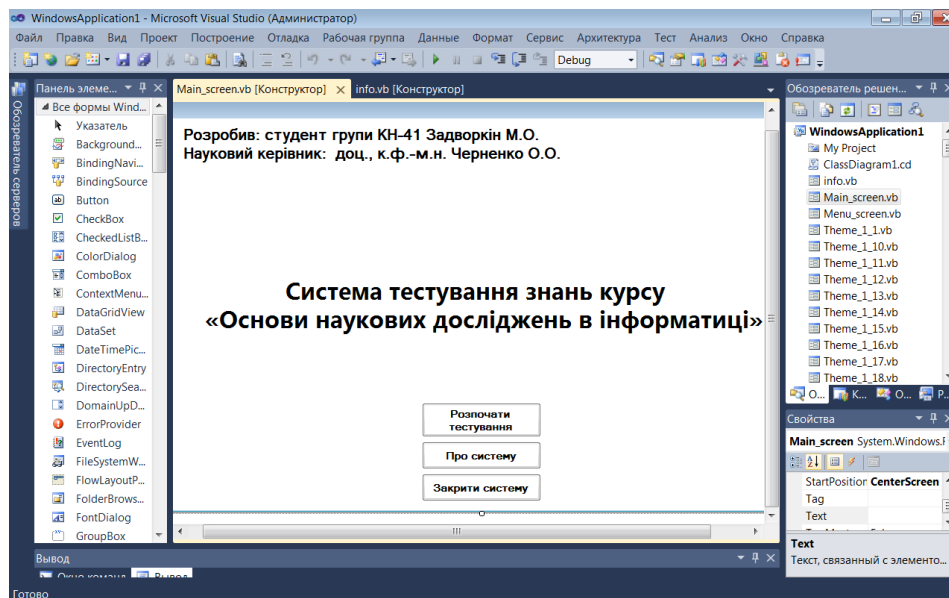


Рисунок 4.2 – Головний екран системи тестування знань.

Після відображення початкового екрану, наступним етапом є написання програмного коду для першого тестового питання з матеріалу лекції. (рис. 4.3)

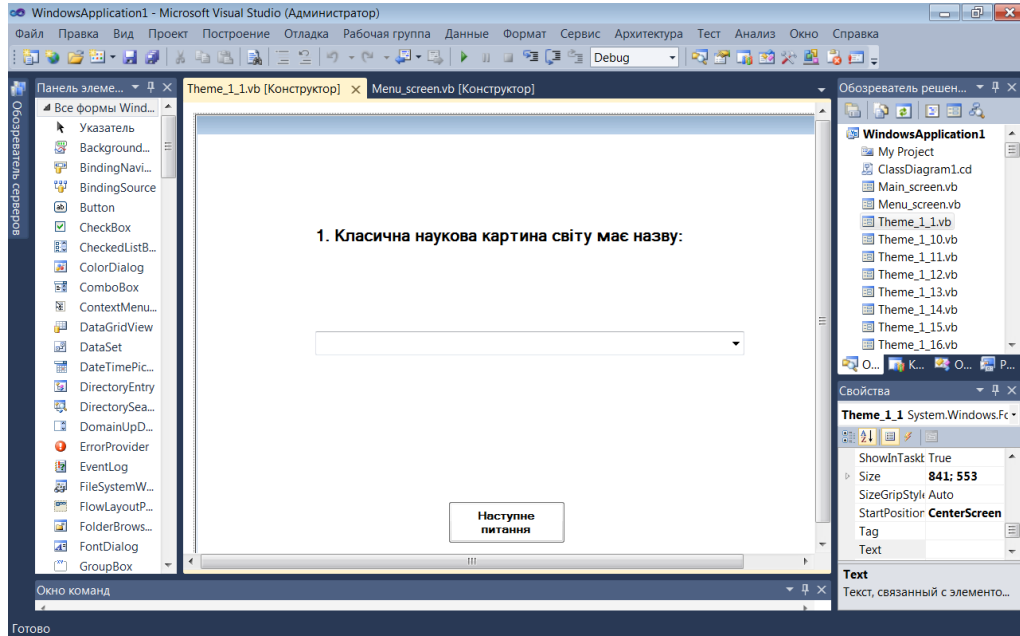


Рисунок 4.3 – Перше питання з лекційного матеріалу.

Після першого вікна з лекційним матеріалом, реалізується наступне вікно. (рис. 4.4)

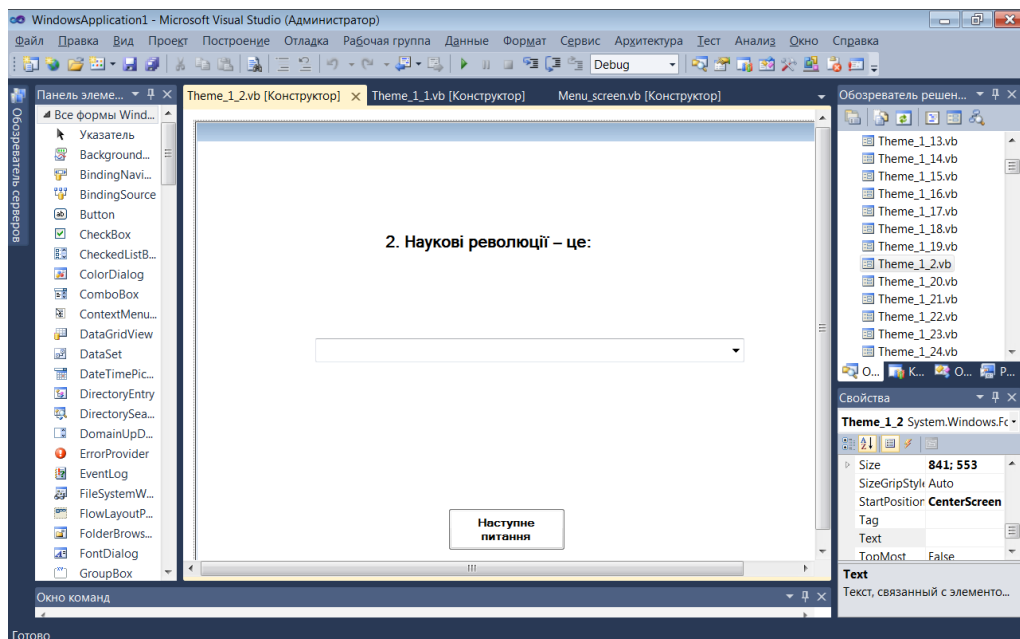


Рисунок 4.4 – Наступне вікно.

Весь вміст системи тестування знань запрограмовано за допомогою програмних засобів Microsoft Visual Studio та мови програмування Visual Basic. (рис. 4.5)

```

2 Partial Class Theme_1_1
24 Private Sub InitializeComponent()
25     Me.Label1 = New System.Windows.Forms.Label()
26     Me.Button1 = New System.Windows.Forms.Button()
27     Me.ComboBox1 = New System.Windows.Forms.ComboBox()
28     Me.SuspendLayout()
29
30     'Label1
31     .
32     Me.Label1.AutoSize = True
33     Me.Label1.Font = New System.Drawing.Font("Microsoft Sans Serif", 12.0!, System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, CType(204, Byte))
34     Me.Label1.ForeColor = System.Drawing.SystemColors.ActiveCaptionText
35     Me.Label1.Location = New System.Drawing.Point(143, 114)
36     Me.Label1.Name = "Label1"
37     Me.Label1.Size = New System.Drawing.Size(474, 25)
38     Me.Label1.TabIndex = 3
39     Me.Label1.Text = "1. Класична наукова картина світу має назву:"
40
41     'Button1
42     .
43     Me.Button1.BackColor = System.Drawing.Color.White
44     Me.Button1.Cursor = System.Windows.Forms.Cursors.Hand
45     Me.Button1.Font = New System.Drawing.Font("Microsoft Sans Serif", 7.8!, System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, CType(204, Byte))
46     Me.Button1.Location = New System.Drawing.Point(315, 468)
47     Me.Button1.Name = "Button1"
48     Me.Button1.Size = New System.Drawing.Size(147, 53)
49     Me.Button1.TabIndex = 4
50     Me.Button1.Text = "Початок виконання"
51     Me.Button1.UseVisualStyleBackColor = False
52
53     'ComboBox1
54     .
55     Me.ComboBox1.Font = New System.Drawing.Font("Microsoft Sans Serif", 10.8!, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(204, Byte))
56     Me.ComboBox1.FormattingEnabled = True
57     Me.ComboBox1.Items.AddRange(New Object() {"а) механістичної", "б) квантово-механічної", "в) електродинамічної", "г) енергетичної"})
58     Me.ComboBox1.Location = New System.Drawing.Point(148, 247)
59     Me.ComboBox1.Name = "ComboBox1"
60     Me.ComboBox1.Size = New System.Drawing.Size(539, 30)
61     Me.ComboBox1.TabIndex = 9
62
63     'Theme_1_1
64     .
65     Me.AutoScaleDimensions = New System.Drawing.SizeF(8.0!, 16.0!)
66     Me.BackColor = System.Drawing.Color.White
67     Me.ClientSize = New System.Drawing.Size(835, 525)
68

```

Рисунок 4.5 – Код системи тестування знань.

Додатково для інформативності розроблено вікно з інформацією про систему тестування знань. (рис. 4.6)

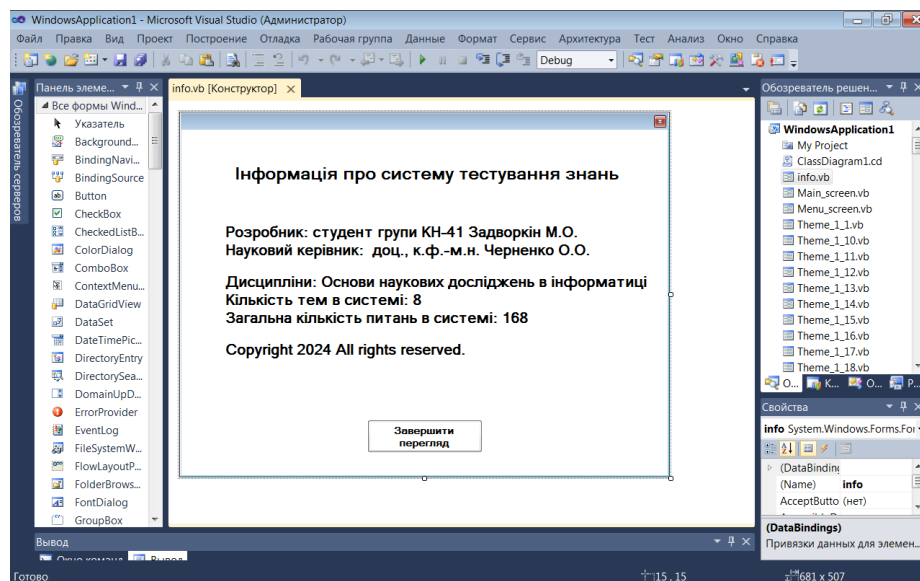


Рисунок 4.6 – Вікно інформації

4.3. Опис роботи системи тестування знань

Після входу в систему тестування знань, користувача вітає головний екран.
(рис. 4.7)

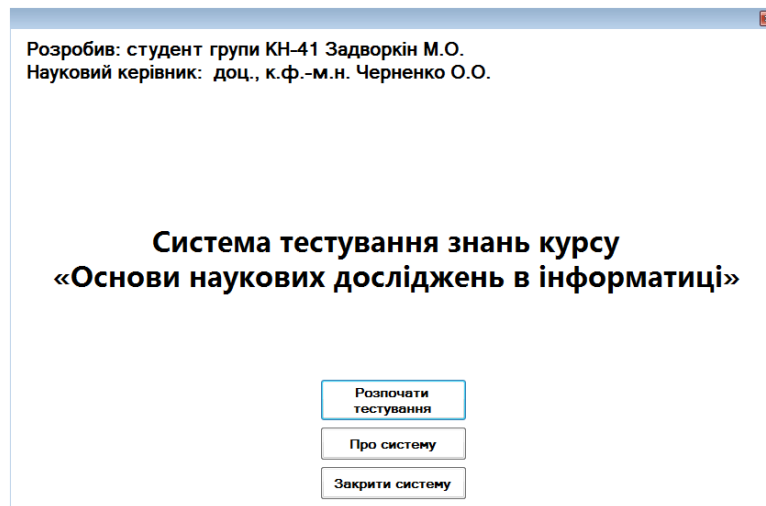


Рисунок 4.7 – Головний екран системи тестування знань.

Далі, користувач обирає з меню одну з 8-ми тем, натискає кнопку початку тестування, система автоматично запускає перше питання. Користувач може відповісти на нього, вибравши будь-який з доступних варіантів відповіді, на малюнку показано перше питання першої теми.
(рис. 4.8)

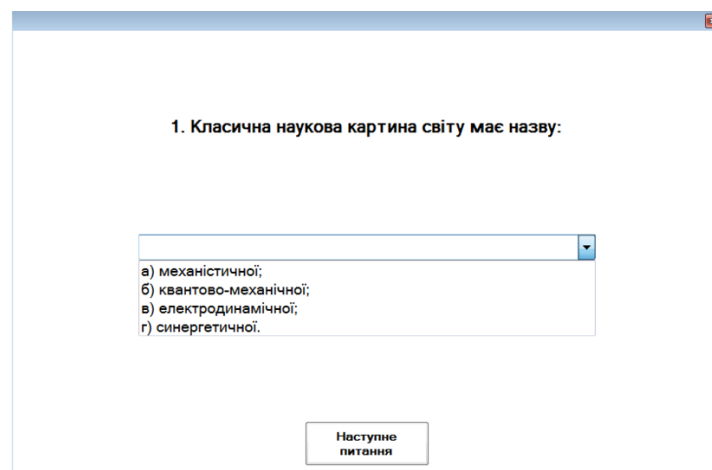


Рисунок 4.8 – Перше завдання першої теми.

Після того, як користувач обрав свій варіант відповіді, система тестування знань перевіряє цю відповідь і надає додаткову інформацію користувачеві через відповідне вікно. (рис. 4.9)

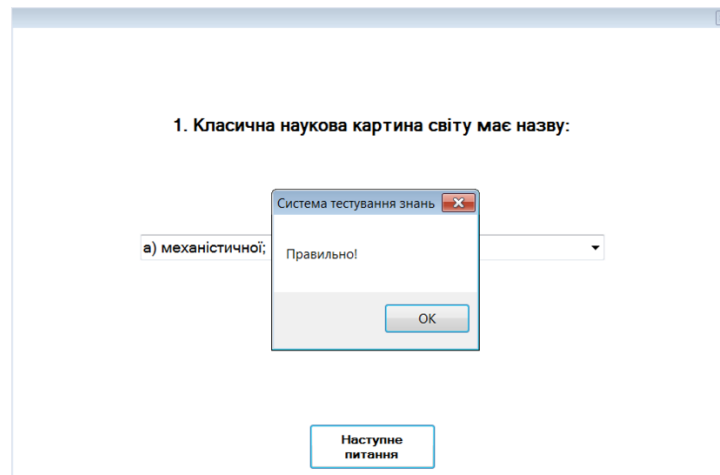


Рисунок 4.9 – Повідомлення у випадку правильної відповіді.

У разі неправильної відповіді користувача, система відображає наступне попередження. (рис. 5.0)

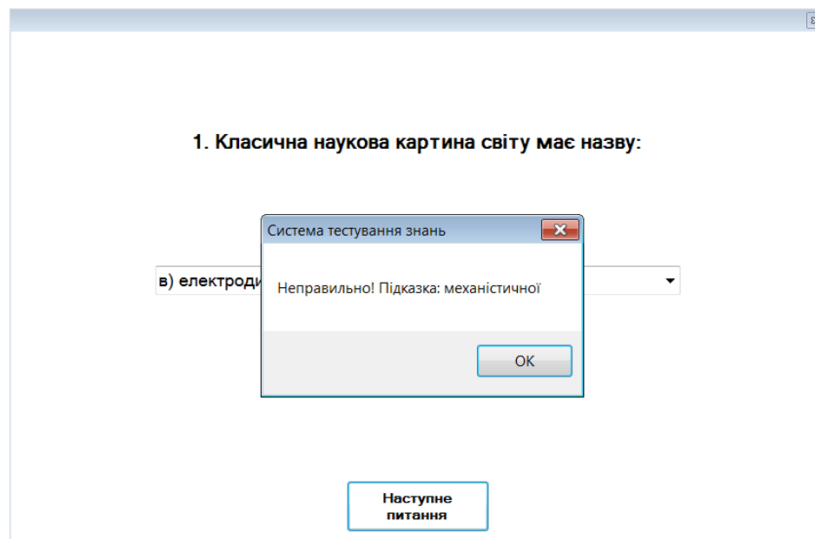


Рисунок 5.0 – Сповіщення в разі хибної відповіді.

Після закінчення теми 1, користувач обирає з меню одну з 8-ми тем, натискає кнопку початку тестування, система автоматично запускає перше питання. Користувач може відповісти на нього, вибравши будь-який з доступних варіантів відповіді, на малюнку показано перше питання другої теми. (рис. 5.1)

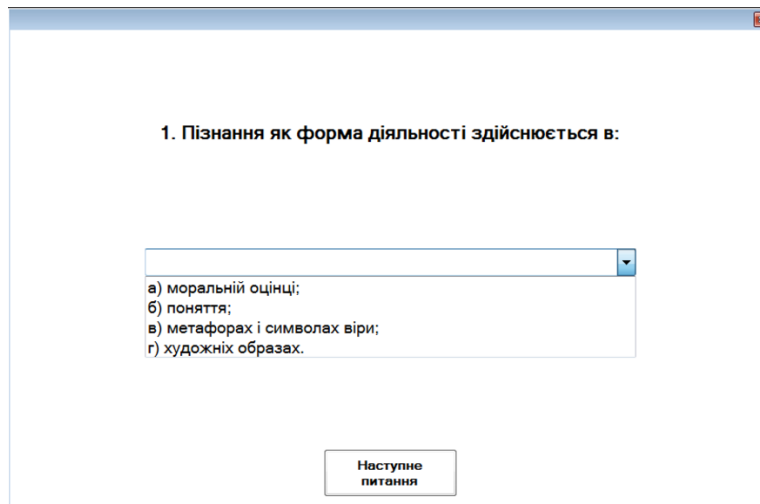


Рисунок 5.1 – Перше завдання другої теми.

Після того, як користувач обрав свій варіант відповіді, система тестування знань перевіряє цю відповідь і надає додаткову інформацію користувачеві через відповідне вікно. (рис. 5.2)

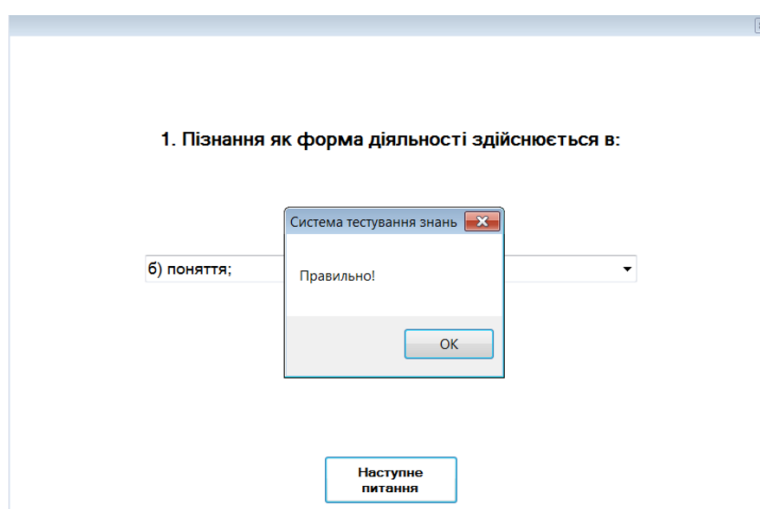


Рисунок 5.2 – Повідомлення у випадку правильної відповіді.

У разі неправильної відповіді користувача, система відображає наступне попередження. (рис. 5.3)

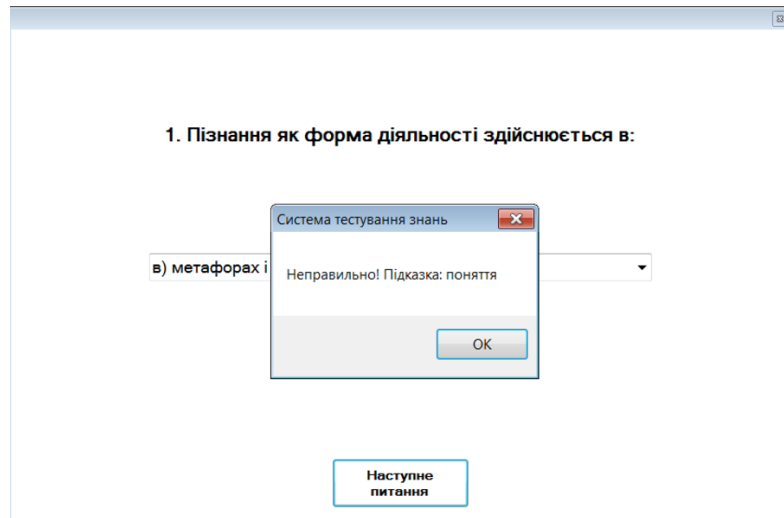


Рисунок 5.3 – Сповіщення в разі хибної відповіді.

Після закінчення теми 2, користувач обирає з меню одну з 8-ми тем, натискає кнопку початку тестування, система автоматично запускає перше питання. Користувач може відповісти на нього, вибравши будь-який з доступних варіантів відповіді, на малюнку показано перше питання третьої теми. (рис. 5.4)

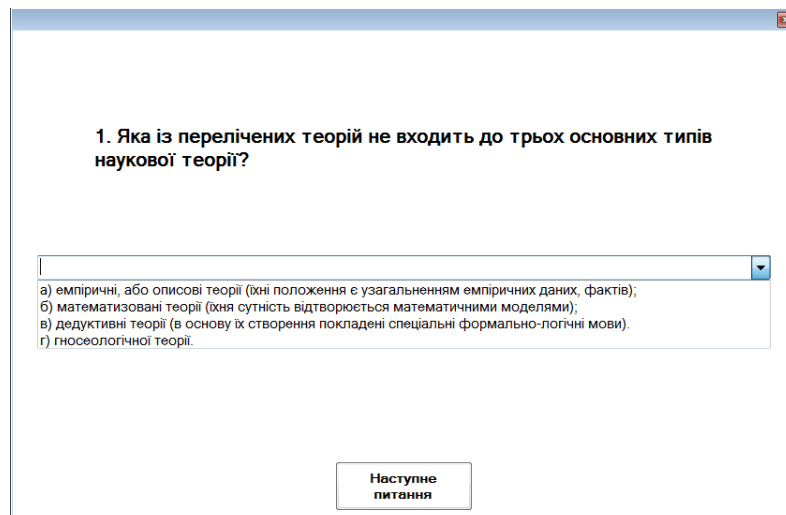


Рисунок 5.4 – Перше завдання третьої теми.

Після того, як користувач обрав свій варіант відповіді, система тестування знань перевіряє цю відповідь і надає додаткову інформацію користувачеві через відповідне вікно. (рис. 5.5)

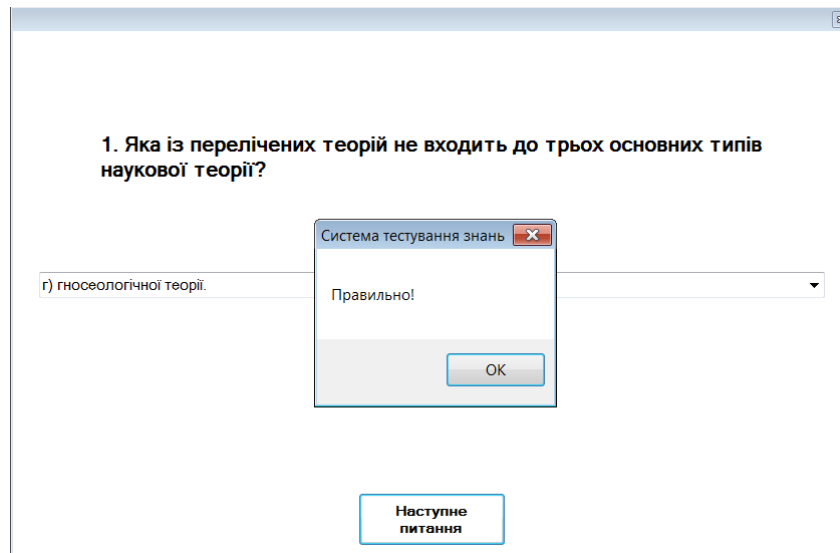


Рисунок 5.5 – Повідомлення у випадку правильної відповіді.

У разі неправильної відповіді користувача, система відображає наступне попередження. (рис. 5.6)

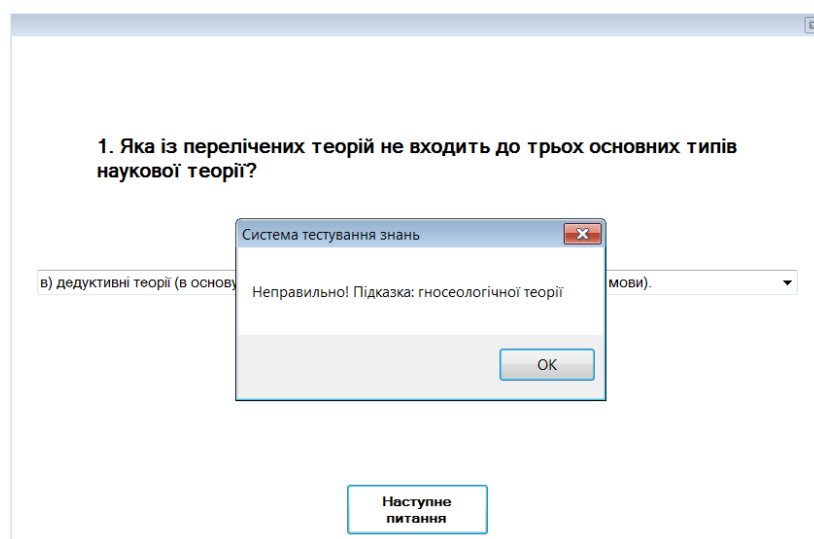


Рисунок 5.6 – Сповідження в разі хибної відповіді.

Після закінчення теми 3, користувач обирає з меню одну з 8-ми тем, натискає кнопку початку тестування, система автоматично запускає перше питання. Користувач може відповісти на нього, вибравши будь-який з доступних варіантів відповіді, на малюнку показано перше питання четвертої теми. (рис. 5.7)

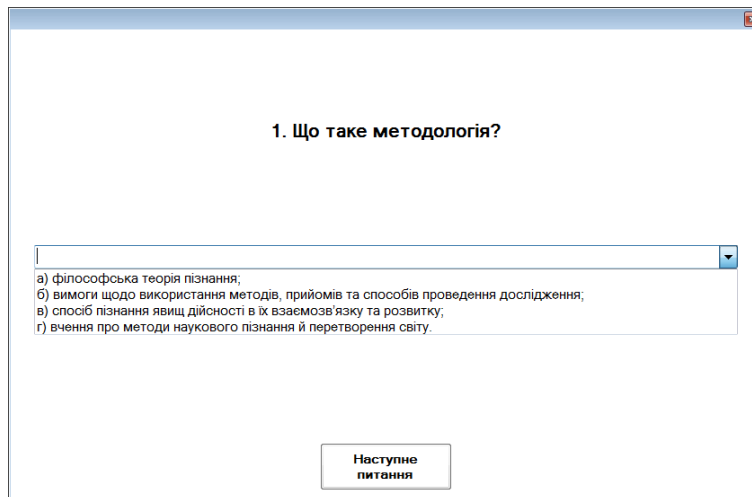


Рисунок 5.7 – Перше завдання четвертої теми.

Після того, як користувач обрав свій варіант відповіді, система тестування знань перевіряє цю відповідь і надає додаткову інформацію користувачеві через відповідне вікно. (рис. 5.8)

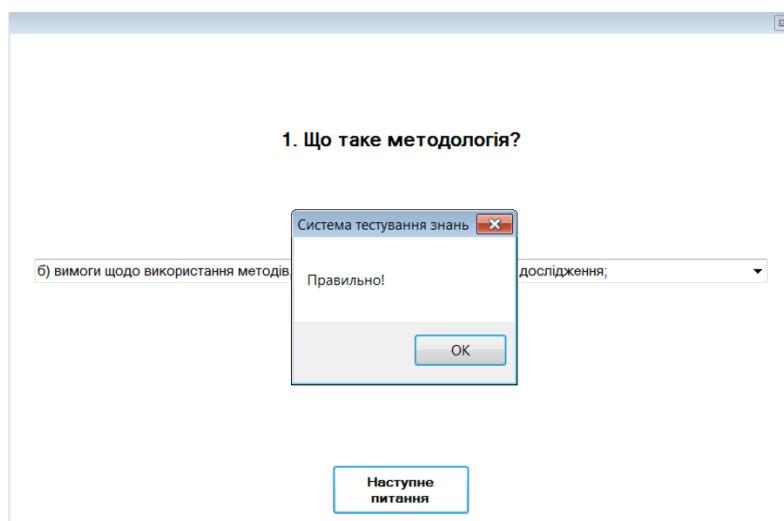


Рисунок 5.8 – Повідомлення у випадку правильної відповіді.

У разі неправильної відповіді користувача, система відображає наступне попередження. (рис. 5.9)

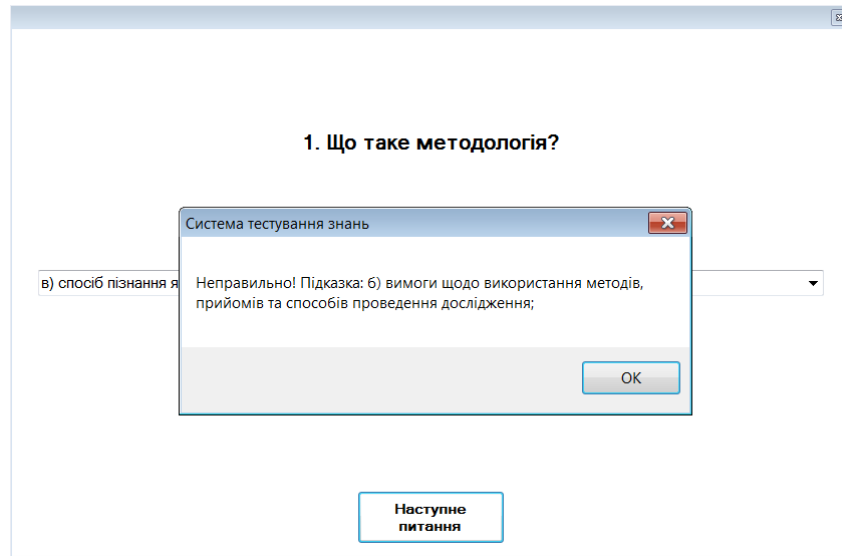


Рисунок 5.9 – Сповіщення в разі хибної відповіді

Після закінчення теми 4, користувач обирає з меню одну з 8-ми тем, натискає кнопку початку тестування, система автоматично запускає перше питання. Користувач може відповісти на нього, вибравши будь-який з доступних варіантів відповіді, на малюнку показано перше питання п'ятої теми. (рис. 6.0)

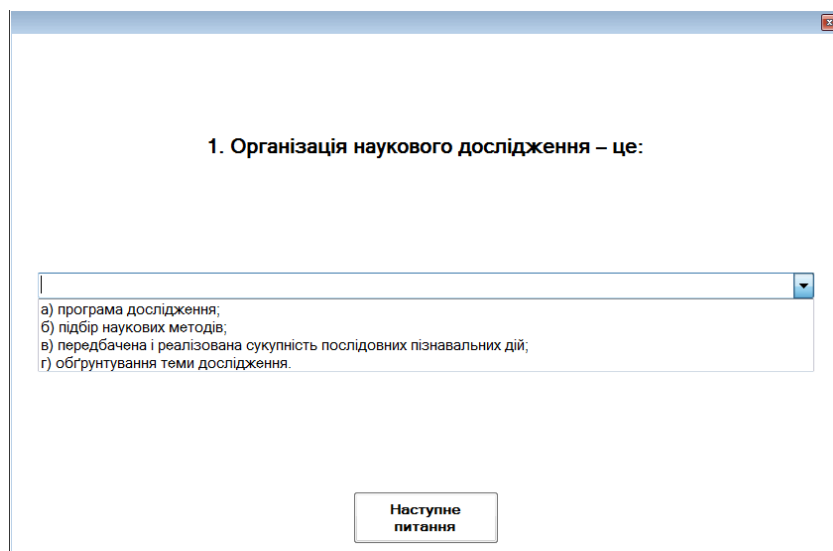


Рисунок 6.0 – Перше завдання п'ятої теми.

Після того, як користувач обрав свій варіант відповіді, система тестування знань перевіряє цю відповідь і надає додаткову інформацію користувачеві через відповідне вікно. (рис. 6.1)

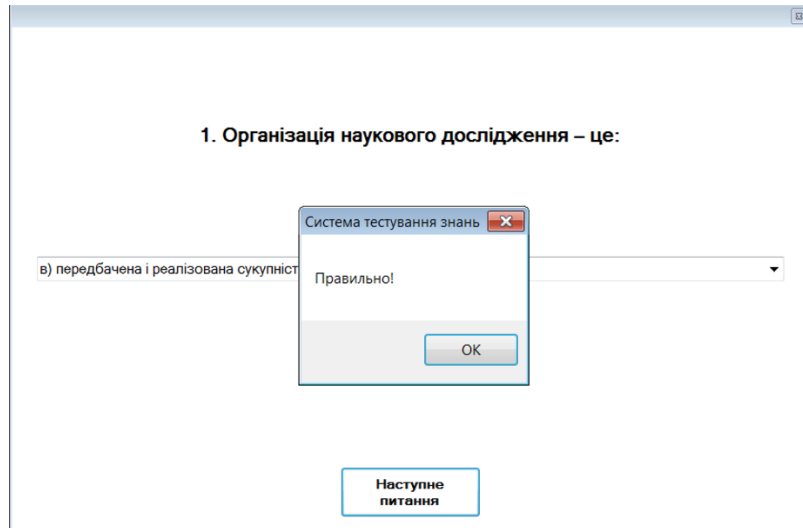


Рисунок 6.1 – Повідомлення у випадку правильної відповіді.

У разі неправильної відповіді користувача, система відображає наступне попередження. (рис. 6.2)

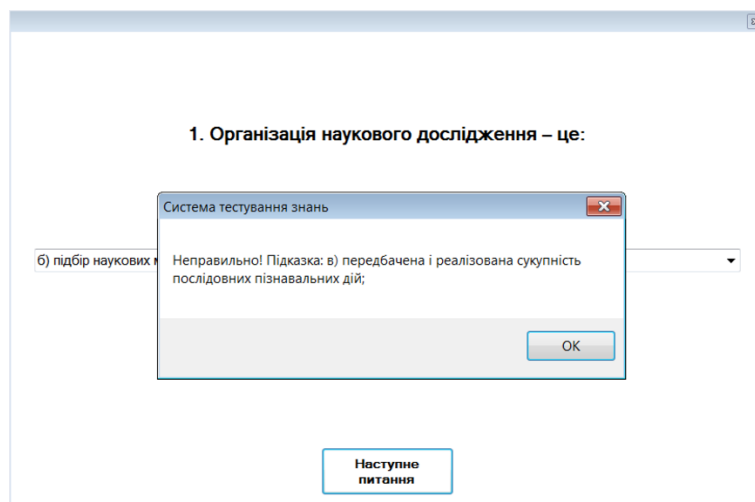


Рисунок 6.2 – Сповідження в разі хибної відповіді

Після закінчення теми 5, користувач обирає з меню одну з 8-ми тем, натискає кнопку початку тестування, система автоматично запускає перше питання. Користувач може відповісти на нього, вибравши будь-який з доступних варіантів відповіді, на малюнку показано перше питання шостої теми. (рис. 6.3)

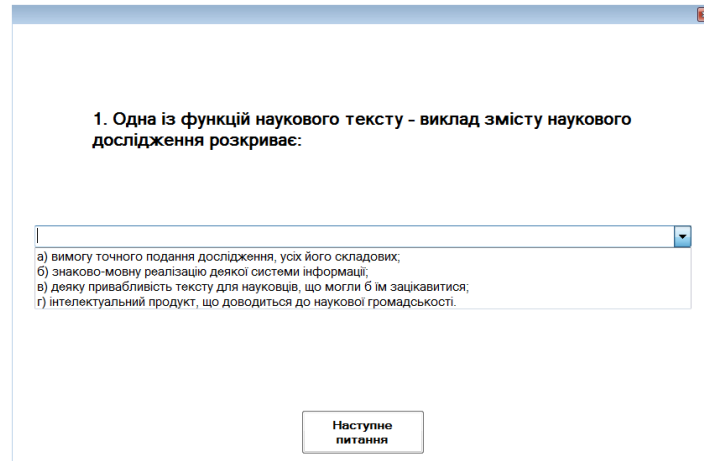


Рисунок 6.3 – Перше завдання шостої теми.

Після того, як користувач обрав свій варіант відповіді, система тестування знань перевіряє цю відповідь і надає додаткову інформацію користувачеві через відповідне вікно. (рис. 6.4)

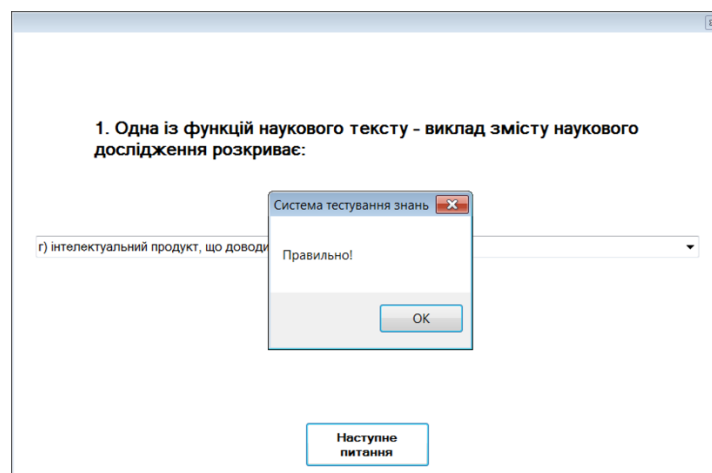


Рисунок 6.4 – Повідомлення у випадку правильної відповіді.

У разі неправильної відповіді користувача, система відображає наступне попередження. (рис. 6.5)

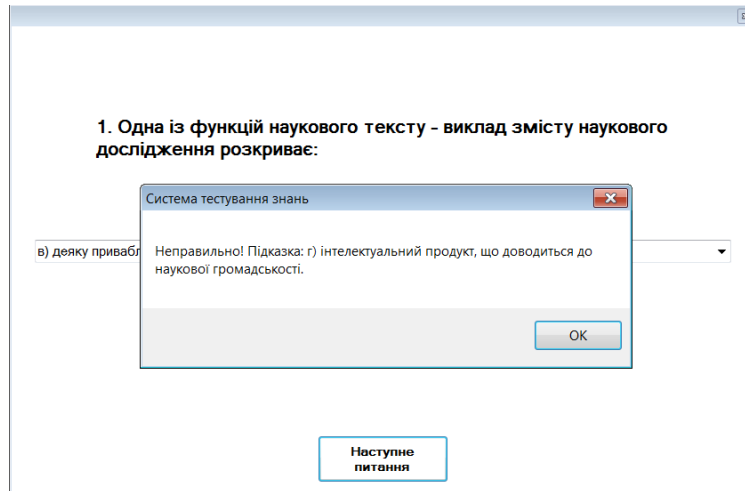


Рисунок 6.5 – Сповіщення в разі хибної відповіді

Після закінчення теми 6, користувач обирає з меню одну з 8-ми тем, натискає кнопку початку тестування, система автоматично запускає перше питання. Користувач може відповісти на нього, вибравши будь-який з доступних варіантів відповіді, на малюнку показано перше питання сьомої теми. (рис. 6.6)

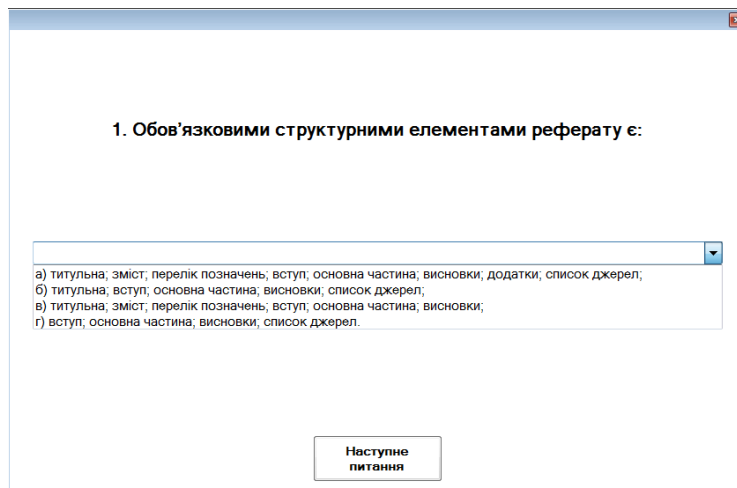


Рисунок 6.6 – Перше завдання сьомої теми.

Після того, як користувач обрав свій варіант відповіді, система тестування знань перевіряє цю відповідь і надає додаткову інформацію користувачеві через відповідне вікно. (рис. 6.7)

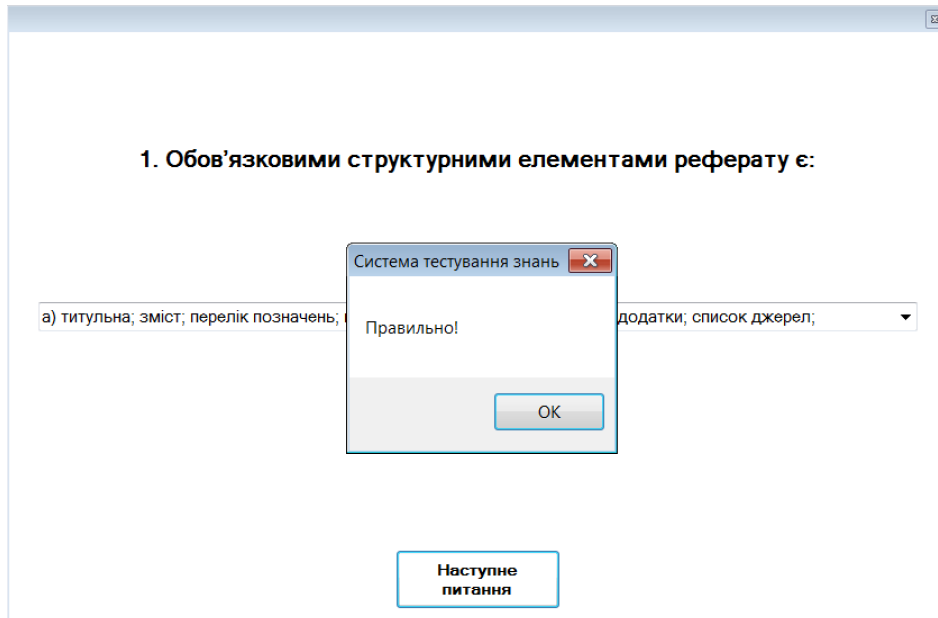


Рисунок 6.7 – Повідомлення у випадку правильної відповіді.

У разі неправильної відповіді користувача, система відображає наступне попередження. (рис. 6.8)

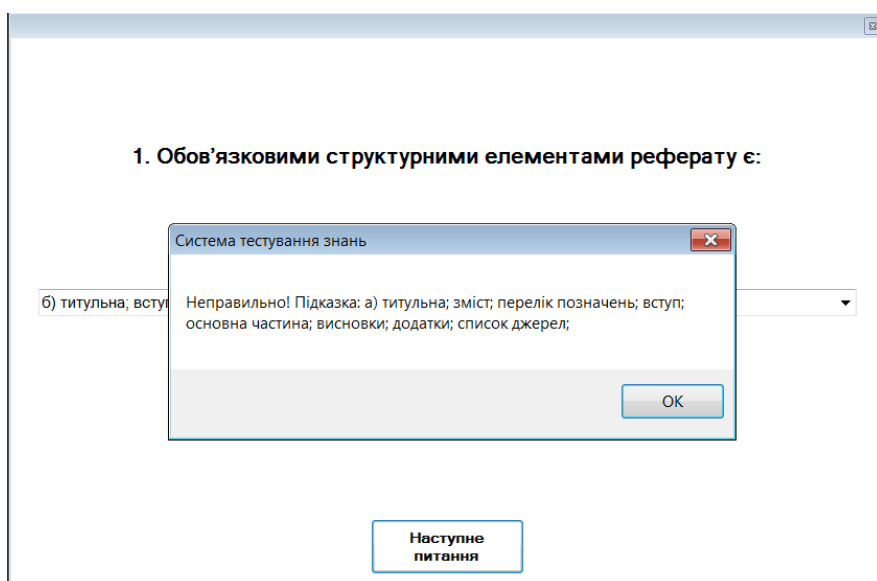


Рисунок 6.8 – Сповідження в разі хибної відповіді

Після закінчення теми 7, користувач обирає з меню одну з 8-ми тем, натискає кнопку початку тестування, система автоматично запускає перше питання. Користувач може відповісти на нього, вибравши будь-який з доступних варіантів відповіді, на малюнку показано перше питання восьмої останньої теми. (рис. 6.9)

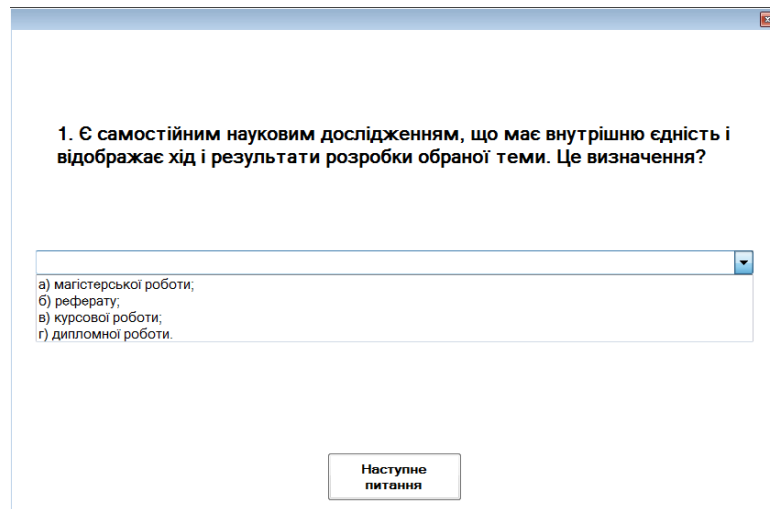


Рисунок 6.9 – Перше завдання восьмої останньої теми.

Після того, як користувач обрав свій варіант відповіді, система тестування знань перевіряє цю відповідь і надає додаткову інформацію користувачеві через відповідне вікно. (рис. 7.0)

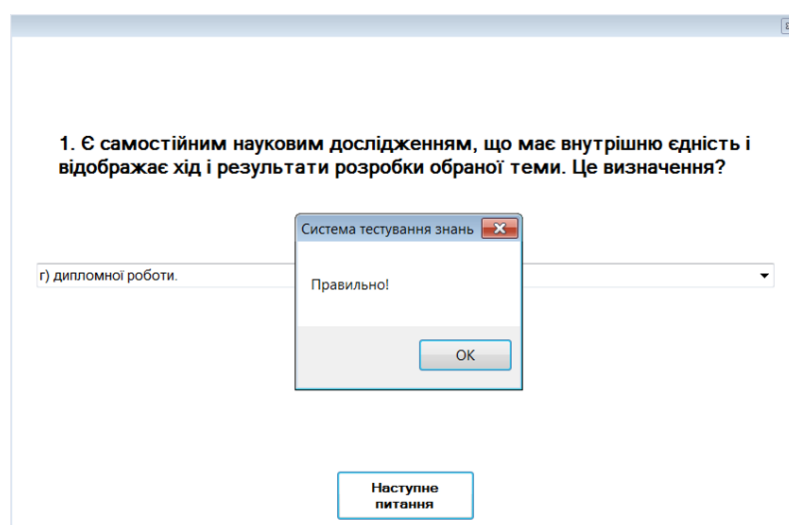


Рисунок 7.0 – Повідомлення у випадку правильної відповіді.

У разі неправильної відповіді користувача, система відображає наступне попередження. (рис. 7.1)

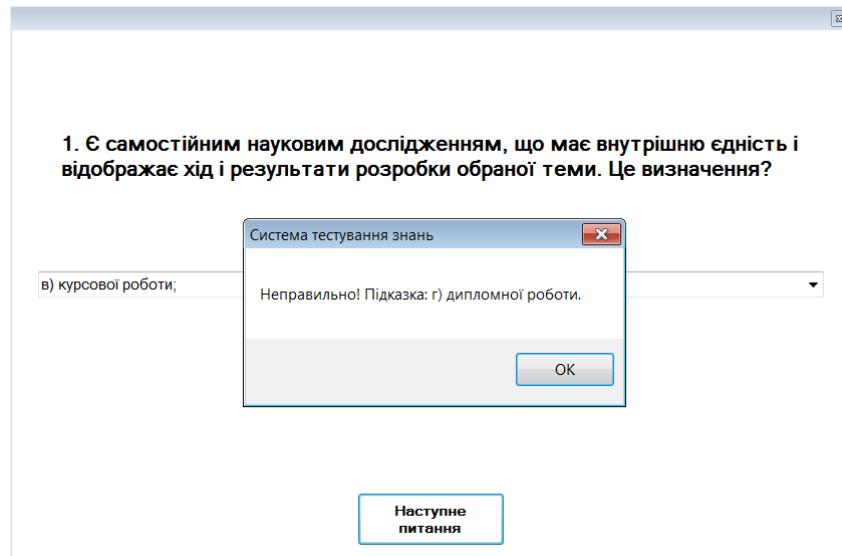


Рисунок 7.1 – Сповіщення в разі хибної відповіді

В системі також є список тем перед початком тренінгу, користувачеві необхідно вибрати тему яка його цікавить і розпочнеться тестування. (рис. 7.2)

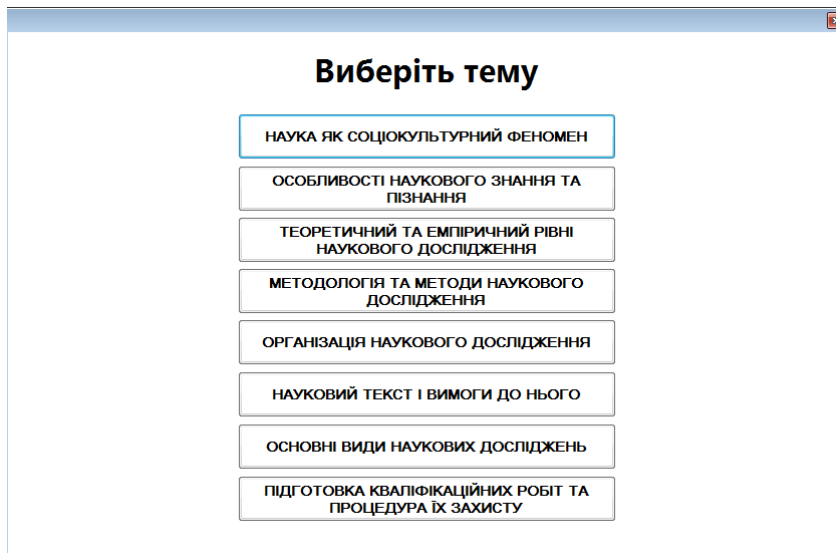


Рисунок 7.2 – Список тем.

Якщо на стартовому екрані системи тестування знань натиснути кнопку “Про систему”, користувач отримає вікно з детальною інформацією про систему. (рис. 7.3)

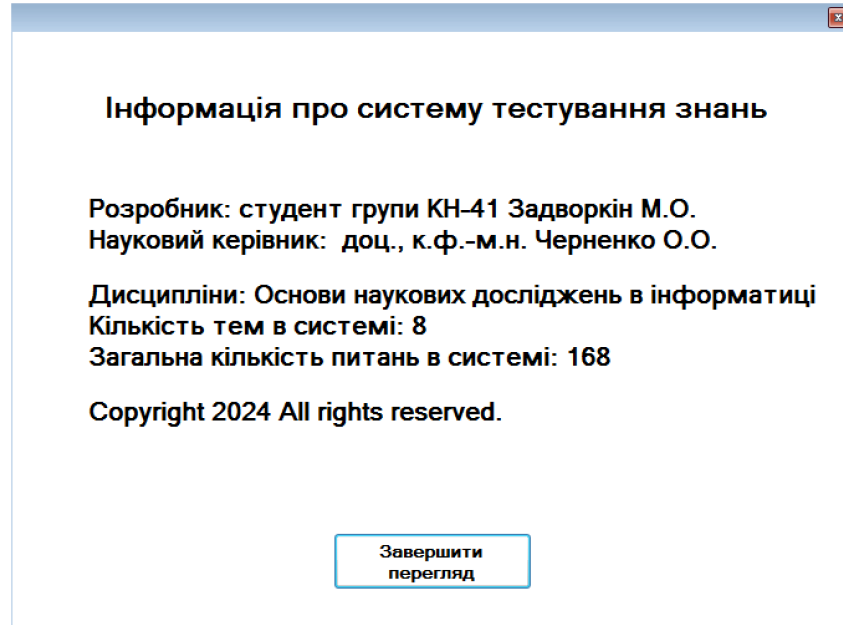


Рисунок 7.3 – Вікно інформації.

Після проходження тестування, в кінці користувачеві виводиться кількість правильних та неправильних відповідей. (рис. 7.4)

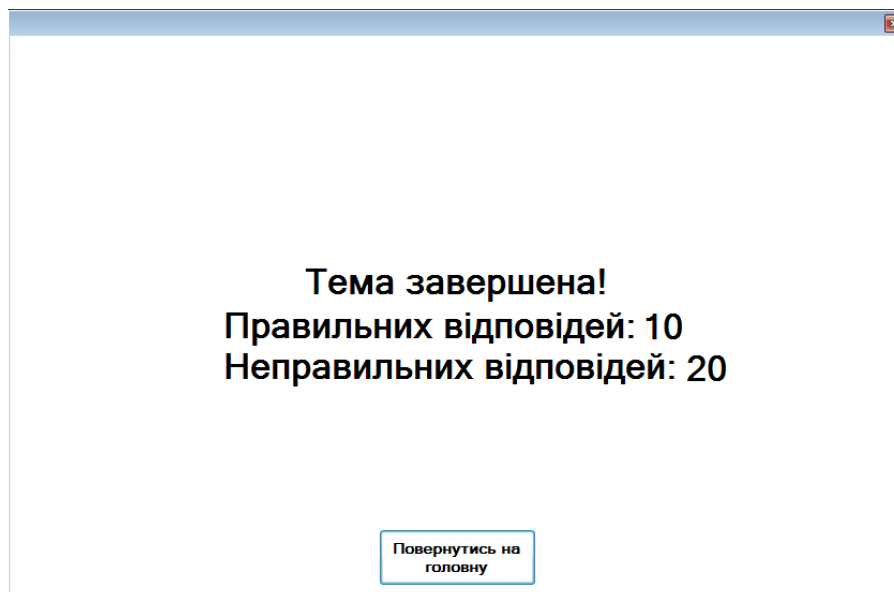


Рисунок 7.4 – Підсумкове вікно системи тестування знань з кількістю відповідей.

ВИСНОВКИ

Отже, дистанційне навчання сьогодні є необхідною та важливою складовою сучасної освітньої системи. Його важливість базується на численних факторах, які відкривають нові можливості для освіти та розвитку у всьому світі. Системи тестування знань залишаються актуальними в сучасному світі, де освіта і постійне навчання відіграють ключову роль у розвитку особистості та професійному успіху. Ці системи надають можливість об'єктивно виміряти рівень засвоєння матеріалу, виявляти слабкі місця та визначати напрямки подальшого розвитку.

В результаті виконання кваліфікаційної роботи було створено систему тестування знань з теми «Основи наукових досліджень в інформатиці».

Основні результати кваліфікаційної роботи:

1. Розроблено зручну та інформативну систему тестування знань студентів з теми «Основи наукових досліджень в інформатиці».
2. Розроблено вікна, що включають тестові завдання та питання з теми тестування, на які користувач повинен надавати вірні або невірні відповіді.
3. Створено вікно, в якому описана детальна інформація про систему тестування знань.
4. Впроваджено зручне меню вибору, за допомогою якого в користувача є можливість вибрати одну з 8-ми тем, загальна кількість питань системи – 168.
5. Додано кінцеві вікна після кожної теми, які містять інформацію про результати проходження поточної теми.

Система тестування знань з теми «Основи наукових досліджень в інформатиці» повністю готова користувачам для використання та навчання.

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 навчання [Електронний ресурс]. – Режим доступу: <https://life.pravda.com.ua/society/2020/07/2/241517/>
3. Приклади комп'ютерних систем тестування знань [Електронний ресурс]. – Режим доступу: <https://funmath1234.blogspot.com/p/blog-page.html>
4. Основи наукових досліджень [Електронний ресурс]. – Режим доступу: https://shron1.chtyvo.org.ua/Romanchykov_Volodymyr/Osnovy_naukovykh_do_slidzhen.pdf
5. Тестова система оцінювання знань студентів [Електронний ресурс]. – Режим доступу: <https://ap.uu.edu.ua/article/279>
6. Онлайн система тестування знань [Електронний ресурс]. – Режим доступу: <http://testing-system.vntu.net/>
7. Основи наукових досліджень: Конспект лекцій [Електронний ресурс]. – Режим доступу: <https://essuir.sumdu.edu.ua/retrieve/62596/Kolisnichenko.doc>
8. Рівні та види наукових досліджень [Електронний ресурс]. – Режим доступу: <http://www.tsatu.edu.ua/ros1/wp-content/uploads/sites/20/lekcija-1.teoretychni-osnovy-naukovykh-doslidzhen-z-ahronomiyi.pdf>
9. Методології розробки програмного забезпечення [Електронний ресурс]. – Режим доступу: <https://wezom.com.ua/ua/blog/metodologija-razrabotki-programmnogo-obespechenija>
10. 9 основних методологій розробки [Електронний ресурс]. – Режим доступу: <https://agile.yakubovsky.com/ua/2015/10/9-osnovnykh-metodolohiy-rozrobky-prohramnoho-zabezpechennya-agile/>

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. Development with Visual Studio [Електронний ресурс]. – Режим доступу: <https://learn.microsoft.com/en-us/visualstudio/get-started/vb/?view=vs-2022>
14. Microsoft Visual Basic [Електронний ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/Visual_Basic
15. 9. Програмування з використанням Visual BASIC [Електронний ресурс]. – Режим доступу: https://kursoviks.com.ua/bd_kompyuterni/article_post/65-lektsiya-osnovi-programuvannya-movoyu-visual-basic-6-0
16. Створення програми на VB в Visual Studio [Електронний ресурс]. – Режим доступу: <https://learn.ztu.edu.ua/mod/page/view.php?id=9976>
17. Getting Started With vb on Visual Studio Code [Електронний ресурс]. – Режим доступу: <https://medium.com/@adebiyiadedotun9/getting-started-with-c-on-visual-studio-code-5a76dcca1110>
18. 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.
19. ReSharper: The Visual Studio Extension for .NET [Електронний ресурс]. – Режим доступу: <https://www.jetbrains.com/resharper/>
20. Install Visual Studio [Електронний ресурс]. – Режим доступу: <https://www.csharptutorial.net/csharp-tutorial/install-visual-studio/>

ДОДАТОК А. КОД ПРОГРАМИ