

Проектування адаптивної системи контролю знань студентів з  
дисципліни "Мови інформаційного обміну"

2021 р.

## ЗМІСТ

ВСТУП.....	3
РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ ПРОЕКТУВАННЯ АДАПТИВНОЇ СИСТЕМИ КОНТРОЛЮ ЗНАНЬ СТУДЕНТІВ.....	5
1.1. Огляд моделей адаптивного контролю знань .....	5
1.2. Система адаптивного контролю знань .....	9
1.3. Розробка алгоритму адаптивного контролю знань .....	17
РОЗДІЛ 2. ПРАКТИЧНА РЕАЛІЗАЦІЯ СИСТЕМИ АДАПТИВНОГО КОНТРОЛЮ ЗНАНЬ .....	20
2.1. Розподіл модулів між системами .....	20
2.2. Розробка модулю обміну даними з системою Moodle.....	21
2.3. Розробка модулю вибору питань .....	24
ВИСНОВКИ.....	28
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	29

## ВСТУП

**Актуальність.** В останнє десятиліття роль інформаційних технологій в освіті швидко і суттєво змінилася із появою систем електронного навчання. Системи електронного навчання збільшили свою цінність із ростом і популярністю комп'ютерних мереж через Всесвітню павутину та Інтернет. Поєднання використання Інтернету з потенційними методами викладання та навчання пропонує нові виклики та можливості у дистанційній освіті та електронному навчанні. Електронне навчання відіграє важливу роль у наданні навчальних матеріалів учням. Адаптивне електронне навчання - це новий підхід, який може зробити систему електронного навчання більш ефективною, адаптуючи подання інформації та загальну структуру зав'язків до окремих користувачів відповідно до їхніх знань та поведінки. Адаптивне електронне навчання базується на припущенні, що кожен учень має різні характеристики учня і що різні освітні параметри можуть бути більш придатними для одного типу учнів, ніж для іншого. Коли зміст курсу може бути наданий в гнучкому вигляді, адаптованому до характеристик окремих учнів через систему електронного навчання, система може доставити зміст курсу таким чином, щоб він використовував характеристики учня для оптимізації результатів навчання. Метою адаптивного електронного навчання є надання відповідної інформації необхідному студенту в необхідний час. Це означає, що адаптивна система електронного навчання здатна відстежувати використання та автоматично розміщувати вміст для кожного з користувачів для досягнення найкращих результатів навчання. Адаптивна система підтримується студентською моделлю, яка побудована на основі цілей, уподобань та знань студента. Потім модель студента використовується для адаптації режиму взаємодії системи електронного навчання відповідно до потреб користувача. Адаптивні системи електронного навчання можуть підвищити зручність використання матеріалу і, таким чином, зробити систему електронного навчання більш ефективною, що

покращить засвоєння студентами знань та призведе до кращих результатів навчання.

**Об'єкт:** адаптивний контроль знань студентів з дисципліни «Мови інформаційного обміну».

**Предмет:** адаптивна система контролю знань студентів з дисципліни «Мови інформаційного обміну».

**Мета дослідження:** розробити адаптивну систему контролю знань студентів з дисципліни «Мови інформаційного обміну».

**Завдання:**

1. Проаналізувати літературу з адаптивного контролю знань студентів.
2. Проаналізувати придатність системи керування навчанням Moodle до адаптивного контролю знань.
3. Проаналізувати інструментальні засоби та технології розробки систем взаємодії з Moodle.
4. Скласти алгоритм контролю знань для адаптивної системи контролю знань з дисципліни «Мови інформаційного обміну».
5. Скласти технічне завдання та розробити адаптивну систему контролю знань студентів з дисципліни «Мови інформаційного обміну».
6. Розрахувати собівартість розробки системи контролю знань.

**Наукова новизна:** полягає у розробці адаптивної системи контролю знань студентів з дисципліни «Мови інформаційного обміну».

**Практичне значення:** полягає у розробці адаптивної системи контролю знань студентів, використання якої може поліпшити ефективність навчання студентів дисципліни «Мови інформаційного обміну».

**Структура і обсяг роботи.** Наукова робота складається зі вступу, трьох розділів, загальних висновків, списку використаних джерел (36 найменування).

# РОЗДІЛ 1.

## ТЕОРЕТИЧНІ ОСНОВИ ПРОЕКТУВАННЯ АДАПТИВНОЇ СИСТЕМИ КОНТРОЛЮ ЗНАНЬ СТУДЕНТІВ

### 1.1. Огляд моделей адаптивного контролю знань

Традиційне навчання базується на лінійному процесі, коли студенти повинні вивчати всі запропоновані знання, тему за темою. Потім студенти повинні скласти іспити, щоб отримати підтвердження від професорів, що вони зберегли ці знання. Початковий період навчання дуже тривалий, зазвичай до 17 років (школа та університет). Є багато іспитів, деякі з яких дуже важкі, і мають великий вплив на життя студентів. Але вся ця дуже важка робота не гарантує того, що учні мають усі або навіть більшість знань, необхідних для вирішення проблем, які виникають у їхній післяшкільній діяльності, у реальному світі.

Адаптації в навчальному середовищі базуються на добре організованих моделях та процесах. Великий обсяг інформації в адаптивних системах електронного навчання необхідний для представлення знань у галузі та для моделювання навчальної поведінки студентів. Цю інформацію можна розділити на три основні моделі: модель домену, модель студента та адаптивну модель [3].

Модель домену містить інформацію про область знань змісту курсу для підтримки адаптивного курсу. Модель домену діє як сховище даних, яке складається з тем, вмісту, сторінок або вузлів та навігаційних посилань, пов'язаних із структурою дизайну представлених даних. Однак модель домену може також містити інформацію про студентів, яка має відношення до навчальної діяльності, таку як робочі процеси, учасники та ролі.

Є дві важливі частини доменної моделі: зміст курсу та система доставки. Система доставки повинна мати можливість підтримувати всі типи вмісту та бути адаптованою до різних вимог до змісту курсу. Найважливішим аспектом

моделі домену є взаємозв'язок між елементами курсу - навігаційними посиланнями між сторінками - які використовуються для прийняття рішення про адаптацію. Модель домену зосереджена на розробці гіпермедійної структури, яка відповідає потребам та характеристикам користувачів. Ця модель розробляє структури вмісту гіпермедіа, щоб захистити користувачів від таких перешкод, як проблема "загублених у просторі", коли вони входять до кожного вузла. Ці структури вмісту роблять студентам зручним пошук потрібних даних та тем [6].

Модель студента. Основним компонентом адаптивних систем електронного навчання є модель студента. Іноді її називають моделлю навчання або моделлю користувача. Він містить всю інформацію про студентів, наприклад, їхні знання в області, поведінку, рівень навчання та іншу інформацію. Студентська модель не тільки збирає загальну інформацію про користувачів, але й підтримує діючі облікові записи користувачів у системі [9].

Інформація, що стосується конкретного домену, та інформація, яка не залежить від домену, - це дві основні групи інформації, зібраної в студентських моделях, заснованої на взаємозв'язку з конкретним предметом. Інформаційна модель для конкретного домену називається моделлю знань студента. Він описує рівень знань студентів, їх розуміння предметних знань чи елементів навчальної програми, помилки, допущені студентами, процес розвитку знань студентів, записи поведінки, записи оцінювання чи оцінювання тощо.

Інформація, що не залежить від домену, - це інформація про навички учнів, тому вона базується на їх поведінці. Це може включати цілі навчання (для оцінки досягнень учнів), когнітивні можливості, такі як навичка індуктивного міркування та асоціативна навичка навчання, мотиваційні стани, які рухають учнів, передумови та досвід та переваги.

Адаптивна модель включає адаптивну теорію адаптивної системи електронного навчання шляхом поєднання доменної моделі зі студентською. Процес адаптивного моделювання починається з вибору репрезентативних вузлів шляхом аналізу потреб студента з моделі студента. Вузли можна

класифікувати на різні типи знань: базові знання, включаючи знання визначень, формул та інших питань; процедурні знання, що стосуються відносин між етапами; та концептуальні знання, посиляючись на відносини між поняттями, які вкладають деталі в загальну картину [18]. Кожен вид знань вимагає різних стратегій, тому вузли будуть представлені учням по-різному [18]. Наступним кроком є прийняття рішення про те, які навчальні об'єкти з яких вузлів повинні бути представлені, щоб вони могли використовуватись учнями до закінчення роботи з цим вузлом. Останнім кроком є повторення процесу, поки кожен вузол не буде повністю обраний.

Адаптивне навчання базується на концепції, яка називається "Знання вчасно" (JIT-Knowledge). Загальний обсяг зовнішніх джерел знань, навіть у певних областях, стає постійно більшим. Неможливо, беручи до уваги обмеження людського мозку, набути його шляхом традиційного навчання, тема за темою. Це означає, що насправді великим обсягом знань ніхто не користується, і багато проблем не вирішуються, оскільки необхідні знання не засвоюються. Електронна публікація знань дозволяє виявляти та засвоювати лише знання, що стосуються існуючих проблем.

Професором Л.А. Растригиним [27] було запропоновано розглядати процес навчання як процес управління складною системою. Аналогічно можна уявити і процес управління адаптивним контролем знань (рис. 1.1).

Блок "Алгоритм контролю" виконує наступні функції:

- аналіз діяльності студента (перевірка правильності його відповідей і виконуваних дій);
- управління процесом контролю знань на основі обраного методу;
- визначення результатів контролю, яке зазвичай зводиться до виставлення оцінки студенту.

База знань (БЗ) містить методи і / або моделі процесу контролю, а також сукупність знань предметної області. База даних (БД) включає набори питань і завдань, призначених для перевірки знань студента і / або дані для формування завдань. Контрольні завдання можуть також генеруватися автоматично на

основі БЗ. База даних і база знань спільно з моделлю студента утворюють репозиторій системи контролю.

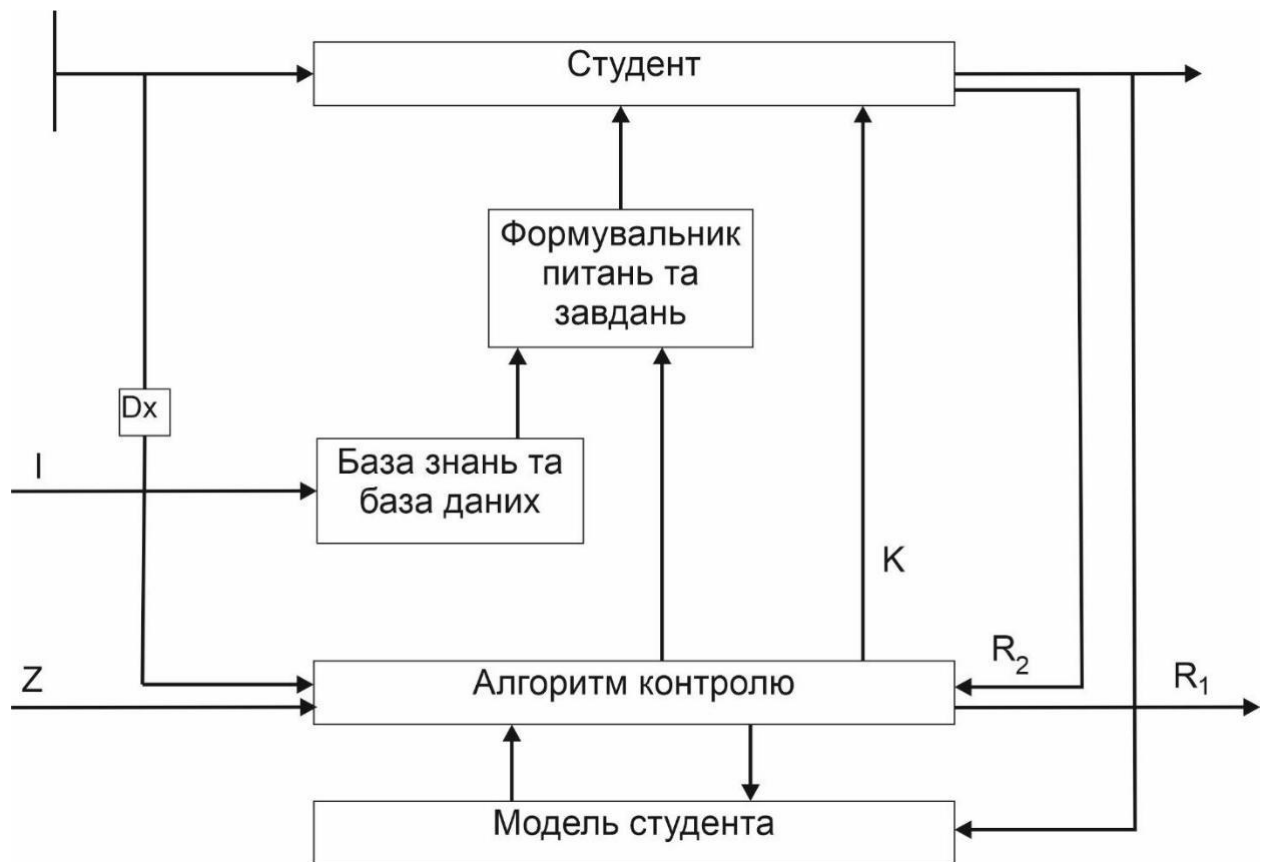


Рис. 1.1. Модель адаптивного контролю знань.

Модель студента включає різноманітну інформацію про студента: передісторія навчання; результати поточної роботи (тип виконаних завдань, час виконання завдань, число звернень за допомогою тощо); особистісні психологічні характеристики (тип і спрямованість особистості, репрезентативна система, здатність до навчання, рівень занепокоєння-тривоги, особливо пам'яті та ін.); загальний рівень підготовленості [25].

Формувальник питань і завдань використовується для формування і видачі студенту чергового завдання (питання або завдання).

Контроль знань здійснюється наступним чином: студент виконує запропоноване завдання, і результат його роботи поміщається в модель студента. Блок "Алгоритм контролю" на основі аналізу відповіді студента,



цілей контролю  $Z$  і використовуваного методу проведення контролю, враховуючи зовнішні ресурси  $R1$  (наприклад, можливості системи контролю) та внутрішні ресурси студента  $R2$  (наприклад, час контролю), а також стан середовища  $Dx$ , визначає параметри завдання, яке повинно бути запропоновано студенту. Формувальник питань і завдань, отримавши від "Алгоритму контролю" дані про параметри наступного завдання, вибирає з БД і / або БЗ необхідну інформацію  $I$ , формує текст завдання і видає його студенту. У найпростішому випадку робота цього блоку зводиться до вибору потрібного питання або завдання з бази даних. При деяких видах контролю (наприклад, при поточному КЗ або самоперевірці) може бути передбачена зворотний зв'язок, який полягає у видачі коментаря на відповідь студента.

## 1.2. Система адаптивного контролю знань

Організація підсистеми оцінювання знань є однією з найбільш складних завдань при синтезі системи адаптивного тестування знань учнів в реальному масштабі часу. При оцінюванні знань необхідно врахувати індивідуальні здібності учнів, правильність виконання кожного із завдань, з огляду на кількість що допускаються кожним учнем помилок, а також ймовірність вгадування правильної відповіді, яких навчають. В даному розділі пропонується модель оцінювання знань при адаптивному тестуванні, що враховує відмічені критерії.

При проходженні адаптивного тесту в кожен момент часу учні виконують деякі завдання. За результатами виконання кожного із завдань висуваються умови для вибору наступного завдання з бази знань. Виходячи зі сказаного вище, побудуємо диференціальну модель процесу оцінювання знань за допомогою тестування.

Диференціальна модель являє собою диференціальне рівняння, отримане в результаті дослідження будь-якого реального явища або процесу, в даному випадку процесу оцінювання тесту [6]. Отже, нехай деякий учень проходить

адаптивне тестування. Йому пропонується виконати ланцюжок з  $N$  завдань, причому результат виконання тесту  $R$  в кожен момент часу (при виконанні кожного наступного завдання) залежить від результатів, отриманих при виконанні попередніх завдань. Таким чином, результат виконання тесту змінюється безперервно і, більш того, він диференціюючи як функція, що залежить від кількості виконаних в даний момент завдань. Звичайно, це твердження є спрощенням реальної ситуації, оскільки  $R$  - ціле число. Для побудови моделі проведемо аналіз процесу адаптивного тестування. Розглянемо процес проходження тестування та виділимо основні параметри, що впливають на його результат.

1. Для отримання об'єктивної оцінки знань під час проходження тестувань рекомендується використовувати безперервну шкалу оцінювання.
2. Однією з основних проблем тестування є ймовірність вгадування правильних варіантів відповіді.
3. При проведенні адаптивного тестування кожному із завдань присвоюється певний рівень складності  $(z_i)$ .
4. При оцінюванні знань застосовуються різні системи оцінювання (4-бальна, 12-бальна, 100 бальна, літерна). Для отримання результату в заданій системі оцінювання знань, введемо спеціальний параметр  $B$ -бальність системи.

При обліку введених параметрів, що впливають на результат проходження тесту, динаміка зміни результату виконання тесту при виконанні кожного з завдань може бути описана за допомогою функціоналу

$$R = \sum_{i=1}^N r_i \cdot c_i \cdot z_i \cdot B, \quad (1.1)$$

де  $N$  - кількість тестових завдань, які виконуються учнем;  $r_i$  - коефіцієнт оцінювання завдань різних форм. Безперервна величина, змінюється в діапазоні  $[0,1]$ ;  $z_i$  - рівень складності  $i$ -го завдання;  $c_i$  - ймовірність вгадування правильної відповіді на  $i$ -е завдання;  $B$  - бальність системи.

Завдання полягає в тому, щоб вказати відповідні формули для введених вище величин коефіцієнта оцінювання знань при виконанні заданих різних форм  $r_i$  і ймовірності вгадування правильної відповіді  $c_i$ .

Для того, щоб об'єктивно оцінити знання під час тестування з використанням тестових завдань різних форм пропонується використовувати для кожної з форм завдань свою методику розрахунку оцінки. Пропонується використання безперервної шкали оцінювання на інтервалі  $[0,1]$ . Визначимо коефіцієнт оцінювання знань для відповідей на тестові завдання різних форм окремо [7].

Для оцінювання одноальтернативного тестового завдання досить застосування відомої і широко використовується дихотомічної шкали оцінювання, в якій 1 відповідає правильній відповіді, 0 - неправильного. Отже, в цьому випадку коефіцієнт  $r_i$  приймає одне із значень 0 або 1. багатоальтернативного тестові завдання. У разі багатоальтернативного тестового завдання необхідно враховувати не тільки правильність відповіді на завдання в цілому, а й кількість варіантів відповіді, обраних учнем вірно / невірно. В даному випадку коефіцієнт  $r_i$  пропонується розраховувати за формулою (1.2):

$$\text{---} \quad 1.2$$

де - кількість правильних варіантів відповіді в  $i$ -м завданні; - кількість правильних варіантів відповіді, обраних учнем при виконанні  $i$ -го завдання;  
 - кількість неправильних варіантів відповіді, обраних учнем при виконанні  $i$ -го завдання.

При виконанні завдання на встановлення відповідності кожну пару відповідей можна розглядати як окремий варіант відповіді, і при обчисленні результату виконання завдання слід враховувати, скільки пар було вибрано правильно. Отже, коефіцієнт  $r_i$  необхідно розраховувати за формулою:

$$\text{---} \quad 1.3$$

де - кількість пар для зіставлення в  $i$ -м завданні; - кількість вірно складених пар в  $i$ -м завданні.

При

оцінюванні  
завдань на  
встановлення  
правильної  
послідовності

можливий тільки один свідомо правильних відповідь. Отже, для оцінювання даної форми тестових завдань досить використовувати дихотомічне шкалу оцінювання, коефіцієнт  $r_i$  приймає значення 0 або 1. Однією з форм відкритих тестових завдань є заповнення таблиць. Кожна клітинка таблиці є окремим варіантом відповіді та, якщо одна з осередків заповнена неправильно, така відповідь не можна зараховувати як повністю неправильний. Для визначення коефіцієнта оцінювання завдань на заповнення таблиць рекомендується використовувати показову функціональну залежність, представлену формулою:

—

1.4

де  $n_i$  - кількість осередків, які пропонується заповнити якого навчають в  $i$ -м завданні;  $m_i$  - кількість осередків, які навчають, заповнив правильно при виконанні  $i$ -го завдання.

Багатокрокові тестові завдання складаються з набору завдань (набору кроків), що вирішуються послідовно, коли перехід до наступного кроку завдання здійснюється тільки після правильної відповіді на попередній крок. Це дає можливість тому, кого навчають аналізувати не тільки завдання в цілому, а й розбиратися в кожній складовій завдання. Завдяки цьому той, якого навчають одразу може побачити, де їм допущена помилка і в подальші кроки отримати правильні вихідні дані, тобто помилки в завданнях не будуть накопичуватися. Багатокрокове завдання вважається пройденим, якщо на кожному його кроці отримано правильну відповідь. Для об'єктивного оцінювання відповіді і глибини знань учня використовується лічильник допускаються учнем помилок, кількість яких враховується при виставленні оцінки.

Однак постає питання про оцінювання таких питань. Для них проста дихотомічна шкала не підходить. Рекомендована формула для обчислення коефіцієнта правильності для багатокрокового тестового завдання, на кожному

з кроків якого знаходиться одноальтернативне завдання або завдання на встановлення правильної послідовності, має вигляд:

$$\frac{\Sigma}{\Sigma}$$

де  $i$  - номер кроку,  $m_i$  - кількість помилок, допущених на  $i$ -му кроці,  $n$  - кількість кроків.

У разі використання на будь-якому з кроків тестового завдання на відповідність або багатоальтернативного тестового завдання формули (1.5) недостатньо.

Доцільно використовувати формули (1.6) і (1.7) відповідно.

$$\frac{\Sigma}{\Sigma(j)}$$

де  $j$  - номер спроби проходження кроку, якщо на ньому була допущена помилка;  $\Sigma$  - кількість пар для складання на  $i$ -му кроці;  $\Sigma(j)$  - кількість вірно складених пар на  $i$ -му кроці при  $j$ -й спробі.

$$\frac{\Sigma(i)}{\Sigma(i, j)}$$

де  $\Sigma(i)$  - кількість правильних варіантів відповіді на  $i$ -му кроці при  $j$ -й спробі;

$\Sigma(i, j)$  - кількість правильних варіантів відповіді, обраних тестованим на  $i$ -му кроці;  $\Sigma(i, j)$  - кількість неправильних варіантів відповіді, обраних тестованим на  $i$ -му кроці при  $j$ -й спробі.

Використання вибірових відповідей пов'язане з можливістю неадекватної оцінки знань учня в разі випадкового введення правильної відповіді. Оцінимо цю ймовірність для відповідей на тестові завдання різних типів.

Для відповіді на тестові завдання закритого типу розглянемо окремо відповідь на одноальтернативні та багатоальтернативні тестові завдання. Одноальтернативні тестові завдання представляють собою завдання, в яких  $q$  варіантів відповіді, один з яких правильний. Ймовірність вгадування правильної відповіді  $c_i$  в даному випадку визначається за формулою:

$$\frac{1}{q}$$

При виконанні багатоальтернативних тестових завдань якого навчають, пропонується з  $q$  варіантів відповіді вибрати  $Q_1$  правильних ( $0 \leq Q_1 \leq q$ ) Імовірність введення кожної відповіді для вибірки довжиною  $Q_1$  визначається як  $\frac{Q_1}{M}$ , де  $M$  - загальна кількість можливих відповідей, які може дати той, якого навчають. У разі, якщо наперед відомо кількість правильних варіантів відповіді, значення  $M$  розраховується за формулою:

$$M = \frac{q!}{Q_1!(q-Q_1)!} \quad 1.9$$

Загальна кількість можливих відповідей  $M$  на багатоальтернативність завдання при заздалегідь невідомому числі  $Q_1$  дорівнює кількості всіх можливих варіантів вибору відповіді. Отже,

$$\sum_{Q_1=0}^q \frac{q!}{Q_1!(q-Q_1)!} = 2^q \quad 1.10$$

З комбінаторики відомо [8], що  $\sum_{Q_1=0}^q \frac{q!}{Q_1!(q-Q_1)!} = 2^q$

Значення  $Q_1$  невідомо тому кого навчають, отже, можна вважати всі  $M$  варіантів можливої відповіді рівноімовірними. Тільки один з можливих варіантів відповіді є правильним. Таким чином, ймовірність вгадування правильної відповіді визначається за формулою (1.11):

$$P = \frac{1}{M} \quad 1.11$$

При складанні тестового завдання на встановлення послідовності можливі два варіанти подання списку елементів послідовності. У першому випадку, кого навчають необхідно скласти послідовність включивши всі запропоновані елементи. У другому випадку - пропонується вибрати послідовність елементів, включивши в неї тільки необхідні елементи. Розглянемо обидва випадки.

Завдання, в якому все  $q$  елементів послідовності входять у відповідь. Для завдань, в яких необхідно скласти послідовність з усіх запропонованих

елементів, завдання визначення кількості можливих комбінацій відповіді зводиться до знаходження числа перестановок елементів послідовності. Таким чином, кількість  $M$  можливих відповідей визначається за формулою



Таким

чином,  
ймовірність  
вгадування  
правильної  
відповіді  
для  
списку

визначається за формулою (1.12):

$$— \quad 1.12$$

Завдання, в якому з  $q$  запропонованих елементів  $l$  входить в правильну послідовність. Для розглянутого випадку мінімальне значення  $l$  дорівнює 2, тому що послідовність меншої довжини не має сенсу. Кількість  $M$  можливих варіантів відповіді можна визначити як розміщення з  $q$  по  $l$ . Слід також врахувати, що значення  $l$  змінюється від 2 до  $q$ . Таким чином, кількість  $M$  різних варіантів перестановок визначається за формулою (1.13):

$$\sum_{l=2}^q \sum_{i=1}^l \frac{q!}{(q-l)!} \quad 1.13$$

Правильним є єдиний варіант відповіді, тому ймовірність випадкового введення правильної відповіді  $c_i$  визначається за формулою (1.14):

$$\frac{\sum_{i=1}^m c_i}{\sum_{i=1}^m c_i} \quad 1.14$$

При виконанні тестових завдань на встановлення відповідності можливе подання відповіді одним із таких способів:

1. Є два списки по  $q_1$  елементів. Необхідно скласти  $q_1$  пар.
2. Є два списки по  $q_1$  і  $q_2$  елементів відповідно ( $q_2 < q_1$ ). Необхідно скласти  $m$  пар.
3. Є два списки по  $q_1$  і  $q_2$  елементів відповідно ( $q_2 \leq q_1$ ). Необхідно скласти  $q_3$  пар ( $q_3 < q_1$ ).

Розглянемо кожен із способів окремо.

При встановленні відповідності між елементами двох множин з однаковою кількістю елементів загальна кількість можливих варіантів введення

відповіді  $M = (q_1!)^2$  Отже, в цьому випадку ймовірність випадкового введення  
правильної відповіді визначається за формулою (1.15):

( )

1.15

У разі, коли необхідно встановити відповідність між  $q_1$  парами, причому кількість елементів в одному з множин  $q_2$  ( $q_2 \leq q_1$ ), кількість можливих варіантів відповіді зросте і буде визначатися за формулою (1.16):

$$\frac{q_1!}{(q_1 - q_2)!} \quad 1.16$$

Імовірність вгадування відповіді в такому випадку визначається за формулою:

$$\frac{q_2!}{q_1!} \quad 1.17$$

Для отримання завдання на встановлення відповідності з найменшою вірогідністю вгадування рекомендується додавати «зайві» елементи в обидва безлічі елементів. Нехай є два безлічі, що складаються з  $q_1$  і  $q_2$  елементів відповідно. При встановленні складання для  $q_3$  пар ( $q_3 < q_1$ ,  $q_3 < q_2$ ) загальна кількість можливих відповідей можна визначити по формулі (1.18):

$$\frac{q_1! q_2!}{(q_1 - q_3)! (q_2 - q_3)!} \quad 1.18$$

Імовірність вгадування відповіді в разі, коли в завданні на встановлення відповідності в обидва безлічі елементів додані «зайві» елементи, визначається за формулою (1.19):

$$\frac{(q_1 - q_3)! (q_2 - q_3)!}{q_1! q_2!} \quad 1.19$$

При виконанні відкритих тестових завдань навчається не пропонуються варіанти відповіді, тому ймовірність вгадування правильної відповіді прагне до нуля.

При оцінюванні ймовірності вгадування правильної відповіді під час виконання багатокрокового тестового завдання спочатку необхідно оцінити ймовірність вгадування правильної відповіді на кожному з  $n$  кроків окремо. Загальний результат в даному випадку визначається за формулою:

$$P = \prod_{i=1}^n p_i \quad 1.20$$

### 1.3. Розробка алгоритму адаптивного контролю знань

Доцільність адаптивного контролю впливає з міркувань раціоналізації традиційного тестування. Підготовленому студенту немає необхідності давати легкі завдання. Легкі завдання не володіють помітним розвивають потенціалом, в той час як важкі завдання у більшості студентів знижують навчальну мотивацію. Переробимо модель адаптивного контролю знань (рис. 1.2).



Рис. 1.2. Перероблена модель адаптивного контролю знань.

Тести будуть різної важкості тому у моделі студента буде попадати інформація яку частину завдання вирішив студент, де в нього помилки. Результати тестування у яких будуть попадати до алгоритму контролю. На відмінність від стандартного алгоритму, який присутній у відповідному модулі системи Moodle, розробляємий алгоритм контролю буде включати в себе нейронну мережу яка буде навчатися на даних, які будуть додаватися модулем модель студента. Так безпосередньо від модуля студент у результаті роботи алгоритму контролю буде сформована модель кожного студента. Далі з бази даних та знань за допомогою алгоритму контролю будуть обиратися необхідні питання та завдання.

Всі завдання у базі знань мають наступні показники: тема, до якого відноситься тестове питання та важкість завдання. Завдання до бази знань

будуть вноситися викладачем за допомогою стандартних засобів системи Moodle. Рівень важкості питань буде встановлюватися за допомогою модулю системи Moodle Adaptive Quiz. Для взаємодії системи Moodle та системи адаптивного контролю знань до кожного питання додано признак «видимість». Та розроблений спеціальний модуль для системи Moodle, який під керуванням алгоритму контролю системи адаптивного контролю знань буде робити видимими для Moodle потрібний для даного студента набір питань.

Алгоритм контролю дає команду формувальнику питань сформувати набір питань які будуть додаватися з бази даних та знань.

Формувальник питань діє за наступним алгоритмом адаптивного тестування:

0. Запросити наступного кандидата: встановити  $D=0$ ,  $L=0$ ,  $H=0$  і  $R=0$ .

1. Знайдіть наступне питання приблизно складністю ( $D$ ).

2. Встановіть  $D$  при фактичному калібруванні цього питання.

3. Задати питання цього пункту.

4. Отримати відповідь.

5. Оцініть цю відповідь.

6. Підрахуйте взяті позиції:  $L = L + 1$

7. Додайте використані труднощі:  $H = H + D$

Якщо відповідь неправильна,

8. Оцінюйте складність наступного питання:  $D = D - 2 / L$

Якщо відповідь правильна,

9. Підрахуйте правильні відповіді:  $R = R + 1$

10. Оцініть складність наступного питання:  $D = D + 2 / L$

Якщо ви не готові вирішити пройти / не скласти,

11. Перейдіть до кроку 1.

Якщо ви готові прийняти рішення про передачу / невдачу,

12. Обчисліть неправильні відповіді:  $W = L - R$

13. Оціночна міра:  $B = H / L + \log_e (R / W)$

Якщо  $W = 0$ , то  $B = H / L + \log_e (R-0.5 / W + 0.5)$

Якщо  $R = 0$ , то  $B = H / L + \log_e (R + 0,5 / W - 0,5)$

Щоб отримати більш точну оцінку  $B$ , див. Оцінка мір з відомими труднощами предмета

14. Помилка оцінки:  $S = \sqrt{L / (R * W)}$

Якщо  $W = 0$ , то  $S = \sqrt{L / (R - 0,5 * W + 0,5)}$

Якщо  $R = 0$ , то  $S = \sqrt{L / (R + 0,5 * W - 0,5)}$

15. Порівняйте міру  $B$  із стандартним пропуском / відмовою  $T$ .

16. Якщо  $(T - S) < B < (T + S)$ , перейдіть до кроку 1.

17. Якщо  $(B - S) > T$ , то здайте.

18. Якщо  $(B + S) < T$ , то не вдасться.

20. Перейдіть до кроку 0.

У результаті формується список питань тестових питань які необхідно вирішити студенту на періодичному контролі. Також на періодичному контролі частину завдань буде з використанням блоку практичних завдань. Таким чином студент буде виконувати ті завдання які відповідають його ступню знань по предмету.

## РОЗДІЛ 2. ПРАКТИЧНА РЕАЛІЗАЦІЯ СИСТЕМИ АДАПТИВНОГО КОНТРОЛЮ ЗНАНЬ

### 2.1. Розподіл модулів між системами

Нашою метою є розробка системи адаптивного контролю знань, яка би проводила цей контроль не тільки за допомогою стандартного алгоритму контролю який заложений у плагін Moodle. Для цього необхідні додаткові дані, розширена база даних для побудови моделі студента та нейронна мережа для аналізу цих даних.

Виходячи з того, що Moodle періодично оновлюється, додавання складних плагінів до Moodle може привести до краху системи через логічну несумісність оновлень. Тому доцільно розділити систему адаптивного контролю знань на дві частини: систему Moodle та розроблену власноруч систему адаптивного контролю знань. Останню розташувати незалежно від системи Moodle та розробити плагін для зв'язку з Moodle (рис. 2.1).

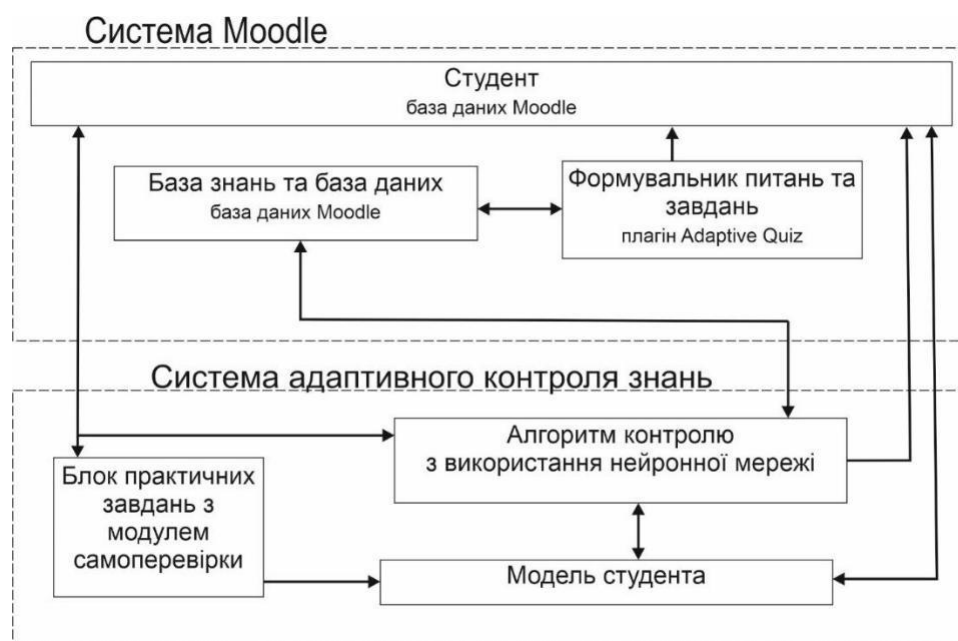


Рис. 2.1. Розподіл модулів між системами

Розглянемо, які модулі ми залишимо у системі Moodle, а які розробимо у окремій системі адаптивного контролю знань. Основні дані про студента, результати тестувань, результати інших видів діяльності знаходяться у базі даних системи Moodle. Там дані постійно поновлюються та змінюються у результаті дій користувачів системи. Тому модуль «Студент» ми залишимо у системі Moodle. Як було розглянуто вище за формування питань тестів та адаптивне тестування відповідають стандартні модулі системи Moodle: «База знань та база даних» і «Формувальник питань та завдань». Для зв'язку з ними ми розробимо модуль, який буде зв'язувати ці модулі та модуль системи адаптивного контролю знань (САКЗ) «Алгоритм контролю». Для проведення та автоматичної перевірки практичних завдань розробимо модуль у САКЗ. Цей модуль буде викликатися по гіперпосиланню зі сторінки предмету системи Moodle. Результати роботи будуть зберігатися у базу даних «модель студента». Для обміну даними між базою «модель студента» САКЗ та базою «Студент» системи Moodle у обох системах розробимо модулі обміну інформацією. База даних «модель студента» необхідна для роботи модуля «Алгоритм контролю». Для оперативного моніторингу результатів адаптивного тестування необхідно передбачити керування «модулем Студент» із модулю «Алгоритм контролю».

## 2.2. Розробка модулю обміну даними з системою Moodle

Для розробки системи обміну даними спочатку необхідно розглянути архітектуру системи Moodle та виявити об'єкти необхідні для обміну даними.

Спочатку ми розглянемо загальну архітектуру LAMP, на якій базується Moodle, перш ніж розглядати внутрішні компоненти рівня VLE.

Moodle був розроблений на платформі LAMP з відкритим кодом, що складається з Linux (операційна система), Apache (веб-сервер), MySQL (база даних) та PHP (мова програмування). Завдяки переносимості цих компонентів та модульності самого Moodle (саме це означає «М»), він може підтримувати



широкий спектр операційних систем, систем баз даних та веб-серверів. На наступній схемі показано простий огляд загальної архітектури (рис. 3.2)

Найнижчий рівень - це операційна система. Хоча Linux є найкращою платформою, підтримуються інші похідні Unix, такі як Solaris та AIX, а також Windows та Mac OS X (бажано варіанти серверів для виробничих сайтів). Потрібно встановити певні бібліотеки - див. Розділ 1, Встановлення Moodle. Користувач робить запити через інтерфейс веб-браузера або мобільний додаток Moodle (наприклад, для відображення навчального ресурсу). Веб-браузер передає запит веб-серверу, який викликає модуль PHP, який відповідає за дзвінок. Модуль PHP викликає базу даних з дією (запит, оновлення, вставка або видалення), яка повертає запитувані дані. Виходячи з цієї інформації, модуль PHP повертає дані (зазвичай у формі HTML або JavaScript-коду) на веб-сервер, який передає інформацію, що відображається назад, у браузер або додаток користувача.

Moodle розрізняє код (здебільшого написаний на PHP, HTML та CSS) та дані (переважно значення, додані через інтерфейс Moodle).

Бібліотеки Moodle, модулі (наприклад, ресурси та дії), блоки, плагіни та інші сутності представлені в коді. Він завжди зберігається у файльовій системі в каталозі Moodle, який називається `dirroot`, який був вказаний під час процесу встановлення в попередньому розділі. Код включає всі елементи, які мають справу з операційними процесами (сервер) та інтерфейс (інтерфейс користувача).

Курси Moodle, користувачі, ролі, групи, оцінки та інші дані, такі як навчальні ресурси, додані викладачами, повідомлення на форумі, додані студентами, та системні налаштування, додані адміністратором, в основному зберігаються в базі даних Moodle. Однак такі файли, як фотографії користувача або завантажені завдання, зберігаються в іншому каталозі Moodle, відомому як `moodledata`, який знаходиться в каталозі, який називається `dataroot`. Інформація про файли (такі метадані, як ім'я, місцезнаходження, остання модифікація,

ліцензія та розмір) зберігається у базі даних, яка посилається на відповідні файли.

Навіть копіювання файлу з однієї папки в іншу або додавання файлу вручну порушить узгодженість вашої системи, і подальша поведінка не може бути передбачена. Внутрішньо Moodle використовує механізм, який називається хешування SHA1. Moodle повністю підтримує імена файлів Unicode, а також уникає надмірного зберігання, коли один і той же файл використовується двічі. Системні файли - файли, необхідні для запуску Moodle - знаходяться в ряді каталогів у розділі `dirroot` (кореневий каталог вашої інсталяції Moodle). Для спрощення ідентифікації будь-яких налаштувань у розділі адміністрування передбачений засіб пошуку, який знаходиться під ієрархічним меню адміністрування Сайту.

Під час пошуку за будь-яким терміном Moodle відображає результати у розгорнутому вигляді, що дозволяє негайно змінити налаштування. Наприклад, під час пошуку за "календарем" у результаті з'являються численні розділи, які можна відразу змінити в кожному розділі, а не переходити до кожного окремого розділу, щоб внести зміни.

Повна документація Moodle знаходиться в Інтернеті на сайті [docs.moodle.org](https://docs.moodle.org). Якщо ви хочете надати власну документацію, змініть кореневий параметр документа Moodle Docs в Appearance | Документи Moodle. Для кожної версії Moodle публікуються окремі документи Moodle. Попередній знімок екрану - з версії 2.0. Будуть створені нові документи для Moodle версії 2.1, 2.2 тощо.

Спільнота Moodle постійно зростає, і на момент написання статті мала вже понад 1 мільйон зареєстрованих користувачів (так, це 1 мільйон!), з яких від 5 до 10 відсотків є активними. Якщо ви не можете знайти рішення будь-якої з проблем Moodle в Moodle Docs, скористайтеся функцією пошуку [moodle.org](https://moodle.org) у верхній частині екрана в настрої `l e. o r g`. У порядку пріоритетності пошук видає вже згадані документи Moodle, найактивніші форуми користувачів та Moodle Tracker, який відстежує всі проблеми та запити на функції

(tracker.Moodle.Org). Пошук на форумах Moodle часто може призвести до великої кількості посилань. Щоб звужити

пошуковий простір, використовуйте Розширений пошук у блоці Пошук на форумах. Якщо ви все ще не можете знайти рішення своєї проблеми, що трапляється порівняно рідко, опублікуйте запитання на відповідному форумі, і хтось, ймовірно, допоможе вам далі.

Робота з файлами у веб-програмах не завжди є простою. Хоча Moodle надає користувацький інтерфейс для виконання цього завдання, іноді потрібно, щоб, довелось обійти цей механізм та використовувати інші засоби.

### 2.3. Розробка блоку практичних завдань з модулем самоперевірки

САТ вдає із себе програмний комплекс [2]. При цьому в більшості випадків він орієнтований на ситуацію, коли необхідно проводити збір даних - програмні реалізації рішень завдань - з різних вузлів мережі, комп'ютерів авторів програм, в той же час результати тестування всіх робіт повинні бути доступні викладачеві. Таким чином, САТ повинна дотримуватися клієнт-серверної архітектури додатків [3]. Очевидно, на оцінку ефективності та зручності САТ впливають характеристики клієнтського, серверного додатків комплексу, а також каналу зв'язку між ними [3].

Клієнтську програму може накладати частина обмежень на клієнтів САТ [3]. Можливо кілька випадків якості клієнтської програми.

У першому випадку клієнтську програму відсутнє як таке, таким чином, вся діяльність користувачів повинна відбуватися на одному і тому ж комп'ютері. У разі ж існування клієнтського програмного забезпечення (ПО) на оцінку САТ впливають обмеження, що накладаються на це ПЗ, такі як критерії до апаратної або програмної частин системи або кваліфікацію користувача.

Аналогічно клієнтського ПЗ серверне ПЗ накладає власні обмеження на програмну і апаратну складову сервера САТ [3], окремі вимоги виставляють вимоги до кваліфікації адміністратора сервера. Не виключений випадок

відсутності окремого серверного ПО, тоді існують дві різних категорії. У першій категорії знаходяться програмний комплекс, в якому серверне ПЗ не існує як окремий пакет, але певні настройки в програмі САТ виділяють сервер від клієнтів. У другій категорії, всі вузли САТ рівноправні, в такому випадку вся звітність як по проведеним тестам, так і по роботі системи доступна всім клієнтам в рівній мірі.

Як і будь-яке ПЗ на САТ накладаються певні вимоги до документації проекту [4] як для кінцевих користувачів системи, так і для адміністраторів, які забезпечують працездатність системи. Однак обсяг документації та освітлені аспекти функціонування САТ можуть істотно відрізнитися. Документація може обмежуватися лише технічними вимогами ПО, так і описом використання системи. Важливий аспект оцінки САТ - це ситуація, в якій використовується САТ. Це може бути проведення навчальних занять, змагань та конкурсів з програмування або використання в якості контрольно-вимірювальних матеріалів [5]. Залежно від цього призначення САТ може бути представлена як окремий електронний ресурс, спеціалізована служба, інтегрована в яку-небудь оболонку системи електронного навчання або як стандартна функція цієї системи. Компіляція програмного коду може відбуватися як на клієнтській машині як елемент підготовки до тестування, так і сервері як частина перевірки програмного рішення. Саме тестування також може відбуватися як на сервері, так і на клієнтських машинах, що залежить від функціональних можливостей конкретної САТ і поставлених вимогах до продуктивності і безпеки тестування. Крім того, існує можливість використання декількох серверних елементів САТ на різних машинах, таким чином дистанціюючи функцію управління системою від безпосереднього тестування додатків. Деякі САТ підтримують можливість використання окремих публічних або корпоративних серверів для забезпечення тестування на них, в такому випадку дистанційоване тестування виступає як окремий сервіс. Можливість розподілу обчислень САТ впливає на її кінцеву оцінку.

Важливим аспектом САТ є її продуктивність і максимальний обсяг використовуваних для обчислень ресурсів [5]. В архітектурі САП може бути закладено обмеження на характеристики даних для тестованих додатків, або обсяг доступних тестів додатків за одиницю часу. Багато САТ як програмного рішення для тестування використовують програмний код, який буде скомпільовано вже всередині системи, щоб уникнути помилок, пов'язаних з відмінностями в архітектурі середовища тестування і середовища компіляції. Таким чином, однією з характеристик САП стає перелік мов програмування, які підтримуються системою. При цьому, так як мови програмування розвиваються, з'являються нові версії та мови, важливий не тільки обсяг цієї множини, але і можливість його доповнення [1, 2]. Підтримка мов програмування може вдосконалюватися двома шляхами: оновленнями від розробників САТ і налаштуванням адміністраторів. Залежно від обсягу та актуальності переліку підтримуваних мов програмування, можливості модифікації цієї опції адміністратором САТ, а також якістю і своєчасністю оновлень може змінитися кінцева оцінка САТ.

При виборі САТ необхідно враховувати, як саме необхідно контролювати діяльність користувачів САТ [3-5]. На що впливає мета використання САТ. Моніторинг тестування може демонструвати результати лише після повного проведення тестування ПО або ж демонструвати процес тестування, відображаючи інформацію за окремими тестами і самого процесу роботи ПО і САТ. У той же час такий моніторинг вимагає додаткових ресурсів, а контроль лише результатів набагато економічніше. Обсяг інформації за результатами тестування також може відрізнитися, в залежності від САТ крім результатів самого тестування ПО може бути надана інформація щодо процесу тестування, порівняння з іншими рішеннями або ж дані пов'язані з автором ПО. Розрізняються і форми подачі інформації звітності, САТ може володіти функцією налаштування виведеної інформації, форми виведення і способу передачі користувачеві. Звітність і можливість її налаштування впливає на кінцеву оцінку САТ.

Блок практичних завдань розроблений за допомогою мови програмування Python. Запуск вікна рішення завдань виконується із системи Moodle за посиланням «Завдання до лабораторної роботи». Далі за посиланням відкривається сторінка серверу перевірок відповідної лабораторної роботи. Після натискання на кнопку «Перевірити» запускається скрипт перевірки валідності коду. Відповідно до теми лабораторної роботи кількість вікон до введення коду студента може відрізнятись. Система перевірки лабораторних робіт дозволяє перевіряти код студентів на мовах обміну даними XML та json. Також підтримується перевірка запитів то бази даних на мові SQL. Перевірка ведеться за декількома тестами. Сервер перевірок лабораторних робіт організований на мові програмування Python. Як було описано вище мова Python дозволяє розробити як серверну частину - систему відображення сторінок, систему роботи з базою даних, системою формування вебсервісів, так і систему перевірки коду на мовах обміну даними. Також, як буде описано нижче, Python дозволяє розробити алгоритм контролю тестування студентів. Результати перевірки виводяться студентові на сторінку, записуються в базу даних для аналізу профілю студента, та передаються в плагін у системі Moodle для додавання оцінки до профілю студента. За результатами аналізу сервер перевірок завдань можна розділити на такі складові частини (рис. 2.2)



Рис. 2.2. Схема серверу перевірки завдань

## ВИСНОВКИ

Аналіз наукової літератури та інтернет-джерел допоміг з'ясувати, що в більшості вузів на сучасному етапі застосовується контроль знань у вигляді тестових завдань. Доцільність адаптивного контролю впливає з міркувань раціоналізації традиційного тестування. Підготовленому студенту немає необхідності давати легкі завдання. Легкі завдання не володіють помітним розвивають потенціалом, в той час як важкі завдання у більшості студентів знижують навчальну мотивацію. Використання завдань, що відповідають рівню підготовленості, істотно підвищує точність вимірювань і мінімізує час індивідуального тестування. Як основу для побудови системи адаптивного контролю знань ми використовували систему управління навчанням Moodle. Ми додали до системи посилання на окремий сервер, на якому розташовані модулі для підтримки системи адаптивного тестування. Системи адаптивного тестування була розроблена на окремому сервері. Були розглянуті фактори, які впливають на вибір мови програмування для реалізації адаптивної системи контролю навчання. Серед них: . Розроблений модуль повинен отримувати інформацію від системи онлайн навчання. Модуль повинен мати окрему базу даних для аналізу. Розроблений модуль повинен мати здатність до самонавчання або навчання з вчителем та мати високу швидкість обробки даних для видачі рекомендацій. З великої кількості мов програмування ми була вибрана мова, у якій є наявність математичного блоку для аналізу даних та розробки нейронних мереж, та мов, які використовуються у серверному програмуванні – мова Python. Представляється перспективним підключення до системи адаптивного контролю навчання додаткових модулів, які дозволять функції адаптивного контролю.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Brusilovsky P., Peylo C. Adaptive and intelligent web-based educational systems. *International Journal of Artificial Intelligence in Education*, 13, 2003. С. 156-169.
2. Chollet F. Deep Learning with Python, Manning Publications Co, 2018. 386 с.
3. Cisar M., Dragica R., Branko M. Computer Adaptive Testing of Student Knowledge. URL: [https://www.researchgate.net/publication/49619355\\_Computer\\_Adaptive\\_Testing\\_of\\_Student\\_Knowledge](https://www.researchgate.net/publication/49619355_Computer_Adaptive_Testing_of_Student_Knowledge). (дата звернення: 12.10.2020).
4. Contester. Система для проведення турнірів и індивідуального рішення задач по олімпіадному програмуванню. URL: <http://www.contester.ru>
5. Dall'Acqua L. A model for an adaptive e-Learning environment. *Proceedings of the World Congress on Engineering and Computer Science (WCEES)*. 2009. Vol. 42, № 5.
6. Dekson, D.E., Suresh, E.S.M. Adaptive E-Learning techniques in the development of teaching Electronic Portfolio – A survey. *International Journal of Engineering Science and Technology*. 2010. 2(9), pp. 4175-4181.
7. Dempster, A.P. A generalization of Bayesian inference. *Journal of the Royal Statistical Society, Series B*. 1968. 30, pp. 205-247.
8. Ju, Gin-Fon N., Bork, A., The Implementation of an Adaptive Test on the Computer. *Fifth IEEE International Conference on Advanced Learning Technologies (ICALT'05)*, Kaohsiung, Taiwan, 2005, pp. 822-823, <http://doi.ieeecomputersociety.org/10.1109/ICALT.2005.274>
9. Kadan, A. Interactive Internet-service for Computer Testing in Mathematics *Pattern Recognition and Information Processing: Proceedings of The Seventh International Conference*. Vol. 1. – Minsk: United Institute of Informatics Problem of National Academy of Sciences of Belarus, 2003. С.178 – 181.



10. Kardan, S., Kardan, A. Towards a More Accurate Knowledge Level Estimation. *2009 Sixth International Conference on Information Technology: New Generations*, Las Vegas, Nevada, 2009. C. 1134-1139.
11. Lewis, J.R. Psychometric evaluation of the PSSUQ using data from five years of usability studies. *International Journal of Human-Computer Interaction*, 14, 2002. C. 463-488.
12. Limongelli C., Sciarrone F., Temperini M. Adaptive Learning with the LS-Plan System: A Field Evaluation. *IEEE transactions on learning technologies*, vol. 2, no. 3, 2009. pp. 203-215
13. Lutz M. Learning Python, Fifth Edition. London: O'Reilly Media, Inc, 2019. 1594 с.
14. Mathworks, Computer Adaptive Test Demystified: веб-сайт. URL: <http://www.mathworks.com/matlabcentral/fileexchange/12467-computeradaptive-test-demystified-gre-pattern>.
15. Moodle. Система управления обучением веб-сайт. URL: <https://moodle.org/>
16. Mustafa, Y.E.A., Sharif, S.M. An approach to adaptive E-Learning hypermedia system based on learning styles (AEHS-LS): implementation and evaluation. *International Journal of Library and Information Science* 2011, 3(1), C. 15-28.
17. Paramythis, A., Loidl-Reisinger, S. Adaptive learning environments and eLearning standards. URL: <http://www.fim.uni-linz.ac.at/staff/paramythis/papers/ecel2003.pdf>
18. Shute, V. & Towle, B. Adaptive e-learning. *Educational Psychologist*, 38(2), 2003. C.105-114.
19. Sonamthiang, S., Naluedomkul, K., Cercone N. Initializing student models using Dempster-Shafer theory. *Proceedings of International Conference on Computer and Advanced Technology in Education (CATE2006)*. No.4, 2006

20. Vatcharaporn, E., Supaporn, L. Student Modelling in Adaptive E-Learning Systems. *Knowledge Management & E-Learning: An International Journal*, Vol.3, No.3, 2009
21. Адаптивные модели диалога в компьютерных обучающих системах: монография/ Гданский Н. И., Рысин М. Л., Леванов Д. Н., Альтиментова Д. Ю. Москва: Изд-во РГСУ, 2013. 100 с.
22. Богорев В.В. Психолого-педагогические основы системы адаптивного обучения. *Наука и школа*. 2001. № 2. С. 12–15.
23. Вергазов Р.И. Система компьютерного тестирования знаний: дис. ... канд. техн. наук. Пенза, 2006.
24. Вишнякова С. М. Профессиональное образование: словарь, ключевые понятия, термины актуальная лексика. 2-е изд., доп. и испр. Москва : НМЦ СПО, 2005. 558 с.
25. Воробейчикова О. В. Структурированные тесты как средства контроля знаний. *Информатика в образовании*. 2001. № 7. С. 14–17.
26. Выготский Л. С. Педагогическая психология. Москва : Изд-во Моск. гор. пед. ун-та, 2008. 671 с.
27. Галинский Д. Н. Адаптивное тестирование в оценке качества образования. *Теории, содержание и технологии высшего образования в условиях глобализации образовательного процесса*. Оренбург: Изд-во ОГПУ, 2006. С. 11–18.
28. Гальперин П. Я. Психология как объективная наука / под ред. А.И. Подольского. Москва : Ин-т практической психологии, 2004. – 480 с.
29. Гданский Н.И., Альтиментова Д.Ю. Применение адаптации при тестировании знаний в условиях компьютерного обучения. *Преподаватель – XXI век*. 2014. С.115–126.
30. Гданский Н.И., Рысин М.Л., Альтиментова Д.Ю. Адаптивные алгоритмы тестирования знаний информационно-тестирующих приложений на основе стандартных LMS-систем. *Ученые записки Российского государственного социального университета*. 2013. № 5 (120). Том 2. С. 77–82.

31. Гданский Н. И., Альтиментова Д. Ю. Адаптивные методы тестирования знания при компьютерных формах обучения: монография. Москва: Издательство Российского государственного социального университета 2015. 219 с.

32. Гданский Н. И., Альтиментова Д. Ю., Рысин М. Л. Многокритериальные адаптивные алгоритмы для тестирования знаний. *Матер. 9 Междунар. науч.-практ. конф. «Наука и инновации – 2013»* Пшемысль. 2013. Т.9. Педагогические науки. Пшемысль: Nauka i studia. С. 3–9.

33. Гданский Н. И., Рысин М. Л., Альтиментова Д.Ю. Особенности преподавания программирования в условиях современного рынка труда в ИТтехнологиях. *Ученые записки Российского государственного социального университета*. 2013. № 5 (120). Том 2. С. 73–77.

34. Гданский Н. И., Рысин М. Л., Леванов Д. Н. Разработка новых адаптивных моделей диалога в компьютерных обучающих системах. *Труды Междунар. науч.-практ. конф. «Наука и инновации – 2013»* Пшемысль: Nauka i studia, 2013.

35. Герасимов В.П., Карпенко Г.Г. Информационные технологии в образовании, управлении, научно-исследовательской деятельности. Невинномысск: Невинномыс. город. типография, 2006. 221 с.

36. Горбач Н.А., Павлюк А.А. Интеллектуальный анализ данных в прогнозировании успешности учебной деятельности обучаемых вузов. *Информатизация и информационная безопасность правоохранительных органов*. Москва, 2004. С. 21–26.