

Центральноукраїнський національний технічний університет  
Механіко-технологічний факультет  
Кафедра кібербезпеки та програмного забезпечення

”Допущено до захисту”  
Завідувач кафедри кібербезпеки  
та програмного забезпечення  
д.т.н., професор  
\_\_\_\_\_ Олексій СМІРНОВ  
“ \_\_\_\_ ” \_\_\_\_\_ 2023 р.

**ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА**  
**за першим (бакалаврським) рівнем вищої освіти**  
на тему

**“Програмне забезпечення системи підсистеми контролю знань  
в хмарних навчаючих системах”**

Виконав здобувач вищої освіти  
IV курсу, групи КІ-19  
ОПП «Комп’ютерна інженерія»  
спеціальності 123 «Комп’ютерна інженерія»  
\_\_\_\_\_ Цішевський М.С.  
« \_\_\_\_ » \_\_\_\_\_ 2023 р.

Керівник проекту  
кандидат технічних наук  
\_\_\_\_\_ Буравченко К.О.  
« \_\_\_\_ » \_\_\_\_\_ 2023 р.

Рецензент \_\_\_\_\_  
\_\_\_\_\_

Центральноукраїнський національний технічний університет  
Факультет Механіко-технологічний  
Кафедра Кібербезпеки та програмного забезпечення  
Освітній ступінь бакалавр  
Галузь знань . 12 “Інформаційні технології”  
Спеціальність 123 “Комп’ютерна інженерія”  
Освітньо-професійна (освітньо-наукова) програма “Комп’ютерна інженерія”

ЗАТВЕРДЖУЮ

Завідувач кафедри

д.т.н., проф.

Олексій СМІРНОВ

« 17 » січня 2023 року

## ЗАВДАННЯ НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ РОБОТУ ЗА ПЕРШИМ (БАКАЛАВРСЬКИМ) РІВНЕМ ВИЩОЇ ОСВІТИ ЗДОБУВАЧА ВИЩОЇ ОСВІТИ

*Цішевському Михайлу Станіславовичу*

(прізвище, ім'я, по батькові)

1. Тема роботи *Програмне забезпечення системи підсистеми контролю знань в хмарних навчаючих системах*

2. Керівник роботи *Буравченко Костянтин Олегович, канд. техн. наук*

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу № 7-02 від 5.01.2023 року

3. Строк подання студентом роботи до захисту *23.05.2023 р.*

4. Мета та завдання випускної кваліфікаційної роботи: *Метою роботи є розробка програмного забезпечення системи підсистеми контролю знань в хмарних навчаючих системах*

5. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

*1. Призначення та область використання.*

*2. Перегляд аналогічних існуючих систем.*

*3. Опис і обґрунтування проектних рішень.*

*4. Етапи програмування системи.*

*5. Впровадження системи в промислову експлуатацію.*

*6. Висновки*

6. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

*Структурна схема системи* *1 аркуш*

*Функціональна схема системи* *1 аркуш*

*Діаграма процесів* *1 аркуш*

*Блок-схема алгоритму роботи додатку* *2 аркуша*

7. Дата видачі завдання « 17 » січня 2023 р.

### КАЛЕНДАРНИЙ ПЛАН

| № з/п | Назва етапів випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти | Строк виконання етапів випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти | Примітка |
|-------|---|---|----------|
| 1.    | Аналіз існуючих систем  | 10.03.2023 р.   |          |
| 2.    | Постановка задачі, оформлення ТЗ  | 15.03.2023 р.   |          |
| 3.    | Розробка моделі компонента  | 20.03.2023 р.   |          |
| 4.    | Розробка структур даних   | 25.03.2023 р.   |          |
| 5.    | Розробка алгоритмів зв'язку та відображення   | 30.03.2023 р.   |          |
| 6.    | Програмування алгоритмів  | 10.04.2023 р.   |          |
| 7.    | Оформлення ПЗ   | 17.04.2023 р.   |          |
| 8.    | Попередній захист роботи  | 23.05.2023 р.   |          |
|       |   |   |          |
|       |   |   |          |
|       |   |   |          |
|       |   |   |          |

Дата видачі завдання  
« 17 » січня 2023 р.

Підпис керівника

Буравченко К.О.  
(прізвище та ініціали)

Завдання прийнято до виконання  
« 17 » січня 2023 р.

Підпис здобувача

Цішевський М.С.  
(прізвище та ініціали)

## АНОТАЦІЯ

**Цішевський М.С. Програмне забезпечення системи підсистеми контролю знань в хмарних навчаючих системах. 123 Комп'ютерна інженерія. Центральноукраїнський національний технічний університет. Кропивницький. 2023.**

В даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи підсистеми контролю знань в хмарних навчаючих системах.

Метою розробки є програмне забезпечення системи підсистеми контролю знань в хмарних навчаючих системах.

Результат роботи – програмна реалізація системи підсистеми контролю знань в хмарних навчаючих системах.

В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.

Розроблено зручний інтерфейс користувача. Наведені інструкції по роботі з програмними засобами.

Програма може використовуватися на ПЕОМ архітектури IBM PC з ОС Windows 10/11.

Програму розроблено в середовищі PHP.

**Ключові слова:** комп'ютерна інженерія, підсистема контролю знань, хмарні навчаючі системи

## ABSTRACT

**Tsishevskiy M.S. Software of the knowledge control subsystem system in cloud-based educational systems. 123 Computer engineering. Central Ukrainian National Technical University. Kropyvnytskyi. 2023.**

In this graduation thesis for the first (bachelor) level of higher education, software is developed, which is intended for the system of the knowledge control subsystem in cloud-based educational systems.

The goal of the development is the software of the knowledge control subsystem in cloud-based educational systems.

The result of the work is the software implementation of the knowledge control subsystem in cloud-based educational systems.

In the process of working on the software model, an analysis of existing hardware and software was performed. All components of the developed software are fully described.

A convenient user interface has been developed. Instructions for working with software tools are provided.

The program can be used on PCs of IBM PC architecture with Windows 10/11 OS.

The program is developed in the PHP environment.

**Keywords:** computer engineering, knowledge control subsystem, cloud-based learning systems

## ЗМІСТ

|   |    |
|---|----|
| ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ .....   | 2  |
| ВСТУП.....  | 3  |
| 1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ .....   | 5  |
| 1.1 Призначення системи.....  | 5  |
| 1.2 Область застосування.....   | 6  |
| 2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ .....  | 11 |
| 2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти..... | 11 |
| 2.2 Обґрунтування вибору засобів для побудови системи та мови програмування.....  | 20 |
| 2.3 Розгорнута постановка завдання .....  | 25 |
| 3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ .....   | 27 |
| 3.1 Опис функціонування системи .....   | 27 |
| 3.2 Розробка структурної схеми.....   | 43 |
| 3.3 Розробка функціональної схеми .....   | 59 |
| 3.4 Розробка діаграми процесів.....   | 61 |
| 4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....  | 64 |
| 4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....   | 64 |
| 4.2 Захист розробленого програмного забезпечення.....   | 80 |
| 5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ .....  | 83 |
| 6 ОСНОВНІ ВИСНОВКИ.....   | 89 |
| СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....  | 91 |

|                 |                        |                 |              |             |  |                   |              |                |
|-----------------|------------------------|-----------------|--------------|-------------|--|-------------------|--------------|----------------|
|                 |                        |                 |              |             | <b>ВКРБ-123.23.0007.00.00.ПЗ</b>   |                   |              |                |
| <b>Вим.</b>     | <b>Арк.</b>            | <b>№ докум.</b> | <b>Підп.</b> | <b>Дата</b> | <i>Програмне забезпечення системи підсистеми контролю знань в хмарних навчаючих системах</i> | <b>Літ.</b>       | <b>Аркуш</b> | <b>Аркушів</b> |
| <i>Розроб.</i>  | <i>Цішевський М.С.</i> |                 |              |             |  | <b>Б</b>          | 1            | 101            |
| <i>Перев.</i>   | <i>Буравченко К.О.</i> |                 |              |             |  |                   |              |                |
| <i>Н.контр.</i> | <i>Гермак В.С.</i>     |                 |              |             |  |                   |              |                |
| <i>Затв.</i>    | <i>Смірнов О.А.</i>    |                 |              |             |  |                   |              |                |
|                 |                        |                 |              |             |  | <i>ЦНТУ КІ-19</i> |              |                |

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

|           |   |
|-----------|---|
| АНС       | – автоматизована система, яка навчає            |
| АНК       | – автоматизований навчальний курс               |
| БД        | – база даних                                    |
| БЗ        | – база знань                                    |
| БПЗ       | – базове програмне забезпечення                 |
| ОТ        | – обчислювальна техніка                         |
| ІС        | – інструментальна система                       |
| КНП       | – комп'ютерна навчальна програма                |
| КЗНП      | – комп'ютерні засоби навчального призначення    |
| ЛОМ       | – локальна обчислювальна мережа                 |
| ЛКПЗ      | – локальне комунікаційне програмне забезпечення |
| ЛППЗ      | – локальне прикладне програмне забезпечення     |
| МН        | – модель навчаємого                             |
| МПредмОбл | – модель предметної області                     |
| ПредмОбл  | – предметна область                             |
| ППЗ       | – прикладне програмне забезпечення              |
| ХАНС      | – хмарна автоматизована навчальна система       |
| СКЗ       | – система контролю знань                        |
| СУБД      | – система управління базою даних                |

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-123.23.0007.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 2    |

## ВСТУП

**Актуальність теми.** Програмні засоби навчального призначення по принципах використання можна умовно розділити на навчальні системи, наповнені знаннями про конкретну предметну область, і інструментальні системи, призначені для наповнення їхніми знаннями про довільну предметну область із метою створення навчальної системи [1]. Найбільш перспективними з погляду співвідношення кінцевого результату й працезатрат на створення й підтримку є інструментальні системи, які прийнята називати автоматизованими навчальними системами (АНС) [1, 5]. До основних переваг АНС відносяться:

- можливість використання переваг індивідуального навчання [2];
- інтенсифікація навчання [8];
- можливість індивідуальної адаптації курсу навчання до потреб тих, кого навчають, або умов навчання [6];
- можливість використання й тиражування передового досвіду [6];
- підвищення доступності утворення [5];
- навчання навичкам самостійної роботи [6];
- розвантаження викладача від ряду рутинних, повторюваних дій (читання лекцій, перевірки контрольних робіт і т.д.) [6];
- можливість використання в рамках дистанційного навчання, перенавчання й підвищення кваліфікації [3].

У цей час завдяки розвитку обчислювальних мереж і телекомунікаційних технологій АНС одержали можливість вийти на новий рівень. При переході від локальних навчальних систем до хмарних якісно змінюються функціональні можливості навчальної системи. Організація хмарних АНС (ХАНС) вимагає пророблення мережних аспектів роботи системи, пов'язаних з наданням віддаленого доступу до системи, підтримкою хмарних даних і об'єднанням мережних ресурсів для рішення вартих перед системою завдань.

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРБ-123.23.0007.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 3    |

Одним з важливих завдань при створенні ХАНС є організація контролю знань. Більшість існуючих АНС і систем контролю знань мають обмежену кількість форм подання відповідей і двобальну систему оцінки. Це обумовлено простотою аналізу вибіркових відповідей і відсутністю формальних методів аналізу й диференційованої оцінки відповідей тих, кого навчають, на контрольні питання. Однак такий підхід обмежує можливості розроблювача курсу відносно використання різних варіантів тестових питань і аналізу відповідей тих, кого навчають. У зв'язку із цим тематика досліджень, що зачіпають організацію контролю знань у ХАНС, є актуальною.

**Мета й завдання дослідження.** Метою роботи є програмне забезпечення системи підсистеми контролю знань в хмарних навчаючих системах.

Для досягнення поставленої мети визначена програма дослідження, що складається з наступних завдань:

- Огляд існуючих систем підсистеми контролю знань в хмарних навчаючих системах.
- Дослідження системи підсистеми контролю знань в хмарних навчаючих системах.
- Програмна реалізація системи підсистеми контролю знань в хмарних навчаючих системах.

**Практична цінність отриманих результатів** полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі підсистеми контролю знань в хмарних навчаючих системах.

Таким чином, виходячи з вищеперерахованого, програмне забезпечення системи підсистеми контролю знань в хмарних навчаючих системах, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти.

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРБ-123.23.0007.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 4    |

# 1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

## 1.1 Призначення системи

Система призначена для реалізації підсистеми контролю знань в хмарних навчаючих системах.

При створенні підсистеми контролю знань необхідно:

- зробити систему максимально відкритою;
- забезпечити переносимість тестів на рівні їхніх вихідних текстів;
- по можливості врахувати всі типи питань і відповідей;
- реалізувати методи оцінки відповідей різних типів, які дозволять диференційовано оцінювати відповіді того, кого навчають.

Для рішення цього завдання потрібно:

- проаналізувати різні типи тестових питань і відповідей;
- ввести метрику для визначення ступеня подібності відповідей того, кого навчають і еталонних відповідей;
- розробити формат опису тестів, що враховує різні варіанти відповідей і питань і можливість підключення зовнішніх модулів.

Створення мови опису тестів вирішує завдання уніфікації СКЗ. Розробка розвиненого засобу опису дозволяє, з одного боку, уніфікувати інтерфейс, використовуючи ту саму оболонку для різних тестів. Це приведе до зниження працезатрат користувача на вивчення системи й дозволить йому зосередитися на відповідях на питання. З іншого боку, відкритість формату опису тесту дає можливість створювати різні оболонки для того самого набору контрольних завдань, адаптуючи систему для різних операційних систем, апаратних платформ і вимог по організації інтерфейсу.

Для того щоб відокремити опис тесту від його алгоритмічної реалізації й зробити його стерпним, мова опису тесту повинен бути декларативною.

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРБ-123.23.0007.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 5    |

Будь-яка система контролю знань містить у собі множина пропонованих питань, правильні відповіді на них і правила визначення коректності відповідей, отриманих у ході опитування. Тому опис тесту складається з наступних частин:

- опис правил формування тесту зі списку контрольних завдань і методів оцінки результатів тесту в цілому;
- опис питань;
- опис еталонів відповідей і методів оцінок відповідей, отриманих у ході опитування.

Таким чином розробка програмного забезпечення підсистеми контролю знань в хмарних навчаючих системах є актуальною задачею, яка потребує вирішення у даній бакалаврській роботі.

## 1.2 Область застосування

Визначимо область застосування системи, яка розробляється. Областю застосування є автоматизовані навчальні системи (АНС). Багато дослідників вважають АНС найбільш перспективним напрямком в області створення комп'ютерних засобів навчального призначення по цілому ряді причин.

### Переваги автоматизованих навчальних систем

Переваги АНС перед іншими видами навчальних систем у першу чергу визначаються тим, що фактично АНС – це інструментальний комплекс для створення комп'ютерних засобів навчального призначення. Розходження між довільною навчальною системою й інструментальним комплексом для створення навчальних систем полягає в тому, що інструментальний комплекс не містить знань по конкретній предметній області.

Створення будь-якої навчальної програми вимагає більших працезатрат [6]. Застосування АНС дозволяє скоротити трудомісткість і строки розробки за рахунок наявності готових механізмів для виконання функцій навчання [5].

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-123.23.0007.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 6    |

Крім здешевлення й прискорення розробки навчальної системи застосування АНС має наступні переваги:

- у систему можна вкласти знання по довільній предметній області;
- АНС надає можливість тією чи іншою мірою настроїти систему на вимоги викладача;
- застосування однієї й тієї ж АНС для різних курсів приводить до уніфікації інтерфейсу й, як наслідок, до зменшення часу на вивчення правил роботи із системою, затрачуване тими, кого навчають.

Визначимо, що розуміють під терміном автоматизована навчальна система, і якими ознаками вона повинна характеризуватися.

### **Поняття автоматизованої навчальної системи**

Через відсутність однозначної класифікації й сформованої термінології до 2010 р. не склалося єдиної думки щодо визначення автоматизованих навчальних систем. Існує множина визначень АНС. Простежимо за еволюцією цих визначень із 60-х років ХХ століття й до наших днів.

На початку робіт зі створення навчальних систем відповідно до глобальної мети їхнього створення під АНС розумівся комплекс програмних і апаратних засобів, призначених для автоматизації процесу навчання [6] (1963).

Пройшло десятиліття, і розуміння суті АНС змінилося. Як підтвердження можна привести роботу [6] (1976), у якій АНС розглядається як взаємозалежний комплекс засобів інформаційного, математичного й програмного забезпечення, організований на базі ЕОМ і призначений для керування процесом навчання.

Ще десятьма роками пізніше (1985) АНС визначалися вже як функціонально взаємозалежний набір підсистем учбово-методичного, інформаційного, математичного й інженерно-технічного забезпечення на базі засобів обчислювальної техніки, призначений для оптимізації процесу навчання в різних його формах і працюючий у діалоговому режимі колективного користування [6].

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРБ-123.23.0007.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 7    |

До теперішнього часу сформувалося відношення до АНС як до "організованого на базі ЕОМ комплексу засобів технічного, лінгвістичного, учбово-методичного й програмного забезпечення, призначеного для діалогової навчальної взаємодії й утворюючу програмну оболонку, пристосовану для заповнення навчальним матеріалом користувачем-непрограмістом" [6] (1998).

Як бачимо, у різний час давалися схожі визначення АНС, але розставлялися різні акценти. Зсув цих акцентів – від автоматизації процесу навчання до керування навчанням, потім до оптимізації навчання, і, зрештою, до створення оболонок, що підтримують діалог з тими, кого навчають – відбиває зміну розуміння місця й ролі навчальних систем в освітньому процесі.

Підсумовуючи все вищесказане й з огляду на тенденцію, що намітилася, повернення до раніше поставленої мети автоматизації навчання, можна дати ще одне визначення: АНС – це інструментальний комплекс, що включає математичні, методологічне й програмне забезпечення й призначений для заповнення знаннями користувачем-непрограмістом з метою створення навчальної системи й, в остаточному підсумку, реалізації автоматизованого навчання.

З обліком хмарності й на підставі розглянутої схеми навчання систематизуємо вимоги, яким повинна відповідати ХАНС.

### **Вимоги, пропоновані до ХАНС**

Основні вимоги, пропоновані до ХАНС, можна умовно розбити на дві групи:

#### **1. Вимоги, обумовлені хмарним характером системи:**

- організація прозорої взаємодії користувачів і системи через обчислювальну мережу;
- можливість об'єднання різномірних програмних засобів, хмарних у мережі, для рішення поставлених завдань;
- підтримка хмарних даних.

#### **2. Вимоги, обумовлені функціональним призначенням АНС:**

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРБ-123.23.0007.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 8    |

- облік мети навчання й завдання критерію досягнення цієї мети;
- можливість заповнення АНС знаннями по предметній області;
- забезпечення каналу зворотного зв'язка з тими, кого навчають;
- адаптація системи до того, кого навчають.

Для виконання цих вимог ХАНС повинна:

- мати засоби організації віддаленого доступу до системи;
- включати засоби міжпроцесної взаємодії;
- містити інструментарій, що дозволяє настроїти ХАНС на предметну область шляхом внесення в неї прикладних знань;
- мати засоби попереднього визначення рівня підготовки того, кого навчають і налаштування системи на певний рівень кваліфікації;
- містити засоби створення й використання різних стратегій навчання й різноманітних моделей того, кого навчають;
- включати засоби контролю процесу навчання;
- підтримувати різні форми й види навчання (лекції, лабораторні й контрольні роботи, практичні заняття й т.п.).

Реалізація цих базових вимог представляє складний і трудомісткий процес, тому навіть найбільш передові АНС відповідають лише деяким із зазначених вимог [6].

Існуючі в цей час реально діючі АНС не можуть характеризуватися як хмарні, так як якнайбільше, що вони забезпечують, – це можливість віддаленого доступу до навчального сервера [6]. Тому при аналізі існуючих АНС має сенс ураховувати тільки вимоги, обумовлені функціональним призначенням навчальних систем.

Ті ж системи, які можна зарахувати в розряд хмарних, не можуть вважатися автоматизованими навчальними системами. Наприклад, Центр дистанційного утворення МГІЙМ функціонує на основі системи Lotus Notes. Фактично це система документообігу з можливістю обміну інформацією з електронної пошти, постачена засобами автоматизованого контролю знань. Але

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРБ-123.23.0007.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 9    |

результати контролю знань того, кого навчають не впливають на поведження системи. Очевидно, що цих функцій недостатньо для того, щоб називатися АНС.

Більше того, переважна більшість систем дистанційного утворення, що працюють в Internet, взагалі не можна віднести до розряду навчальних систем, так як вони "є нічим іншим, як просто мережею статичних гіпертекстових сторінок" [6].

### **Основні напрямки досліджень**

У цей час в усьому світі активно ведуться роботи з дослідження й розробки АНС [6]. Основні напрямки цих досліджень:

- створення нових форм подання й способів зберігання знань (навчального матеріалу), стратегій активації й використання цих знань;
- розробка формальних і когнітивних моделей придбання знань;
- формування моделей поведження тих, кого навчають (student modelling);
- створення нових стратегій навчання студентів і вивчення навчального матеріалу (teaching and learning strategies).

Таким чином, виходячи з вищеперерахованого, програмне забезпечення системи підсистеми контролю знань в хмарних навчаючих системах, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти.

|      |      |          |        |      |                                  |           |
|------|------|----------|--------|------|----------------------------------|-----------|
|      |      |          |        |      | <b>ВКРБ-123.23.0007.00.00.ПЗ</b> | Арк.      |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | <b>10</b> |

## 2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

### 2.1 Огляд існуючих систем, технологій, архітектур, програмних рішень за профілем теми випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти

На цей час є багато вітчизняних інструментальних систем загального призначення, призначених для створення навчальних програм: КАДІС, АДОПІС, КОБРА, УРОК, АНСМІКРО, СЦЕНАРІЙ, "Наставник" і інші, порівнянних по функціональних можливостях з такими закордонними ІС як: Costoc, LinkWay, Quest, TenCore [2].

Проаналізуємо й порівняємо деякі використовувані в цей час інструментальні оболонки навчальних систем (те, що прийнято називати автоматизованими навчальними системами).

Як об'єкти порівняння були обрані наступні системи: КАДІС [1]; АНСМІКРО [2]; "ЕКСТЕРН" [3]; КОБРА [3]; "Фея" [5]. Ці системи, з одного боку, одержали певне поширення за рамки кола розроблювачів, і, з іншого боку, демонструють різні підходи до рішення основних завдань, що стоять перед навчальними системами.

Результати порівняння цих систем наведені в таблиці 2.1. Аналіз показав такі переваги цих систем, як:

- розвинені можливостями подання інформації;
- наявність засобів налаштування системи;
- можливість віддаленого доступу (по мережі);
- різноманітні форми подання й аналізу відповідей навчаємого;
- можливість збору статистики навчання.

Таблиця 2.1 – Інструментальні оболонки навчальних систем

| Властивості                                       | КАДІС                         | ЕКСТЕРН                                    | КОБРА         | АНСМІКРО                                   | Фея                                 |
|---|-------------------------------|--|---------------|--|-------------------------------------|
| Методологія керування навчанням                   | Праці В.П. Беспалько [2]      | Заснована на ПЗ, програма авторською мовою | Відсутня      | Заснована на ПЗ, програма авторською мовою | Програма візуально або мовою Pascal |
| <b>1. Віддалений доступ</b>                       | Немає                         | Так  | Так, розвитий | Так  | Так                                 |
| <b>Загальні якісні показники</b>                  |                               |  |               |  |                                     |
| <b>2. Керування навчанням</b>                     |                               |  |               |  |                                     |
| Облік мети навчання                               | Так                           | Так  | Немає         | Немає                                      | Програма <sup>2</sup>               |
| Критерій досягнення мети                          | Задано, можна міняти значення | Задано, можна міняти значення              | Немає         | Немає                                      | Програма мовою Pascal               |
| Сценарій курсу                                    | Так                           | Так  | Так           | Так  | Так                                 |
| Адаптація до того, кого навчають                  | Немає                         | Програма авторською мовою                  | Немає         | Немає                                      | Програма мовою Pascal               |
| <b>3. Зворотний зв'язок з тими, кого навчають</b> |                               |  |               |  |                                     |
| Облік реакції навчаємого на дії системи           | Так, програма візуально       | Так, Запрограмований                       | Немає         | Немає                                      | Програма візуально                  |

Продовження таблиці 2.1

| Властивості   | КАДІС                                 | ЕКСТЕРН                   | КОБРА                  | АНСМІКРО                                      | Фея   |
|---|---------------------------------------|---------------------------|------------------------|---|---|
| Контекстна допомога   | Програма візуально                    | Програма авторською мовою | Включена у файли даних | Програма авторською мовою                     | Програма візуально                            |
| Контроль знань  | Так                                   | Так                       | Так                    | Так   | Так   |
| <b>4. Можливості налаштування</b>                             |                                       |                           |                        |   |   |
| На предметну область  | Так                                   | Так                       | Так                    | Так   | Так   |
| На викладача (зміна моделей і методів навчання)               | Обмежена модифікація методів навчання | Програма авторською мовою | Немає                  | Ні, але може бути запрограмована мовою Pascal | Ні, але може бути запрограмована мовою Pascal |
| На навчаємого (адаптація до рівня підготовки й мети навчання) | Немає                                 | Програма авторською мовою | Немає                  | Ні, але може бути запрограмована мовою Pascal | Ні, але може бути запрограмована мовою Pascal |

Продовження таблиці 2.1

| Властивості                               | КАДІС                   | ЕКСТЕРН                                     | КОБРА                                | АНСМІКРО                                 | Фея                          |
|---|-------------------------|---|--------------------------------------|--|------------------------------|
| <b>Додаткові характеристики</b>           |                         |   |                                      |  |                              |
| <b>5. Типи організації навчання:</b>      |                         |   |                                      |  |                              |
| Ліберальний                               | Так                     | Так   | Так                                  | Так                                      | Програма                     |
| Консервативний                            | Так                     | Так   | Так                                  | Так                                      | Програма                     |
| Адаптивний                                | Немає                   | Програма авторською мовою                   | Немає                                | Немає                                    | Програма мовою Pascal        |
| <b>6. Модель навчаємого</b>               | Немає                   | Так (фіксуємий алгоритм і набір параметрів) | Немає                                | Немає                                    | Немає                        |
| <b>7. Предметна область</b>               |                         |   |                                      |  |                              |
| Модель предметної області                 | У вигляді графа         | Немає                                       | Немає                                | Немає                                    | Немає                        |
| Принцип організації навчального матеріалу | Гіпертекст і гіпермедіа | Набір інформаційних і контрольних кадрів    | Файли форматів doc, txt, htm, gif... | Набір інформаційних і контрольних кадрів | Система меню, вікон і кадрів |
| Місце зберігання навчального матеріалу    | База даних (БД) системи | БД системи                                  | БД системи                           | БД системи й зовнішні файли              | БД системи                   |

Продовження таблиці 2.1

| Властивості                            | КАДІС                                      | ЕКСТЕРН  | КОБРА                         | АНСМІКРО   | Фея  |
|--|--|--|-------------------------------|--|--|
| <b>8. Контроль знань</b>               |  |  |                               |  |  |
| Тестування                             | Так  | Так  | Так                           | Так  | Так  |
| Адаптивний контроль                    | Немає                                      | Програма авторською мовою                                | Немає                         | Програма (мовою Pascal)  | Програма мовою Pascal  |
| Контроль із визначенням причин помилок | Немає                                      | Немає  | Немає                         | Програма (мовою Pascal)  | Програма мовою Pascal  |
| Реалізовані схеми контролю знань       | ФЧВ, квитки й гранична схема               | Програма авторською мовою                                | Фіксований тест               | Програма (на авт. мові)  | ФЧВ  |
| Алгоритм проведення контролю           | Фіксований                                 | Фіксовано або програма авторською мовою                  | Фіксований                    | Фіксований або програма  | Фіксований або програма  |
| Типи відповідей                        | Вибіркові, конструюемі й числові відповіді | Вибіркові, конструюемі й вибірково-конструюемі відповіді | Вибіркові й числові відповіді | Мат. І хім. Формули, Текст, графіки, структурні схеми, вектора й ін. | Рядок символів, вибіркові (вибір номера відповіді або області на екрані) |

Продовження таблиці 2.1

| Властивості   | КАДІС  | ЕКСТЕРН   | КОБРА       | АНСМІКРО  | Фея                   |
|---|--|---|-------------|---|-----------------------|
| Аналіз відповідей   | Вимагає перерахування всіх можливих еталонів | Перерахування всіх можливих еталонів або шаблон відповіді | Примітивний | Розвитий (можна використовувати шаблони відповідей) | Примітивний           |
| Підсистема введення відповідей                            | Проста                                       | Проста  | Проста      | Складна   | Проста                |
| <b>9. Статистика</b>                                      | Так  | Так   | Так         | Так   | Так                   |
| <b>Додаткові характеристики АНС як програмного засобу</b> |  |   |             |   |                       |
| <b>10. Параметри</b>                                      |  |   |             |   |                       |
| Вбудовані   | Так  | Так   | Так         | Так   | Так                   |
| Обумовлені користувачем                                   | Немає  | Немає   | Немає       | Так   | Немає                 |
| <b>11. Необхідність програми</b>                          | Візуальне програма                           | Програма авторською мовою                                 | Немає       | Візуальне програма                                  | Програма мовою Pascal |
| <b>12. Підключення зовнішніх модулів</b>                  | Так, через стандартний інтерфейс             | Немає   | Так         | Так   | Так                   |

– Якщо не зазначена мова програмування, а система допускає різні можливості (наприклад, авторська мова й Pascal), значить можна використовувати й ту, і іншу мову.

– ФВЧ – фіксоване число питань, довільна вибірка питань із тесту.

Але ці системи не позбавлені недоліків:

1. Використовувана методологія навчання або програма (ЕКСТЕРН, "Фея"), або вже закладена в алгоритм функціонування системи (КАДІС, КОБРА, АНСМІКРО), причому обґрунтування цієї методології відсутній у всіх розглянутих системах, за винятком системи КАДІС.

2. Можливості організації адаптивного керування навчанням відсутні (КАДІС, КОБРА, АНСМІКРО) або обмежені необхідністю програмування (ЕКСТЕРН, "Фея").

3. Відсутність моделі навчаємого звужує можливості системи по адаптації до того, кого навчають. Модель навчаємого є тільки в системі ЕКСТЕРН, але алгоритм її формування запрограмований і заснований на статистичному наборі вбудованих параметрів.

4. Можливості організації адаптивного контролю відсутні (КАДІС, КОБРА) або обмежені необхідністю програмування (ЕКСТЕРН, "Фея", АНСМІКРО).

5. Тільки система КАДІС підтримує модель предметної області (у вигляді графа), але при викладі навчального матеріалу однаково зводить цю модель до списку тем (навчальних елементів).

6. Всі розглянуті системи є "річчю в собі", тобто вимагають зберігання навчальних і контролюючих матеріалів (тестів) у базі дані системи, не забезпечуючи переносимість матеріалів в іншу оболонку.

7. Форми подання відповідей навчаємого у кожній системі обмежені, методи визначення правильності відповідей запрограмовані й не підлягають зміні.

Не можна не відзначити, що всі складові частини АНС у тому або іншому виді реалізовані хоча б в одній з розглянутих систем. Але немає системи, що відповідає б всім вимогам, пропонованим до АНС.

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРБ-123.23.0007.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 17   |

Аналіз існуючих навчальних систем показує, що переважна більшість їх є електронними підручниками, доповненими, у найкращому разі, системами тестового контролю знань [6]. Крім цього, є невелика кількість навчальних систем (наприклад, [6]), які направляють процес навчання не по заздалегідь написаному сценарію, а використовують у своїй роботі знання викладача (звичайно, у вигляді експертної системи) і зворотний зв'язок з тими, кого навчають. Але ці системи написані під конкретну предметну область.

На жаль, нові науково-методичні розробки слабо відбиваються на реальних програмних продуктах. Аналіз запропонованих продуктів і технологій виявив наступну ситуацію:

– серед множини програмних продуктів не було ні однієї системи, що здійснювала б повну підготовку того, кого навчають хоча б по одній дисципліні;

– всі програмні продукти або висвітлюють якийсь фрагмент предметної області, або виконують одну (у найкращому разі – декілька) функцій навчальної системи, або призначені для формування окремих умінь і навичок рішення завдань;

– у більшості програмних продуктів (за винятком інструментальних засобів КОНСПЕКТ, АССОЛЬ, ПА9) не передбачене налаштування системи на довільну предметну область і вимоги викладача, тобто навчальний матеріал включений до складу системи й не підлягає зміні, а послідовність його подачі запрограмована;

– жодна із представлених навчальних програм не містить моделі того, кого навчають, тобто індивідуалізація процесу навчання здійснюється тільки за рахунок індивідуальної роботи того, кого навчають із програмою.

#### **Системи контролю знань**

У цей час існує велика кількість систем контролю знань (СКЗ), як виконаних у вигляді окремих програмних продуктів (наприклад, ITEMAN, RASCAL, RSP, The Examiner testing system, FastTEST professional, C-Quest, CONTEST, ГРАММАТЕЙ-КЛАС, ПЗЛСТАР, "Контроль знань", "Екзаменатор",

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРБ-123.23.0007.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 18   |

"Атестація"), так і вбудованих у навчальні системи (наприклад, у всіх АНС передбачений контроль знань). Відсутність стандарту або хоча б загальноприйнятих норм і правил написання таких систем привело до того, що практично кожна з них є "річчю в собі" і використовується, за рідкісним винятком, тільки її власними розроблювачами.

Аналіз існуючих СКЗ виявив наступні недоліки цих систем:

1. Обмежена кількість типів відповідей. Не у всіх системах реалізовані навіть основні форми подання відповідей [1].

2. Відсутність формальних методів диференційованої оцінки відповідей. Існують тестові системи, у яких оцінка може бути диференційованою, але це реалізується неформальними способами:

– задається кілька еталонів, один із яких є абсолютно правильним, а інші – частково правильними (для них вказується більше низький бал, чим для абсолютно правильного еталона);

– тому, кого навчають, надається  $N$  спроб відповіді: якщо він відразу дав правильну відповідь, вона одержує максимальний бал, якщо з  $i$ -ї спроби ( $i < N$ ), то оцінка зменшується пропорційно значенню  $i$ .

3. Слабкі можливості параметричного налаштування тесту.

4. Відсутність відкритої архітектури. Закритість систем обумовлюється:

– зберіганням тестів у базі дані системи;

– відсутністю можливості підключення зовнішніх модулів для подання питань, одержання відповідей і оцінки відповідей;

– обмеженою кількістю алгоритмів формування тесту й керування проведенням тестування.

5. Багато які СКЗ надають можливості віддаленого тестування (через Internet), але можливості таких систем ще більш обмежені в порівнянні з локальними СКЗ.

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-123.23.0007.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 19   |

## 2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

Для написання програмного забезпечення підсистеми контролю знань в хмарних навчаючих системах була обрана мова програмування PHP.

PHP – це мова серверних скриптів (server scripting language), що вбудовується в HTML, який інтерпретується і виконується на сервері. PHP працює як частина Web-сервера, і цим самим схожий на ASP від Microsoft або Coldfusion від Allaire. Синтаксис PHP дуже схожий на синтаксис таких мов програмування C або Perl. Люди, що мають деякий досвід програмування, дуже швидко зможуть почати писати програми на PHP. У цій мові немає строгої типізації даних і немає необхідності в діях з виділення/звільнення пам'яті. Програми, написані на мові програмування PHP, читаються достатньо легко. На відміну від Perl-программ PHP-коду властива легка читабельність та зрозумілість.

PHP є препроцесором HTML. До того, як сервер відправить файл браузеру, його проглядає препроцесор-інтерпретатор. Для того, щоб це відбувалося, файли, які піддаються обробці препроцесором, повинні мати визначене розширення (звичайно це .phtml або .php3, але ці значення можна поміняти) і містити (хоча це не обов'язкова вимога) код для препроцесора. Перед відправкою сторінки PHP-код програється на сервері і браузеру видається результат у вигляді знову таки HTML-сторінки, яка може сильно відрізнятись від тієї, що зберігається на сервері. Звичайні ж сторінки, що мають розширення .html/.htm Web-сервер буде відправляти браузеру без будь-якої обробки. Основна відмінність від CGI-скриптів, написаних на інших мовах, типу Perl або C – це те, що в CGI-програмах розробник самостійно пише HTML-код, що виводиться, а, використовуючи PHP – вбудовує свою програму-скрипт в готову HTML-сторінку, використовуючи відкриваючий і закриваючий теги (<?php та ?>). PHP називається мовою серверних скриптів на відміну від Javascript/jscript/vbscript, які є мовами клієнтських скриптів. Це означає, що PHP-скрипт виконується на

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-123.23.0007.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 20   |

сервері, а клієнтові передається результат його роботи, тоді як в JavaScript , код повністю передається на клієнтську машину і лише там виконується браузером.

При написанні на інших мовах програмування, наприклад, на Perl або C – замість того, щоб створювати програму, яка займається формуванням HTML-коду і містить незліченну безліч призначених для цього команд, PHP-програміст має можливість створювати HTML-код з декількома упровадженими командами PHP. Код PHP відділяється спеціальними початковим і кінцевим тегами, які дозволяють процесору PHP визначати початок і кінець ділянки HTML-коду, що містить PHP-скрипт.

Значною відмінністю PHP від якого-небудь коду, що виконується на стороні клієнта, наприклад, Javascript, є те, що PHP-скрипти виконуються на сервері. PHP-скрипт розміщений на сервері, клієнт отримує тільки результат виконання скрипта, причому клієнт не має можливості з'ясувати, який саме код виконується. Існує також можливість конфігурувати власний сервер так, щоб HTML-файли оброблялися процесором PHP, так що клієнти навіть не в змозі дізнатися, чи отримують вони звичайний HTML-файл, чи це результат виконання скрипта.

Мова програмування PHP досить проста для освоєння, але разом з тим здатна задовольнити запити професійних програмістів. Хоча PHP, головним чином, призначений для роботи в середовищі web-серверів, область його застосування не обмежується тільки цим.

Можливості мови PHP дуже великі. Головним чином, область застосування PHP сфокусована на написання скриптів, що працюють на стороні сервера; таким чином, PHP здатний виконувати все те, що виконує будь-яка інша програма CGI, наприклад, обробляти дані форм, генерувати динамічні сторінки або посилати і приймати cookies. Окрім вищенаведеного PHP здатний виконувати і безліч інших завдань.

Існують три основні області, де використовується PHP.

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРБ-123.23.0007.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 21   |

1. Створення скриптів для виконання на стороні сервера. Саме таким чином мова програмування PHP найширше використовується. Все, що може знадобитися програмісту – це парсер PHP (у вигляді програми CGI або серверного модуля), вебсервер і браузер. Для того, щоб існувала можливість проглядати результати виконання PHP-скриптів у браузері, потрібен працюючий веб-сервер і встановлений PHP.

2. Створення скриптів для виконання в командному рядку. Існує можливість створити PHP-скрипт, здатний запускатися незалежно від веб-серверу та браузера. Все, що потрібно – це парсер PHP. Такий спосіб використання PHP ідеально підходить для скриптів, які повинні виконуватися регулярно, наприклад, за допомогою cron (на платформах Unix або Linux) або за допомогою планувальника завдань (Task Scheduler) на платформах Windows. Ці скрипти також можуть бути використані в завданнях простої обробки даних.

Існує величезна кількість документації і списків розсилки, до яких можна звернутися у разі виникнення яких-небудь питань.

### **Обґрунтування щодо вибору методу представлення та збереження даних у системі. Опис СУБД**

У програмному забезпеченні, що розробляється при виконанні бакалаврської роботи інформація представляється у вигляді даних, які зберігаються в пам'яті ЕОМ у вигляді організованої структури – бази даних.

База даних є структурованою сукупністю даних. Для запису, вибірки і обробки даних, що зберігаються в комп'ютерній базі даних, необхідна система управління базою даних, якою і є ПЗ MySQL. Оскільки комп'ютери чудово справляються з обробкою великих об'ємів даних, управління базами даних грає центральну роль в обчисленнях. Реалізовано таке управління може бути різноманітним – як у вигляді окремих утиліт, так і у вигляді коду, що входить до складу інших застосувань.

MySQL – це система управління реляційними базами даних.

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРБ-123.23.0007.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 22   |

У реляційній базі даних дані зберігаються та можуть бути представлені у вигляді різних таблиць та між ними можуть бути сформовані зв'язки. Таблиці зв'язуються між собою за допомогою відносин, завдяки чому забезпечується можливість об'єднувати при виконанні запиту дані з декількох таблиць. SQL як частину системи MySQL можна охарактеризувати як мову структурованих запитів плюс найбільш поширена стандартна мова, використовувана для доступу до баз даних.

Завдяки існуючим зв'язкам в реляційних БД існують такі переваги :

- інформація не дублюється;
- легко проводити зміни в таблиці;
- в не реляційних БД важко передати всі залежності, тобто зв'язати один з одним дані з різних таблиць;
- існує можливість запобігти помилковим зв'язкам між різними даними із таблиць, а необхідний об'єм пам'яті зменшується до мінімуму.

Взаємодія із програмним забезпеченням відбувається за допомогою використання серверу баз даних MySQL.

MySQL – це невеликий, компактний багато потоковий сервер баз даних. MySQL характеризується великою швидкістю, стійкістю і легкістю у використанні.

Розробку і супровід MySQL, найпопулярнішої SQL-базы даних з відкритим кодом, здійснює компанія MySQL AB, що надає різні сервіси для СУБД MySQL. Програмне забезпечення MySQL – це ПЗ з відкритим кодом.

Це означає, що застосовувати і модифікувати його може будь-який охочий. Таке ПЗ можна отримувати по Internet і використовувати безкоштовно. При цьому кожен користувач може вивчити початковий код і змінити його відповідно до своїх потреб. Тобто MySQL-сервер є безкоштовним для некомерційного використання.

MySQL є ідеальним рішенням для малих і середніх застосувань. Початкові тексти сервера компілюються на безлічі платформ. Найбільш повно можливості

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРБ-123.23.0007.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 23   |

сервера виявляються на Unix-серверах, де є підтримка багато-поточності, що дає значний приріст продуктивності. У варіанті під Windows, MySQL може запускатися як звичайний процес на Windows.

MySQL підтримує мову запитів SQL в стандарті ANSI 92, і окрім цього має безліч розширень до цього стандарту, якого немає ні в одній іншій СУБД.

Короткий перелік можливостей MySQL:

1. Підтримується необмежена кількість користувачів, що одночасно працюють з базою даних.
2. Кількість рядків в таблицях може досягати 50 млн.
3. Швидке виконання команд. Можливо MySQL найшвидший сервер з тих, що існують.
4. Проста і ефективна система безпеки.

MySQL дійсно дуже швидкий сервер, але для досягнення цього розробникам довелося пожертвувати деякими вимогами до реляційних СУБД. У MySQL відсутні:

1. Не реалізована підтримка транзакцій. Натомість пропонується використовувати Lock/unlock TABLE.
2. Немає підтримки зовнішніх (foreign) ключів.
3. Немає підтримки тригерів і процедур, що зберігаються.
4. Немає підтримки уявлень (VIEW).

MySQL є дуже швидким, надійним і легким у використанні. Спочатку сервер MySQL розроблявся для управління великими базами даних з метою забезпечити вищу швидкість роботи в порівнянні з аналогами, що існували на той момент. І ось вже протягом декількох років даний сервер успішно використовується в умовах промислової експлуатації з високими вимогами. Не дивлячись на те що MySQL постійно удосконалюється, він вже сьогодні забезпечує широкий спектр корисних функцій. Завдяки своїй доступності, швидкості і безпеці MySQL дуже добре підходить для доступу до баз даних по Internet.

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-123.23.0007.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 24   |

ПЗ MySQL є системою клієнт-сервер, яка містить багато потоковий SQL-сервер, що забезпечує підтримку різних обчислювальних машин баз даних, а також декілька різних клієнтських програм і бібліотек, засоби адміністрування і широкий спектр програмних інтерфейсів (API). Також поставляється сервер MySQL у вигляді багато потокової бібліотеки, який можна підключити до призначеного для користувача застосування і отримати компактний, швидший і легший в управлінні продукт.

Доступно також велика кількість програмного забезпечення MySQL розробленого сторонніми розробниками. СУБД MySQL підтримується мовою програмування PHP.

### 2.3 Розгорнута постановка завдання

Згідно з технічним завданням на випускню кваліфікаційну роботу за першим (бакалаврським) рівнем вищої освіти, реалізації підлягає програмне забезпечення, яке призначено для системи підсистеми контролю знань в хмарних навчаючих системах.

В процесі розробки випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти необхідно виконати наступний обсяг роботи:

а) провести аналіз існуючих систем-аналогів для виявлення їх позитивних і негативних якостей. Результати аналізу врахувати в подальших розробках;

б) вибрати та обґрунтувати методику побудови системи контролю роботи технологічного обладнання на виробництві в автоматизованому режимі. Розробити функціональну та структурну схеми системи;

в) розробити програмне забезпечення системи, що дозволить реалізувати поставлену технічним завданням задачу. Побудувати блок-схеми алгоритмів програми та підпрограми;

г) організувати інтерфейс користувача з метою формування та виводу на екран ЕОМ повідомлень про некоректні дії користувача та нестандартні ситуації

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРБ-123.23.0007.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 25   |

в роботі технологічного обладнання;

д) розробити рекомендації по організаційних та методичних заходах, які забезпечать впровадження системи в промислову експлуатацію та її подальшу успішну експлуатацію;

е) провести розрахунки по визначенню економічної ефективності розробленої системи;

ж) розробити заходи по охороні праці при впровадженні та експлуатації системи, а також розробити заходи з цивільного захисту;

з) сформулювати висновки про виконаний обсяг робіт та одержані результати.

ВКРБ-123.23.0007.00.00.ПЗ

Арк.

26

| Вим. | Арк. | № докум. | Підпис | Дата |
|------|------|----------|--------|------|
|      |      |          |        |      |

## 3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

### 3.1 Опис функціонування системи

#### Опис системи контролю знань

Хмарна автоматизована навчальна система повинна мати наступні властивості.

З одного боку, бути якнайближче до традиційного індивідуального навчання з викладачем. Для цього система повинна підтримувати зворотний зв'язок з тим, кого навчають, мати здатність адаптуватися до його рівня й потреб, об'єктивно оцінювати його знання. З іншого боку, така система повинна включати ті можливості, які надає обчислювальна техніка: різні форми подання інформації, моделювання процесів і явищ, використання електронних довідників, доступ до більших обсягів інформації.

#### Створення мережного середовища ХАНС

Мережне середовище хмарної навчальної системи містить у собі локальну обчислювальну мережу й глобальну (Internet). Опишемо базову технологію доступу в Інтернет. Мережі ISDN здатні багато в чому вирішити проблеми доступу в мережу Internet [6].

**ISDN (Integrated Services Digital Network)** – цифрова мережа з інтеграцією служб. Дозволяє сполучити послуги телефонного зв'язку й обміну даними.

Основне призначення ISDN – передача даних зі швидкістю до 64 кбіт/с по 4-кілогерцовій провідній лінії й забезпечення інтегрованих телекомунікаційних послуг (телефон, факс, та ін.). Використання для цієї мети телефонних проводів має дві переваги: вони вже існують і можуть використовуватися для подачі живлення на термінальне встаткування. Вибір 64 кбіт/с стандарту визначається наступними міркуваннями. При смузі частот 4 кГц, відповідно до теореми Котельникова, частота дискретизації повинна бути не нижче 8 кГц. Мінімальне

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРБ-123.23.0007.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 27   |

число двійкових розрядів для подання результатів стробування голосового сигналу за умови логарифмічного перетворення дорівнює 8. Таким чином, у результаті перемножування цих чисел і виходить значення смуги В-каналу ISDN. Базова конфігурація каналів має вигляд  $2 \times B + D = 2 \times 64 + 16 = 144$  кбіт/с. Крім В-каналів і допоміжного D-каналу ISDN може запропонувати й інші канали з більшою пропускну здатністю: канал Н10 зі смугою 384 кбіт/с, Н11 – 1536 і Н12 – 1920 кбіт/с (реальні швидкості цифрового потоку). Для первинних каналів (1544 і 2048 кбіт/с) смуга D-каналу може становити 64 кбіт/с. Для об'єднання в мережі ISDN різних видів трафіка використовується технологія TDM (Time Division Multiplexing, мультіплексування за часом). Для кожного типу даних виділяється окрема смуга, що називається елементарним каналом (або стандартним каналом). Для цієї смуги гарантується фіксована, погоджена частка смуги пропускання. Виділення смуги відбувається після подачі сигналу CALL по окремому каналі, що називається каналом позаканальної сигналізації. У стандартах ISDN визначаються базові типи каналів, з яких формуються різні користувальницькі інтерфейси.

Таблиця 3.1 – Опис базових типів каналів стандарту ISDN

| Тип | Смуга        | Опис  |
|-----|--------------|---|
| A   | —            | Аналогова телефонна лінія, 4кГц.                                  |
| B   | 64 кбіт/с    | передача даних або 1 телефонна лінія (1 потік оцифрованого звуку) |
| C   | 8/16 кбіт/с  | передача даних  |
| D   | 16/64 кбіт/с | Канал позаканальної сигналізації (керування іншими каналами)      |
| E   | 64 кбіт/с    | Внутрішня сигналізація ISDN                                       |
| H0  | 384 кбіт/с   | передача даних  |
| H10 | 1472 кбіт/с  | передача даних  |
| H11 | 1536 кбіт/с  | передача даних  |
| H12 | 1920 кбіт/с  | передача даних  |

У більшості випадків застосовуються канали типів В і D.

Із зазначених типів каналів формуються інтерфейси, найбільше поширення одержали наступні типи:

### **Інтерфейс базового рівня**

Basic Rate Interface (BRI) – інтерфейс абонента, що надає для зв'язку апаратури, і ISDN-станції два В-канали й один D-канал. Інтерфейс базового рівня описується формулою  $2B+D$ . У стандартному режимі роботи BRI можуть бути одночасно використані обоє В-канали (наприклад, один для передачі даних, інший для передачі голосу) або один з них. При одночасній роботі каналів вони можуть забезпечувати з'єднання з різними абонентами. Максимальна швидкість передачі даних для BRI інтерфейсу становить 128кб/с. D-канал використовується тільки для передачі керуючої інформації. У режимі AO/DI (Always On/Dynamic ISDN) смуга 9600 біт/с D-каналу використовується в якості постійно включеного виділеного каналу X.25, як правило, що підключається до Інтернет. При необхідності, використовується для доступу до Інтернет смуга розширюється шляхом включення одного або двох В-каналів. Цей режим, хоча й стандартизований (під найменуванням X.31), але не знайшов широкого поширення. Для вхідних з'єднань BRI підтримується до 7 адрес (номерів) які можуть призначатися різними ISDN-пристроями, що розділяє одну абонентську лінію. Додатково, забезпечується режим сумісності зі звичайними, аналоговими абонентськими пристроями – абонентське встаткування ISDN, як правило, допускає підключення таких пристроїв і дозволяє їм працювати прозорим образом. Цікавим побічним ефектом такого «псевдоаналогового» режиму роботи стала можливість реалізації так званого симетричного X2 – модемні протоколи фірми US-Robotics, що дозволяли передачу даних поверх лінії ISDN в обидва боки на швидкості 56 кбіт/с.

Інтерфейс базового рівня BRI (англ. Basic Rate Interface) – забезпечує користувачеві надання двох цифрових каналів (ОЦК) по 64 кбіт/с (канал В) і однополосний канал сигналізації D зі швидкістю передачі даних 16 кбіт/с. Таким

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРБ-123.23.0007.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 29   |

чином, максимальна швидкість передачі в інтерфейсі BRI (2B+D) становить  $R_{bmax}=128+16=144$  кбіт/с [1] Найпоширеніший тип сигналізації – DSS1 (Euro ISDN). Використовується два магістральних режими портів BRI щодо станції або телефонів – S/TI й NT. Режим S/TI – порт емулює роботу ISDN телефону, режим NT – емулює роботу станції. Окреме доповнення – використання ISDN телефону з додатковим живленням у цьому режимі, так як стандартно не всі порти (і карти HFC) дають живлення по ISDN шлейфі – англ. inline power. Кожний із двох режимів може бути « точка-багатоточка» point-to-multi-point (PTMP) він же MSN, або «точка-точка» point-to-point (PTP). У першому режимі для пошуку адресата призначення на шлейфі використовуються номери MSN, які, як правило, збігаються з виділеними провайдером телефонії міськими номерами. Провайдер повинен повідомити передані їм MSN. Іноді провайдер використовує так звані «технічні номери» – проміжні MSN. У другому режимі BRI порти можуть поєднуватися в транк – умовну трубу, по якій передані номери можуть використовуватися в багатоканальному режимі.

ISDN технологія використовує три типи інтерфейсу BRI: U, S і T:

– U – одна кручена пара, прокладена від комутатора до абонента, що працює в повному або пол-дуплексі. До U-інтерфейсу можна підключити тільки 1 пристрій, називаний NT-1 (або NT-2) Network Termination – мережне закінчення.

– S/T інтерфейс (S0). Використовуються дві кручені пари, передача й прийом. Може бути обтиснута як в RJ-45 так і в RJ-11 гніздо/кабель. До гнізда S/T інтерфейсу можна підключити одним кабелем (шлейфом) за принципом шини до 7 ISDN пристроїв – телефонів, модемів, факсів, названих TE1 (Terminal Equipment 1). Кожний пристрій слухає запити в шині й відповідає на прив'язаний до нього MSN. Принцип роботи багато в чому схожий на SCSI.

– NT-1, NT-2 – Network Termination, мережне закінчення. Перетворює одну пару U в один (NT-1) або два (NT-2) 2-х парних S/T інтерфейси (з роздільними парами для прийому й передачі). По суті S і T це однакові на вид інтерфейси, різниця в тому, що по S інтерфейсі можна подати живлення для TE

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРБ-123.23.0007.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 30   |

пристроїв, телефонів наприклад, а по Т – немає. Більшість NT-1 і NT-2 перетворювачів уміють і те й інше, тому інтерфейси найчастіше називають S/T.

### **Інтерфейс первинного рівня**

Primary Rate Interface (PRI) – використовується для підключення до широкополосних магістралей, що зв'язують місцеві й центральні АТС або мережні комутатори. Інтерфейс первинного рівня поєднує 23 В-каналу й один D-канал для стандарту T1 23B+D (розповсюджений у Північній Америці і Японії); або 30 В-каналів і один D-канал для стандарту E1 30B+D (розповсюджений у Європі). Елементарні канали PRI можуть використовуватися як для передачі даних, так і для передачі оцифрованого телефонного сигналу.

Інтерфейс первинного рівня (Primary Rate Interface, PRI) – стандартний інтерфейс мережі ISDN, що визначає дисципліну підключення станцій ISDN до широкополосних магістралей, що зв'язують місцеві й центральні АТС або мережні комутатори. Інтерфейс первинного рівня поєднує 23 В-каналу й один D-канал для стандарту T1 ( $23B + D = 24 * 64 = 1536$  [kBit/s]) або 30 В-каналів для голосу або даних, один D-канал для сигналізації й один Н-канал для службових даних стандарту E1 ( $30B + D + H = 32 * 64 = 2048$  [kBit/s]).

### **Архітектура мережі ISDN**

Мережа ISDN складається з наступних компонентів:

- мережні термінальні пристрої (NT, Network Terminal Devices);
- лінійні термінальні пристрої (LT, Line Terminal Equipment);
- термінальні адаптери (TA, Terminal adapters);
- абонентські термінали.

Абонентські термінали забезпечують користувачам доступ до послуг мережі. Існує два види терміналів: TE1 (спеціалізовані ISDN-термінали), TE2 (неспеціалізовані термінали). TE1 забезпечує пряме підключення до мережі ISDN, TE2 вимагають використання термінальних адаптерів (TA).

Розглянувши стандарт ISDN, перейдемо до продовження опису розробленої системи.

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРБ-123.23.0007.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 31   |

Виділяють три варіанти підключення окремих комп'ютерів і ЛОМ до Internet. Для доступу в Internet окремих користувачів можна застосовувати ISDN BRI-адаптери, які встановлюються в стандартне гніздо шини персонального комп'ютера (ISA, PCI або PC-Card). Для зв'язку із провайдером звичайно використовується Point-to-Point Protocol (PPP), а для автентифікації вхідних у мережу користувачів – протоколи PAP і CHAP.

Крім внутрішніх адаптерів, існують зовнішні термінальні адаптери або зовнішній ISDN-модем, які призначені для конвертації послідовного інтерфейсу ПК, звичайного мосту/маршрутизатора або іншого не ISDN-пристрою, у формат ISDN BRI.

Для мереж ISDN можливе застосування активних або пасивних адаптерів. Активний адаптер побудований на основі процесора зі своєю оперативною пам'яттю й орієнтований на виконання комунікаційного програмного забезпечення. Він дозволяє значно менше використовувати ресурси ЦП файлового сервера. Пасивний адаптер ISDN аналогічний звичайному мережному адаптеру й використовує ресурси ЦП сервера. Звичайно активні адаптери дорожче пасивних, але й більше продуктивні.

Для підключення ЛОМ до Internet звичайно використовується маршрутизатор, що дозволяє розділити внутрішньої й зовнішню IP-мережі й здійснюючої функції брандмауера. Самі маршрутизатори можуть бути реалізовані програмним шляхом на серверах NetWare (IntranetWare), Windows або UNIX. Необхідно також програмне забезпечення типу NetWare Multiprotocol Router for ISDN, або аналогічне для Windows або UNIX.

Іншим варіантом є використання апаратного маршрутизатора, що виконаний у вигляді окремого пристрою й має один або кілька портів для підключення ЛОМ і один або трохи WAN-портів. Продуктивність і широта функціональних можливостей (зокрема, підтримка різноманітних протоколів) залежать від вартості маршрутизатора.

|      |      |          |        |      |  |                                  |      |
|------|------|----------|--------|------|--|----------------------------------|------|
|      |      |          |        |      |  | <b>ВКРБ-123.23.0007.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |  |                                  | 32   |

## Налаштування АНС

Налаштування АНС відбувається шляхом заповнення системи предметними даними, створення моделі предметної області, зміни значень параметрів у файлах конфігурації. Система підтримує вбудовані й користувальницькі параметри.

Для вбудованих параметрів визначені назви, алгоритми обробки й, у більшості випадків, множина припустимих значень. Ці параметри можна розділити на групи по принципах їхньої обробки:

1. Параметри з певними значеннями. Ці параметри можуть приймати значення з фіксованого списку значень, для кожного з яких передбачений свій алгоритм обробки. Наприклад, параметр СПОСІБ ВИДАЧІ для системи контролю знань зі значеннями ОДИН і ВСЕ. Якщо такий параметр не заданий користувачем у явному виді, йому привласнюється певне значення за замовчуванням.

2. Параметри з певним алгоритмом. Значення таких параметрів задаються користувачем довільно, і передбачений єдиний алгоритм обробки, що не залежить від значення. Наприклад, параметр PROCESS визначає командний рядок, що буде запущений при порушенні сінапса.

3. Параметри з довільними значеннями. Значення таких параметрів задають обмеження відповідних характеристик роботи системи. Наприклад, параметр TIMEOUT задає максимальний час відповіді на питання.

Користувальницькі параметри визначаються викладачем при заповненні системи, для них задається ім'я й вираження для розрахунку значення. Вираження може бути арифметичним або логічним, у якості операндів можуть виступати назви вбудованих і користувальницьких параметрів і імена функцій, визначених у системі. Ці параметри описують довільні характеристики складових предметної області й моделі того, кого навчають і призначені, зокрема, для використання в правилах бази знань.

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-123.23.0007.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 33   |

## Підсистема контролю знань

Ядро СКЗ оформлене у вигляді бібліотеки функцій. На основі цієї бібліотеки реалізовані два варіанти СКЗ:

- система TS, для якої написаний віконний інтерфейс під ОС Windows;
- система IPDO, що є консольним додатком і призначена для віддаленого тестування того, кого навчають в інтерактивному режимі з використанням мережі Internet.

## Особливості системи контролю знань

СКЗ характеризується наступними особливостями:

1. Реалізація основних форм подання відповідей, а саме: закрита (вибіркова); відкрита (числова і текстова); на впорядкування елементів; на відповідність (звичайно, елементів двох множин). Відповідь на закритий тест реалізована як множина, відповідь на впорядкування елементів – як список.

2. Множина типів відповідей розширено за рахунок того, що:

- як числова відповідь розглядається арифметичне вираження;
- використовуються дворівневі схеми, у яких елементами множин і списків є списки або множини, що дає широкі можливості по поданню зв'язків між елементами. Ці схеми:

а) дозволяють моделювати відповіді на відповідність, не обмежуючи кількість вихідних множин;

б) можуть використовуватися, наприклад, для аналізу синтаксичного опису графічних образів;

- для реалізації тих форм відповідей, які не включені в систему, передбачена можливість виклику зовнішніх процесів.

3. Використання диференційованих методів для оцінки відповідей.

4. Можливість параметричного налаштування, що полягає у виборі:

- способу формування тесту;
- способу оцінки тесту;
- способу оцінки відповіді.

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРБ-123.23.0007.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 34   |

## 5. Відкритість архітектури системи:

### а) зовнішня:

- використання декларативної мови опису тестів;
- зберігання описів тестів і завдань на тестування в зовнішніх файлах;
- можливість підключення зовнішніх модулів:
  - для подання питань;
  - для одержання відповідей того, кого навчають.

б) внутрішня, котра забезпечується можливістю підключення зовнішніх модулів:

- для одержання оцінок відповідей;
- для керування формуванням тесту;
- для керування проведенням тестування.

Подання текстових питань у вигляді шаблону здійснюється за допомогою спеціальної програми. Для одержання відповідей, подання яких виходить за рамки внутрішніх можливостей системи, були написані графічний редактор і редактор формул.

Зовнішнє керування роботою СКЗ дає можливість вибору довільних схем проведення тестування, у тому числі, адаптивних. Для організації зовнішнього керування підсистемою контролю знань створені бази знань, що реалізують:

- формування тесту на основі параметрів питань;
- тестування у відповідність із граничною й адаптивною схемами.

Для забезпечення переносимості тестів створена мова опису тестів [4].

### **Мова опису тестів**

#### **Вимоги, пропоновані до мови описи тестів**

Основні властивості, який повинен володіти мова опису контрольних завдань, що впливають:

- простота опису тесту;
- облік максимально великої кількості можливих варіантів відповідей і питань;

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРБ-123.23.0007.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 35   |

- переносимість створюваних тестів на різні платформи (на рівні їхніх текстів);
- можливість підключення й використання зовнішніх модулів (для демонстрації питань, одержання відповідей і для визначення правильності отриманих відповідей).

### **Формат мови опису тестів**

Для того щоб відокремити опис тесту від його алгоритмічної реалізації й зробити його стерпним, мова повинна бути декларативною.

Будь-яка система контролю знань містить у собі множину пропонованих питань, правильні відповіді на них і правила визначення коректності відповідей, отриманих у ході опитування. Тому опис тесту складається з наступних частин:

1. Опис питань.
2. Опис еталонів, методів і оцінок відповідей тих, кого навчають.
3. Опис правил формування тесту зі списку контрольних завдань і методів оцінки тесту в цілому.

За результатами аналізу систем контролю знань у третьому розділі були виділені наступні типи питань і відповідей:

- **Типи питань:** ТЕКСТ, КОМАНДА.
- **Типи відповідей:** ТЕКСТ, МНОЖИНА, СПИСОК, ВИРАЖЕННЯ.

Всі відповіді, подання яких виходить за рамки зазначених типів (рисунок, мовне повідомлення й ін.), можуть бути отримані в рамках процесу – командного рядка, зазначеного у відповідному питанні.

Опис тесту може містити дані по декількох темах. Він зберігається в текстовому файлі і являє собою послідовність блоків описів питань, відповідей і типів контролю. Кожна фраза (блок) починається з одного із ключових слів **КОНТРОЛЬ ПредмОбл ТЕМИ, СПИСОК ТЕМ, ПИТАННЯ й ВІДПОВІДЬ** і закінчується ключовим словом **КІНЕЦЬ**. У середині кожної фрази в довільному порядку розташовуються ключові слова, обумовлені типом фрази, і дані, що

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРБ-123.23.0007.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 36   |

відносяться до цих ключових слів. Необов'язкові параметри опису тесту можуть бути опущені (у даному нижче описі вони полягають у квадратні дужки).

Коментарі. Тип коментарю – рядковий. Як символ початку коментарю використовується символ вартий на початку рядка.

Роздільники. Роздільниками лексем є пробіл, кома, табуляція й символ перекладу рядка. Крім того, символи '=' і ':' (знак рівності й двочка) є спеціальними роздільниками; вони розглядаються як роздільники поряд із пробілами й табуляціями, але повинні стояти в місцях, визначених синтаксисом опису.

Ідентифікатор є довільною послідовністю символів довжиною не більше 255. Якщо ідентифікатор містить пробіли, він повинен бути укладений у подвійні лапки.

Система нечутлива до регістра: ідентифікатори й ключові слова можуть бути написані як прописними, так і малими літерами. Текст, укладений у подвійні лапки, сприймається так, як він записаний, тобто з урахуванням регістра.

Як дані система сприймає:

- ідентифікатори – послідовності символів, не утримуючих пробілів;
- рядки – довільні послідовності символів, укладені в подвійні лапки;
- вирази – розрахункові послідовності символів (констант і ідентифікаторів, що мають значення), розділені знаками арифметичних операцій.

#### Блок опису питання

ПИТАННЯ = <ім'я питання>

[ТЕМА = <тема>]

ТИП = <тип питання>

ЗМІСТ = <текст питання> | <командний рядок>

[ВЛАСТИВІСТЬ <ідентифікатор властивості 1> = "<значення 1>"

ВЛАСТИВІСТЬ <ідентифікатор властивості 2> = "<значення 2>"

ВЛАСТИВІСТЬ <ідентифікатор властивості n> = "<значення n>"]

КІНЕЦЬ;

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-123.23.0007.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 37   |

<ім'я питання> – ідентифікатор питання (назва).

<тема> – ідентифікатор теми (назва).

<тип питання> – одне із ключових слів: ТЕКСТ, КОМАНДА. Якщо тип питання ТЕКСТ, то на екран виводиться рядок <текст питання> (параметр ЗМІСТ). Якщо тип питання КОМАНДА, то запускається командний рядок.

ВЛАСТИВІСТЬ – характеристика питання, що вводиться й обумовлена користувачем: <ідентифікатор властивості і> – назва і-ої властивості, <значення і> – вираження, що породжує значення (довільно типу) цієї характеристики для даного питання. Властивості можуть бути використані:

- при формуванні списків питань для тестування;
- при створенні моделі того, кого навчають;
- для керування питаннями типу КОМАНДА

ТЕМА – оголошення теми може зустрічатися в будь-якому місці тексту. Залежно від місця розташування розрізняються глобальне й локальне оголошення теми. Глобальним вважається оголошення, розташоване безпосередньо перед початком фрази, його дія поширюється на всі питання до наступного оголошення (якщо в питанні не введене локальне оголошення). Локальним вважається оголошення, розташоване усередині блоку опису питання, і воно діє в межах цього блоку.

### Блок опису відповіді

ВІДПОВІДЬ = <ідентифікатор питання>

[ТИП = <тип відповіді>]

ЕТАЛОН = <список> | <множина> | <текст> | <вираження>

[ЕТАЛОН = <список> | <множина> | <текст> | <вираження> ...]

БАЛ = min : max ОЦІНКА = <тип оцінки>

[ВЛАСТИВІСТЬ <ідентифікатор властивості 1> = "<значення 1>"

ВЛАСТИВІСТЬ <ідентифікатор властивості 2> = "<значення 2>"

ВЛАСТИВІСТЬ <ідентифікатор властивості n> = "<значення n>"]

КІНЕЦЬ;

<тип відповіді> – одне із ключових слів: СПИСОК, МНОЖИНА, ТЕКСТ, ВИРАЖЕННЯ:

– <список> – список розділених комами варіантів, ув'язнений у квадратні дужки, наприклад [a, b, c].

– <множина> – множина розділених комами варіантів, укладених у фігурні дужки, наприклад: {a, b, z}.

– <текст> – довільна послідовність символів, укладена в подвійні лапки, наприклад: "Фізика високих температур".

– <вираження> – арифметичне вираження, наприклад:  $(a+b)/2$ .

Тип комбінованої відповіді (множина списків або список множин) визначається зовнішніми дужками.

Наприклад:

[{a, b}, {c}, {d, e}] – СПИСОК

{[a], [b, c, d], [e]} – МНОЖИНА

ЕТАЛОН – правильна відповідь (еталон). Кожному питанню можуть відповідати кілька еталонів. Відповідь того, кого навчають рівняється з усіма еталонами й максимальним ступенем подібності відповіді й еталонів береться як остаточний результат.

БАЛ – максимальний і мінімальний бал, якому можна одержати за правильну або неправильну відповідь на дане питання (розділяються двочкою).

<тип оцінки> – описує спосіб визначення правильності відповіді:

– ДОРІВНЮЄ (<величина погрішності>): діє по-різному залежно від типу відповіді. Для відповіді типу ВИРАЖЕННЯ величина погрішності визначає розкид значення відповіді щодо еталона, а для інших типів вона змінюється в інтервалі [0,1]. Якщо відповідь укладається в інтервал (еталон) $\pm$ (величина погрішності), то відповідь вважається правильним (оцінка дорівнює max), у протилежному випадку відповідь невірна (оцінка дорівнює min).

– ПОДІБНІСТЬ (<величина погрішності>): не використовується для відповіді типу ВИРАЖЕННЯ.

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | VKPB-123.23.0007.00.00.P3 | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 39   |

Якщо ступінь подібності відповіді й еталона не більше зазначеної погрішності, то відповідь вважається правильним (оцінка дорівнює max), інакше відповідь невірна (не зовсім вірний) і оцінка O обчислюється так:

$$O = \max - (1 - \delta)(\max - \min),$$

де: –  $\delta$  – ступінь подібності відповіді й еталона,  $\delta \in [0,1]$ ;

– max, min – максимальна й мінімальна оцінки за відповідь. Величина погрішності може задаватися арифметичним вираженням.

### **Блок вибору тем**

Цей блок дає можливість обмежити список тим, що беруть участь у контролі, і визначити порядок видачі тим для контролю.

СПИСОК ТЕМ = <список тим> | ВСІ

[СПОСІБ ВИДАЧІ = ОДИН | ВСІ]

КІНЕЦЬ;

СПИСОК\_ТЕМ – список тим через кому або всі теми (ключове слово ВСЕ). Теми для опитування вибираються в тому порядку, у якому вони зазначені.

СПОСОБ ВИДАЧІ – видається по одній темі (ОДИН) або всі (ВСІ) теми відразу у вигляді списку тем. В останньому випадку користувач сам може вибирати послідовність тим для відповідей. (За замовчуванням – ОДИН).

### **Блок опису контролю по темі**

Цей блок дозволяє вказувати способи формування списку питань по темі, видачі цих питань тому, кого навчають, і підрахунку підсумкової оцінки. Можна вказувати різні способи для різних типів контролю.

КОНТРОЛЬ ПредмОбл ТЕМИ = <назва теми>

[СПИСОК = {<список питань>} | ВСІ]

ТИП = <тип контролю 1>

[СПОСІБ ВИБОРУ = ВИПАДКОВО\_БЕЗ\_ПОВТОРІВ |  
ВИПАДКОВО ПІДРЯД | КОРИСТУВАЧ]

[СПОСІБ ВИДАЧІ = ОДИН | ВСІ]

[КІЛЬКІСТЬ ПИТАНЬ = <N\_queryes >]

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-123.23.0007.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 40   |

[КІЛЬКІСТЬ ВІДПОВІДЕЙ = <N\_answers>] ОЦІНКА = <число> |  
<відсоток>

[ТИП\_ОЦІНКИ=СУМА | ВІДСОТОК | ШКАЛА <нижчий  
бал>:<вищий бал>]

[ТИП = <тип контролю 2>...]

[ТИП = <тип контролю 3> ...]

КІНЕЦЬ;

СПИСОК – для кожної теми може бути зазначений конкретний список ідентифікаторів питань (через кому) або ключове слово ВСЕ.

Якщо даний параметр опущений, то це означає, що питання будуть вибиратися із усього списку питань по даній темі. ТИП – можуть бути зазначені різні типи контролю.

Тип контролю – одне із ключових слів:

– ЗАВЕРШАЛЬНИЙ – підсумковий контроль, за результатами якого визначається, чи здана тема. Наприклад, якщо тема здана (отримана оцінка не менше тієї, що зазначено в пропозиції ОЦІНКА), то питання по цій темі тому, кого навчають, більше задаватися не будуть.

– ПРОМІЖНИЙ – контроль знань із виставлянням оцінки по темі. Наприклад, якщо проміжний контроль проводиться кілька разів по одній й тій же темі, то програма може зберігати останню оцінку й середньої арифметичне всієї отриманої оцінок по даній темі.

– ВИБІРКОВИЙ – за результатами контролю виставляється одна загальна оцінка за всі теми, що беруть участь у вибіркового контролі. Може використовуватися при підсумковому контролі знань за курсом.

СПОСІБ ВИБОРУ – спосіб вибору питання зі списку:

– ВИПАДКОВО БЕЗ ПОВТОРІВ – питання вибирається зі списку випадковим образом, але питання не повторюються (значення за замовчуванням).

– ВИПАДКОВО – питання вибирається випадковим образом із зазначеного (або загального) списку.

– КОРИСТУВАЧ – користувач сам вибирає питання з повного списку питань. Може використовуватися для самоконтролю.

– ПІДРЯД – вибирається Nqueries питань підряд з повного списку.

СПОСІБ ВИДАЧІ – спосіб видачі питань зі списку:

– ОДИН – користувачеві видається по одному питанню зі списку обраних питань. За замовчуванням видається по одному питанню.

– ВСІ – відразу видається весь список питань по темі, користувач сам вибирає чергове питання зі списку.

КІЛЬКІСТЬ\_ПИТАНЬ – кількість питань, що задаються. Якщо параметр опущений, задаються всі питання зі списку; якщо при цьому відсутній список питань, то задаються всі питання по темі.

КІЛЬКІСТЬ\_ВІДПОВІДЕЙ – кількість отриманих відповідей, при якому вважається що відбулося опитування. Якщо параметр опущений, необхідно відповісти на всі задані питання (опитування не можна перервати).

ОЦІНКА – оцінка (число або відсоток правильних відповідей), при якій тема вважається зданою.

ТИП\_ОЦІНКИ – спосіб визначення загальної оцінки O за тему на підставі оцінок отриманих за кожну відповідь:

– СУМА (використовується за замовчуванням).

– ШКАЛА <min> : <max> – оцінка нормалізується відповідно до зазначеної розмірності бальної шкали.

– ВІДСОТОК.

### **Контроль знань у режимі віддаленого доступу**

Послідовність проведення контролю знань у режимі віддаленого доступу визначається використовуваним програмним забезпеченням. Система TS дозволяє проводити тестування в рамках ЛОМ, при цьому її можливості повністю аналогічні можливостям локального режиму. Для системи IPDO було розглянуто три варіанти організації віддаленого тестування: застосування Telnet, CGI і Java:

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-123.23.0007.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 42   |

– Telnet має на увазі емуляцію термінала, тому питання й відповіді можуть бути тільки текстові й числові. Цей підхід дозволяє використовувати бази знань для організації тестування за будь-якою схемою, у тому числі, інтерактивної (гранична й адаптивна схеми).

– CGI-скрипт має більше широкі можливості подання питань (текст і графічне зображення формату GIF) і відповідей. Для введення відповідей у вигляді формул і графіків були створені Internet-аналоги графічного редактора й редактори формул. Але формування HTML-сторінки при такому підході відбувається один раз, після чого вона відправляється тому, кого навчають. Тому з його допомогою можливе тестування тільки за лінійною схемою.

– Java-апплет, що реалізує ще один варіант інтерфейсу до СКЗ, дозволяє системі не тільки проводити віддалене тестування за будь-якою схемою (у тому числі, інтерактивною), але має також можливості для віддаленого адміністрування користувачів і використання вищезгаданих Internet-додатків.

### 3.2 Розробка структурної схеми

З метою експериментальної перевірки застосовності запропонованих методів створення ХАНС був розроблений макетний програмний комплекс, до складу якого входять:

- підсистема керування (монітор), що організує роботу системи в цілому;
- підсистема контролю знань;
- набір баз знань, що реалізують керування процесом навчання й контролем знань.

Програмний комплекс призначений для створення хмарних повнофункціональних навчальних систем по довільній предметній області.

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-123.23.0007.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 43   |

## Структура хмарного програмного комплексу

Хмарність системи має два основних аспекти: наявність допоміжного програмного забезпечення (ДПЗ), що виконує окремі внутрішні функції АНС, і робота в режимі віддаленого доступу.

Для обміну даними по мережі Internet потрібно додаткове програмне забезпечення, що відповідає за створення HTML-сторінок і обмін інформацією з ядром PDO. При створенні СКЗ IPDO були розглянуті різні варіанти такого ПЗ, а саме:

1. Протокол Telnet, емулюючий термінал користувача. Для цього потрібен мінімум додаткового програмного забезпечення (властиво драйвер Telnet на машині клієнта), цей підхід має широку функціональність, але має небагаті виразні можливості.

2. Використання CGI-інтерфейсу. При цьому на серверній машині необхідна наявність інтерпретатора мови Perl або PHP і сервера, що підтримує протокол CGI. На машині клієнта повинен бути встановлений браузер Internet Explorer (або будь-який інший, підтримуючий протокол CGI). Основне переваго цього підходу – переносимість: написаний скрипт без яких би то не було змін працює на різних платформах і дозволяє переглядати HTML-сторінки як за допомогою Internet Explorer, так і за допомогою Netscape Communicator. Недоліками такого рішення є скромні функціональні можливості (лінійна схема тестування й статична інформація) і обмежена кількість одночасно працюючих з ним користувачів – не більше 50.

3. Використання Java-аплета. Цей підхід надає найбільш розвинені функціональні можливості аж до адаптивного тестування.

Дамо короткий опис перерахованих вище технологій.

### TELNET

TELNET (TELEcommunication NETwork) – мережний протокол для реалізації текстового інтерфейсу по мережі (у сучасній формі – за допомогою транспорту TCP). Назву «telnet» мають також деякі утиліти, що реалізують

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРБ-123.23.0007.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 44   |

клієнтську частину протоколу. Призначення протоколу TELNET у наданні досить загального, двонаправленого, восьмибітного байт-орієнтованого засобу зв'язку. Його основне завдання полягає в тім, щоб дозволити термінальним пристроям і термінальним процесам взаємодіяти один з одним. Передбачається, що цей протокол може бути використаний для зв'язку виду термінал-термінал («зв'язування») або для зв'язку процес-процес («хмарні обчислення»).

З'єднання TELNET – це TCP з'єднання, використовуване для передачі даних, з різною керуючою інформацією.

Протокол TELNET базується на трьох основних ідеях: перша, концепція «Віртуального Мережного Термінала» (Network Virtual Terminal, NVT); друга, принцип обмовлених опцій; третя, симетричний вид терміналів і процесів.

1. Коли встановлюється TELNET з'єднання, передбачається, що на кожному кінці з'єднання породжується й завершується «Віртуальний Мережний Термінал» або ВСТ. ВСТ – це уявлюваний пристрій яке надає стандартне, доступне через мережу, проміжне подання класичного термінала. Це усуває необхідність в «серверному» і «клієнтському» вузлах для зберігання інформації про характеристики кожного термінала й домовленостей про взаємодію. Всі вузли, як клієнтський, так і серверний, відображають свої локальні характеристики пристрою для того, щоб виступати в мережі як ВСТ, і кожний міг прийняти схоже відображення з іншої сторони. ВСТ призначений для відомості балансу між надмірним обмеженням і надмірними можливостями. «Користувальницький» хост – це звичайний хост із прив'язаним до нього фізичним терміналом, а «серверний» хост – це звичайний хост, що надає якийсь сервіс. Як альтернативну точку зору, можна розглядати випадок коли з'єднуються рівні хости: термінал-термінал або процес-процес. У такий спосіб будемо вважати «користувальницьким» хостом той хост який ініціює з'єднання.

2. Принцип обмовлених опцій охоплює той факт, що багато які хости швидше за все будуть хотіти надати додаткові сервіси до або після їхньої доступності у ВСТ, і багато користувачів захочуть мати складні термінали з

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРБ-123.23.0007.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 45   |

елементами вишуканості, замість мінімальних, для одержання таких додаткових сервісів. Незалежні від, але структуровані в TELNET протоколі різні «опції» санкціоновані й можуть бути використані з «DO, DON'T, WILL, WON'T» структурою для того, щоб дозволити користувачеві й серверу сходитися у використанні більше продуманого (або відмінного) набору угод для їх TELNET з'єднання. Такі опції включають зміну набору символів, режиму еха, і т.д. Базова стратегія для налагодження використання опцій – це на одній зі сторін (або на обох) ініціювати запит: чи буде певна опція мати будь який ефект. Інша сторона може або прийняти, або відкинути запит. Якщо запит приймається, то опція негайно набуває чинності; якщо ж опція відкидається, то зв'язаний аспект з'єднання залишається як специфіковане для VST. Очевидно, що сторона може завжди відкидати запит на включення, і ніколи не повинна відкидати запит на відключення деякої опції починаючи з моменту коли сторони домовилися про підтримку VST. Синтаксис обмовленої опції повинен бути таким, щоб якщо обидві сторони запросять одночасно опцію, те кожний буде розглядати запит з іншої сторони як позитивне підтвердження цієї опції.

3. Симетричність синтаксису узгодження може потенційно привести до нескінченного циклу узгодження – коли кожна сторона бачить вхідні команди не підтверджує їх, але для підтвердження надсилає новий запит. Для запобігання таких циклів, необхідно дотримуватися наступних правил:

– Сторони можуть запитувати тільки зміну статусу опції; тобто сторона може не посилати «запит» тільки для того, щоб повідомити, що дана опція підтримується.

– Якщо сторона одержує щось, що інтерпретується як запит перемикавання в деякий режим у якому ця сторона вже перебуває, то на такий запит не потрібно відправляти підтвердження.

– Щораз, коли одна сторона відправляє команду опції на іншу сторону, не важливо запитуючи або підтверджуючи, і використання опції повинне мати будь який ефект на обробку даних, що відсилаються першою стороною другій стороні,

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-123.23.0007.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 46   |

то команда опції повинна бути вставлена в потоці даних у тій місці, з якого бажано, щоб опція набула чинності. (Слід зазначити, що пройде якийсь час між передачею запиту й одержанням підтвердження, і воно може бути негативним. Таким чином, хост може буферизувати дані, після відправлення запиту опції й до одержання відповіді із прийняттям або відкиданням опції, для того, щоб сховати «період невизначеності» від користувача.)

Імовірно, що відразу після установки TELNET з'єднання, запити опцій будуть шквалом передаватися в обох напрямках з'єднання, через те, що кожна сторона буде намагатися одержати найкращий сервіс від іншої сторони. Крім того, опції можуть бути використані для динамічної зміни характеристик з'єднання для того, щоб відповідати локальним умовам, що змінюються. Наприклад, ВСТ використовує дисципліну передачі, що добре підходить для багатьох додатків «рядок за раз» (таких як BASIC), але погано підходить для додатків «символ за раз» (таких як NLS). Сервер міг би виділити додатковий процесорний час необхідне для дисципліни «символ за раз» якщо це підходить для локального процесу й домовитися про відповідну опцію. Хоча, замість того, щоб додатково не витратити процесорний час, можна перемкнутися (тобто домовитися) повернутися назад до ВСТ, коли детальний контроль більше не потрібний.

Можливо що запити, ініційовані процесами, будуть сприяти незакінченому циклу запиту, у випадку коли процес відмовляє на повторний запит опції. Для виключення такої ситуації відхилені запити не повинні бути повторені, поки що або не змінитися. Операційно, це може означати, що процес почав виконувати іншу програму або користувач дав іншу команду, або щось трапилося в контексті даного процесу або опції. Необхідно взяти за правило те, що повторний запит може бути посланий тільки як результат більш пізньої інформації з іншої сторони або у випадку локального втручання людини.

Проектувальники опції не повинні себе почувати стиснутими трохи обмеженим синтаксисом для обмовленої опції. Зміст простого синтаксису

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРБ-123.23.0007.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 47   |

полягає в тому, щоб спростити прийняття опції. Якщо деяка специфічна опція вимагає більше складної структури погоджень чим наявна в «DO, DON'T, WILL, WON'T», то необхідно спочатку домовитися через існуючу структуру погоджень, а після того, як обидві сторони впевняться в розумінні цієї опції, використовувати вільно більше екзотичний синтаксис. Наприклад, одна зі сторін могла б надіслати запит на зміну (установку) довгі рядки. Якщо він приймається, то може бути використаний відмінний від базового синтаксис для того, щоб фактично домовитися про довжину рядка; такі « під-узгодження» могли б включати поля для мінімально, максимально й бажаних довжин рядка. Важливо те, що такі розширені узгодження ніколи не повинні починатися, поки «стандартні» переговори не привели до того, що обидві сторони можуть розуміти розширений синтаксис.

У підсумку, WILL XXX посилає однією із двох сторін, для того щоб показати бажання (пропозицію) сторони виконувати опцію XXX, DO XXX і DON'T XXX є підтвердженням і відкиданням опції XXX відповідно, на запит WILL XXX; аналогічно, DO XXX відправляє для того, щоб показати бажання іншої сторони (тобто одержувача DO) почати виконувати опцію XXX, WILL XXX і WON'T XXX є підтвердженням і відкиданням опції XXX відповідно, на запит DO XXX. Так як ВСТ – це те, що залишається коли ніякі опції не включені, відповіді DON'T і WON'T гарантують що з'єднання залишиться в стані яким обидві сторони можуть управляти. Таким чином, всі хости можуть реалізувати свої TELNET процеси так, щоб вони взагалі не знали про опції які не підтримуються, просто повертаючи відкидання (тобто відмовляючись від опції) на будь-який запит опції, що даний процес не може зрозуміти.

Протокол TELNET був зроблений максимально симетричним стосовно зв'язування користувач-сервер, для того, щоб він легко й природно покривал випадки користувач-користувач і сервер-сервер. Бажано, але не обов'язково, щоб опції зберігали цей принцип. У кожному разі явно зізнається, що симетрія – це принцип, а не правило.

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРБ-123.23.0007.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 48   |

## **CGI**

CGI (Common Gateway Interface – «загальний інтерфейс шлюзу») – стандарт інтерфейс, використовуваний для зв'язку зовнішньої програми з веб-сервером. Програму, що працює по такому інтерфейсі разом з веб-сервером, прийнято називати шлюзом, хоча багато хто воліють назви «скрипт» (сценарій) або «CGI-програма».

Сам інтерфейс розроблений таким чином, щоб можна було використовувати будь-яку мову програмування, що може працювати зі стандартними пристроями уведення/виводу. Такими можливостями володіють навіть скрипти для убудованих командних інтерпретаторів операційних систем, тому в тих випадках, коли немає потреби в складній функціональності, можуть використовуватися навіть такі прості командні скрипти.

Всі скрипти, як правило, поміщають у каталог `cgi-bin` сервера, але це не обов'язково: скрипт може розташовуватися де завгодно, але при цьому більшість веб-серверів вимагають спеціального налаштування.

У веб-сервері Apache, наприклад, таке налаштування може вироблятися за допомогою загального файлу налаштувань `httpd.conf` або за допомогою файлу `.htaccess` у тім каталозі, де втримується цей скрипт. Також Apache дозволяє запускати всі скрипти, що мають розширення `.cgi`.

CGI є одним з найпоширеніших засобів створення динамічних веб-сторінок.

### **Java-апплет**

Java-апплет – це прикладна програма на Java у формі байта-коду. Java-апплети виконуються у веб-браузері з використанням віртуальної Java машини (JVM), або в Sun's AppletViewer, автономному інструменті для тестування апплетів. Java-апплети звичайно пишуться мовою програмування Java, але можуть бути написані й на інших мовах, які компілюються в байт код Java, таких, як Python. Апплети використовуються для надання інтерактивних можливостей веб-додатків, які не можуть бути надані HTML. Так як байт-код Java платформо-

|      |      |          |        |      |                                  |           |
|------|------|----------|--------|------|----------------------------------|-----------|
|      |      |          |        |      | <b>ВКРБ-123.23.0007.00.00.ПЗ</b> | Арк.      |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | <b>49</b> |

незалежний, то Java-аплети можуть виконуватися за допомогою плагінів браузером багатьох платформ, включаючи Microsoft Windows, UNIX, Apple Mac OS і GNU/Linux. Такі програми з відкритим вихідним кодом, як applet2app, можуть бути використані для перетворення аплета в самостійні програми на Java або файли, що виконуються, Linux і Windows.

Java Servlet іноді неофіційно порівнюють із серверним аплетом, але він відрізняється своєю мовою, функціями й кожною з описаних тут характеристик аплетів.

Код аплета завантажується з веб-сервера, і браузер або вставляє аплет у веб-сторінку, або відкриває нове вікно із власним користувацьким інтерфейсом аплета. Аплет може бути відображений на веб-сторінці шляхом використання застарілого HTML елемента `applet`, або елемента, що рекомендується, `object`. Цим визначається місце розташування й джерело аплета.

Головний клас аплета розширює клас `java.applet.Applet` або, якщо створюється Swing аплет, `javax.swing.JApplet`. Клас повинен перевизначити методи створення користувацького інтерфейсу усередині себе. (`Applet` є нащадком `Panel`, що, у свою чергу, є нащадком `Container`).

#### Переваги використання Java-аплетів

- кроссплатформеність;
- аплет може працювати на «всіх» установлених до цього часу версіях Java, а не тільки з останньою версією; однак, якщо аплет вимагає останню версію JRE, то клієнт буде змушений чекати більше тривалого завантаження;
- аплет підтримується більшістю браузерів;
- він кешується в більшості браузерів, а тому буде швидко завантажуватися при поверненні на веб-сторінку; але аплет може зберігатися в кеші й створювати проблеми після виходу нових версій;
- він може мати повний доступ до машини, на якій виконується, якщо користувач згодний на це;

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРБ-123.23.0007.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 50   |

– аплет може поліпшити використання: після першого запуску аплета, коли JVM уже виконується й швидко запускається, переважно в постійних користувачів Java, однак JVM прийдеться перезапускати щораз, коли запускається новий браузер;

– він може запуснитися з порівнянною (але звичайно повільніше) швидкістю на інших компілюємих мовах, таких як C++, але в багато разів швидше, ніж JavaScript;

– він може перенести роботу із сервера до клієнта, роблячи Інтернет-рішення з, більшим числом користувачів/клієнтів.

Java-аплет має наступні недоліки:

– він вимагає установки Java-розширення (plug-in), що не у всіх браузерах доступно за замовчуванням;

– він не може запуснитися доти, поки не запусниться віртуальна Java-Машина, і це може забрати значний час при першому запуску;

– створення й дизайн гарного користувальницького інтерфейсу з використанням аплетів вважається більше складним завданням, чим за допомогою технології, заснованої на HTML;

– для підвищення безпеки аплета, обмежується доступ до користувальницької системи: зокрема, він не має прямого доступу до диска клієнта або буферу обміну;

– деякі організації допускають тільки програмне забезпечення, установлене адміністраторами. У результаті багато користувачів не можуть бачити аплети за замовчуванням.

– аплети можуть зажадати використання певного JRE.

Розглянувши технології, перейдемо до опису структурної схеми.

З урахуванням функціональної структури ХАНС, хмарного характеру системи й наявності додаткового програмного забезпечення загальна структура програмного комплексу буде виглядати так (рисунок 3.1).

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРБ-123.23.0007.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 51   |

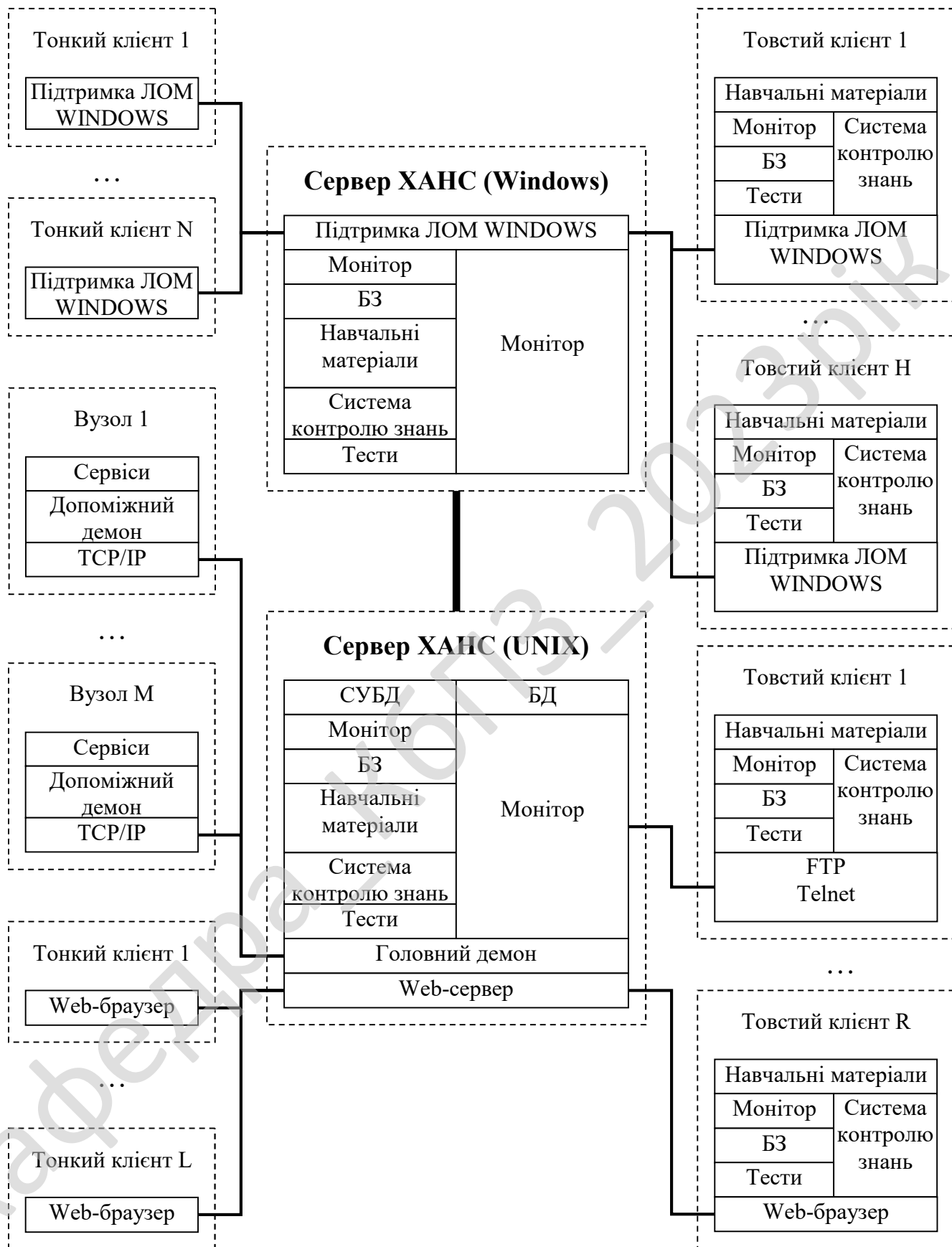


Рисунок 3.1 – Структурна схема хмарної автоматизованої навчальної системи (ХАНС)

У програмному забезпеченні активно використовується технологія товстих, тонких клієнтів і демонів.

Приведемо опис цих технологій.

### **Товстий клієнт**

Товстий або Rich-клієнт в архітектурі клієнт-сервер – це комп'ютер, що забезпечує розширену функціональність незалежно від центрального сервера. Як правило, сервер у цьому випадку є лише сховищем даних, а вся робота з обробки й подання цих даних переноситься на машину клієнта.

### **Тонкий клієнт**

Тонкий клієнт (thin client) у комп'ютерних технологіях – бездисковий комп'ютер-клієнт у мережах із клієнт-серверною або термінальною архітектурою, що переносить всі або більша частина завдань по обробці інформації на сервер.

У цей час (2010 р.) під терміном «тонкий клієнт» мається на увазі досить широкий з погляду системної архітектури ряд пристроїв, які поєднуються загальною властивістю: можливість роботи в термінальному режимі. Таким чином, для роботи тонкого клієнта необхідний термінальний сервер. Цим тонкий клієнт відрізняється від товстого клієнта, що, навпроти, робить обробку інформації незалежно від сервера, використовуючи останній в основному лише для зберігання даних. Прикладом тонкого клієнта може служити комп'ютер із браузером, що використовується для роботи з веб-додатками.

Крім загального випадку, варто виділити апаратний тонкий клієнт (наприклад, Windows-термінал) – спеціалізований пристрій, принципово відмінний від ПК. Апаратний тонкий клієнт не має жорсткого диска, використовує спеціалізовану локальну ОС (одне із завдань якої організувати сесію з термінальним сервером для роботи користувача), не має у своєму складі рухливих деталей, виконується в спеціалізованих корпусах з повністю пасивним охолодженням.

Для розширення функціональності тонкого клієнта прибігають до його «стовщення», наприклад, додають можливості автономної роботи, зберігаючи

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРБ-123.23.0007.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 53   |

головну відмінність – роботу в сесії з термінальним сервером. Коли в клієнті з'являються рухливі деталі (жорсткі диски), з'являються можливості автономної роботи, він перестає бути тонким клієнтом у чистому виді, а стає універсальним клієнтом.

Тонкий клієнт у більшості випадків має мінімальну апаратну конфігурацію, замість жорсткого диска для завантаження локальної спеціалізованої ОС використовується DOM (DiskOnModule) [модуль із роз'ємом IDE, флеш-пам'яттю й мікросхемою, що реалізує логіку звичайного жорсткого диска – в BIOS визначається як звичайний жорсткий диск, тільки розмір його звичайно на 2-3 порядки менше]. У деяких конфігураціях системи тонкий клієнт завантажує операційну систему по мережі із сервера, використовуючи протоколи PXE, BOOTP, DHCP, TFTP і Remote Installation Services (RIS).

### Демон

Демон (daemon) – у системах класу UNIX – програма, що працює у фоновому режимі без прямого спілкування з користувачем.

Демони звичайно запускаються під час завантаження системи. Типові завдання демонів: сервери мережних протоколів (HTTP, FTP, електронна пошта й ін.), керування устаткуванням, підтримка черг друку, керування виконанням завдань за розкладом і т.д. У технічному змісті демоном вважається процес, предком якого є `init` – кореневий процес UNIX.

Іноді слово `daemon` інтерпретують як акронім **Disk and execution monitor**, але саме слово виникло на зорі Unix набагато раніше цієї розшифровки. Операційні системи сімейства BSD використовують як логотип зображення демона.

У системах Solaris 10 і OpenSolaris для керування демонами використовується Service Management Facility.

У системах Windows аналогічний клас програм називається «Служби» (Services), які, втім, теж іноді називають демонами.

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-123.23.0007.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 54   |

**Служби Windows** (Windows Service, сервіси) – додатки, які запускаються автоматично системою при запуску Windows і користувача, що виконуються поза залежністю від статусу. Має загальні риси з концепцією демонів в Unix.

### Режими роботи

У більшості випадків службам заборонена взаємодія з консоллю або робочим столом користувачів (як локальних, так і віддалених), однак для деяких сервісів можливе виключення – взаємодія з консоллю (сесією з номером 0, у якій зареєстрований користувач локально або при запуску служби `mstsc` із ключем `/console`).

Існує кілька режимів для Сервісів:

- заборонений до запуску;
- ручний запуск (по запиту);
- автоматичний запуск при завантаженні комп'ютера;
- автоматичний (відкладений) запуск (уведений в Windows Vista і Windows Server 2008);
- обов'язковий сервіс/драйвер (автоматичний запуск і неможливість (для користувача) зупинити сервіс).

### Фоновий режим

Windows пропонує програму Service Control Manager, з її допомогою можна управляти створенням, видаленням, запуском і зупинкою служб. Додаток, що має статус сервісу, повинне бути написане таким чином, щоб воно могло приймати повідомлення від Service Control Manager. Потім, одним або декількома викликами API, ім'я служби й інші атрибути, такі, як його опис, реєструються в Service Control Manager.

### Запуск, зупинка й зміна служб Windows

Служби і їхні атрибути можуть бути змінені в консолі керування MMC: Панель керування Windows -> «Administrative Tools» -> «Services».

### Керування запуском служб при старті Windows

Список служб перебуває у вітці Реєстру Windows

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-123.23.0007.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 55   |

«HKEY\_LOCAL\_MACHINE \ SYSTEM \ CurrentControlSet \ Services».

Значення параметра «Start» мають тип «REG\_DWORD» і можуть приймати значення:

0 – Низькорівневі драйвери, наприклад драйвери дисків, які завантажуються на самому ранньому етапі завантаження – завантаження ядра.

1 – Драйвери, які завантажуються після ініціалізації ядра ОС.

2 – Служби, які повинні бути завантажені Диспетчером Керування Сервісами (дорівнює параметру – "Авто").

3 – Служби, що запускаються Диспетчером Керування Сервісами, тільки у випадку одержання явної інструкції на завантаження (дорівнює параметру – "Вручну").

4 – Служби, які не завантажуються (дорівнює параметру – "Відключене").

### Керування роботою служб із командного рядка

Керування службами можливо за допомогою командного рядка: зупинка служби – «net stop service\_name », запуск служби – «net start service\_name ». Наприклад, запуск служби «Диспетчер черги друку» – «net start spooler».

Видалення служби: «sc delete „Ім'я служби“ » (лапки необхідні, якщо ім'я служби містить пробіли). З видаленням потрібно бути гранично обережним, так як віддалену службу повернути назад дуже й дуже важко.

Установка режиму запуску служби: «sc config „Ім'я служби“ start= параметр запуску». Параметри запуску: auto (автоматично), demand (вручну), disabled (відключена).

### Права користувача й особливості реалізації

Сервіси Windows за замовчуванням запускаються від імені користувача «LocalSystem», що має повні права в системі (переважаючими права навіть облікового запису «Administrator»). Робочим каталогом буде каталог Windows (визначається змінної «WinDir», що звичайно дорівнює «C:\WINNT» або «C:\WINDOWS»), а каталог для зберігання тимчасових файлів буде «%WinDir%\TEMP».

|      |      |          |        |      |  |  |  |  |      |
|------|------|----------|--------|------|--|--|--|--|------|
|      |      |          |        |      |  |  |  |  | Арк. |
|      |      |          |        |      |  |  |  |  | 56   |
| Вим. | Арк. | № докум. | Підпис | Дата |  |  |  |  |      |
|      |      |          |        |      |  |  |  |  |      |

ВКРБ-123.23.0007.00.00.ПЗ

Арк.

56

Оскільки це не справжній користувач, а «системний», то з'являються деякі труднощі, коли додатку необхідно зберегти дані, що ставляться до користувача (user-specific data), оскільки домашньої директорії цього користувача не існує.

Важливо також те, що у випадку якщо служба працює від імені локального користувача (реальний користувач створений для службових цілей) якщо пароль такого користувача змінений, сервіс не буде запускатися доти, поки пароль для сервісу теж не буде змінений.

Протокол обміну даними між ядром ХАНС і ДПЗ здійснює динамічну реконфігурацію системи при порушенні топології мережі з метою підвищення надійності функціонування хмарної системи.

### **Протокол обміну даними з динамічної реконфігурацією системи**

Перед протоколом обміну даними коштують наступні завдання:

1. Організація хмарних обчислень.
2. У випадку виявлення порушення топології мережі – реконфігурація системи, що полягає в запуску аналогічного процесу на одному з доступних вузлів і перенапрямку потоку даних знову запущеному процесу.

Конфігурація системи описується в конфігураційному файлі, формат якого наведений нижче:

period = t count = N

Ім'я\_вузла\_1 = Адреса\_1

Ім'я\_вузла\_2 = Адреса\_2

...

Ім'я\_вузла\_N = Адрес\_N

Services = M

Ім'я\_процесу\_1 = Процес\_1

timeout = t1

sites = ім'я\_вузла\_11[, ім'я\_вузла\_12,..ім'я\_вузла\_1i]

...

Ім'я\_процесу\_M = Процес\_M

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРБ-123.23.0007.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 57   |

timeout = tm

sites = ім'я\_вузла\_т1[, ім'я\_вузла\_т2,..ім'я\_вузла\_тj]

(У квадратні дужки укладені необов'язкові параметри). Тут:

- period=t – час відгуку обслуговуючого процесу;
- count=N – кількість вузлів;
- M – кількість процесів;
- Ім'я\_вузла\_ – ідентифікатор вузла і;
- Ім'я\_процесу\_і – ідентифікатор процесу і;
- timeout – період часу очікування відповіді;
- sites – список ідентифікаторів вузлів, на яких може бути запущений даний процес.

Для реалізації протоколу потрібні процеси-демони: головний демон на сервері АНС і допоміжні демони на тих вузлах, на яких розташоване допоміжне програмне забезпечення (ДПЗ). Фактично процеси ДПЗ є сервісами, для яких повинен бути описаний інтерфейс.

Система хмарної обробки характеризується невеликою довжиною повідомлень і істотним часом, що потрібно на їхню обробку. Повідомлення, якими обмінюються сервер і допоміжні процеси, уміщаються в одному пакеті.

Формат пакету обміну даними наступний:

| Адреса відправника | Ім'я відправника | Адреса одержувача | Ім'я одержувача | Код | Довжина пакета даних | Дані |
|--------------------|------------------|-------------------|-----------------|-----|----------------------|------|
|--------------------|------------------|-------------------|-----------------|-----|----------------------|------|

Адреса й ім'я відправника необхідні для того, щоб процес-одержувач знав, кому відправляти відповідь. Адреса й ім'я одержувача визначає сам процес-одержувач. Код інтерпретується залежно від того, кому адресований пакет. Якщо адресат – допоміжний демон, то це код команди, що йому потрібно виконати.

Якщо ж пакет адресований головному демонові, то це відгук (підтвердження готовності) або код завершення допоміжного процесу.

Для виключення втрати запитів головний демон повинен підтримувати чергу запитів (у порядку надходження, FIFO). Черга запитів зберігає наступну інформацію:

1. Ідентифікатор запиту.
2. Час надходження запиту.
3. Адреса й ім'я відправника.
4. Ім'я адресуємого процесу.
5. Властиво запит.
6. Ідентифікатор обслуговуючого процесу.
7. Ім'я вузла, на якому обслуговується запит.
8. Таймаут, протягом якого очікується відповідь на запит.

Відповідь (результат обробки або повідомлення про помилку). Запит віддаляється із черги, якщо він обслужений або якщо він не може бути обслужений через відсутність сервісу, якому був адресований запит.

### 3.3 Розробка функціональної схеми

Функціональна структура програмного комплексу наведена на рисунку 3.2. У завдання монітора й системи контролю знань, крім зазначених на рисунку 3.2, входить формування завдання для процесора бази знань (для зовнішнього керуючого модуля) і зчитування результатів його роботи з файлу обміну даними. Система контролю знань (СКЗ) також організує взаємодію з тим, кого навчають, оцінює його відповіді й записує результати в базу даних.

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-123.23.0007.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 59   |

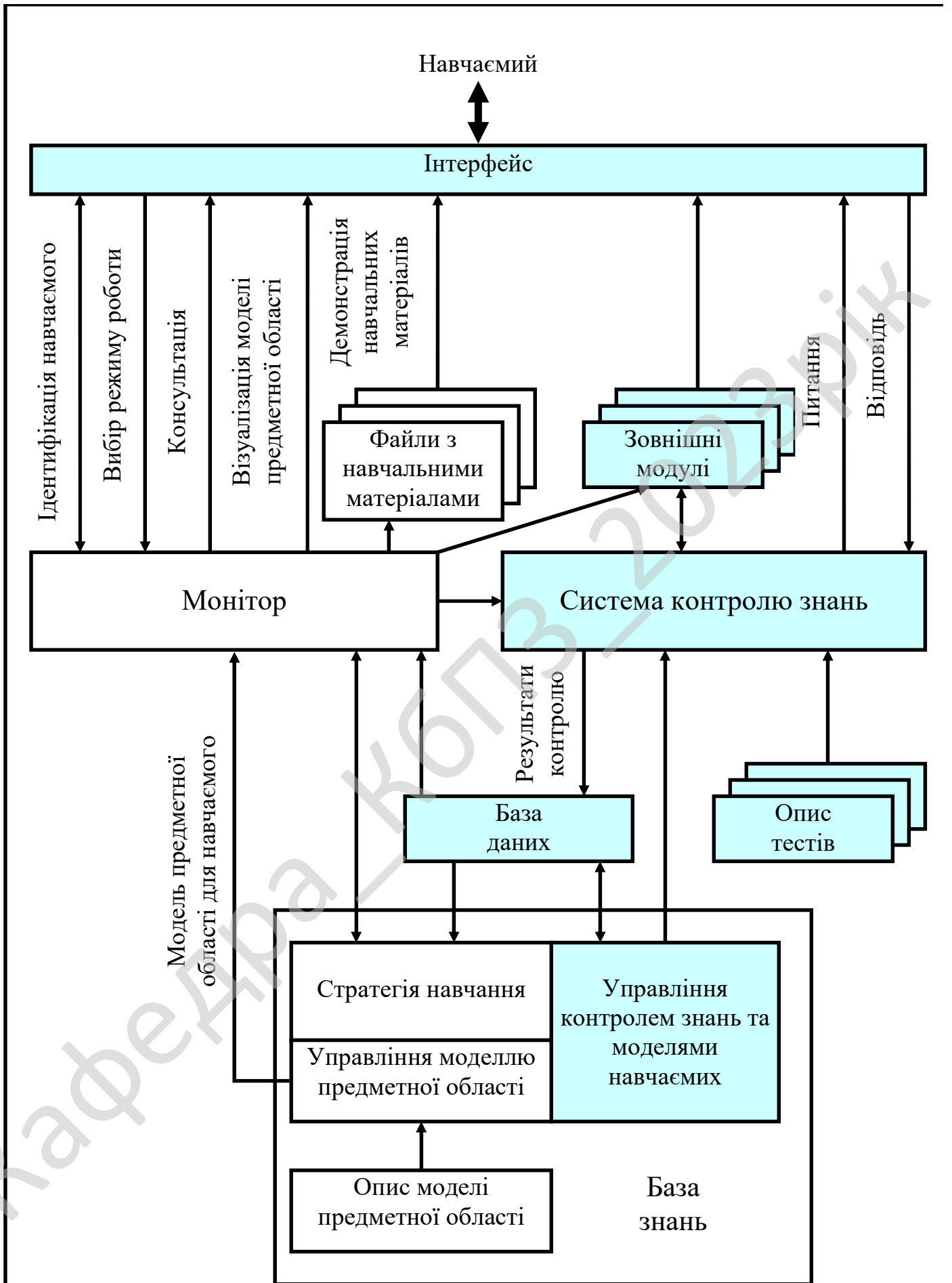


Рисунок 3.2 – Функціональна схема системи

Основні особливості цього комплексу:

**1. Орієнтація на роботу в мережі.** Система забезпечує роботу в режимі "клієнт-сервер" у рамках ЛОМ і через Internet. Включає спеціалізований протокол обміну даними в рамках ЛОМ для підтримки хмарних обчислень.

**2. Відкритість.** Принцип відкритості має на увазі можливість налаштування системи на вимоги конкретного користувача. Тут можна виділити три аспекти:

– Зміна режимів і параметрів функціонування програмного забезпечення, що входить до складу ядра системи. Це реалізовано шляхом винесення параметрів у файл конфігурації.

– Можливість використання будь-яких навчальних матеріалів на машинних носіях і методів подачі цих матеріалів: електронний довідник (від перегляду текстових файлів до підключення гіпертексту), запуск демонстраційних програм і т.д.

– Організація зовнішнього керування процесом навчання. Це досягається шляхом винесення в зовнішні файли (у базу даних і в базу знань) правил організації керування навчанням і всіма можливими параметрами й характеристик, які можна виміряти в процесі роботи з тим, кого навчають. Набори правил можна модифікувати. Значення характеристик і параметрів доступно для зміни й для використання в правилах БЗ.

**3. Переносимість** на різні платформи на рівні вихідних текстів програм.

Система контролю знань реалізована у двох варіантах: як додаток під ОС Windows (система TS) і як консольний додаток (система IPDO).

### 3.4 Розробка діаграми процесів

Діаграма процесів розробленої системи зображена на рисунку 3.3.

З рисунку видно, що на початку роботи програми відбувається процес початку/кінця програми.

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРБ-123.23.0007.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 61   |

Він взаємодіє з наступними процесами:

- Процес ідентифікації навчального.
- Процес збереження результатів в базі даних.

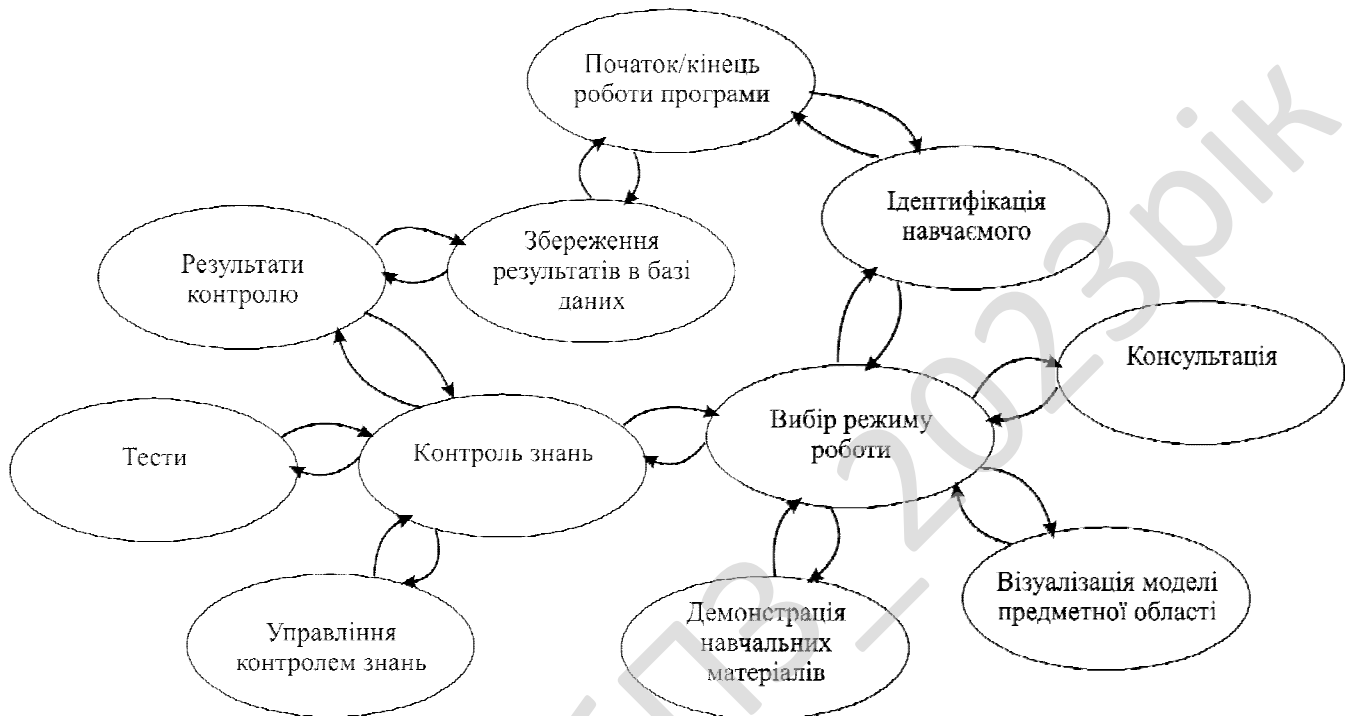


Рисунок 3.3 – Діаграма процесів системи

Процес ідентифікації навчального взаємодіє з процесом вибору режиму роботи.

Процес вибору режиму роботи взаємодіє у свою чергу з наступними процесами:

- Процесом консультування навчального викладачем.
- Процесом візуалізації моделі предметної області, який дозволяє більш наглядно донести учбовий матеріал.
- Процесом демонстрації навчальних матеріалів, який дозволяє студенту, перш ніж здати тест, до нього підготуватися.
- Процесом контролю знань.

Процес контролю знань є основним процесом, який виконується у системі.

Він взаємодіє з наступними процесами:

- Процесом управління контролем знань.
- Процесом тестування.
- Процесом отримання результатів контролю.

Процес отримання результатів контролю взаємодіє з процесом збереження результатів в базі даних.

Цей процес є завершуючим у системі, й він взаємодіє з процесом початку/кінця роботи програмного забезпечення, розробленого у результаті виконання бакалаврської роботи.

Таким чином, розглянувши опис системи, структурну, функціональну схеми системи, та діаграму взаємодії процесів перейдемо до опису блок-схем основної програми, та підпрограм, які використовуються, для реалізації системи.

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-123.23.0007.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 63   |

## 4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ

### 4.1 Блок-схеми та опис алгоритмів функціонування системи

У цьому розділі описана реалізація повнофункціональної ХАНС, заснованої на принципі відкритої архітектури. Визначено вимоги до програмного комплексу і його структура.

Основні особливості розробленого комплексу:

1. Орієнтація на роботу в мережі. Система забезпечує роботу в режимі "клієнт-сервер" по ЛОМ і через Internet з використанням різних технологій. Вона включає спеціалізований протокол обміну даними для підтримки хмарних обчислень. Цей протокол дозволяє визначити працездатність вузлів мережі й допоміжного програмного забезпечення й виконує реконфігурацію системи у випадку виникнення збоїв.

2. Відкритість, що включає в себе:

– Зміну режимів і параметрів функціонування програмного забезпечення, що входить до складу ядра системи, шляхом налаштування параметрів у файлах конфігурації.

– Можливість використання різних навчальних матеріалів на машинних носіях і методів подачі цих матеріалів.

– Організація зовнішнього керування процесом навчання. Це досягається шляхом винесення в зовнішні файли (у базу даних і в базу знань) правил організації керування навчанням і всіма можливими параметрами й характеристик, які можна виміряти в процесі роботи з тим, кого навчають.

3. Переносимість на різні платформи на рівні вихідних текстів програм.

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРБ-123.23.0007.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 64   |

Для системи розроблена декларативна мова опису тестів, що дозволяє зберігати опис тестів у зовнішні стосовно системи файли, що робить тест незалежним від програмного забезпечення.

На основі спроектованої мови опису тестів і розроблених диференційованих методів створена система контролю знань, у якій реалізовані розглянуті раніше форми подання відповідей.

Відкритість архітектури СКЗ також забезпечується можливістю підключення зовнішніх модулів для виконання наступних функцій:

- реалізації тих форм відповідей, які не включені в систему;
- одержання оцінок відповідей;
- керування формуванням тесту;
- керування проведенням тестування.

У рамках СКЗ створені бази знань, що реалізують:

- формування тесту на основі параметрів питань для різних схем тестування;
- проведення тестування у відповідність із лінійною, граничною й адаптивною схемами.

Система контролю знань реалізована у двох варіантах: як додаток під ОС Windows і як консольний додаток.

З використанням системи консольного додатка в режимі віддаленого доступу через Internet проведене тестування студентів. Результати експериментів підтверджують скорочення часу, затрачуваного викладачем на проведення контрольних заходів.

Аналіз використовуваних у тестах типів відповідей показав, що укладачі тестів віддають перевагу розширеним типам відповідей (дворівневі схеми й вільно-конструюємі відповіді: текст, графіки, формули, арифметичні вирази).

Структурна схема організації хмарних обчислень наведена на рисунку 4.1. При одержанні запиту сервер за допомогою списку сервісів визначає, що цей

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРБ-123.23.0007.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 65   |

запит адресований допоміжному програмному забезпеченню (ДПЗ), і передає його головному демонові.

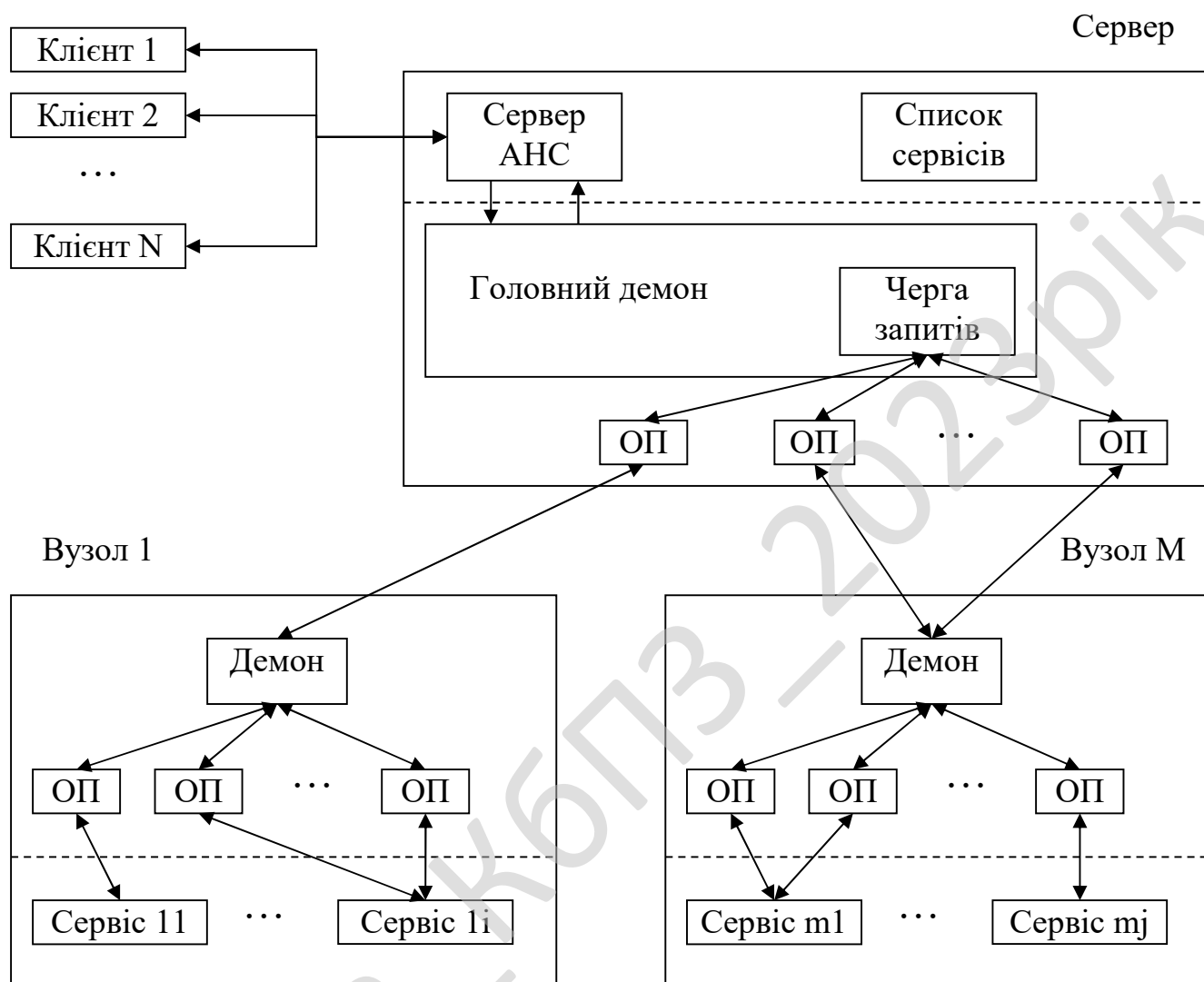


Рисунок 4.1 – Схема організації обміну даними між ядром АНС і ДПЗ

На рисунку 4.2 наведена блок-схема роботи основної програми, у якій реалізовано алгоритм роботи головного демона. Для організації багатокористувальницької роботи він породжує обслуговуючий процес (ОП) для кожного запиту.

Через певні проміжки часу (period), ОП повинен надсилати демонові відгук на підтвердження своєї працездатності. Якщо демон за визначений час не одержує цей відгук, він вважає, що ОП "завис" і починає процедуру

реконфігурації (рисунок 4.3). При неможливості обробки запиту головний демон зберігає запит у файлі трасування, а сервер видає повідомлення тому клієнтові, від якого прийшов запит.

Допоміжні демони також породжують обслуговуючі процеси для виконання запитів. Алгоритм роботи допоміжного демона наведений на рисунку 4.4.

Блок-схема алгоритму роботи основної програми зображена на рисунку 4.2. Після запуску програми відбувається зчитування конфігурації підсистеми контролю знань в хмарній навчальній системі.

Коли конфігурація зчитана, відбувається запит готовності вузлів хмарної навчальної системи.

Наступним кроком виконання програмного забезпечення є запуск процесів допоміжного програмного забезпечення (ДПЗ).

Після цього відбувається перевірка запуску усіх процесів.

Якщо процеси запуснені не всі, тоді відбувається виведення повідомлення про помилку, й перехід до повторного опитування вузлів хмарної навчальної системи на готовність.

Якщо ж процеси запуснені усі, тоді відбувається запуск підсистеми контролю знань студентів.

Після цього перевіряється чи є запит на тестування.

Якщо його немає, тоді програма чекає, коли він з'явиться.

Якщо ж запит є, тоді відбувається виконання наступних дій:

– Постановка запиту на тестування у чергу.

– Породження обслуговуючого процесу, для даної сесії контролю перевірки знань.

Після цього йде ряд перевірок умов виконання тестування.

Спершу відбувається перевірка на відповідь від обслуговуючого процесу. Якщо відповідь є то відбувається передача відповіді відправнику, й програма переходить на вибір виходити з програми чи ні.

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-123.23.0007.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 67   |

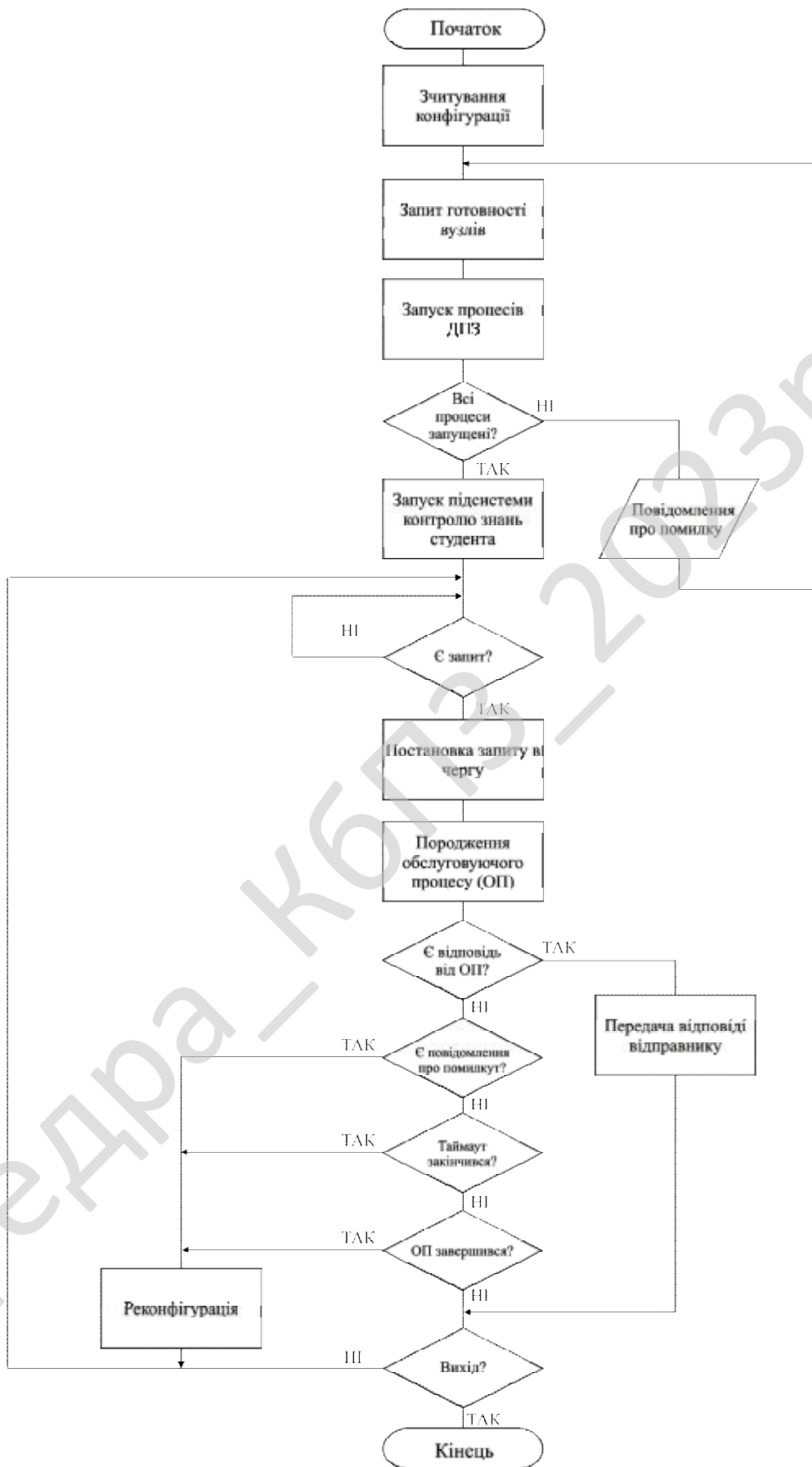


Рисунок 4.2– – Блок-схема основної програми

Якщо ж відповіді немає то відбувається перевірка на повідомлення про помилку.

Якщо таке повідомлення існує то відбувається реконфігурація системи, й програма переходить на запит на тестування.

Якщо ж такого повідомлення не існує, тоді перевіряється таймаут сесії закінчився, або ні.

Якщо він закінчився, то відбувається реконфігурація системи, й програма переходить на запит на тестування.

Якщо таймаут не закінчився, то перевіряється завершився обслуговуючий процес чи ні.

Якщо обслуговуючий процес завершився, то відбувається реконфігурація системи, й програма переходить на запит на тестування.

Якщо ж він не завершився, то користувач обирає виходити йому із системи, або ні.

Якщо він обирає вихід то програма завершує свою роботу.

У іншому випадку, тобто, якщо він хоче ще працювати з системою, то програма переходить на запит на тестування.

Розглянувши блок-схему основної програми, зображеної на рисунку 4.2, перейдемо до розгляду блок-схеми підпрограми реконфігурації, зображеної на рисунку 4.3.

Підпрограма починає діяти наступним чином.

Спершу опитується збійний вузол.

Якщо він доступний, то процес відмічається як збійний.

Якщо ж він недоступний, то вибирається процес, який потрібно перезапустити.

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-123.23.0007.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 69   |

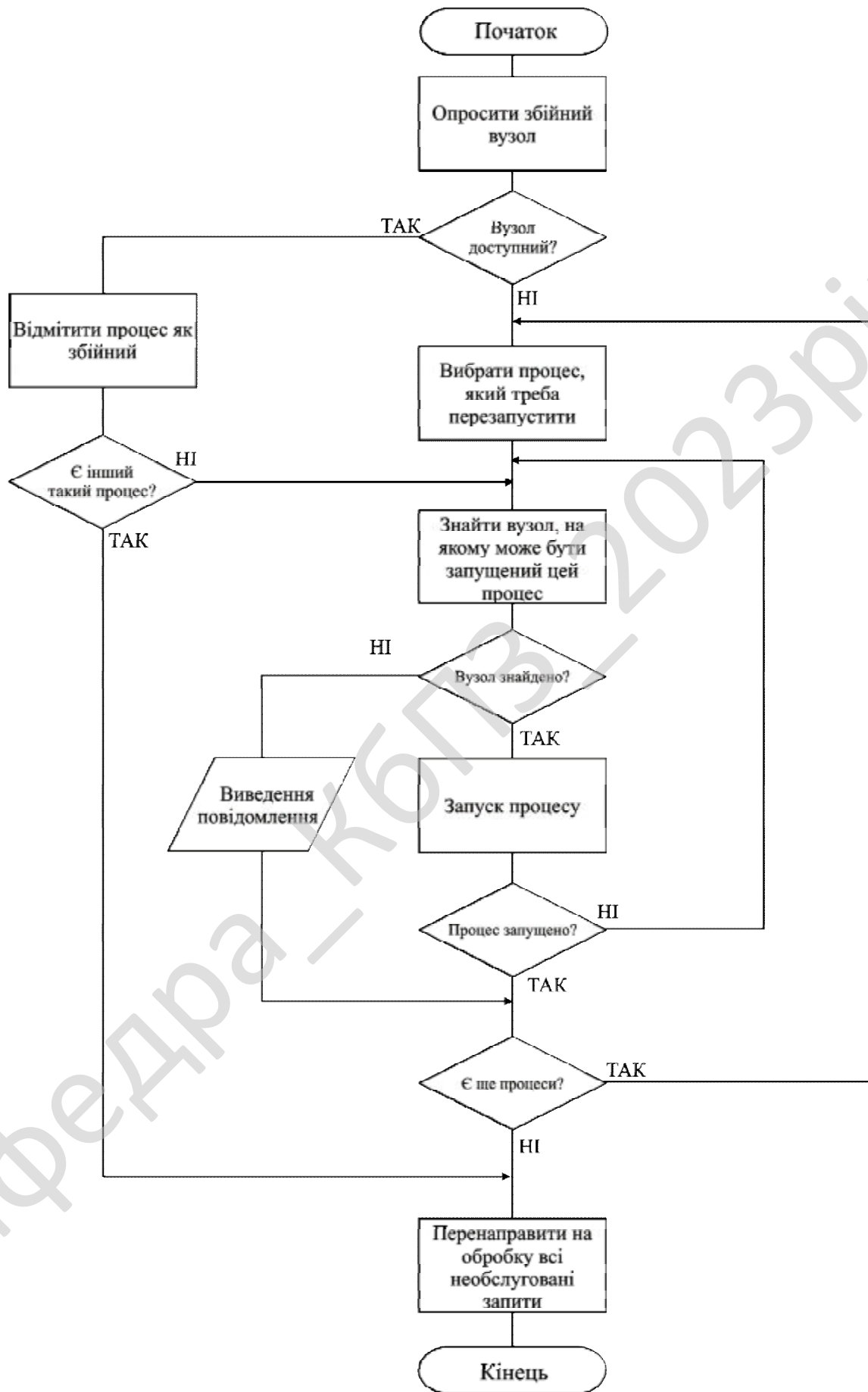


Рисунок 4.3 – Блок-схема підпрограми реконфігурації

|      |      |          |        |      |
|------|------|----------|--------|------|
|      |      |          |        |      |
| Вим. | Арк. | № докум. | Підпис | Дата |

ВКРБ-123.23.0007.00.00.ПЗ

Арк.

70

Після відмічення процесу як збійного, відбувається перевірка, чи є такий інший процес.

Якщо він є то відбувається перехід на перенаправлення на обробку усіх необслугованих запитів.

Якщо ж його немає, то відбувається знаходження вузла мережі, на якому може бути запущений цей процес.

Якщо такий вузол не знайдений, то відбувається виведення повідомлення про те, що вузол не знайдено, й відбувається перехід до вибору, чи є ще процеси.

Якщо ж такий вузол знайдено, то процес запускається.

Після цього відбувається перевірка, процес запущено, чи ні.

Якщо він не запущений то відбувається перехід на знаходження вузла мережі, на якому може бути запущений цей процес.

Якщо ж він може бути запущений, то відбувається перевірка, чи є ще процеси.

Якщо вони є то відбувається перехід на процес, який потрібно перезапустити.

Якщо ж процесів немає, то усі необслуговані запити перенаправляються на обробку.

На цьому підпрограма ре конфігурації закінчує свою роботу.

Блок-схема підпрограми допоміжного демона зображена на рисунку 4.4.

Вона працює наступним чином.

Спершу відбувається перевірка чи є запит на тестування.

Якщо його немає, то підпрограма чекає, коли він з'явиться.

Якщо він є, тоді відбувається перевірка, чи цей запит є запитом на готовність.

Якщо він є запитом на готовність, то відбувається повернення відгуку основній програмі, й система переходить у стан чекання запиту.

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРБ-123.23.0007.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 71   |

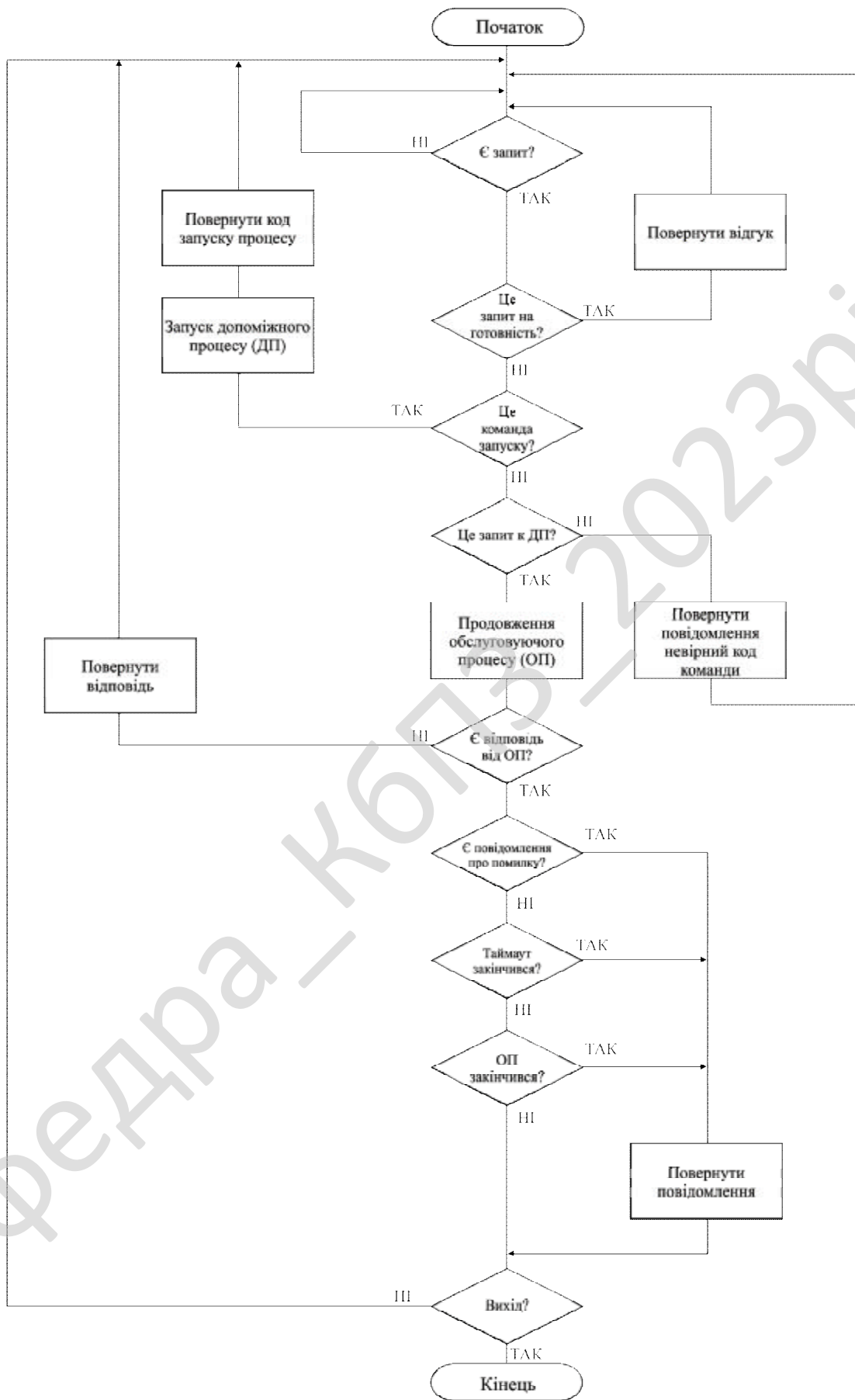


Рисунок 4.4 – Блок-схема підпрограми допоміжного демона

Якщо ж це запит не на готовність, то відбувається перевірка чи є цей запит командою запуску.

Якщо це так, то відбувається виконання наступних дій:

- Запуск допоміжного процесу.
- Повернення коду запуску процесу.
- Перехід у стан чекання запиту.

Якщо ж це не команда запуску, то відбувається перевірка, чи є це запит до допоміжного процесу.

Якщо він не є цим запитом, то відбувається виконання наступних дій:

- Повернення повідомлення про невірний код команди.
- Перехід у стан чекання запиту.

Якщо він є цим запитом, то обслуговуючий процес продовжується.

Після цього відбувається перевірка, чи є відповідь від обслуговуючого процесу.

Якщо її немає, то відбувається виконання наступних дій:

- Повертається відповідь.
- Перехід у стан чекання запиту.

Якщо ж відповідь від обслуговуючого процесу є, то відбувається ряд перевірок:

- Перевірка на повідомлення про помилку.
- Перевірка на закінчення таймаута.
- Перевірка на закінчення обслуговуючого процесу.

В усіх цих перевірках, якщо вони виконуються, то повертається повідомлення у систему, про успішну працю допоміжного демона, який реалізує процедуру контролю знань, а саме тестування рівня знань студентів.

Після цього підпрограма закінчує свою роботу.

Нижче наведемо фрагмент коду програми, який відображає головну сторінку, реалізований мовою PHP.

```
//index.php -- Програма на PHP яка відображає головну сторінку
```

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-123.23.0007.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 73   |

```

<?
session_start();
include('lib/config.php');
include(PATH_TO_LIBRARY_FILE.'functions.php');
include(PATH_TO_LIBRARY_FILE.'app.php');
include(PATH_TO_LIBRARY_FILE.'menu.php');
if(isset($_SESSION['valid_user'])) {
    $status=$_SESSION['valid_user'];
    //генерація посилань згідно профілю користувача
    switch($_SESSION['valid_user']) {
        case "1":
            //Сторінка зареєстрованого студента
                $body['link1']='home';
                $body['name_link1']='Завдання';
                $body['link2']='home';
                $body['name_link2']='Матеріали';
                $body['link3']='';
                $body['name_link3']='Форум';
                $body['link4']='';
                $body['name_link4']='Чат';
                break;
            //Сторінка викладача
            case "2":
                $body['link1']='home';
                $body['name_link1']='Завдання';
                $body['link2']='add';
                $body['name_link2']='Додати';
                $body['link3']='';
                $body['name_link3']='Форум';
                $body['link4']='';
                $body['name_link4']='Чат';
                break;
            case "3":
            //Сторінка адміністратора
                $body['link1']='home&user_status=1';
                $body['name_link1']='Студенти';
                $body['link2']='add&user_status=1';
                $body['name_link2']='NEW студент';
                $body['link3']='home&user_status=2';
                $body['name_link3']='Викладачі';
                $body['link4']='add&user_status=2';
                $body['name_link4']='NEW Викладач';
    }
}

```

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРБ-123.23.0007.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 74   |

```

        break;
    }
} else {
//Сторінка незареєстрованого користувача
    $body['link1']='home';
    $body['name_link1']='Головна';
    $body['link2']='home&special';
    $body['name_link2']='Спеціальності';
    $body['link3']='user_view';
    $body['name_link3']='Анкета';
    $body['link4']='home&history';
    $body['name_link4']='Історія';
    $status='';
}
if(isset($_GET['act'])){
    $act=$_GET['act'];
    if($act=='error'){
    } else {
        if(isset($_SESSION['valid_user'])){
//Вийти з під профілю
            if($act=='log_out'){
                $status='';
                session_unset();
            }
            if($act=='edit' || $act=='add' || $act=='motor' || $act=='new_test' || $act=='test' ||
            $act=='test_rezalts'){
                } else {
                    $act='home';
                }
            } else {
                if($act=='user_view' || $act=='motor_user'){
                } else {
                    $act='home';
                }
            }
        }
    } else {
        $act='home';
    }
}
//виклик файлу необхідні підпрограми з параметрами
$actions=PATH_TO_LIBRARY_FILE.$act.$status.'.php';

```

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРБ-123.23.0007.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 75   |

```

include($actions);
//виведення відповідної сторінки
$body['content']=parse_file(PATH_TO_HTML_FILE.$html.'.html', $variables);
echo parse_file(PATH_TO_HTML_FILE.'common.html', $body);
?>
//motor2.php -- Підпрограма додавання та редагування тестів викладачем
<?
php
$er='';
$flag_er='';
//реєстрація нового тесту
if(isset($_POST['create_test'])) {
    $mas= array('desk', 'qmax', 'subject_name');
    $name='&new_name='.$_SESSION['user_id'];
    for($w=0;$w<3;$w++) {
        if(!isset($_POST[$mas[$w]]) || empty($_POST[$mas[$w]])) {
            $er.='&er_'. $mas[$w];
        } else {
            $name.='&new_'. $mas[$w]. '='. $_POST[$mas[$w]];
        }
    }
    if($er!='') {
        if(empty($_POST['edit'])) {
            $act="add";
        } else {
            $act="edit&edit=".$_POST['edit'];
        }
        $flag_er='act='.$act.$er.$name.'&';
    } else {
        $desk=addslashes($_POST['desk']);
        $qmax=addslashes($_POST['qmax']);
        $subject_name=addslashes($_POST['subject_name']);
        // додавання нових записів
        if(empty($_POST['edit'])) {
            //додавання даних до таблиці реєстрації тестів
            $query_into = "
                INSERT INTO
                    `reg`
                SET
                    `desk` = '". $desk."',
                    `qmax` = '". $qmax."',
                    `subject_id` = '". $subject_name.",

```

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРБ-123.23.0007.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 76   |

```

        `author` = '".$_SESSION['user_id']."';
query_mysql($query_into);
// визначення номеру останнього тесту
$query_max = "
    SELECT
        `no`
    FROM
        `reg`
    WHERE
        1
    ORDER BY
        `no` DESC";
$rez_max = query_mysql($query_max);
$array_for_max=mysql_fetch_assoc($rez_max);
$max=$array_for_max['no'];
// створення таблиці питань для визначеного тесту
$create_table = "
    CREATE TABLE
        `st`.`".$max."` (
            `no` int(10) unsigned NOT NULL AUTO_INCREMENT ,
            `qust` text NOT NULL ,
            `true` varchar( 6 ) NOT NULL default '',
            `dif` tinyint( 1 ) unsigned NOT NULL default '1',
    PRIMARY KEY
        (`no`));";
query_mysql($create_table);
$flag_er='act=new_test&no='.$max.'&qmax='.$qmax;
// редагування запису
} else {
    $query_update="
        UPDATE
            `reg`
        SET
            `desk` = '". $desk."',
            `qmax` = '". $qmax."',
            `subject_id` = '". $subject_name."',
            `author` = '". $_SESSION['user_id']."'
        WHERE
            `no` = '". $_POST['edit']."'
        LIMIT 1 ";

    query_mysql($query_update);

```

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРБ-123.23.0007.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 77   |

```

    }
}
//заповнення тесту питаннями
elseif(isset($_POST['save'])) {
    $count = $_POST['count'];
    $number_query = 1;
    $add_query_insert = "('',
                        '$_POST['qust'].$number_query.'',
                        '$_POST['true'].$number_query.'',
                        '$_POST['dif'].$number_query.'')";

    while($count>1) {
        $count--;
        $number_query++;
        $add_query_insert.= ", ('',
                            '$_POST['qust'].$number_query.'',
                            '$_POST['true'].$number_query.'',
                            '$_POST['dif'].$number_query.'')";
    }
    $query_insert = "
        INSERT INTO
            `$_POST['edit']`.``
            (`no`, `qust`, `true`, `dif`)
        VALUES
            `$_POST['edit']`.``
        ";
    query_mysql($query_insert);
}
// зміна тесту та питань
elseif(isset($_POST['save_edit'])) {
    $query_sel_qust="
        SELECT
            `qmax`
        FROM
            `reg`
        WHERE
            `no` = '$_POST['edit']`.``
        LIMIT 1 ";
    $rez_sel_qust = query_mysql($query_sel_qust);
    $num_sel_row_qust = mysql_fetch_assoc($rez_sel_qust);
    $old_qmax = $num_sel_row_qust['qmax'];
    $qmax=$_POST['qmax'];

```

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРБ-123.23.0007.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 78   |

```

$query_update="
    UPDATE
        `reg`
    SET
        `desk` = '".$_POST['desk']."',
        `qmax` = '$qmax.',
        `subject_id` = '".$_POST['subject_name']."',
        `author` = '".$_SESSION['user_id']."'
    WHERE
        `no` = '".$_POST['edit']."'
    LIMIT 1 ";
query_mysql($query_update);
$new_qmax = $qmax-$old_qmax;
$add_empty_fild = '';
//додати нові поля питань
if($new_qmax>0){
    $add_empty_fild .= "
        INSERT INTO
            '".$_POST['edit']."'
            (`no`, `qust`, `true`, `dif`)
        VALUES ('', '', '', '1')";
    $new_qmax--;
    while($new_qmax>0){
        $add_empty_fild .= ", ('', '', '', '1')";
        $new_qmax--;
    }
    query_mysql($add_empty_fild);
}
// видалити питання
elseif($new_qmax<0){
    while($qmax!=$old_qmax){
        $del_fild = "DELETE
            FROM
                '".$_POST['edit']."'
            WHERE
                `no` = '$old_qmax.'"
            LIMIT 1 ";
        query_mysql($del_fild);
        $old_qmax--;
    }
}
// зберегти в БД зміни в тесті та питаннях

```

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-123.23.0007.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 79   |

```

if ($qmax==$old_qmax||$new_qmax==0) {
    $count = $old_qmax;
    $number_query = 1;
    while ($count>0) {
        $query_up=      "UPDATE
                        `".$_POST['edit']."`
                        SET
                        `qust` = '".$_POST['qust'].$number_query."',
                        `true` = '".$_POST['true'].$number_query."',
                        `dif` = '".$_POST['dif'].$number_query.'"
                        WHERE
                        `no` = ".$number_query."
                        LIMIT 1 ;";

        $count--;
        $number_query++;
        query_mysql ($query_up);
    }
}
header("Location: ../index.php?". $flag_er.$_POST['many_vars']);
?>

```

## 4.2 Захист розробленого програмного забезпечення

Захист розробленого програмного забезпечення буде відбуватися за допомогою CRYPTON – алгоритм симетричного блочного шифрування (розмір блоку 128 біт, ключ довжиною до 256 біт), розроблений південнокорейським криптологом Чьо Лім Хун з південнокорейської компанії Future Systems, яка з кінця 1980-х років працює на ринку забезпечення мереж і захисту інформації. Алгоритм був розроблений в 1998 році в якості шифру – учасника конкурсу AES. Як зізнавався автор, конструкція алгоритму спирається на алгоритм SQUARE[1]. В алгоритмі Crypton немає традиційних для блочних шифрів мережі Фейстеля. Основу даного шифру становить так звана SP-мережа (повторювана циклова функція, що складається із замін-перестановок, орієнтована на розпаралелену нелінійну обробку всього блоку даних). Крім високої швидкості, перевагами таких алгоритмів є полегшення дослідження стійкості шифру до методів

|      |      |          |        |      |                                  |           |
|------|------|----------|--------|------|----------------------------------|-----------|
|      |      |          |        |      | <b>ВКРБ-123.23.0007.00.00.ПЗ</b> | Арк.      |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | <b>80</b> |

диференціального та лінійного криптоаналізу, що є на сьогодні основними інструментами розтину блочних шифрів. На конкурс AES була представлена версія алгоритму Crypton v0.5. Однак, як казав Чьо Лім Хун, йому не вистачало часу для розробки повної версії. І вже на першому етапі конкурсу AES в ході аналізу алгоритмів, версія Crypton v0.5 була замінена на версію Crypton v1.0. Відмінність нової версії від первинної полягала в зміні таблиці замін та в модифікації процесу розширення ключа.

Як і інші учасники конкурсу AES, Crypton призначений для шифрування 128-бітових блоків даних[2]. При шифруванні використовуються ключі шифрування для декількох фіксованих розмірів – від 0 до 256 біт з кратністю 8 бітів. Структура алгоритму Crypton – структура «Квадрата» – багато в чому схожа на структуру алгоритму Square, створеного в 1997 році. Криптографічні перетворення для алгоритмів з даною структурою можуть бути виконані як для цілих рядків і стовпців масиву, так і над окремими його байтами. (Варто зазначити, що алгоритм Square був розроблений авторами майбутнього переможця конкурсу AES – авторами алгоритму Rijndael – Вінсентом Ріджменом і Джоан Дейменом.)

### **Шифрування**

Алгоритм Crypton являє 128-бітовий блок шифруємих даних у вигляді байтового масиву  $4 \times 4$ , над якими в процесі шифрування проводиться кілька раундів перетворень. У кожному раунді передбачається послідовне виконання наступних операцій:

- Таблична заміна  $\gamma$ ;
- Лінійне перетворення  $\pi$ ;
- Байтова перестановка  $\tau$ ;
- Операція  $\sigma$ .

### **Таблична заміна $\gamma$**

Алгоритм Crypton використовує 4 таблиці замін. Кожна з яких заміщає 8-бітне вхідне значення на вихідне такого ж розміру.

|      |      |          |        |      |                                  |           |
|------|------|----------|--------|------|----------------------------------|-----------|
|      |      |          |        |      | <b>ВКРБ-123.23.0007.00.00.ПЗ</b> | Арк.      |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | <b>81</b> |

### **Лінійне перетворення $\pi$**

Тут використовується 4 спеціальні константи. Ці константи об'єднані в маскуючі послідовності

### **Байтова перестановка $\tau$**

Дана перестановка перетворює найпростішим чином рядок даних у стовпець.

### **Операція $\sigma$**

Дана операція є побітовим складанням всього масиву даних з ключем раунду. Зауважимо, саме 12 раундів шифрування рекомендується автором алгоритму Чьо Хун Лімом, проте сувора кількість раундів не встановлена.

## 5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

Розроблене програмне забезпечення реалізує підсистему контролю знань в хмарних навчаючих системах.

Для використання системи потрібні наступні параметри комп'ютерного обладнання:

- Розподіляюча здатність екрану не менше 800x600 точок.
- Глибина палітри не гірше 8 біт на піксель.
- Процесор І300 або вище.
- Звукова карта.
- Відео карта з 4Мб пам'яті або більше.
- ОЗП – 128Мб та більше.
- Жорсткий диск не менше 2Гб.
- Операційна система Microsoft Windows 10/11.

На час проходження навчання студент отримує індивідуальний пароль для доступу до навчальних матеріалів на нашому сайті.

В процесі вивчення студент може познайомитися з теоретичними аспектами, а також освоїти практичні інструменти, необхідні при роботі, в системі.

Залежно від форми навчання – дистанційне або змішане – студент отримує практичні завдання для самостійно виконання або бере участь в очній складовій навчання, яка проходить у вигляді дво- триденного тренінгу, направлено за закріплення теоретичних знань і уточнення виниклих питань. У деяких курсах також заплановані групові форми роботи і виконання завдань.

Дистанційне навчання припускає активне спілкування слухачів як з координатором і викладачами курсу, так і слухачів між собою. Ви завжди можете

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-123.23.0007.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 83   |

звернутися до нас по питаннях організації курсу і змісту матеріалу. Також передбачена можливість спілкування на Форумі.

Час на вивчення матеріалів курсів в дистанційній формі обмежено. Темп навчання слухач може визначати самостійно в рамках встановлених термінів проведення курсів.

Після закінчення курсу передбачено виконання випускної роботи (ВР). На основі результатів ВР, а також загальної оцінки успішності слухача ухвалюється рішення про видачу сертифікату.

Створена комп'ютерна тестова система поєднує в собі простоту, надійність та винахідливість. Найголовніший здобуток – інтуїтивна доступність інтерфейсу для користувача, яка забезпечується науково обґрунтованими способами подання інформації (шрифт, колір, розмір, фон і нічого зайвого). Студенти можуть з першого разу опанувати систему. Винахідливість використовується для захисту системи. Видиму частину захисту складають підсистеми чергування запитань і відповідей, а прихована частина захищає систему від несанкціонованого втручання. Надійність системи забезпечує збереження усієї інформації про тестові випробування, а також постійно зростаючий масштаб завдань і відповідей на них.

Нижче зображені деякі з розроблених в результаті виконання бакалаврської роботи HTML-сторінок, які відображають та підкреслюють головні аспекти роботи системи дистанційного навчання.

Отже, на рисунку 5.1 зображена HTML-сторінка, яка призначена для реєстрації користувачів в підсистемі контролю знань (ПКЗ). HTML-сторінка містить форму для заповнення інформаційних полів, причому поля, які позначені червоною зірочкою є обов'язковими для заповнення. Інші поля можуть заповнюватися чи ні за бажанням користувача. Дана сторінка є доступною для незареєстрованого користувача і викликається шляхом переходу за посиланням «Анкета», котре знаходиться в верхньому та дублюється нижньому меню посилань.

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРБ-123.23.0007.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 84   |

Підсистема контролю знань

ГОЛОВНА | СПЕЦІАЛЬНОСТІ | АНКЕТА | ІСТОРІЯ

Вхід для ПАРТНЕРІВ

Логін:

Пароль:

Ласкаво просимо у Підсистему Контролю Знань!

### Анкета реєстрації в Підсистемі Контролю Знань

Поля відмічені \* повинні бути заповнені

\* Ім'я Фамілія:

\* Спеціальність:

\* Email:

Освіта:

Стать:  Чоловічий  Жіночий

Контактний телефон:

Адреса:

Примітки:

ГОЛОВНА | СПЕЦІАЛЬНОСТІ | АНКЕТА | ІСТОРІЯ

Рисунок 5.1 – HTML-сторінка анкети реєстрації користувачів в ПКЗ

Нижче, на рисунку 5.2 зображена HTML-сторінка, яка призначена для редагування запису щодо студента, який навчається в ПКЗ. На зображеній сторінці відображається інформація про студента у вигляді заповнених інформаційних полів форми. Поле для введення паролю при редагуванні не відображається, оскільки це закодована секретна інформація. Після проведення змін (редагування якого-небудь або усіх полів) потрібно натиснути кнопку «Зберегти» для збереження змінених даних. Важливим моментом є те, що після натиснення описаної вище кнопки, дані про пароль стираються з БД і потрібно ввести новий пароль.

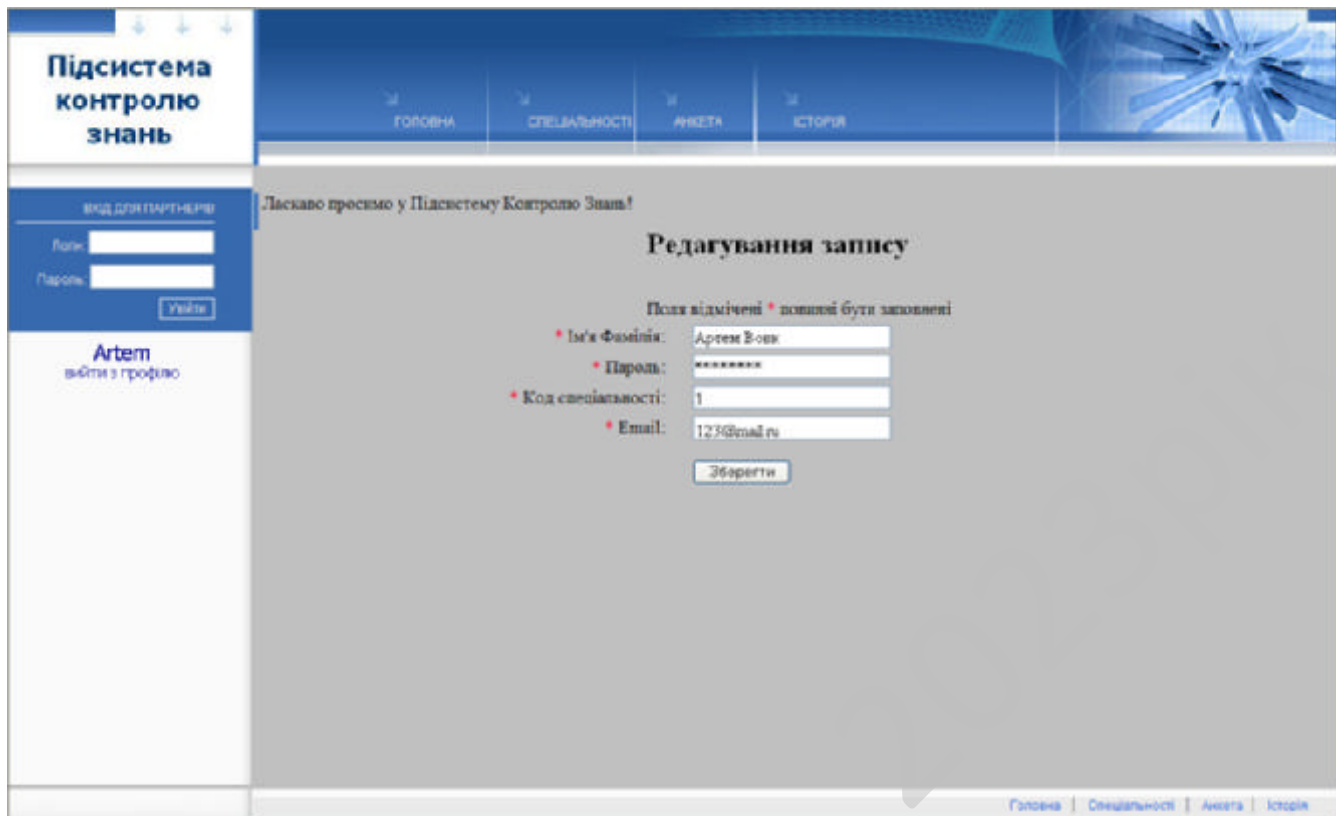


Рисунок 5.2 – HTML-сторінка редагування запису щодо студента

На рисунку 5.3 зображена HTML-сторінка, яка призначена для виведення на екран у вигляді таблиці списку усіх тестів, які доступні визначеному студенту згідно його спеціальності.

Причому, якщо студент вже пройшов тест у таблиці відображується його оцінка, інакше з'являється посилання «Пройти тест». У таблиці, окрім описаної інформації відображені дані про назву тесту, предмет до якого відноситься даний тест, а також кількість питань, яка є у цьому тесті.

Перехід до даної сторінки здійснюється одразу після того, як відбувається успішний процес авторизації студенту. Також можливий перехід до вказаної сторінки шляхом натиснення на посилання «Завдання» зі сторінок, які доступні користувачу, а саме: «Матеріали», «Форум» та «Чат».

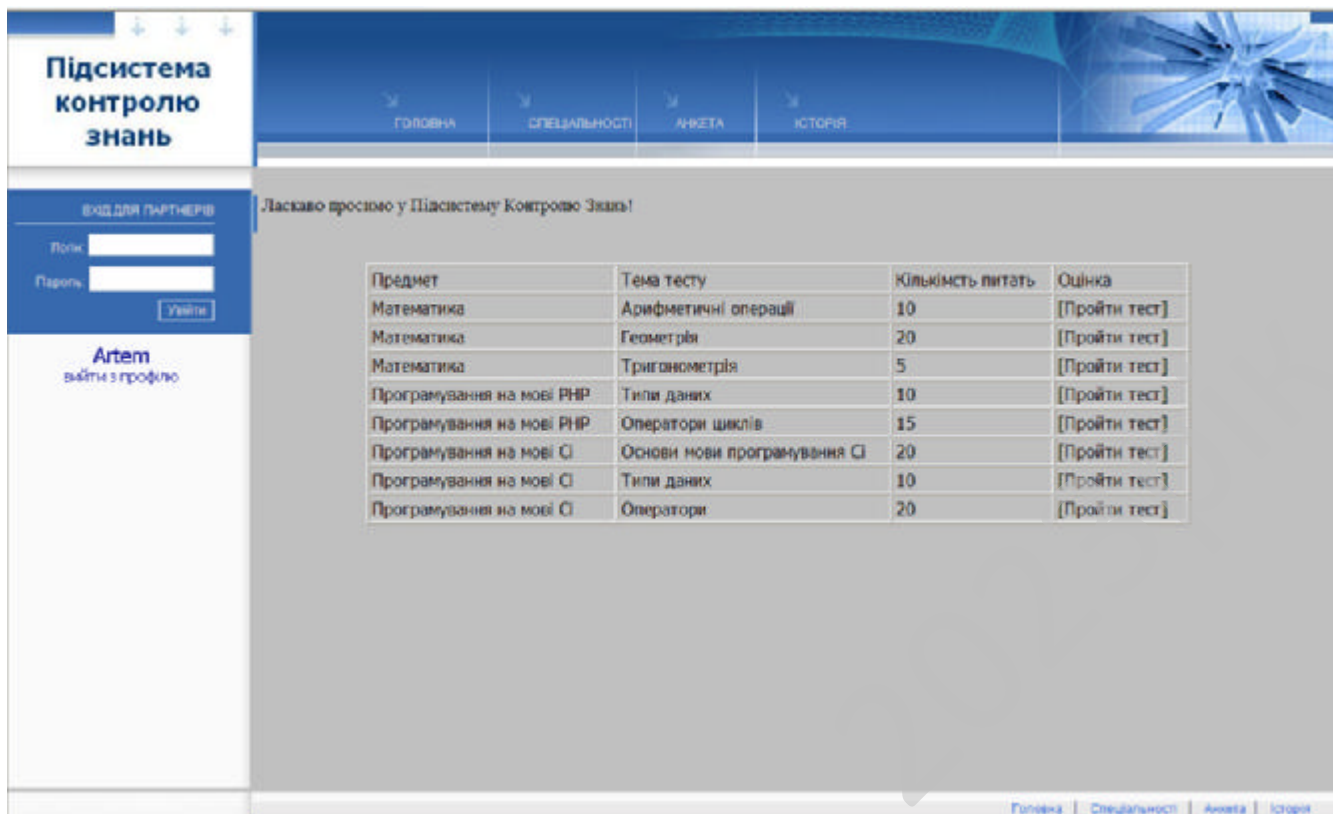


Рисунок 5.3 – HTML-сторінка виведення списку тестів студента

На рисунку 5.4 зображена HTML-сторінка, яка призначена для виведення на екран у вигляді форми-таблиці тестових питань відповідей до нього (у вигляді елементів checkbox-ів). Згори перед описаною таблицею відображується назва теми того тесту, який на даний момент складає студент.

Питання відображаються із бази даних у випадковому порядку та захищені від повторів. Кожне питання тесту містить п'ять різних варіантів відповідей. На одне питання може бути декілька правильних відповідей, які обираються шляхом натиснення студентом на відповідний елемент checkbox, який відповідає відображеній поряд відповіді. Кнопка «Подача запиту», яка відображається внизу HTML-сторінки (одразу після закінчення таблиці питань) дозволяє відправити дані (відповіді обрані студентом) для їхнього аналізу та обробки до потрібного PHP сценарію.

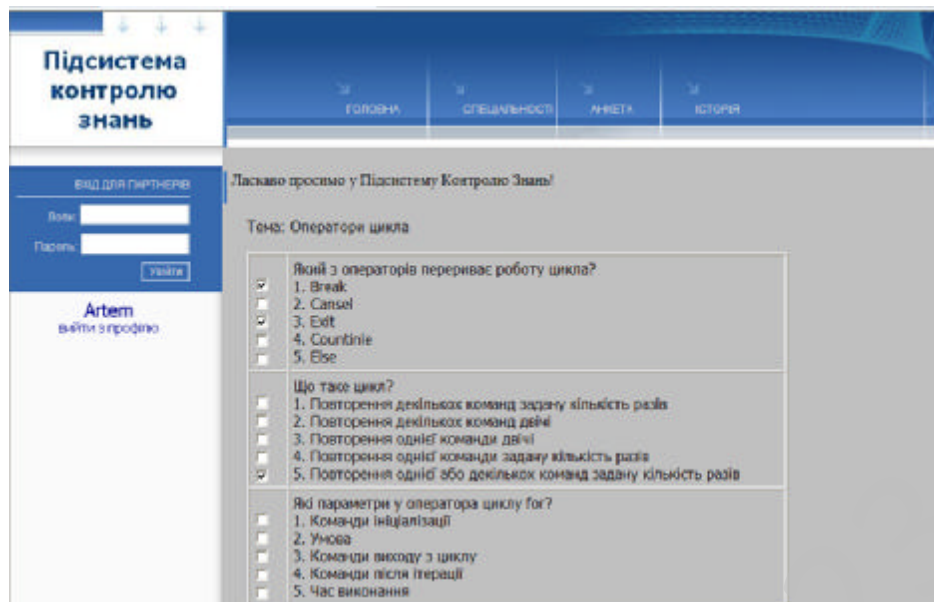


Рисунок 5.4 – HTML-сторінка проходження студентом тестування

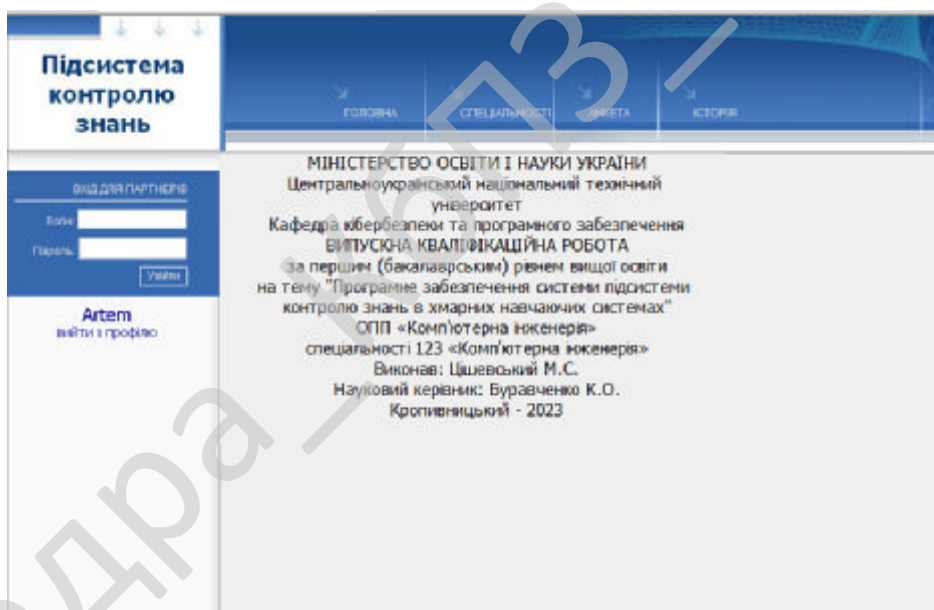


Рисунок 5.5 – Авторське право

Таким чином, програмне забезпечення, яке було розроблене в результаті виконання бакалаврської роботи має простий, дружній та зручний інтерфейс користувача. Це забезпечує легкість у освоєнні роботи програмного продукту та зручність у його використанні.

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРБ-123.23.0007.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 88   |

## 6 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти, призначено для системи підсистеми контролю знань в хмарних навчаючих системах.

В межах України в недостатній мірі представлені вітчизняні розробки в цій області.

Рішення завдання полягало у вирішенні наступних задач:

– Був проведений огляд існуючих систем підсистеми контролю знань в хмарних навчаючих системах.

– Досліджена система підсистеми контролю знань в хмарних навчаючих системах.

– На основі отриманих результатів досліджень створена програмна реалізація системи підсистеми контролю знань в хмарних навчаючих системах.

Розроблені під час виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання підсистеми контролю знань в хмарних навчаючих системах.

Розроблене програмне забезпечення має простий, дружній та зручний інтерфейс користувача, що забезпечує легкість у освоєнні роботи програмного продукту, зручність у використанні, і не потребує особливих спеціальних знань.

При створенні програмного забезпечення було використано об'єктно-орієнтований підхід, що відповідає сучасним тенденціям у галузі розробки комерційних програмних систем.

Програма реалізована на мові високого рівня РНР. Дана мова програмування дозволяє найбільш ефективно обробляти дані призначені для системи підсистеми контролю знань в хмарних навчаючих системах. Це дозволило мінімізувати строк розробки програмного забезпечення, і, як слід,

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРБ-123.23.0007.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 89   |

зменшити витрати на його розробку. Запропоноване програмне забезпечення ділиться на загальне програмне забезпечення, що поставляється із засобами обчислювальної техніки й спеціальне програмне забезпечення, що спеціально розроблене для даної конкретної системи й включає програми, що реалізують її функції.

Програма призначена для виконання під управлінням багатозадачної операційної системи Windows 10/11.

Даються необхідні рекомендації з установки розробленого програмного забезпечення.

Для підвищення рівня безпеки запропоновано застосовувати алгоритм CRYPTON.

В цілому створене програмне забезпечення підтверджує правильність використаних проектних рішень та повністю відповідає вимогам технічного завдання. Створене програмне забезпечення має потенційну можливість для подальшого вдосконалення і застосування у різних галузях.

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-123.23.0007.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 90   |

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Аванесов В .С. Научные проблемы тестового контроля знаний / Монография. -м.: Исследовательский центр проблем качества подготовки специалистов,1994.-135 с.

2. Агеев В.Н. Электронные учебники и автоматизированные обучающие системы. – М.: 2001. – 79 с.

3. Аджемов А.С. Единое образовательное пространство на основе инфотелекоммуникационных технологий // Сети и системы связи, 2001, №11. – с. 20-23.

4. Александров Г.Н. Программированное обучение и новые информационные технологии обучения. // Информатика и образование, 1993, №5. – с. 7-19.

5. Кривошеев А.О., Голомидов Г.С., Таран А.Н. Перспективные Internet-технологии информационного обеспечения образовательных услуг // Российский НИИ информационных систем, 2000.

6. Морозевич А.И., Комличенко В.Н., Гедранович В.В. Стратегия автоматизации управления познавательной деятельностью на основе информационной модели образовательного процесса // Информационные технологии, 2000, № 5.– с. 47-52.

7. Kovalenko O., Poperehnyak S., Grinenko S., Grinenko O., Radivilova T. «Methods for Assessing the Maturity Levels of Software Ecosystems». *CEUR Workshop Proceedings* Volume 2654, 2019, Pages 251-261. Режим доступа: <https://www.scopus.com/record/display.uri?eid=2-s2.0-85083214331&origin=AuthorNamesList&txGid=5633fba897776a6e0f3d5633fbcd3fbe> (Scopus).

8. Kovalenko O., Drieieva H., Simakhin V., Bondar S., Drieiev O., Zhumadilova M. «Multifractal Properties of Traffic Generator Based on Markov Chains ». *CEUR Workshop Proceedings* Volume 2588, 2019, Pages 567-579. Режим

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-123.23.0007.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 91   |

доступу: <https://www.scopus.com/record/display.uri?eid=2-s2.0-85083214331&origin=AuthorNamesList&txGid=176e2cada8976a6e0f3d5633fbcd3fbc> (**Scopus**).

9. Kovalenko Oleksandr Qualitative risk analysis of software development / Oleksandr Kovalenko, Jamil Al-Azzeh, Oleksii Smirnov, Anna Kovalenko, Serhii Smirnov // Asian Journal of Information Technology. – Volume 17 Issue 3. – Medwell Journals. – 2018. – P. 218-230. ISSN: 1682-3915. URL: <http://medwelljournals.com/abstract/?doi=ajit.2018.218.230> Doi: ajit.2018.218.230

10. Kovalenko Oleksandr, The mathematical model of the testing technology for DOM XSS vulnerabilities / O. Kovalenko, O. Smirnov, A. Kovalenko, S. Smirnov, V. Vialkova // Scientific & practical cyber security journal (SPCSJ) Volume 2 Issue 1, P. 22-28. Georgia. Tbilisi. Scientific Cyber Security Association (SCSA), 2018 ISSN: 2587-4667. URL: <https://journal.scsa.ge/wp-content/uploads/2018/12/04-3-o.kovalenko-a.kovalenko-o.smirnov-s.smirnov-v.vialkova.pdf>

11. Коваленко А.В. Технология тестирования DOM XSS уязвимости / А.В. Коваленко, А.С. Коваленко, А.А. Смирнов, С.А. Смирнов // Scientific & practical cyber security journal (SPCSJ) Volume 1. Issue 1. P. 38-45 Georgia. Tbilisi. Scientific Cyber Security Association (SCSA), 2017 ISSN: 2587-4667. URL: <https://journal.scsa.ge/wp-content/uploads/2018/12/8-dom-xss-testing-technology-vulnerabilities.pdf>

12. Коваленко О.В. Моделі та методи розроблення програмного забезпечення комп'ютерних систем для підвищення безпеки даних: монографія / О.В. Коваленко // К.: Вид. «КОД» – 2019. – 305 с.

13. Коваленко А.В. Методы качественного анализа и количественной оценки рисков разработки программного обеспечения / А.В. Коваленко, А.А. Смирнов // Информационные технологии в управлении, образовании, науке и промышленности: монография / Под редакцией профессора В.С. Пономаренко. – Х.: Видавець Рожко С.Г., 2016. – 566 с.

14. Коваленко А.В. Разработка метода управления рисками разработки

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРБ-123.23.0007.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 92   |

программного обеспечения / А.В. Коваленко, А.А. Смирнов // Інформаційні технології: проблеми та перспективи: монографія / За загальною редакцією В.С. Пономаренка. – Х.: Видавець Рожко С.Г., 2017. – 447 с.

15. Коваленко А.В. Комплекс математических моделей технологии тестирования web-приложений / А.В. Коваленко, А.А. Смирнов // Інформаційні технології: сучасний стан та перспективи: монографія / За загальною редакцією В.С. Пономаренка. – Х.: ТОВ «ДІСА ПЛЮС», 2018. – 461 с.

16. Коваленко А.В. Задачи распознавания ситуаций в ERP системах / А.В. Коваленко, А.А. Смирнов, А.С. Коваленко // Збірник наукових праць "Системи обробки інформації". – Випуск 4(120). – Х.: ХУПС – 2014. – С. 161-164.

17. Коваленко А.В. Методы качественного анализа и количественной оценки рисков разработки программного обеспечения / А.В. Коваленко, А.А. Смирнов // Збірник наукових праць "Системи обробки інформації". – Випуск 5(142). – Х.: ХУПС – 2016. – С. 153-157.

18. Коваленко А.В. Проблемы анализа и оценки рисков информационной деятельности / А.В. Коваленко, А.А. Смирнов, Н.Н. Якименко, А.П. Доренский // Збірник наукових праць "Системи обробки інформації". – Випуск 3(140). – Х.: ХУПС – 2016. – С. 40-42.

19. Коваленко А.В. Метод качественного анализа рисков разработки программного обеспечения / А.В. Коваленко, А.А. Смирнов, Н.Н. Якименко, А.П. Доренский // Наука і техніка Повітряних Сил Збройних Сил України. – Випуск 2(23). – Харків: ХУПС. – 2016. – С. 150-158.

20. Коваленко А.В. Метод количественной оценки рисков разработки программного обеспечения / А.В. Коваленко, А.А. Смирнов, Н.Н. Якименко, А.П. Доренский // Збірник наукових праць Харківського університету Повітряних Сил. Випуск 2 (47). – Харків: ХУПС. – 2016. – С. 128-133.

21. Коваленко А.В. Использование псевдобулевых методов бивалентного программирования для управления рисками разработки программного

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРБ-123.23.0007.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 93   |

обеспечения / А.В. Коваленко, А.А. Смирнов // Системи управління, навігації та зв'язку. – Випуск 1 (37). – Полтава: ПолтНТУ. – 2016. – С. 98-103.

22. Коваленко А.В. Метод управління ризиками розробки програмного забезпечення / А.В. Коваленко // Системи управління, навігації та зв'язку. – Випуск 2 (38). – Полтава: ПолтНТУ. – 2016. – С. 93-100.

23. Коваленко А.В. Технологія тестування уязвимості к SQL ін'єкціям / А.В. Коваленко // Системи управління, навігації та зв'язку. – Випуск 5 (45). – Полтава: ПолтНТУ. – 2017. – С. 66-71.

24. Коваленко А.В. Масштабирование имитационной модели технологии тестирования безопасности / А.В. Коваленко // Системи управління, навігації та зв'язку. – Випуск 6 (46). – Полтава: ПолтНТУ. – 2017. – С. 181-184.

25. Коваленко А.В. Имитационная модель технологии тестирования безопасности Web-приложений / А.В. Коваленко // Системи управління, навігації та зв'язку. – Випуск 1 (47). – Полтава: ПолтНТУ. – 2018. – С. 114-123.

26. Коваленко О.В. Методи якісного аналізу та кількісної оцінки ризиків розроблення програмного забезпечення / О.В. Коваленко // Системи управління, навігації та зв'язку. – Випуск 3 (49). – Полтава: ПолтНТУ. – 2018. – С. 116-125.

27. Коваленко О.В. Управління ризиками розроблення програмного забезпечення за умови обмеженості коштів виділених на усунення помилок безпеки / О.В. Коваленко // Техніка в сільськогосподарському виробництві, галузеве машинобудування, автоматизація. – Випуск 31. – Кропивницький: ЦНТУ. – 2018. – С. 128-140.

28. Коваленко О.В. GERT-мережевий синтез технології тестування на вразливість WEB-додатків / О.В. Коваленко // Захист інформації. – Випуск 20(2) – К.: НАУ. – 2018. – С. 89-94.

29. Коваленко О.В. Імітаційна модель технології тестування безпеки на основі положень теорії масштабування / О.В. Коваленко // Безпека інформації. – Випуск 24 (2). – К.: НАУ. – 2018. – С. 110-117.

30. Коваленко О.В. Оцінка ефективності технології тестування

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРБ-123.23.0007.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 94   |

безпеки / О.В. Коваленко // Вчені записки Таврійського національного університету імені В.І. Вернадського. Серія: Технічні науки. Том 29 (68) № 2, 2018. – С. 137-141

31. Коваленко О.В. Методи та засоби управління безпекою додатків / О.В. Коваленко // Інформаційно-керуючі системи на залізничному транспорті. №4, 2018. – С. 41-44.

32. Коваленко О.В. Розробка інформаційної технології передтестової компіляції та розподілу доступу / О.В. Коваленко // Системи управління, навігації та зв'язку. – Випуск 4 (50). – Полтава: ПолтНТУ. – 2018. – С. 115-119.

33. Коваленко О.В. Аналіз та дослідження інформаційних технологій розробки програмного забезпечення/ О.В. Коваленко // Вчені записки Таврійського національного університету імені В.І. Вернадського. Серія: Технічні науки. Том 29 (68) № 5, 2018. – С. 131-137.

34. Коваленко О.В. Удосконалений метод управління ризиками розроблення програмного забезпечення на основі напівмарковської моделі прийняття рішень/ О.В. Коваленко // Сучасні інформаційні системи. – Випуск 2(3). – Харків. – 2018. – С. 41-48.

35. Коваленко О.В. Математичні моделі технології тестування DOM XSS вразливості та вразливості до SQL ін'єкцій / О.В. Коваленко // Вісник Черкаського державного технологічного університету. Серія : Технічні науки №4, 2018. – С. 29-36.

36. Коваленко О.В. Математична модель технології тестування вразливості до SQL ін'єкцій / О.В. Коваленко // Системи управління, навігації та зв'язку. – Випуск 6 (58). – Полтава: ПолтНТУ. – 2019. – С. 43-47.

37. Коваленко О.В. Математична модель технології тестування комплексу DOM XSS вразливостей для аналітичної оцінки часових витрат / О.В. Коваленко // Центральноукраїнський науковий вісник. Технічні науки. № 2(33). с. 173-180, 2019.

38. Коваленко А.В. Проблемы анализа и оценки рисков

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРБ-123.23.0007.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 95   |

информационной деятельности / А.В. Коваленко, А.А. Смирнов // Збірник наукових праць II міжнародної науково-практичної конференції «Актуальні питання забезпечення кібернетичної безпеки та захисту інформації». м. Київ. 24-27 лютого 2016 р. – Київ: Європейський університет. – 2016. – С. 138-139.

39. Коваленко А.В. Анализ и оценка рисков разработки программного обеспечения / А.В. Коваленко, А.А. Смирнов // Збірник тез «Securitea informationala 2015-2016». Conferenta internationala (editia a XII-a). Chisinau. Moldova. 3 martie 2016. – Chisinau: ADSEM. – 2016. – Р. 96-102.

40. Коваленко А.В. Исследование источников и причин риска разработки программного обеспечения, этапов и работ, при выполнении которых возникает риск / А.В. Коваленко, А.А. Смирнов // Збірник тез VII всеукраїнської науково-практичної конференції "Інформатика та системні науки (ІСН-2016)". м. Полтава. 10-12 березня 2016 р. – Полтава.: ПУЕТ – 2016. – С. 264-266.

41. Коваленко А.В. Оценка показателя чистой приведенной стоимости для количественной оценки рисков проекта разработки программного обеспечения / А.В. Коваленко, А.А. Смирнов // Збірник тез науково-практичної конференції “Проблеми кібербезпеки інформаційно-телекомунікаційних систем”. м. Київ. 10-11 березня 2016 р. – Київ: КНУ ім. Тараса Шевченко – 2016. – С. 81-82.

42. Коваленко А.В. Методика структурной идентификации рисков разработки программного обеспечения / А.В. Коваленко, А.А. Смирнов // Збірник тез Міжнародної науково-практичної конференції «Інформаційна безпека та комп'ютерні технології» (IS&CT). м. Кіровоград. 24-25 березня 2016 р. – Кіровоград: КНТУ. – 2016. – С. 71-72.

43. Коваленко А.В. Методы качественного анализа рисков разработки программного обеспечения / А.В. Коваленко, А.А. Смирнов // Збірник тез першої міжнародної науково-практичної конференції «Проблеми науково-технічного та правового забезпечення кібербезпеки у сучасному світі» (ПНПЗК-

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРБ-123.23.0007.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 96   |

2016). м. Харків. 30 березня – 1 квітня 2016 р. – Харків: НТУ «ХПІ». – 2016. – С. 6-7.

44. Коваленко А.В. Структурная идентификация рисков разработки программного обеспечения / А.В. Коваленко, А.А. Смирнов // Збірник тез XVIII міжнародного науково-практичного семінару «Комбінаторні конфігурації та їх застосування». м. Кіровоград. 15-16 квітня 2016 р. – Кіровоград: КНТУ. – 2016. – С. 175-182.

45. Коваленко А.В. Исследование разработанной методики структурной идентификации рисков разработки программного обеспечения / А.В. Коваленко, А.А. Смирнов // Збірник тез VIII міжнародної науково-практичної конференції “Проблеми і перспективи розвитку ІТ-індустрії”. м. Харків. 28-29 квітня 2016 р. – Харків: ХНЕУ. – 2016. – С. 49.

46. Коваленко А.В. Исследование дерева рисков разработки программного обеспечения / А.В. Коваленко, А.А. Смирнов // Збірник тез III міжнародної науково-практичної конференції «Інформаційна та економічна безпека» (INFECO-2016)». м. Харків. 28-30 квітня 2016 р. – Харків: ХННІ ДВНЗ «УБС». – 2016. – С. 174-178.

47. Коваленко А.В. Методы качественного анализа и количественной оценки рисков разработки программного обеспечения / А.В. Коваленко, А.А. Смирнов, А.С. Коваленко // Сборник тезисов XII международной конференции "Стратегия качества в промышленности и образовании". г. Варна. Болгария. 30 мая – 02 июня 2016 г – Варна. ТУВ. – 2016. – С. 585-589.

48. Коваленко А.В. Разработка метода управления рисками разработки программного обеспечения / А.В. Коваленко, А.А. Смирнов, А.С. Коваленко // Матеріали Всеукраїнської науково-практичної конференції «Кібербезпека в Україні: правові та організаційні питання». м. Одеса, 21 жовтня 2016 р. – Одеса : ОДУВС, 2016. – С.146-148.

49. Коваленко А.В. Метод управления рисками разработки программного обеспечения с использованием псевдобулевых методов

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРБ-123.23.0007.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 97   |

бивалентного програмування / А.В. Коваленко, А.А. Смирнов // Матеріали Всеукраїнської науково-практичної конференції «Актуальні задачі та досягнення у галузі кібербезпеки». м. Кропивницький, 23-25 листопада 2016 року – Кропивницький: ЦНТУ, 2016. – С. 162.

50. Коваленко А.В. Псевдобулевы методы бивалентного програмування для управління ризиками розробки програмного забезпечення / А.В. Коваленко, А.А. Смирнов, А.С. Коваленко, С.А. Смирнов // Збірник наукових праць III міжнародної науково-практичної конференції «Актуальні питання забезпечення кібернетичної безпеки та захисту інформації». м. Київ. 22-25 лютого 2017 р. – Київ: Європейський університет. – 2017. – С. 158-162.

51. Коваленко А.В. Метод управління ризиками розробки програмного забезпечення / А.В. Коваленко, А.А. Смирнов // Збірник тез II науково-практичної конференції “Проблеми кібербезпеки інформаційно-телекомунікаційних систем”. м. Київ. 23-24 березня 2017 р. – Київ: КНУ ім. Тараса Шевченка – 2017. – С. 203-205.

52. Коваленко А.В. Алгоритмы анализа уязвимостей при управлении ризиками розробки програмного забезпечення / А.В. Коваленко, А.А. Смирнов, А.С. Коваленко // Conferenta internationala (editia a XIII-a). «Securitatea informationala 2017». Chisinau. Republic of Moldova. 4-5 aprilie 2017. – Chisinau: ADSEM. – 2017. – P. 19-22.

53. Коваленко А.В. Алгоритм анализа DOM XSS уязвимости при управлении ризиками розробки програмного забезпечення / А.В. Коваленко, А.А. Смирнов, А.С. Коваленко // Збірник тез дев'ятого міжнародного науково-практичного семінару «Комбінаторні конфігурації та їх застосування». м. Кропивницький 7-8 квітня 2017 р. – Кропивницький: ГЛА НАУ. – 2017. – С. 125-127.

54. Коваленко А.В. Алгоритм анализа уязвимости SQL Injection для управления ризиками розробки програмного забезпечення / А.В. Коваленко,

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРБ-123.23.0007.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 98   |

А.А. Смирнов, А.С. Коваленко // Збірник тез другої міжнародної науково-технічної конференції «Проблеми науково-технічного та правового забезпечення кібербезпеки у сучасному світі» (ПНПЗК-2017). м. Харків. 10-12 квітня 2017 р. – Харків: НТУ «ХПІ». – 2017. – С. 27.

55. Коваленко А.В. Метод управління ризиками розробки програмного забезпечення на основі алгоритмів аналізу уязвимостей / А.В. Коваленко, А.А. Смирнов, А.С. Коваленко // Збірник тез Міжнародної науково-практичної конференції «Інформаційна безпека та комп'ютерні технології» (IS&CT). м. Кіровоград. 20-22 квітня 2017 р. Кіровоград: КНТУ. – 2017. – С. 92.

56. Коваленко А.В. Алгоритмы анализа DOM XSS уязвимости и уязвимости SQL Injection при управлении ризиками розробки програмного забезпечення / А.В. Коваленко, А.А. Смирнов, А.С. Коваленко // Збірник тез ІХ міжнародної науково-практичної конференції “Проблеми і перспективи розвитку ІТ-індустрії”. м. Харків. 20-21 квітня 2017 р. – Харків: ХНЕУ. – 2017. – С. 61.

57. Kovalenko O.V. Method of testing the dom xss vulnerability / Kovalenko Oleksandr, Kovalenko Anna, Smirnov Oleksii, Smirnov Serhii // International Conference «information technologies, systems and networks ITSН-2017». Chisinau, Republic of Moldova. 17 – 18 October 2017. – Chisinau: Academy of Sciences of Moldova, Military Academy of Armed Forces “Alexandru cel Bun”. – 2017. – P. 7.

58. Коваленко О.В. Метод тестування DOM XSS уразливості / О.В. Коваленко, О.А. Смірнов, А.С. Коваленко, С.А. Смірнов // Збірник тез всеукраїнської науково-практичної інтернет-конференції «Автоматика та комп'ютерно-інтегровані технології у промисловості, телекомунікаціях, енергетиці та транспорті». м. Кропивницький. 16-17 листопада 2017 р. – Кропивницький: ЦНТУ. – 2017. – С. 198-199.

59. Коваленко О.В. GERT-модель технології тестування DOM XSS уразливості / О.В. Коваленко, А.С. Коваленко, О.А. Смірнов,

С.А. Смірнов // Збірник наукових праць IV міжнародної науково-практичної конференції «Актуальні питання забезпечення кібернетичної безпеки та захисту інформації». м. Київ. 21-24 лютого 2018 р. – Київ: Європейський університет. – 2018. – С. 65-70.

60. Коваленко О.В. Технології тестування уразливостей Web-застосунків з використанням GERT-моделі / О.В. Коваленко, А.С. Коваленко, О.А. Смірнов, С.А. Смірнов // Збірник тез всеукраїнської науково-практичної конференції "Комп'ютерні інтелектуальні системи та мережі (КІСМ-2018)". м. Кривий Ріг. 21-23 березня 2018 р. – Кривий Ріг.: ДВНЗ КНУ – 2018. – С. 227-230.

61. Коваленко А.В. Тестирование уязвимости Web-приложений к атаке вида межсайтовый скриптинг / А.В. Коваленко, А.С. Коваленко, А.А. Смирнов, С.А. Смирнов // Збірник тез «Securitea internationala 2018». Conferenta internationala (editia a XIV-a). Chisinau. Moldova. 20-21 martie 2018. – Chisinau: ADSEM. – 2018. – Р. 54-56.

62. Коваленко А.В. Комплекс математических моделей технологии тестирования web-приложений / А.В. Коваленко, А.С. Коваленко, А.А. Смирнов, С.А. Смирнов // Збірник тез X міжнародної науково-практичної конференції “Проблеми і перспективи розвитку ІТ-індустрії”. м. Харків. 19-20 квітня 2018 р. – Харків: ХНЕУ. – 2018. – С. 38.

63. Коваленко О.В. Розробка методу передтестової компіляції й розподілу доступу / О.В. Коваленко, А.С. Коваленко, О.А. Смірнов, С.А. Смірнов // Збірник наукових праць III міжнародної науково-практичної конференції “Інформаційна безпека та комп'ютерні технології”, м. Кропивницький. 19-20 квітня 2018 р. – Кропивницький: ЦНТУ. – 2018. – С. 214-215.

64. Коваленко О.В. Оцінка ефективності технологій тестування безпеки уразливостей DOM XSS й SQL-ін'єкцій / О.В. Коваленко, А.С. Коваленко, О.А. Смірнов, С.А. Смірнов // Сборник тезисов XIV международной

|      |      |          |        |      |                                  |      |
|------|------|----------|--------|------|----------------------------------|------|
|      |      |          |        |      | <b>ВКРБ-123.23.0007.00.00.ПЗ</b> | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                                  | 100  |

конференції "Стратегія якості в промисловості і освіті", Варна, Болгарія. 04-07 червня 2018 г – Варна. ТУВ. – 2018. – С. 271-274.

65. Коваленко О.В. Аналіз основних підходів математичного моделювання та методологій для забезпечення максимальних показників безпеки програмного забезпечення / О.В. Коваленко, А.С. Коваленко // Збірник наукових праць всеукр. наук.-практ. конф. здобувачів вищої освіти й молодих учених «Комп'ютерна інженерія і кібербезпека : досягнення та інновації», м. Кропивницький. 27-29 листопада

Кафедра КБПЗ – 2023 рік

|      |      |          |        |      |                           |      |
|------|------|----------|--------|------|---------------------------|------|
|      |      |          |        |      | ВКРБ-123.23.0007.00.00.ПЗ | Арк. |
| Вим. | Арк. | № докум. | Підпис | Дата |                           | 101  |

Додаток А  
(обов'язковий)

Технічне завдання

Зміст

|       |   |   |
|-------|---|---|
| 1     | Найменування та область застосування.....             | 2 |
| 2     | Підстава для розробки.....                            | 2 |
| 3     | Мета та призначення розробки.....                     | 2 |
| 4     | Джерела розробки.....                                 | 2 |
| 5     | Технічні вимоги.....                                  | 2 |
| 5.1   | Вміст проекту.....                                    | 2 |
| 5.2   | Показники призначення.....                            | 3 |
| 5.3   | Вимоги до функціональних характеристик.....           | 3 |
| 5.4   | Вимоги до архітектури.....                            | 3 |
| 5.5   | Вимоги до надійності.....                             | 3 |
| 5.6   | Умови експлуатації.....                               | 4 |
| 5.7   | Вимоги до складу та параметрів технічних засобів..... | 4 |
| 5.8   | Вимоги до інформаційної і програмної сумісності.....  | 4 |
| 5.8.1 | Обладнання.....                                       | 4 |
| 5.8.2 | Мова програмування.....                               | 4 |
| 5.8.3 | Вхідні дані.....                                      | 5 |
| 5.8.4 | Вихідні дані.....                                     | 5 |
| 6     | Вимоги до програмної документації.....                | 5 |
| 7     | Перелік документів, що розробляються.....             | 5 |
| 8     | Етапи розробки.....                                   | 6 |
| 9     | Порядок контролю та приймання.....                    | 6 |

|           |                 |             |        |      |   |      |       |         |
|-----------|-----------------|-------------|--------|------|---|------|-------|---------|
|           |                 |             |        |      | <b>ВКРБ-123.23.0007.00.00.ТЗ</b>  |      |       |         |
| Вим.      | Арк.            | № документа | Підпис | Дата |   |      |       |         |
| Розробив  | Цішевський М.С. |             |        |      | <i>Програмне забезпечення системи<br/>підсистеми контролю знань в<br/>хмарних навчальних системах</i> | Літ. | Аркуш | Аркушів |
| Перевірів | Буравченко К.О. |             |        |      |   | Б    | 1     | 6       |
| Н. Контр. | Гермак В.С.     |             |        |      | ЦНТУ КІ-19  |      |       |         |
| Затв.     | Смірнов О.А.    |             |        |      |   |      |       |         |

## 1 Найменування та область застосування

Це технічне завдання розповсюджується на розробку системи підсистеми контролю знань в хмарних навчаючих системах.

## 2 Підстава для розробки

Підставою для розробки служить завдання на випускню кваліфікаційну роботу за першим (бакалаврським) рівнем вищої освіти, видане на кафедрі кібербезпеки та програмного забезпечення (нак. № 7-02 від 5.01.2023 року).

## 3 Мета та призначення розробки

Метою випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти є розробка програмного забезпечення системи підсистеми контролю знань в хмарних навчаючих системах.

## 4 Джерела розробки

Джерелом цієї випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти є стосовна до теми література і існуючі аналоги.

## 5 Технічні вимоги

### 5.1 Склад продукції

Складниками розробки є:

- вибір і обґрунтування методів реалізації проекту;

|      |      |             |        |      |                           |      |
|------|------|-------------|--------|------|---------------------------|------|
|      |      |             |        |      | ВКРБ-123.23.0007.00.00.ТЗ | Арк. |
| Вим. | Арк. | № документа | Підпис | Дата |                           | 2    |

- розробка програмної частин системи, а також розробка взаємодії системи з ОС та з користувачем;
- розробка програми, що реалізує спроектовані алгоритми роботи системи.

## 5.2 Показники призначення

Система повинна забезпечувати:

- системи підсистеми контролю знань в хмарних навчаючих системах;
- цілісність даних у процесі роботи та при зберіганні;
- простий, інтуїтивно зрозумілий інтерфейс.

## 5.3 Вимоги до функціональних характеристик

Розроблене програмне забезпечення не повинно мати обмежень на версію драйверів та операційної системи.

## 5.4 Вимоги до архітектури

Компонент, що розробляється повинен використовувати системні засоби та апаратні засоби, що на даному етапі розвитку обчислювальної техніки найбільше поширені.

## 5.5 Вимоги до надійності

Програмні модулі написані по всім правилам, які стосуються стандартних викликів процедур, функцій, методів і форм, визначених технічною документацією на середовище розробки.

|      |      |             |        |      |                                  |      |
|------|------|-------------|--------|------|----------------------------------|------|
|      |      |             |        |      | <b>ВКРБ-123.23.0007.00.00.ТЗ</b> | Арк. |
| Вим. | Арк. | № документа | Підпис | Дата |                                  | 3    |

## 5.6 Умови експлуатації

Робочі місця користувачів ПЗ повинні задовольняти наступним умовам експлуатації:

- температура повітря: 19-20 град. по Цельсію;
- відносна вологість повітря до 80%;
- атмосферний тиск 107 кПа.

## 5.7 Вимоги до складу та параметрів технічних засобів

Програмне забезпечення повинно бути реалізоване на ПЕОМ архітектури IBM PC, працювати в ОС Windows 10/11 і з сумісними з цією платформою пристроями і прикладним програмним забезпеченням.

## 5.8 Вимоги до інформаційної і програмної сумісності

Переносність програмного забезпечення повинна бути забезпечена за рахунок його реалізації стандартного інтерфейсу взаємодії з ОС, що працюють під управлінням ОС Windows 10/11.

### 5.8.1 Обладнання

Комп'ютер Intel® Celeron/8 Mb/1.2 Gb/SVGA 14" 1Mb або сумісні з ним.

### 5.8.2 Мова програмування

Середовище PHP.

|      |      |             |        |      |                           |      |
|------|------|-------------|--------|------|---------------------------|------|
|      |      |             |        |      | ВКРБ-123.23.0007.00.00.ТЗ | Арк. |
| Вим. | Арк. | № документа | Підпис | Дата |                           | 2    |

### 5.8.3 Вхідні дані

Опис алгоритму роботи запропонованої системи.

### 5.8.4 Вихідні дані

Робоча програма.

## 6 Вимоги до програмної документації

Програмна продукція повинна бути представлена у виді опису структури даних, схем та опису алгоритму, а також текстів вихідних модулів програмного забезпечення згідно ЄСПД .

## 7 Перелік документів, що розробляються

- Структурна схема системи – 1 аркуш.
- Функціональна схема системи – 1 аркуш.
- Діаграма процесів – 1 аркуш.
- Блок-схема алгоритму роботи програми – 2 аркуша.
- Пояснювальна записка – 101 аркуш.

## 8 Етапи розробки

8.1 Збір і обробка інформації по темі випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти. Постановка задачі на виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти (складання ТЗ).

|      |      |             |        |      |                                  |      |
|------|------|-------------|--------|------|----------------------------------|------|
|      |      |             |        |      | <b>ВКРБ-123.23.0007.00.00.ТЗ</b> | Арк. |
| Вим. | Арк. | № документа | Підпис | Дата |                                  | 5    |

8.2 Проведення досліджень або експериментальних робіт для уточнення основних положень випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти.

8.3 Розробка функціональних схем, блок схем алгоритмів роботи програмного забезпечення.

8.4 Побудова схем взаємодії даних.

8.5 Створення прототипу ПЗ.

8.6 Віднаходження ПЗ, аналіз отриманих результатів.

8.7 Оформлення пояснювальної записки і виконання робіт по графічній частині.

## 11 Порядок контролю та приймання

11.1 Подання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти на попередній захист 23.05.2023 р.

11.2 Подання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти на захист 1.06.2023 р.

|      |      |             |        |      |                           |      |
|------|------|-------------|--------|------|---------------------------|------|
|      |      |             |        |      | ВКРБ-123.23.0007.00.00.ТЗ | Арк. |
| Вим. | Арк. | № документа | Підпис | Дата |                           | 6    |

Додаток Б  
(обов'язковий)

**Міністерство освіти і науки України**  
**Центральноукраїнський національний технічний університет**

ЗАТВЕРДЖУЮ

Керівник випускної кваліфікаційної роботи за  
першим (бакалаврським) рівнем вищої освіти

\_\_\_\_\_ Буравченко К.О.

*Програмне забезпечення системи підсистеми контролю знань в хмарних  
навчальних системах*

Лістинг програми

Код документу 12

Носій: CD/DVD-диск / USB-флеш-накопичувач

Загальна кількість аркушів: 52

Літера: РП

Кропивницький – 2023 року

//index.php -- Програма на PHP яка відображає головну сторінку

<?

```

session_start();
include('lib/config.php');
include(PATH_TO_LIBRARY_FILE.'functions.php');
include(PATH_TO_LIBRARY_FILE.'app.php');
include(PATH_TO_LIBRARY_FILE.'menu.php');

```

```

if(isset($_SESSION['valid_user'])){
    $status=$_SESSION['valid_user'];

```

//генерація посилань згідно профілю користувача

```

switch($_SESSION['valid_user']){
    case "1":

```

//Сторінка зареєстрованого студента

```

        $body['link1']='home';
        $body['name_link1']='Завдання';
        $body['link2']='home';
        $body['name_link2']='Матеріали';
        $body['link3']='';
        $body['name_link3']='Форум';
        $body['link4']='';
        $body['name_link4']='Чат';
        break;

```

//Сторінка викладача

```

        case "2":
            $body['link1']='home';
            $body['name_link1']='Завдання';
            $body['link2']='add';
            $body['name_link2']='Додати';
            $body['link3']='';
            $body['name_link3']='Форум';
            $body['link4']='';
            $body['name_link4']='Чат';
            break;

```

```

        case "3":

```

//Сторінка адміністратора

```

            $body['link1']='home&user_status=1';
            $body['name_link1']='Студенти';
            $body['link2']='add&user_status=1';
            $body['name_link2']='NEW студент';
            $body['link3']='home&user_status=2';
            $body['name_link3']='Викладачі';
            $body['link4']='add&user_status=2';
            $body['name_link4']='NEW Викладач';
            break;

```

```

        }
    } else {

```

```
//Сторінка незареєстрованого користувача
```

```

$body['link1']='home';
$body['name_link1']='Головна';
$body['link2']='home&special';
$body['name_link2']='Спеціальності';
$body['link3']='user_view';
$body['name_link3']='Анкета';
$body['link4']='home&history';
$body['name_link4']='Історія';
$status='';
}

if(isset($_GET['act'])){
    $act=$_GET['act'];
    if($act=='error'){

        } else {
            if(isset($_SESSION['valid_user'])){

//Вийти з під профілю

                if($act=='log_out'){
                    $status='';
                    session_unset();
                }

                if($act=='edit' || $act=='add' || $act=='motor' || $act=='new_test' || $act=='test'
                ' || $act=='test_rezalts'){

                    } else {
                        $act='home';
                    }
                } else {
                    if($act=='user_view' || $act=='motor_user'){

                    } else {
                        $act='home';
                    }
                }
            }
        } else {
            $act='home';
        }
    }

//виклик файлу необхідні підпрограми з параметрами

$actions=PATH_TO_LIBRARY_FILE.$act.$status.'.php';

include($actions);

//виведення відповідної сторінки

    $body['content']=parse_file(PATH_TO_HTML_FILE.$html.'.html', $variables);
    echo parse_file(PATH_TO_HTML_FILE.'common.html', $body);
?>
```

```
// home.php Підпрограма для створення початкової сторінки для незареєстрованого користувача
```

```
<?
```

```
$home_vars=file_func('home');
```

```
$body['admin_name']= '';
```

```
$body['log_out'] = '';
```

```
include(PATH_TO_LIBRARY_FILE.'login.php');
```

```
if(isset($_GET['history'])){
```

```
    $html='history';
```

```
} else if(isset($_GET['special'])){
```

```
    $html='special';
```

```
} else {
```

```
    $html='glav_user';
```

```
}
```

```
?>
```

Кафедра \_ КБПЗ \_ 2023 рік

```
//functions.php -- Функції для роботи підпрограм
```

```
<?
```

```
// функція для запиту до бази даних
```

```
function select_db_mysql(){
    if($result_mysql = mysql_select_db(SQL_DB)){
        return($result_mysql);
    } else {
        mail("kasper-win@mail.ru", "www.ST ERROR",
mysql_errno().": ".mysql_error());
        //header("Location: ../index.php?act=error");
    }
}

function connect_mysql(){
    if($result_mysql = mysql_connect(SQL_HOST, SQL_LOGIN,
SQL_PASS)){
        return($result_mysql);
    } else {
        mail("kasper-win@mail.ru", "www.ST book ERROR",
mysql_errno().": ".mysql_error());
        //header("Location: ../index.php?act=error");
    }
}

function query_mysql($query){
    if($result_mysql=mysql_query($query)){
        return($result_mysql);
    } else {
        mail("kasper-win@mail.ru", "www.ST ERROR",
mysql_errno().": ".mysql_error());
        //header("Location: ../index.php?act=error");
    }
}

function test_int_for_get($int){
    $int=$_GET[$int];
    if($int!=strval(intval($int))){
        header("Location: index.php");
    } else {
        return $int;
    }
}

function parse($str, $params=array()) {
    foreach($params as $k=>$v) {
        $str = str_replace("%$k%", $v, $str);
    }
    return $str;
}

function parse_file($file_name, $params = array()) {
    $str = join('', file($file_name));
    $bob= parse($str, $params);
    return $bob;
}

function select_table($user_status, $show, $page, $order_by,
$sorts){
    $start_limit=$show*$page-$show;
```

```

$send_limit=$show;
$select="
    SELECT
        *
    FROM
        `log_in`
    WHERE
        `status` = ".$user_status."
    ORDER BY ".$order_by." ".$sorts."
    LIMIT ".$start_limit.", ".$send_limit;
$result_sel = query_mysql($select);

return($result_sel);
}
function file_func($vars){
    $vars=file('templates\\'.$vars.'_vars.html');
    for($u=0;$u<count($vars);$u++){
        $vars[$u]=trim($vars[$u]);
    }
    return $vars;
}

```

// функція визначення імені користувача

```

function user_name() {
$query_admin="
    SELECT `name`
    FROM
        `log_in`
    WHERE
        `user_id` = ".$SESSION['user_id']."";
$result_admin = query_mysql($query_admin);
return($result_admin);
}

```

// для роботи під профілем викладача

```

function select_table_2($show, $page, $order_by, $sorts){
$start_limit=$show*$page-$show;
$end_limit=$show;
$select="
    SELECT
        `reg`.`no`,
        `reg`.`desk`,
        `reg`.`qmax`,
        `log_in`.`name`,
        `subject`.`subject_name`
    FROM
        `reg`, `subject`, `log_in`
    WHERE
        `log_in`.`user_id` = `reg`.`author`
    AND
        `subject`.`subject_id` = `reg`.`subject_id`
    ORDER BY
        ".$order_by." ".$sorts."
    LIMIT
        ".$start_limit.", ".$end_limit;
$result_sel = query_mysql($select);

return($result_sel);
}

```

```

function select_table_test($order_by, $sorts){
$select="
    SELECT
        `log_in`.`kurs_id`,
        `subject`.`subject_name`,

```

```

        `reg`.`no`,
        `reg`.`desk`,
        `reg`.`qmax`,
        `results`.`mark`

FROM
    `log_in`,
    `subject`,
    `special_kurs`,
    `reg`, `results`

WHERE
    `log_in`.`user_id` = ".$SESSION['user_id']."
AND
    `special_kurs`.`kurs_id` = `log_in`.`kurs_id`
AND
    `subject`.`subject_id` =
`special_kurs`.`subject_id`
AND
    `reg`.`subject_id` = `special_kurs`.`subject_id`
AND
    `results`.`no` = `reg`.`no`
AND
    `results`.`user_id` = ".$SESSION['user_id']."

ORDER BY ".$order_by." ".$sorts;
$result_sel = query_mysql($select);

return($result_sel);
}

function select_table_test1($order_by, $sorts, $select_no){
$select1="
SELECT
    `log_in`.`kurs_id`,
    `subject`.`subject_name`,
    `reg`.`no`,
    `reg`.`desk`,
    `reg`.`qmax`

FROM
    `log_in`,
    `subject`,
    `special_kurs`,
    `reg`

WHERE
    `log_in`.`user_id` = ".$SESSION['user_id']."
AND
    `log_in`.`kurs_id` = `special_kurs`.`kurs_id`
AND
    `subject`.`subject_id` =
`special_kurs`.`subject_id`
AND
    `reg`.`subject_id` = `special_kurs`.`subject_id`
    ".$select_no."

ORDER BY ".$order_by." ".$sorts;
$result_sell = query_mysql($select1);

return($result_sell);
}

?>

```

// hom3.php -- Підпрограма виведення таблиці студентів або викладачів

```

<?php
$home_vars=file_func('home');
if($_SESSION['valid_user']=='3'){
    $result_admin=user_name();
    $row_admin=mysql_fetch_assoc($result_admin);
    $body['admin_name']= $home_vars[0].$row_admin['name'];
    $body['log_out'] = $home_vars[1];
} else {
    $body['admin_name']= '';
    $body['log_out'] = '';
    include(PATH_TO_LIBRARY_FILE.'login.php');
}

if(isset($_GET['del'])){

    $sel_del="
        DELETE
        FROM
            `log_in`
        WHERE
            `user_id`=".$del."
        LIMIT 1";
    $rez_del = query_mysql($sel_del);
}

if(!isset($_GET['user_status'])){
    $user_status=1;
} else {
    $user_status=$_GET['user_status'];
}

$sort['user_id']='desc';
$sort['name']='desc';
$sort['kurs_id']='desc';
$sort['email']='desc';

$image['user_id']='';
$image['name']='';
$image['kurs_id']='';
$image['email']='';

if ($sorts==''){
    $sort[$order_by]='desc';
    $image[$order_by]='';
} elseif($sorts=='asc'){
    $sort[$order_by]='desc';
    $image[$order_by]=$home_vars[3];
} else {
    $sort[$order_by]='asc';
    $image[$order_by]=$home_vars[4];
}

//виклик функції для запиту до БД - визначення кількості отриманих завдяки
пошуку записів

$start_rez_select=select_table($user_status, 50000, 1, $order_by, $sorts);
$num_start_results=mysql_num_rows($start_rez_select);
mysql_free_result($start_rez_select);
$variables['results'].=$num_start_results;

// визначення та формування значення кількості записів, що відображаються
на одній сторінці

$col_show_lines=$home_vars[5];
$many_vars_for_show=$vars.$link_order_by.$link_sorts."show=";
if($num_start_results>10){
    if($show==10){

```

```

        $col_show_lines.="10";
    } else {
        $col_show_lines.=$many_vars_for_show.$home_vars[6];
    }
}
if($num_start_results>20){
    $col_show_lines.=" | ";
    if($show==20){
        $col_show_lines.="20";
    } else {
        $col_show_lines.=$many_vars_for_show.$home_vars[7];
    }
}
if($num_start_results>50){
    $col_show_lines.=" | ";
    if($show==50){
        $col_show_lines.="50";
    } else {
        $col_show_lines.=$many_vars_for_show.$home_vars[8];
    }
}
if($num_start_results>10){
    $col_show_lines.=" | ";
}
if($show==50000||$num_start_results<=10){
    $col_show_lines.=" All";
} else {
    $col_show_lines.=$many_vars_for_show.$home_vars[9];
}
$col_show_lines.=$home_vars[10];

// визначення та формування значення поточної сторінки

$col_page=intval($num_start_results/$show);
$test_col_page=$num_start_results%$show;
if($test_col_page>0){
    $col_page+=1;
}
$kol_page_menu=5;
$stop_page=$start_page-1+$kol_page_menu;

$many_vars_for_page=$vars.$link_show.$link_order_by.$link_sorts."page=";
if ($start_page<>1){
    $prev_puck=$start_page-3;
    $col_show_lines.=$many_vars_for_page.($stop_page-
$kol_page_menu). '&start_page='.$prev_puck.$home_vars[11];
}

for ($p=$start_page; $p<=$stop_page ; $p++){
    if ($p>$col_page){
        break 1;
    }
    if ($p==$page){
        $col_show_lines.=' '.$p;
    } else {

$col_show_lines.=$many_vars_for_page.$p.$home_vars[40].$p.$home_vars[12];
    }

}
if ($stop_page<$col_page){
    $next_puck=$start_page+3;
    $col_show_lines.=$many_vars_for_page.($start_page+5). '&start_page='.
$next_puck.$home_vars[13];
}
$col_show_lines.=$home_vars[14];

```

//виклик функції для запиту до БД - отримання потрібних даних

```

$result_select=select_table($user_status, $show, $page, $order_by,
$sorts);
$num_results=mysql_num_rows($result_select);
$many_vars_for_sorts=$vars.$link_show."sorts=";
$many_vars_for_del=$vars.$link_show.$link_sorts.$link_order_by;
$many_vars_for_view=$many_vars_for_del.$link_page.$link_start_page;

$variables['for_add']=$many_vars_for_view;

// формування таблиці згідно з правами адміністратора

if(isset($_SESSION['valid_user'])&&$_SESSION['valid_user']==3){
    if($user_status==1){

        $title_kurs=$home_vars[19].$many_vars_for_sorts.$sort['kurs_id'].$home_var
s[20].$image['kurs_id'];
        } else {
            $title_kurs='';
        }
        $variables['table']=$home_vars[15].$many_vars_for_sorts.$sort['user_
id'].$home_vars[16].$image['user_id'].

        $home_vars[17].$many_vars_for_sorts.$sort['name'].$home_vars[18].$image['n
ame'].

            $title_kurs.

        $home_vars[21].$many_vars_for_sorts.$sort['email'].$home_vars[22].$image['
email'].$home_vars[23];

        for ($j=0; $j<$num_results; $j++){
            $row=mysql_fetch_assoc($result_select);
            if($user_status==1){
                $about_kurs=$row['kurs_id'].$home_vars[25];
            } else {
                $about_kurs='';
            }
            $row['name']=htmlspecialchars(stripslashes($row['name']));
            $row['email']=htmlspecialchars(stripslashes($row['email']));

            $variables['table'].=$home_vars[24].$row['user_id'].$home_vars[25].

            $row['name'].$home_vars[25].

            $about_kurs.

            $row['email'].$home_vars[25].

            $many_vars_for_view.$home_vars[30].$user_status.$home_vars[26].$row['user_
id'].$home_vars[27].

            $many_vars_for_del."del=".$row['user_id']." ".$home_vars[28];
        }
    }
    mysql_free_result($result_select);
    $new='';
    foreach($mas_search as $kk => $vv ){
        $new.=$home_vars[39].$kk.'" value="'. $vv.'"'. $home_vars[40];
    }
    $variables['new']=$new;
    $variables['table'].=$col_show_lines;
    $html=$act;
    ?>

```

```
//motor2.php -- Підпрограма додавання та редагування тестів викладачем
```

```
<?
php
$er='';
$flag_er='';

//реєстрація нового тесту
if(isset($_POST['create_test'])){
    $mas= array('desk', 'qmax', 'subject_name');
    $name='&new_name='.$_SESSION['user_id'];
    for($w=0;$w<3;$w++){
        if(!isset($_POST[$mas[$w]]) || empty($_POST[$mas[$w]])){
            $er.='&er_'.$mas[$w];
        } else {
            $name.='&new_'.$mas[$w].'='.$_POST[$mas[$w]];
        }
    }
    if($er!=''){
        if(empty($_POST['edit'])){
            $act="add";
        } else {
            $act="edit&edit=".$_POST['edit'];
        }
        $flag_er='act='.$act.$er.$name.'&';
    } else {
        $desk=addslashes($_POST['desk']);
        $qmax=addslashes($_POST['qmax']);
        $subject_name=addslashes($_POST['subject_name']);

        // додавання нових записів
        if(empty($_POST['edit'])){

            //додавання даних до таблиці реєстрації тестів
            $query_into = "
                INSERT INTO
                `reg`
                SET
                `desk` = '". $desk."',
                `qmax` = '". $qmax."',
                `subject_id` = '". $subject_name."',
                `author` = '". $_SESSION['user_id']."'";
            query_mysql($query_into);

            // визначення номеру останнього тесту
            $query_max = "
                SELECT
                `no`
                FROM
                `reg`
                WHERE
                1
                ORDER BY
                `no` DESC";
            $rez_max = query_mysql($query_max);
            $array_for_max=mysql_fetch_assoc($rez_max);
            $max=$array_for_max['no'];

            // створення таблиці питань для визначеного тесту
            $create_table = "
                CREATE TABLE
                `st`.'" . $max.'" (
                    `no` int(10) unsigned NOT NULL
                    AUTO_INCREMENT ,
                    `qust` text NOT NULL ,
```

```

default '',
NULL default '1',
PRIMARY KEY
('no`))";

query_mysql($create_table);

$flag_er='act=new_test&no='.$max.'&qmax='.$qmax;

// редагування запису
} else {
    $query_update="
        UPDATE
        `reg`
        SET
        `desk` = ".$desk."',
        `qmax` = ".$qmax."',
        `subject_id` = ".$subject_name."',
        `author` = ".$SESSION['user_id'].",
        WHERE
        `no` = ".$POST['edit'].",
        LIMIT 1 ";
    query_mysql($query_update);
}
}
//заповнення тесту питаннями
elseif(isset($POST['save'])){
    $count = $POST['count'];
    $number_query = 1;

    $add_query_insert = "('',
        ".$POST['qust'].$number_query."',
        ".$POST['true'].$number_query."',
        ".$POST['dif'].$number_query.'")";

    while($count>1){
        $count--;
        $number_query++;
        $add_query_insert.= ", ('',
        ".$POST['qust'].$number_query."',
        ".$POST['true'].$number_query."',
        ".$POST['dif'].$number_query.'")";
    }

    $query_insert = "
        INSERT INTO
        `".$POST['edit']."`
        (`no`, `qust`, `true`, `dif`)
        VALUES
        ".$add_query_insert."
    ";
    query_mysql($query_insert);
}

// зміна тесту та питань

elseif(isset($POST['save_edit'])){

```

```

$query_sel_qust="
    SELECT
        `qmax`
    FROM
        `reg`
    WHERE
        `no` = '". $_POST['edit'] ."'
    LIMIT 1 ";
$rez_sel_qust = query_mysql($query_sel_qust);
$num_sel_row_qust = mysql_fetch_assoc($rez_sel_qust);
$sold_qmax = $num_sel_row_qust['qmax'];
$qmax=$_POST['qmax'];
$query_update="
    UPDATE
        `reg`
    SET
        `desk` = '". $_POST['desk'] ."',
        `qmax` = '". $qmax ."',
        `subject_id` = '". $_POST['subject_name'] ."',
        `author` = '". $_SESSION['user_id'] ."'
    WHERE
        `no` = '". $_POST['edit'] ."'
    LIMIT 1 ";
query_mysql($query_update);

$new_qmax = $qmax-$sold_qmax;
$add_empty_fild = '';

//додати нові поля питань

if($new_qmax>0){
    $add_empty_fild .= "
        INSERT INTO
            `". $_POST['edit'] . "`
            (`no`, `qust`, `true`, `dif`)
        VALUES ('', '', '', '1')";
    $new_qmax--;
    while($new_qmax>0){
        $add_empty_fild .= ", ('', '', '', '1')";
        $new_qmax--;
    }
    query_mysql($add_empty_fild);
}

// видалити питання

elseif($new_qmax<0){
    while($qmax!=$sold_qmax){
        $del_fild = "DELETE
            FROM
                `". $_POST['edit'] . "`
            WHERE
                `no` = '". $sold_qmax ."'
            LIMIT 1 ";
        query_mysql($del_fild);
        $sold_qmax--;
    }
}

// зберегти в БД зміни в тесті та питаннях

if($qmax==$sold_qmax||$new_qmax==0){
    $count = $sold_qmax;
    $number_query = 1;
    while($count>0){

```

```

                                $query_up=      "UPDATE
                                                                `".$_POST['edit']."`
SET
`qust` =
'"$_POST['qust'].$number_query."',
`true` =
'"$_POST['true'].$number_query."',
`dif` =
'"$_POST['dif'].$number_query.'"
                                                                WHERE
`no` = ".$number_query."
                                                                LIMIT 1 ;";

                                $count--;
                                $number_query++;
                                query_mysql($query_up);
                                }
                                }
                                header("Location: ../index.php?".$flag_er.$_POST['many_vars']);
                                ?>

```

Кафедра \_ КБПЗ \_ 2023 рік

//login.php перевіряє логін та пароль на відповідність

```
<?php
// authentication
$variables['error']='';
if(isset($_POST['login']) && isset($_POST['password'])){
    $login=addslashes($_POST['login']);
    $password=$_POST['password'];
    $q=md5($password);
    // Запит до бази даних користуваів
    $query_log="
SELECT *
FROM
`log_in`
WHERE
`name` = '". $login.'"
AND
`pass` = '". $q."'";
$result_log = query_mysql($query_log);
if(mysql_num_rows($result_log)>0){
    $row_log=mysql_fetch_assoc($result_log);
    // Встановлення сесійних змінних

    $_SESSION['valid_user']=$row_log['status']; // рівень доступу
    $_SESSION['user_id']=$row_log['user_id']; // ідентифікатор користувача

    header("Location: index.php?act=home");
} else {
    $variables['error']=join(' ', file(PATH_TO_HTML_FILE.'error_login.html'));
}
mysql_free_result($result_log);
}
?>
```

```
// config.php -- Файл використовуваних констант
```

```
<?
define('SQL_HOST','localhost');
define('SQL_LOGIN','root');
define('SQL_PASS','');
define('SQL_DB','st');
define('SHOW','10');
define('ORDER_BY','user_id');
define('PAGE','1');
define('START_PAGE','1');
define('PATH_TO_LIBRARY_FILE','lib/');
define('PATH_TO_HTML_FILE','templates/');
?>
```

Кафедра \_ КБПЗ \_ 2023 рік

**// home1.php - програма формування таблиці не пройдених тестів та пройдених з відповідною оцінкою для поточного студента**

```
<?php

    $home_vars=file_func('home1');
    $result_admin=user_name();
    $row_admin=mysql_fetch_assoc($result_admin);

    $body['admin_name']= $home_vars[0].$row_admin['name'];
    $body['log_out'] = $home_vars[1];

    if ($order_by=="user_id"){
        $order_by="subject_name";
    }

    $sort['subject_name']='desc';
    $sort['desk']='desc';
    $sort['qmax']='desc';

    $image['subject_name']='';
    $image['desk']='';
    $image['qmax']='';

    if ($sorts==''){
        $sort[$order_by]='desc';
        $image[$order_by]='';
    } elseif($sorts=='asc'){
        $sort[$order_by]='desc';
        $image[$order_by]=$home_vars[3];
    } else {
        $sort[$order_by]='asc';
        $image[$order_by]=$home_vars[4];
    }

//виклик функції для запиту до БД - отримання потрібних даних

    $result_select=select_table_test($order_by, $sorts);
    $num_results=mysql_num_rows($result_select);
    $many_vars_for_sorts=$vars.$link_show."sorts=";

    // формування таблиці

        $variables['table']=$home_vars[15].$many_vars_for_sorts.$sort['subject_name'].$home_vars[16].$image['subject_name'].

    $home_vars[17].$many_vars_for_sorts.$sort['desk'].$home_vars[18].$image['desk'].

    $home_vars[21].$many_vars_for_sorts.$sort['qmax'].$home_vars[22].$image['qmax'].$home_vars[23];
    $select_no='';
        for ($j=0; $j<$num_results; $j++){
            $row=mysql_fetch_assoc($result_select);

    $row['subject_name']=htmlspecialchars(stripslashes($row['subject_name']));

    $variables['table'].=$home_vars[24].$row['subject_name'].$home_vars[25].

    $row['desk'].$home_vars[25].

    $row['qmax'].$home_vars[25].

    $row['mark'].$home_vars[20];
```

```

$select_no.=" AND `reg`.`no` NOT LIKE ".$row['no'];
    }

mysql_free_result($result_select);

//виклик функції для запиту до БД - отримання потрібних даних

$result_select1=select_table_test1($order_by, $sorts, $select_no);
$num_results1=mysql_num_rows($result_select1);

// формування таблиці

    for ($j=0; $j<$num_results1; $j++){
        $row1=mysql_fetch_assoc($result_select1);

        $row1['subject_name']=htmlspecialchars(stripslashes($row1['subject_name']))
);

        $variables['table'].=$home_vars[24].$row1['subject_name'].$home_vars[25].
        $row1['desk'].$home_vars[25].
        $row1['qmax'].$home_vars[25].
        $home_vars[28].$row1['no'].$home_vars[19];

    }

mysql_free_result($result_select1);

$new='';
foreach($mas_search as $kk => $vv ){
    $new.=$home_vars[39].$kk.'" value="'. $vv.'"'. $home_vars[40];
}
$variables['new']=$new;

$html=$act;
?>

```

//test1.php - формує тест, тему якого обрав студент

```

<?
if(isset($_GET['ntest'])){
    $ntest=test_int_for_get('ntest');
} else {
    header("Location: ../index.php");
}
$result_admin=user_name();
$row_admin=mysql_fetch_assoc($result_admin);
$body['admin_name']= $row_admin['name'];

$add_test_vars=file_func('test1');
$body['log_out'] = $add_test_vars[3];
$html='test1';
$select_nam="
        SELECT
            `reg`.`qmax`,
            `reg`.`desk`
        FROM
            `reg`
        WHERE
            `reg`.`no` = ".$ntest;

$result_select = query_mysql($select_nam);
$row_nam=mysql_fetch_assoc($result_select);
$qmax=$row_nam['qmax'];
$variables['qvest']=$row_nam['desk'];

    $select="
        SELECT
            `".$ntest."`.`no`,
            `".$ntest."`.`qust`
        FROM
            `".$ntest`
        ORDER BY rand()
        LIMIT 0, ".$qmax;

$result_select = query_mysql($select);
$variables['table']='';
$rand_test='';

    for ($j=0; $j<$qmax; $j++){
        $row=mysql_fetch_assoc($result_select);
        $row['qust']=nl2br($row['qust']);
        $variables['table'].="<tr><td width='15px'
valign='bottom'><br>
value=1><br>
value=2><br>
value=3><br>
value=4><br>
value=5><br>
value=".$row['no'].">
                                <input type=checkbox name='".$j."_1'
                                <input type=checkbox name='".$j."_2'
                                <input type=checkbox name='".$j."_3'
                                <input type=checkbox name='".$j."_4'
                                <input type=checkbox name='".$j."_5'
                                <input type=hidden name='".$j."_6'
                                </td>
                                <td>
                                ".$row['qust']."</td></tr>";
    }
    $variables['table'].="<tr><td></td><td><input type='hidden' name='qmax'
value='".$qmax."' /><input type='hidden' name='ntest' value='".$ntest."'
/><input type='submit'></td></tr>";
?>

```

**// test\_rezalts1.php -- перевірка тестів та збереження результатів тестування**

```

<?

$qmax=$_POST['qmax'];
$TotalMax='';
$Total='';
$result_admin=user_name();
$row_admin=mysql_fetch_assoc($result_admin);
$body['admin_name']= $row_admin['name'];

$add_test_vars=file_func('test1');
$body['log_out'] = $add_test_vars[3];

$mark='';
for($j=0; $j!=$qmax; $j++){

    $answer='';
    for ($i=1; $i<6; $i++){
        if(isset($_POST[$j."_".$i])){
            $answer.=$_POST[$j."_".$i];
        }
    }

    $select_nam="
        SELECT
            `".$ntest."`.`true`,
            `".$ntest."`.`dif`
        FROM
            `".$ntest."`
        WHERE
            `".$ntest."`.`no`=".$_POST[$j."_".$i];

    $result_select=query_mysql($select_nam);
    $row_nam=mysql_fetch_assoc($result_select);
    if($row_nam['true']===$answer){
        $Total=$Total+$row_nam['dif'];
    }
    $TotalMax=$TotalMax+$row_nam['dif'];

}
$exp=($Total*100)/$TotalMax;
if ($exp>='90')
{
    $mark=5;
} else if($exp>= 75){
    $mark=4;
} else if($exp>= 50){
    $mark=3;
} else {
    $mark=2;
}

$query_update="
    INSERT INTO
        `results`
    SET
        `no` = `".$ntest."`,
        `user_id` = '".$_SESSION['user_id']."`,
        `mark` = `".$mark."`;

    query_mysql($query_update);

$variables['mark']=$mark;
$html='test_rezalts1';
?>

```

```
// new_test2.php створення нового тесту викладачем
```

```
<?

$variables['forms'] = '';
$qmax = test_int_for_get('qmax');
$number = test_int_for_get('no');
$count = 1;
$result_admin=user_name();
$row_admin=mysql_fetch_assoc($result_admin);
$body['admin_name']= $row_admin['name'];

$new_test_vars=file_func('new_test2');
$body['log_out'] = $new_test_vars[0];
$variables['count'] = $qmax;
$variables['id_edit'] = test_int_for_get('no');;

while($qmax>0){
    $variables['forms'].= $new_test_vars[1];
    for($s=2; $s<20; $s++){
        $variables['forms'].= $count.$new_test_vars[$s];
        $variables['qust_add'.$count]='';
        $variables['true_add'.$count]='';
        $variables['checked1'.$count]='checked';

        $variables['er_qust'.$count]='';
        $variables['er_true'.$count]='';
    }
    $qmax -- ;
    $count ++;
}

$html=$act;
?>
```

**// user\_view.php - формування форми та інформаційних полів для відображення анкети реєстрації користувача**

```
<?
$body['admin_name']='';
$body['log_out'] = '';
$add_test_vars=file_func('user_view');

$mas= array('name', 'spec', 'email');
$mas_message= array($add_test_vars[5], $add_test_vars[6],
$add_test_vars[7]);
$serr=0;

for($w=0;$w<3;$w++){
    if(isset($_GET['er_'.$mas[$w]])){

$variables['er_'.$mas[$w]]=$add_test_vars[0].$mas_message[$w].$add_test_va
rs[1];
        $variables[$mas[$w].'_add']='';
        $serr=1;
    } else {
        $variables['er_'.$mas[$w]]='';
        @$_variables[$mas[$w].'_add']=$_GET['new_'.$mas[$w]];
    }
}
if(isset($_GET['er_reg_email'])){
    $variables['er_email']=$add_test_vars[4].$add_test_vars[1];
    $serr=1;
}
//
if($serr==0){
    $variables['name_add']='';
    $variables['spec_add']='';
    $variables['email_add']='';
}
$variables['id_edit']='';

$html=$act;
?>
```

**//add\_test3.php - формування повідомлення про помилки та виведення  
правильно введених значень у разі помилки у іншому полі**

```

<?
if(isset($_GET['user_status'])){
    $user_status=test_int_for_get('user_status');
}
$result_admin=user_name();
$row_admin=mysql_fetch_assoc($result_admin);
$body['admin_name']= $row_admin['name'];

$add_test_vars=file_func('add_test3');
$body['log_out'] = $add_test_vars[3];
if($user_status==1){
    $variables['for_kurs_id']=$add_test_vars[16];
} else {
    $variables['for_kurs_id']=$add_test_vars[15];
}
$mas= array('name', 'pass', 'kurs_id', 'email');
$mas_message= array($add_test_vars[5], $add_test_vars[8],
$add_test_vars[6], $add_test_vars[7]);
$error=0;

for($w=0;$w<4;$w++){
    if(isset($_GET['er_'.$mas[$w]])){

$variables['er_'.$mas[$w]]=$add_test_vars[0].$mas_message[$w].$add_test_vars[1];
        $variables[$mas[$w].'_add']='';
        $error=1;
    } else {
        $variables['er_'.$mas[$w]]='';
        @$variables[$mas[$w].'_add']=$_GET['new_'.$mas[$w]];
    }
}
if(isset($_GET['er_reg_email'])){
    $variables['er_email']=$add_test_vars[4].$add_test_vars[1];
    $error=1;
}
$variables['user_status']=$user_status;
$variables['many_vars']=$link_show.$link_sorts.$link_order_by;
$variables['many_vars_for_view']=$variables['many_vars'].$link_page.$link_start_page;

?>

```

//add3.php - формування форми для додавання нових користувачів системи

```
<?
include(PATH_TO_LIBRARY_FILE.'add_test3.php');
//
if($err==0){
    $variables['name_add']='';
    $variables['kurs)id_add']='';
    $variables['email_add']='';
    $variables['pass_add']='';
}

$variables['id_edit']='';
$variables['record']=$add_test_vars[13];
$html=$act.$_SESSION['valid_user'];
?>
```

Кафедра \_ КБПЗ \_ 2023 рік

// motor.php - обробка помилок та збереження даних у БД

```
<?php
```

```
$mas= array('name', 'spec', 'email');
```

```
$er='';
```

```
$flag_er='';
```

```
$name='';
```

```
// перевірка
```

```
for($w=0;$w<3;$w++){
```

```
    if(!isset($_POST[$mas[$w]]) || empty($_POST[$mas[$w]])){
```

```
        $er.='&er_'.$mas[$w];
```

```
    } else {
```

```
        if($w==2&&!eregii("^[a-zA-Z0-9_\-]+@[a-zA-Z0-9\-]+\.[a-zA-Z0-9\-]+$",  
$_POST['email'])){
```

```
            $er.='&er_reg_email=';
```

```
        }
```

```
        $name.='&new_'.$mas[$w].'='.$_POST[$mas[$w]];
```

```
    }
```

```
}
```

```
if($er!=''){
```

```
    if(empty($_POST['edit'])){
```

```
        $act="add";
```

```
    } else {
```

```
        $act="edit&edit=".$_POST['edit'];
```

```
    }
```

```
    $flag_er='act='.$act.$er.$name.'&';
```

```
} else {
```

```
    $name=addslashes($_POST['name']);
```

```
    $kurs_id=addslashes($_POST['kurs_id']);
```

```
    $email=addslashes($_POST['email']);
```

```
    $pass=md5($_POST['pass']);
```

```
}
```

```
header("Location: ../index.php?".$flag_er);
```

```
?>
```

//edit3.php - формування форми для редагування записів адміністратором

```

<?php
$mas= array('name', 'spec', 'email');

$er='';
$flag_er='';
$name='';

// перевірка
for ($w=0;$w<3;$w++){
    if(!isset($_POST[$mas[$w]]) || empty($_POST[$mas[$w]])){
        $er.='&er_'. $mas[$w];
    } else {
        if($w==2&&!eregi("[a-zA-Z0-9_\-]+@[a-zA-Z0-9_\-]+\.[a-zA-Z0-9\-]+$", $_POST['email'])){
            $er.='&er_reg_email=';
        }
        $name.='&new_'. $mas[$w]. '='. $_POST[$mas[$w]];
    }
}

if($er!=''){
    if(empty($_POST['edit'])){
        $act="add";
    } else {
        $act="edit&edit=".$_POST['edit'];
    }
    $flag_er='act='. $act. $er. $name. '&';
} else {
    $name=addslashes($_POST['name']);
    $kurs_id=addslashes($_POST['kurs_id']);
    $email=addslashes($_POST['email']);
    $pass=md5($_POST['pass']);
}

header("Location: ../index.php?". $flag_er);

?>

```

// app.php - файл встановлення значень по замовченню із файла конфігурації або відповідно до встановлених користувачем

```

<?
/* Connecting, selecting database */
connect_mysql();
select_db_mysql();

$mas_search= array();
if(!isset($_GET['show'])) {
    $show=SHOW;
    $link_show='';
} else {
    $show=test_int_for_get('show');
    $link_show="show=".$show."&";
    $mas_search['show']=$show;
}

if(!isset($_GET['order_by'])) {
    $order_by=ORDER_BY;
    $link_order_by='';
} else {
    $order_by=$_GET['order_by'];
    $link_order_by="order_by=".$order_by."&";
    $mas_search['order_by']=$order_by;
}

if(!isset($_GET['sorts'])) {
    $sorts='';
    $link_sorts='';
} else {
    $sorts=$_GET['sorts'];
    $link_sorts="sorts=".$sorts."&";
    $mas_search['sorts']=$sorts;
}

if(!isset($_GET['page'])) {
    $page=PAGE;
    $link_page='';
} else {
    $page=test_int_for_get('page');
    $link_page="page=".$page."&";
}

if(!isset($_GET['start_page'])) {
    $start_page=START_PAGE;
    $link_start_page='';
} else {
    $start_page=test_int_for_get('start_page');
    $link_start_page="start_page=".$start_page."&";
}
//--
if(empty($_GET['country_id'])) {
    $get_country_id="";
    $link_country_id="";
} else {
    $country_id=test_int_for_get('country_id');
    $link_country_id="country_id=".$country_id."&";
    $get_country_id="
        AND
        `cities`.`country_id`=".$country_id;
}
if(empty($_GET['city_id'])) {
    $get_city_id='';
    $link_city_id='';
}

```

```
} else {
    $city_id=test_int_for_get('city_id');
    $link_city_id="city_id=".$city_id."&";
    $get_city_id="
        AND
            `users`.`city_id`=".$city_id;
}
if(empty($_GET['keywords'])) {
    $keywords='';
    $get_keywords='';
    $link_keywords='';
} else {
    $keywords=htmlspecialchars($_GET['keywords']);
    $link_keywords="keywords=".$keywords."&";
    $get_keywords="
        AND(
            `users`.`first_name`
            LIKE('%$keywords%')
            OR
            `users`.`last_name`
            LIKE('%$keywords%')
            OR
            `users`.`email`
            LIKE('%$keywords%')
        )";
}
$vars="<a href=index.php?". $link_keywords.$link_country_id.$link_city_id;
?>
```

Кафедра КБПЗ – 2023 рік

// motor3.php - обробка помилок та збереження інформації у базу даних введених адміном

```

    <?php
    $mas= array('name', 'pass', 'kurs_id', 'email');
    $er='';
    $flag_er='';
    $name='';
    // перевірка
    for ($w=0;$w<4;$w++){
        if(!isset($_POST[$mas[$w]]) || empty($_POST[$mas[$w]])){
            $er.='&er_'.$mas[$w];
        } else {
            if ($w==2&&!eregi ("^[a-zA-Z0-9_\-]+@[a-zA-Z0-9_\-]+\.[a-zA-Z0-9_\-]+$",
            $_POST['email'])){
                $er.='&er_reg_email=';
            }
            $name.='&new_'.$mas[$w].'='.$_POST[$mas[$w]];
        }
    }
    if($er!=''){
        if(empty($_POST['edit'])){
            $act="add";
        } else {
            $act="edit&edit=".$_POST['edit'];
        }
        $flag_er='act='.$act.$er.$name.'&';
    } else {
        $name=addslashes($_POST['name']);
        $kurs_id=addslashes($_POST['kurs_id']);
        $email=addslashes($_POST['email']);
        $pass=md5($_POST['pass']);
        // додавання нового запису
        if(empty($_POST['edit'])){
            $query_into = "
                INSERT INTO
                `log_in`
                SET
                    `name` = '". $name."',
                    `kurs_id` = '". $kurs_id."',
                    `email` = '". $email."',
                    `status` = '". $_POST['user_status']."',
                    `pass` = '". $pass."'";

            query_mysql($query_into);
            // редагування запису
        } else {
            $query_update="
                UPDATE
                `log_in`
                SET
                    `name` = '". $name."',
                    `kurs_id` = '". $kurs_id."',
                    `email` = '". $email."',
                    `status` = '". $_POST['user_status']."',
                    `pass` = '". $pass.'"
                WHERE
                    `user_id` = '". $_POST['edit']."'
                LIMIT 1 ";

            query_mysql($query_update);
            $name_img[0]=$_POST['edit'];
        }
    }
    header("Location:
    ../index.php?". $flag_er."user_status=".$_POST['user_status'].$_POST['many_vars']
    );
    ?>

```

// add2.php - формування сторінки для додавання нових записів про тест викладачем

```
<?
include(PATH_TO_LIBRARY_FILE.'add_test2.php');
//
if($err==0){
    $variables['no_add']='';
    $variables['desk_add']='';
    $variables['name_add']='';
    $variables['subject_name_add']='';

    $variables['qust_add']='';
    $variables['true_add']='';
    $variables['checked1']='checked';
}
$variables['id_edit']='';
$variables['record']=$add_test_vars[13];
$html=$act.$_SESSION['valid_user'];
?>
```

Кафедра \_ КБПЗ \_ 2023 рік

**//add\_test3.php - формування повідомлення про помилки та виведення правильно введених значень у разі помилки у іншому полі при роботі адміністратору**

```

<?
$result_admin=user_name();
$row_admin=mysql_fetch_assoc($result_admin);
$body['admin_name']= $row_admin['name'];

$add_test_vars=file_func('add_test');
$body['log_out'] = $add_test_vars[3];
if($user_status==1){
    $variables['for_kurs_id']=$add_test_vars[16];
} else {
    $variables['for_kurs_id']=$add_test_vars[15];
}
$mas= array('name', 'spec', 'email');
$mas_message= array($add_test_vars[5], $add_test_vars[6],
$add_test_vars[7]);
$serr=0;

for($w=0;$w<4;$w++){
    if(isset($_GET['er_'.$mas[$w]])){

$variables['er_'.$mas[$w]]=$add_test_vars[0].$mas_message[$w].$add_test_va
rs[1];
        $variables[$mas[$w].'_add']='';
        $serr=1;
    } else {
        $variables['er_'.$mas[$w]]='';
        @$variables[$mas[$w].'_add']=$_GET['new_'.$mas[$w]];
    }
}
if(isset($_GET['er_reg_email'])){
    $variables['er_email']=$add_test_vars[4].$add_test_vars[1];
    $serr=1;
}
$variables['user_status']=$user_status;
$variables['many_vars']=$link_show.$link_sorts.$link_order_by;
$variables['many_vars_for_view']=$variables['many_vars'].$link_page.$link_
start_page;
?>

```

// edit2.php - формування сторінки редагування записів про тест для користувача

```

<?
include(PATH_TO_LIBRARY_FILE.'add_test2.php');
$edit = test_int_for_get('edit');
$new_test_vars=file_func('new_test2');
$count = 1;
$variables['forms'] = '';

//визначення даних для вписування в поля - редагування опису тесту
$sel_edit="
SELECT
    `reg`.`no`,
    `reg`.`desk`,
    `reg`.`qmax`,
    `log_in`.`name`,
    `subject`.`subject_id`
FROM
    `reg`,
    `subject`,
    `log_in`
WHERE
    `reg`.`no`=".$edit."
AND
    `log_in`.`user_id`=`reg`.`author`
AND
    `subject`.`subject_id`=`reg`.`subject_id`
LIMIT 1";

$rez_edit=query_mysql($sel_edit);
$row_edit=mysql_fetch_assoc($rez_edit);
mysql_free_result($rez_edit);

if($err==0){
    $variables['desk_add']=htmlspecialchars(stripslashes($row_edit['desk
']));
    $variables['name_add']=htmlspecialchars(stripslashes($row_edit['name
']));

    $variables['qmax_add']=htmlspecialchars(stripslashes($row_edit['qmax']));
    $variables['subject_name_add']=htmlspecialchars(stripslashes($row_ed
it['subject_id']));
}

//визначення даних для вписування в поля - редагування питань тесту
$query_qust="
SELECT
    `no`,
    `qust`,
    `true`,
    `dif`
FROM
    ".$row_edit['no']."
";

$rez_qust=query_mysql($query_qust);
$num_row_qust=mysql_num_rows($rez_qust);
for($j=0; $j<$num_row_qust; $j++){
    $row_qust=mysql_fetch_assoc($rez_qust);
    $variables['forms'] .= $new_test_vars[1];
    for($s=2; $s<20; $s++){
        $variables['forms'] .= $row_qust['no'].$new_test_vars[$s];
    }
    $variables['qust_add'.$row_qust['no']] = $row_qust['qust'];
    $variables['true_add'.$row_qust['no']] = $row_qust['true'];
    $variables['checked'.$row_qust['dif'].$row_qust['no']] = 'checked';
}

```

```
        $variables['er_qust'.$row_qust['no']]='';  
        $variables['er_true'.$row_qust['no']]='';  
    }  
  
    mysql_free_result($rez_qust);  
  
    $variables['id_edit']=test_int_for_get('edit');  
    $variables['record']=$add_test_vars[14];  
    $html=$act.$_SESSION['valid_user'];  
    ?>
```

Кафедра \_ КБПЗ \_ 2023 рік

**//add\_test2.php - формування повідомлення про помилки та виведення правильно введених значень у разі помилки у іншому полі при роботі викладача**

```

<?
if(isset($_GET['user_status'])){
    $user_status=test_int_for_get('user_status');
}
$result_admin=user_name();
$row_admin=mysql_fetch_assoc($result_admin);
$body['admin_name']= $row_admin['name'];

$add_test_vars=file_func('add_test2');
$body['log_out'] = $add_test_vars[3];

$mas= array('no', 'desk', 'qmax', 'subject_name', 'qust', 'true', 'name');
$mas_message= array($add_test_vars[8], $add_test_vars[5],
$add_test_vars[6], $add_test_vars[7], $add_test_vars[9], $add_test_vars[10]);
$err=0;

for($w=0;$w<7;$w++){
    if(isset($_GET['er_'].$mas[$w])){

$variables['er_'].$mas[$w]=$add_test_vars[0].$mas_message[$w].$add_test_va
rs[1];

        $variables[$mas[$w].'_add']='';
        $err=1;
    } else {
        $variables['er_'].$mas[$w]='';
        @$variables[$mas[$w].'_add']=$_GET['new_'].$mas[$w];
    }
}

//встановлення рівня складності
if(isset($_GET['dif'])){
    $form_checked='checked'. $_GET['dif'];
    $variables[$form_checked]='checked';
}
//
$variables['many_vars']=$link_show.$link_sorts.$link_order_by;
$variables['many_vars_for_view']=$variables['many_vars'].$link_page.$link_
start_page;
?>

```

// home2.php - формування сторінки з даними про студентів для роботи викладача

```

<?php
$home_vars=file_func('home2');
if($_SESSION['valid_user']=='2'){
    $result_admin=user_name();
    $row_admin=mysql_fetch_assoc($result_admin);

    $body['admin_name']= $home_vars[0].$row_admin['name'];
    $body['log_out'] = $home_vars[1];
} else {
    $body['admin_name']= '';
    $body['log_out'] = '';
    include(PATH_TO_LIBRARY_FILE.'login.php');
}

//o-oey delete
if(isset($_GET['del'])){

    $sel_del="
        DELETE
        FROM
            `reg`
        WHERE
            `no`= '". $del. "'
        LIMIT 1";
    $rez_del = query_mysql($sel_del);

    $drop_table="
        DROP TABLE `".
            $del. "`";
    $result_del = query_mysql($drop_table);
}

$sort['no']='desc';
$sort['desk']='desc';
$sort['qmax']='qmax';
$sort['name']='desc';
$sort['subject_name']='desc';

$image['no']='';
$image['desk']='';
$image['qmax']='';
$image['name']='';
$image['subject_name']='';

if ($sorts==''){
    $sort[$order_by]='desc';
    $image[$order_by]='';
} elseif($sorts=='asc'){
    $sort[$order_by]='desc';
    $image[$order_by]=$home_vars[3];
} else {
    $sort[$order_by]='asc';
    $image[$order_by]=$home_vars[4];
}

if($order_by==ORDER_BY){
    $order_by='no';
}

//виклик функції для запиту до БД - визначення кількості отриманих завдяки
пошуку записів
$start_rez_select=select_table_2( 50000, 1, $order_by, $sorts);
$num_start_results=mysql_num_rows($start_rez_select);
mysql_free_result($start_rez_select);

```

```

$variables['results'].=$num_start_results;

// визначення та формування значення кількості записів, що відображаються
на одній сторінці
$col_show_lines=$home_vars[5];
$many_vars_for_show=$vars.$link_order_by.$link_sorts."show=";
if($num_start_results>10){
    if($show==10){
        $col_show_lines.="10";
    } else {
        $col_show_lines.=$many_vars_for_show.$home_vars[6];
    }
}
if($num_start_results>20){
    $col_show_lines.=" | ";
    if($show==20){
        $col_show_lines.="20";
    } else {
        $col_show_lines.=$many_vars_for_show.$home_vars[7];
    }
}
if($num_start_results>50){
    $col_show_lines.=" | ";
    if($show==50){
        $col_show_lines.="50";
    } else {
        $col_show_lines.=$many_vars_for_show.$home_vars[8];
    }
}
if($num_start_results>10){
    $col_show_lines.=" | ";
}
if($show==50000||$num_start_results<=10){
    $col_show_lines.=" All";
} else {
    $col_show_lines.=$many_vars_for_show.$home_vars[9];
}
$col_show_lines.=$home_vars[10];

// визначення та формування значення поточної сторінки
$col_page=intval($num_start_results/$show);
$test_col_page=$num_start_results%$show;
if($test_col_page>0){
    $col_page+=1;
}
$kol_page_menu=5;
$stop_page=$start_page-1+$kol_page_menu;

$many_vars_for_page=$vars.$link_show.$link_order_by.$link_sorts."page=";
if ($start_page<>1){
    $prev_puck=$start_page-3;
    $col_show_lines.=$many_vars_for_page.($stop_page-
$kol_page_menu).'&start_page='.$prev_puck.$home_vars[11];
}

//iz file i show too!!
for ($p=$start_page; $p<=$stop_page ; $p++){
    if ($p>$col_page){
        break 1;
    }
    if ($p==$page){
        $col_show_lines.=' '.$p;
    } else {
}

$col_show_lines.=$many_vars_for_page.$p.$home_vars[40].$p.$home_vars[12];
}

}
if ($stop_page<$col_page){
    $next_puck=$start_page+3;

```

```

        $col_show_lines.=$many_vars_for_page.($start_page+5). '&start_page=' .
$next_puck.$home_vars[13];
    }
    $col_show_lines.=$home_vars[14];

    //виклик функції для запиту до БД - отримання потрібних даних
    $result_select=select_table_2($show, $page, $order_by, $sorts);
    $num_results=mysql_num_rows($result_select);
    $many_vars_for_sorts=$vars.$link_show."sorts=";
    $many_vars_for_del=$vars.$link_show.$link_sorts.$link_order_by;
    $many_vars_for_view=$many_vars_for_del.$link_page.$link_start_page;

    $variables['for_add']=$many_vars_for_view;

    // формування таблиці згідно з правами адміністратора
    if (isset($_SESSION['valid_user']) && $_SESSION['valid_user']==2) {
    /*
    no
    desk
    name
    subject_name
    */
        $variables['table']=$home_vars[15].$many_vars_for_sorts.$sort['no'].
$home_vars[16].$image['no'].

        $home_vars[17].$many_vars_for_sorts.$sort['desk'].$home_vars[18].$image['d
esk'].

        $home_vars[30].$many_vars_for_sorts.$sort['qmax'].$home_vars[31].$image['q
max'].

        $home_vars[21].$many_vars_for_sorts.$sort['subject_name'].$home_vars[22].$
image['subject_name'].
        $home_vars[19].$many_vars_for_sorts.$sort['name'].$home_vars[20].$image['n
ame'].$home_vars[23];

        for ($j=0; $j<$num_results; $j++){
            $row=mysql_fetch_assoc($result_select);

            $row['desk']=htmlspecialchars(stripslashes($row['desk']));
            $row['author']=htmlspecialchars(stripslashes($row['name']));

            $row['subject_name']=htmlspecialchars(stripslashes($row['subject_name']));
            $variables['table'].=$home_vars[24].$row['no'].$home_vars[25].

            $row['desk'].$home_vars[25].

            $row['qmax'].$home_vars[25].

            $row['subject_name'].$home_vars[25].

            $row['name'].$home_vars[25].

            $many_vars_for_view.$home_vars[26].$row['no'].$home_vars[27].

            $many_vars_for_del."del=".$row['no']." ".$home_vars[28];
        }
    }
    mysql_free_result($result_select);
    $new='';
    foreach($mas_search as $kk => $vv ){
        $new.=$home_vars[39].$kk.'" value="'. $vv.'"'.$home_vars[40];
    }
    $variables['new']=$new;
    $variables['table'].=$col_show_lines;
    // end admin_search
    $html=$act;
    ?>

```

**// motor\_user.php - обробка помилок та відправка анкети електронною поштою адміністратору**

```

<?php
$mas= array('name', 'spec', 'email');

$er='';
$flag_er='';
$name='';

// перевірка
for($w=0;$w<3;$w++){
    if(!isset($_POST[$mas[$w]]) || empty($_POST[$mas[$w]])){
        $er.='&er_'.$mas[$w];
    } else {
        if($w==2&&!ereggi("^[a-zA-Z0-9_\-]+\@[a-zA-Z0-9\-]+\.[a-zA-Z0-9\-]+$",
$_POST['email'])){
            $er.='&er_reg_email=';
        }
        $name.='&new_'.$mas[$w].'='.$_POST[$mas[$w]];
    }
}

if($er!=''){
    if(empty($_POST['edit'])){
        $act="user_view";
    } else {
        $act="user_view&edit=".$_POST['edit'];
    }
    $flag_er='act='.$act.$er.$name.'&';
} else {
    $mail_text = '';
    $name=addslashes($_POST['name']);
    $kurs_id=addslashes($_POST['spec']);
    $email=addslashes($_POST['email']);
    $motor_user=file_func('user_view');
    $mas= array('name', 'spec', 'email');
    $mas_message= array($motor_user[5], $motor_user[6], $motor_user[7]);

    for($w=0;$w<3;$w++){
        $mail_text .=
$mas_message[$w].$motor_user[16].$_POST[$mas[$w]].$motor_user[3];
    }
    $mail_text .= $motor_user[9].$motor_user[3];
    $dod_mail_text = '';
    $dod_mas = array('osvita', 'pol', 'tel', 'adres', 'pole_text');
    $dod_mas_message= array($motor_user[10], $motor_user[11],
$motor_user[12], $motor_user[13], $motor_user[14]);

    for($w=0;$w<5;$w++){
        if(!empty($_POST[$dod_mas[$w]])){
            $dod_mail_text .=
$dod_mas_message[$w].$_POST[$dod_mas[$w]].$motor_user[3];
        }
    }
    if($dod_mail_text == ''){
        $mail_text .= $motor_user[15];
    } else {
        $mail_text .= $dod_mail_text;
    }

    mail("kasper-win@mail.ru", "Анкета", $mail_text);
}
header("Location: ../index.php?". $flag_er);
?>

```

//styles.css -- таблиці стилів для HTML

```

body {
    font-family: "Times New Roman", "Times", "serif";
    font-size: 100%;
    margin: 0;
}
img {
    border: 0;
}
.zvezda {
    color: #FF0033;
}
.right_pad {
    padding-right: 20px;
    text-align: right;
}
.left_pad {
    padding-left: 20px;
    text-align: left;
}
.right_pad_10 {
    padding-right: 10%;
    text-align: right;
}
.cent {
    text-align: center;
}
.topp {
    vertical-align: top;
    padding-top: 10px;
}
.for_error {
    color: #FF0033;
    font-weight: bold;
    text-align: center;
}
.for_view {
    line-height: 1.5;
}
.pixel{ background:url(images/1X.jpg) }

.font{
    font-size: 11px;
    font-family: Arial, Helvetica, sans-serif;
    color:#E6E6E6;
    text-align:right;
}
.www {
    color: #1E60D8;
    font-size: 12px;
}
.line{border-bottom:solid 1px white; padding-bottom:5px}

.knopka{ background:#3769AE; border:solid 1px white; text-align:center; }
.niz{padding-bottom:5px}
.adres{
    text-align:right;
    padding-right:30px;
    font-size: 11px;
    font-family: Arial, Helvetica, sans-serif;
    color:#444444
}
A:link, A:visited, A:active {text-decoration:none; font-family:Arial,
Helvetica, sans-serif; color:#1E60D8; font-size:11px}
A:hover {text-decoration:none; font-family:Arial, Helvetica, sans-serif;
color:#B6D3FB; font-size:11px}

```

```
.otstup {padding-right:25;}

A.top, A.top:visited {color:#E0E0E0;}

A.top:hover {color: #FFFfff;}
a.mail:link{ text-decoration:underline}
.upper{
    text-transform:uppercase;
}
```

Кафедра \_ КБПЗ \_ 2023рік

```

//styles1.css -- таблиці стилів для HTML

body {
    font-family: "Times New Roman", "Times", "serif";
    font-size: 100%;
    margin: 0;
}
img {
    border: 0;
}
.zvezda {
    color: #FF0033;
}
.right_pad {
    padding-right: 20px;
    text-align: right;
}
.left_pad {
    padding-left: 20px;
    text-align: left;
}
.right_pad_10 {
    padding-right: 10%;
    text-align: right;
}
.cent {
    text-align: center;
}
.topp {
    vertical-align: top;
    padding-top: 10px;
}
.for_error {
    color: #FF0033;
    font-weight: bold;
    text-align: center;
}
.for_view {
    line-height: 1.5;
}
.pixel{ background:url(images/1X.jpg) }
.font{
    font-size: 11px;
    font-family: Arial, Helvetica, sans-serif;
    color:#E6E6E6;
    text-align:right;
}
.www {
    color: #1E60D8;
    font-size: 12px;
}
.line{border-bottom:solid 1px white; padding-bottom:5px}
.knopka{ background:#3769AE; border:solid 1px white; text-align:center; }
.niz{padding-bottom:5px}
.adres{
    text-align:right;
    padding-right:30px;
    font-size: 11px;
    font-family: Arial, Helvetica, sans-serif;
    color:#444444 }
A:link, A:visited, A:active {text-decoration:none; font-family:Arial,
Helvetica, sans-serif; color:#1E60D8; font-size:11px}
A:hover {text-decoration:none; font-family:Arial, Helvetica, sans-serif;
color:#B6D3FB; font-size:11px}
.otstup {padding-right:25;}
A.top, A.top:visited {color:#E0E0E0;}
A.top:hover {color: #FFFfff;}
a.mail:link{ text-decoration:underline}

```

```

<!-- common.html - оболонка сторінок-->

<html> <head>
    <title>Система тестування</title>
    <meta http-equiv="Тип-зміст" content="text/html; charset=windows-
1251">
    <link rel="stylesheet" type="text/css" href="styles.css">
</head> <body>
<div>
<table border=0 cellpadding=0 cellspacing=0 width="100%" height="100%" >
<tr class="pixel" height="134">
    <td width="210" height="134"> <div style="position: absolute;
top:0px; z-index:100" > </div></td>
    <td><div style="position: absolute; top:0px; left:210px; z-index:100"
> </div></td>
    <td height="134"> <div style="position: absolute; top:67px; left:28%;" ></div>
    <div style="position: absolute; top:90px; left:29%;"><a
href="index.php?act=%%link1%" class="top upper">%%name_link1%</a></div>
    <div style="position: absolute; top:40px; left:36%;"></div>
    <div style="position: absolute; top:67px; left:38%;"></div>
    <div style="position: absolute; top:90px; left:39%;"><a
href="index.php?act=%%link2%" class="top upper">%%name_link2%</a></div>
    <div style="position: absolute; top:40px; left:47%"></div>
    <div style="position: absolute; top:67px; left:49%;" ></div>
    <div style="position: absolute; top:90px; left:50%;"><a
href="index.php?act=%%link3%" class="top upper">%%name_link3%</a></div>
    <div style="position: absolute; top:40px; left:56%;"></div>
    <div style="position: absolute; top:67px; left:58%;"></div>
    <div style="position: absolute; top:90px; left:59%;"><a
href="index.php?act=%%link4%" class="top upper">%%name_link4%</a></div>
    <!--<div style="position: absolute; top:40px; left:54%;"></div>--> </td>
</tr>
<tr>
    <td height="24"> </td>
    <td width="6"></td>
    <td ></td>
</tr>
<tr >
    <td height="110" rowspan="2" valign="top"
background="images/X1.jpg" > <!-- ВХІД ДЛЯ ПАРТНЕРІВ-->
    <table width="210" height="120" border=0 cellpadding=0
cellspacing=6 background="images/lX1.jpg">
        <form action="index.php" method="post">
            <tr>
                <td class="font line" >ВХІД ДЛЯ ПАРТНЕРІВ </td>
                <td width="20"></td>
            </tr>
            <tr>
                <td class="font">Логін:
                <input type="text" maxlength=25 size=15 name="login"></td>
            </tr>
            <tr>
                <td class="font">Пароль:
                <input type="password" maxlength=15 size=15
name="password"></td>
            </tr>
            <tr>
                <td align="right" class="niz"><input type="submit

```

```

name="press" value="Увійти" class="font knopka" ></td>
    </tr>
    </form >
    </table>
    <table align="center">
    <tr><td><b>%admin_name%</b></td></tr>
    <tr><td align="center"><a href="index.php?act=log_out" onClick='return
for_log_out()'><b>%log_out%</b></a></td></tr>
    </table></td>
    <td valign="top">
</td>
    <td colspan="2" rowspan="3" valign="top" > &nbsp; Ласкаво просимо в
систему тестування!<br />
    <br />
    %%content%%</td>
</tr>
<tr>
<tr>
    <td height="110">&nbsp; </td>
</tr>
<tr>
    <td height="80" background="images/X1.jpg"
    <td >&nbsp; </td>
</tr>
<tr height="75">
    <td colspan="2" background="images/niz_left.jpg" height="75" >
<table height=75 width="148" align="center">
    <tr>
    <td height="40">&nbsp;</td>
</tr>
    <tr>
    <td><div align="center"><a href="index.php"></a> </div></td>
    <td><div align="center"> </div></td>
    <td><div align="center"><a href="index.php"></a> </div></td>
    <td><div align="center"> </div></td>
    <td><div align="center"><a href="mailto:kasper-
win@mail.ru"></a> </div></td>
</tr>
    </table></td>
    <td colspan="2" background="images/niz_right.jpg"> <table
width="100%" height="72" cellpadding="0">
    <tr height="10" >
    <td height="18" colspan="5" align="right" class="otstup">
    <a href="index.php?act=%%link1%"> %%name_link1% </a>
><span class="style1">&nbsp; | &nbsp; </span>
    <a href="index.php?act=%%link2%">%%name_link2%
</a>&nbsp; | &nbsp; </span>
    <a href="index.php?act=%%link3%">
%%name_link3%</a>&nbsp; | &nbsp; </span>
    <a href="index.php?act=%%link4%">%%name_link4% </a>
</td>
    </tr>
    <tr >
    <td height="29" width="60" valign="middle"
align="center"></td>
    <td class="text">Розроблено<br>
Вовком Артемом Васильовичем </td>
    <td width="68" valign="middle" align="center" ></td>
    <td class="text" >КНТУ 2010<br>
    Всі права захищені.</td>
    <td width="25" ></td>
    </tr>
    </table></td>
</tr>
</table>
</div> </body> </html>

```

```
<!-- home.html -->
```

```
<script language="JavaScript">
<!--
function for_del(){
    if(confirm("Ви дійсно хочете видалити цей запис?")){
        return true;
    } else{
        return false;
    }
}
function for_log_out(){
    if(confirm("Ви дійсно хочете вийти з даної форми?")){
        return true;
    } else{
        return false;
    }
}
-->
</script>
</head>
<body>
    <div class="right_pad_10">%%results%% записів <br />
    </div>
    %%new%%
    <table align="center" width="90%" border="1">
        %%table%%
    </table>
</form>
```

Кафедра \_ КБПЗ \_ 2023 рік

<!--test.html- сторінка тестування-->

```

<script language="JavaScript">
<!--
function for_log_out(){
    if(confirm("Ви дійсно хочете вийти з даної форми? ")){
        return true;
    } else{
        return false;
    }
}
-->
</script>
</head>
<body>
<form name="add_form" action="index.php?act=motor" method="post"
enctype="multipart/form-data" >
    <input type="hidden" name="edit" value="%%id_edit%%">
    <input type="hidden" name="many_vars" value="%%many_vars%%">
    <table align="center">
        <tr>
            <th <h2 align="center">%%record%%</h2></th>
        </tr>
    </table>
    <table align="center">
        <tr>
            <td colspan="5" align="center" >Поля відмічені <span
class="zvezda">* </span> повинні бути заповнені</td>
        </tr>

        %%er_desk%%
        <tr>
            <td width="180" height="24" class="right_pad"><label
for="desk"><span class="zvezda">* </span>Назва тесту:</label></td>
            <td><input name="desk" id="desk" value="%%desk_add%" type=text
size="50">

            <td>&nbsp;</td>

            %%er_subject_name%%
            <td class="right_pad"><label for="subject_name"><span
class="zvezda">* </span>Предмет:</label></td>
            <td><input name="subject_name" id="subject_name"
value="%%subject_name_add%" type=text size="25"></td>
        </tr>
        <tr>
            <td width="180" height="24" class="right_pad"><label
for="name">Автор тесту:</label></td>
            <td><input name="name" id="name" value="%%name_add%"
type=text size="35">
            <td>&nbsp;</td>

        %%er_qmax%%
            <td class="right_pad"><label for="qmax"><span class="zvezda">*
</span>Кількість питань:</label></td>
            <td><input name="qmax" id="subject_name" value="%%qmax_add%"
type=text size="5"></td>
        </tr>

        <tr>
            <td colspan="5"> <hr width="80%"> </td>
        </tr>

    <table align="center">
        <tr>

```

```

        <td> Введіть питання:</td>
    </tr>
    <tr>
        %%er_qust%%
        <td> <textarea name="qust" id="qust" rows=7 cols="80%"
>%%qust_add%%</textarea></td>
    </tr>
    <tr>
        %%er_true%%
        <td> Комбінація правильних відповідей: <input name="true"
id="true" value="%%true_add%%" type="text" size="5"></td>
    </tr>
    <tr>
        <td> Складність питання: <input name="dif" id="dif"
type="radio" value="1" %%checked1%%> Легке
        <input name="dif" id="dif"
type="radio" value="2" %%checked2%%> Складне
        <input name="dif" id="dif"
type="radio" value="3" %%checked3%%> Дуже складне</td>

    </table>
    <table align="center">
    <tr>
        <td align="center" colspan="2" > <br />
        <input name="save" value="Зберегти" type="submit"></td>
    </tr>
    </td>
    </table>
</form>

```

Кафедра \_ КБПЗ \_ 2023 рік

<!-- add3.html- сторінка для додавання нового користувача-->

```

<script language="JavaScript">
<!--
function for_log_out(){
    if(confirm("Ви дійсно хочете вийти з даної форми?")){
        return true;
    } else{
        return false;
    }
}
-->
</script>
</head>
<body>
<form name="add_form" action="index.php?act=motor" method="post"
enctype="multipart/form-data" >
<input type="hidden" name="user_status" value="%%user_status%%">
<input type="hidden" name="edit" value="%%id_edit%%">
<input type="hidden" name="many_vars" value="%%many_vars%%">
<table align="center">
<tr>
<th> <h2 align="center">%%record%%</h2></th>
</tr>
</table>
<table align="center">
<tr>
<td colspan="2" align="center" >Поля відмічені <span
class="zvezda">* </span> повинні бути заповнені</td>
</tr>
<tr>
<td align="center" colspan="2">%%er_name%%
</td>
<td align="right" width="180" height="24" class="right_pad"><label
for="name"><span class="zvezda">* </span>Ім'я Фамілія:</label></td>
<td align="left" ><input name="name" id="name" value="%%name_add%%" type="text"
size="25">
</td>
</tr>
<tr>
<td align="center" colspan="2">%%er_pass%%
</td>
<td align="right" width="180" height="24" class="right_pad"><label
for="pass"><span class="zvezda">* </span>Пароль:</label></td>
<td align="left" ><input name="pass" id="pass" value="%%pass_add%%" type="text"
size="25">
</td>
</tr>
<tr>
<td align="center" colspan="2">%%for_kurs_id%%
</td>
<td align="right" width="180" height="24" class="right_pad"><label
for="email"><span class="zvezda">* </span>Email:</label></td>
<td align="left" ><input name="email" id="email" value="%%email_add%%" type="text"
size="25"></td>
</tr>
</table>
<table align="center">
<tr>
<td align="center" colspan="2" > <br />
<input name="save" value="Зберегти" type="submit"></td>
</tr>
</table>
</form>

```

<!-- new\_test.html- сторінка для додавання нового тесту-->

```
<script language="JavaScript">
<!--
function for_log_out(){
    if(confirm("Ви дійсно хочете вийти з даної форми?")){
        return true;
    } else{
        return false;
    }
}
-->
</script>
</head>
<body>

    <form name="add_form" action="index.php?act=motor" method="post"
enctype="multipart/form-data" >
    <input type="hidden" name="edit" value="%%id_edit%%">
    <input type="hidden" name="many_vars" value="%%many_vars%%">
    <input type="hidden" name="count" value="%%count%%">
        %%forms%%

    <table align="center">
    <tr>
    <td align="center" colspan="2" > <br />
        <input name="save" value="Зберегти" type="submit"></td>
    </tr>
    </td>
    </table>
</form>
```

Кафедра \_ КБПЗ 2023 рік

<!-- edit2.html- сторінка редагування обраного тесту-->

```

<script language="JavaScript">
<!--
function for_log_out(){
    if(confirm("Ви дійсно хочете вийти з даної форми?")){
        return true;
    } else{
        return false;
    }
}
-->
</script>
</head>
<body>
    <form name="add_form" action="index.php?act=motor" method="post"
enctype="multipart/form-data" >
    <input type="hidden" name="edit" value="%%id_edit%%">
    <input type="hidden" name="many_vars" value="%%many_vars%%">
    <table align="center">
        <tr>
            <th <h2 align="center">%%record%%</h2></th>
        </tr>
    </table>
        <table align="center">
            <tr>
                <td colspan="5" align="center">Поля відмічені <span
class="zvezda">* </span> повинні бути заповнені</td>
            </tr>

            %%er_desk%%
            <tr>
                <td width="180" height="24" class="right_pad"><label
for="desk"><span class="zvezda">* </span>Назва тесту:</label></td>
                <td><input name="desk" id="desk" value="%%desk_add%" type=text
size="50">
            </tr>
            <tr>
                <td colspan="5" align="center">%%er_subject_name%%
                <td class="right_pad"><label for="subject_name"><span
class="zvezda">* </span>Предмет:</label></td>
                <td><input name="subject_name" id="subject_name"
value="%%subject_name_add%" type=text size="25"></td>
            </tr>
            <tr>
                <td colspan="5" align="center">%%er_qmax%%
                <td class="right_pad"><label for="qmax"><span class="zvezda">*
</span>Кількість питань:</label></td>
                <td><input name="qmax" id="subject_name" value="%%qmax_add%"
type=text size="5"></td>
            </tr>

            <tr>
                <td colspan="5"> <hr width="80%"> </td>
            </tr>

            %%forms%%

    <table align="center">
        <tr>
            <td align="center" colspan="2" > <br />
            <input name="save_edit" value="Зберегти" type="submit"></td>
        </tr>
    </table>
</form>

```

```
<!-- test_rezalts1.html- сторінка для виведення результатів тестування-->
```

```
    <script language="JavaScript">
    </script>
    </head>
    <body>

        <table style="border-color:#3366FF" align="center" width="10%"
border="7px">
    <tr>
        <td align="center" bgcolor="#CCCCCC" style="font-size:200%" style="font-
weight:700" style="font:Arial, Helvetica, sans-serif">
        Ваша оцeнка: %%mark%%
    </td>
    </tr>
    </table>
```

Кафедра \_ КБПЗ \_ 2023 рік

<!-- add2.html- базова сторінка додавання нового тесту-->

```

<script language="JavaScript">
<!--
function for_log_out(){
    if(confirm("Ви дійсно хочете вийти з даної форми?")){
        return true;
    } else{
        return false;
    }
} -->
</script>
</head>
<body>
<form name="add_form" action="index.php?act=motor" method="post"
enctype="multipart/form-data" >
<input type="hidden" name="edit" value="%%id_edit%%">
<input type="hidden" name="many_vars" value="%%many_vars%%">
<table align="center">
<tr>
<th <h2 align="center">%%record%%</h2></th>
</tr>
</table>
<table align="center">
<tr>
<td colspan="5" align="center" >Поля відмічені <span
class="zvezda">*</span> повинні бути заповнені</td>
</tr>

%%er_desk%%
<tr>
<td width="180" height="24" class="right_pad"><label
for="desk"><span class="zvezda">*</span>Назва тесту:</label></td>
<td><input name="desk" id="desk" value="%%desk_add%" type=text
size="50">
</tr>
<tr>

%%er_subject_name%%
<td class="right_pad"><label for="subject_name"><span
class="zvezda">*</span>Предмет:</label></td>
<td><input name="subject_name" id="subject_name"
value="%%subject_name_add%" type=text size="25"></td>
</tr>
<tr>

%%er_qmax%%
<td class="right_pad"><label for="qmax"><span class="zvezda">*</span>Кількість питань:</label></td>
<td><input name="qmax" id="subject_name" value="%%qmax_add%"
type=text size="5"></td>
</tr>

<tr>
<td colspan="5"> <hr width="80%"> </td>
</tr>

<table align="center">
<tr>
<td align="center" colspan="2" > <br />
<input name="create_test" value="Створити тест"
type="submit"></td>
</tr>
</table>
</form>

```

**/\* SQL-запит - створення таблиці log\_in в базі даних st, яка містить дані щодо номеру, імені, паролю, статусу та спеціальності користувача\*/**

```
CREATE TABLE `st`.`log_in` (
  `name` varchar( 40 ) NOT NULL default '',
  `pass` varchar( 32 ) NOT NULL default '',
  `status` tinyint( 1 ) NOT NULL default '0',
  `user_id` int( 6 ) unsigned NOT NULL AUTO_INCREMENT ,
  `kurs_id` tinyint( 4 ) default NULL ,
  `email` varchar( 100 ) NOT NULL default '',
  PRIMARY KEY ( `user_id` )
);
```

**/\* SQL-запит - створення таблиці reg (таблиці реєстрації тестів) в базі даних st, яка містить дані щодо тестів \*/**

```
CREATE TABLE `st`.`reg` (
  `no` int( 10 ) unsigned NOT NULL AUTO_INCREMENT ,
  `desk` varchar( 50 ) NOT NULL default '',
  `qmax` tinyint( 1 ) unsigned NOT NULL default '0',
  `tbl` varchar( 10 ) NOT NULL default '',
  `author` int( 11 ) NOT NULL default '0',
  `subject_id` tinyint( 3 ) unsigned NOT NULL default '0',
  PRIMARY KEY ( `no` )
);
```

**/\* SQL-запит - створення таблиці reg (таблиці відповідей) в базі даних st, яка містить дані щодо відповідей про пройдені тести кожним зі студентів \*/**

```
CREATE TABLE `st`.`results` (
  `no` int( 10 ) unsigned NOT NULL default '0',
  `user_id` int( 10 ) unsigned NOT NULL default '0',
  `mark` tinyint( 1 ) unsigned NOT NULL default '0'
);
```

**/\* SQL-запит - створення таблиці special\_kurs в базі даних st, яка містить дані щодо спеціальностей та переліку предметів, які на ній викладаються \*/**

```
CREATE TABLE `st`.`special_kurs` (
  `kurs_id` tinyint( 3 ) unsigned NOT NULL default '0',
  `subject_id` tinyint( 3 ) unsigned NOT NULL default '0'
);
```

**/\* SQL-запит - створення таблиці предметів subject в базі даних st\*/**

```
CREATE TABLE `st`.`subject` (
  `subject_id` tinyint( 3 ) unsigned NOT NULL AUTO_INCREMENT ,
  `subject_name` varchar( 40 ) NOT NULL default '',
  PRIMARY KEY ( `subject_id` )
);
```