

Міністерство освіти і науки, молоді та спорту України
Тернопільський національний економічний університет

На правах рукопису

Мельник Андрій Миколайович

УДК 004.415:378.147

Інформаційна технологія автоматичної генерації тестових
завдань з керованою складністю

05.13.06 – інформаційні технології

Дисертація на здобуття наукового ступеня кандидата
технічних наук

Науковий керівник:
Пасічник Роман Мирославович,
кандидат фізико-математичних наук,
доцент

Тернопіль – 2011

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ.....	5
ВСТУП	6
РОЗДІЛ 1 ТЕСТУВАННЯ ЯК ЗАСІБ КОНТРОЛЮ ЗАСВОЄННЯ ЗНАНЬ В КОМП'ЮТЕРИЗОВАНИХ НАВЧАЛЬНИХ СИСТЕМАХ.....	12
1.1 Комп'ютеризація контролю знань в процесі навчання.....	12
1.2 Технології подання знань та їх використання при автоматичній генерації тестових завдань.....	23
1.2.1 Моделі подання знань	23
1.2.2 Аналіз відомих методів автоматичної генерації тестових завдань	25
1.3 Методи та засоби організації дистанційного навчання із управлінням часовими ресурсами	28
1.4 Постановка задачі дослідження.....	36
Висновки до розділу 1	43
РОЗДІЛ 2 ТЕХНОЛОГІЯ ФОРМАЛІЗАЦІЇ НАВЧАЛЬНИХ МАТЕРІАЛІВ ЗАСОБАМИ РЕЛЯЦІЙНОЇ АЛГЕБРИ.....	46
2.1 Формалізація навчального матеріалу за допомогою деревовидних структур	46
2.2 Методи автоматичної генерації тестових завдань для перевірки теоретичних знань.....	55
2.3 Метод генерації проблемних ситуацій для тестування методологічних знань	68
Висновки до розділу 2	76
РОЗДІЛ 3 ТЕХНОЛОГІЯ ОРГАНІЗАЦІЇ ПРОЦЕСУ НАВЧАННЯ НА ОСНОВІ АДАПТИВНИХ ГЕНЕРАТОРІВ ТЕСТОВИХ ЗАВДАНЬ..	77
3.1 Метод генерації завдань для перевірки вмінь та навичок	77
3.2 Адаптивно-структурний метод оцінки складності тестових завдань	86

3.3	Метод адаптивного управління структурою процесу навчання.....	92
	Висновки до розділу 3	99
РОЗДІЛ 4 ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ АВТОМАТИЧНОЇ ГЕНЕРАЦІЇ		
	ТЕСТОВИХ ЗАВДАНЬ З КЕРОВАНОЮ СКЛАДНІСТЮ	100
4.1	Система SAGT автоматичної генерації тестових завдань	100
4.2	Експериментальні дослідження методів автоматичної генерації тестових завдань закритого типу.....	108
4.3	Експериментальні дослідження методів автоматичної генерації тестових завдань для перевірки засвоєння методологічних знань, вмінь та навичок.....	113
4.4	Експериментальні дослідження адаптивно-структурного методу оцінки складності тестових завдань.....	119
4.5	Експериментальні дослідження методу адаптивного управління структурою процесу навчання.....	126
4.6	Порівняння інформаційних технологій автоматизованого навчання.....	135
	Висновки до розділу 4	140
	ВИСНОВКИ	142
	СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	144
	Додаток А Основні операції класичної алгебри відношень (реляційної алгебри).....	158
	Додаток Б Системи дистанційного навчання	159
	Додаток В Лістинг модуля автоматичної генерації тестових завдань різних видів закритого типу	160
	Додаток Г Графічний інтерфейс модуля управління навчальним контентом ..	170
	Додаток Д Графічний інтерфейс форми «Автоматичне опрацювання конспекту».....	171
	Додаток Е Список тестових завдань, що використано в експериментальних дослідженнях	172

Додаток Ж Показники успішності для типових категорій студентів та розподіл тестових завдань по рівнях складності.....	175
Додаток И Акти впровадження результатів дисертаційної роботи.....	176

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

АГТЗ	Автоматична генерація тестових завдань
АК	Алгебра кортежів
АРИЗ	Алгоритм рішення изобретательських задач
БД	База даних
БЗ	База знань
_C	Вибраний елемент із множини кортежів
ІНС	Інтелектуальна навчальна система
ІСВ	Інтерактивні середовища вивчення
ІТ	Інформаційна технологія
НП	Навчальний процес
ПТ	Параметризовані тести
ПТМ	Понятійно-тезисна модель
ПЗ	Програмне забезпечення
РМД	Реляційна модель даних
РГТ	Ручне генерування тестів
СКБД	Система керування базами даних
ІРТ	Сучасна теорія тестування
ТРИЗ	Теорія рішення изобретательських задач
ІІ	Штучний інтелект
ARIS	Architecture of Integrated Information Systems
SAGT	System automatic generation of tests

ВСТУП

Актуальність теми. Інформаційне суспільство - сучасний етап розвитку цивілізації, яке в значній мірі формується освітою, що підтримує суспільні способи отримання, обробки і розповсюдження інформації. Забезпечення відповідності освіти фахівців динаміці суспільних процесів породжує гостру необхідність в інтенсифікації як самого процесу навчання, так і його організації. Сучасний навчальний процес складно уявити без використання комп'ютерних засобів в управлінні процесом навчання, структуризації навчального контенту, автоматизації формування засобів контролю засвоєння знань.

У сферу досліджень управління процесом навчання та структуризацію навчального контенту з використанням інформаційних технологій зробили вагомий внесок такі науковці як Аванесов В. С., Скіннер Б. Ф., Краудер Н., Прессі С., Згуровський М. З., Томашевський В. М., Федорук П. І., Таран Т. А., Панкратов Д. В, Тонконогий В. М., Атанов Г. А., Андрєєв А. А., Кухаренко В. М., Брусиловський П., Тассо С., Шварц Е., Міллер Р. Проблеми автоматизації процесу тестування і обробки його результатів достатньо повно досліджені в літературі. Однак задачі автоматизації формування банку тестових завдань досліджено значно менше. Серед науковців що працювали над даним питанням варто відзначити Башмакова О. І., Швецова А. Н., Гагаріна О. О., Титенка С. В. Важливою характеристикою цінності методів формування інформаційної бази тестових завдань є часові витрати для їх побудови. Зменшення таких витрат не повинно супроводжуватися втратою якості та варіативності побудованих тестових наборів. Тому поряд із напрацюваннями згаданих авторів невирішеною залишається проблема автоматичного формування множини тестових завдань, які при мінімізації часових затрат на їх розробку, забезпечували достатню варіативність та якість завдань.

Суттєвою складовою якості тестових завдань є їх складність, яка відіграє ключову роль в організації як навчального так і контрольного процесу тестування. При цьому питання управління складністю автоматично згенерованих завдань

залишається недостатньо дослідженим. Таким чином актуальною є задача мінімізації часових витрат на формування тестових завдань при забезпеченні їх достатньої складності та повноти контролю знань шляхом створення інформаційної технології автоматичної генерації тестових завдань з керованою складністю.

Зв'язок роботи з науковими програмами, планами, темами. Науково-дослідна робота за темою дисертації проводилася у відповідності з координаційним планом науково-дослідних робіт і науково-технічних програм Міністерства освіти і науки України, розділ „Математичні методи, інтервальні моделі та інформаційні технології для контролю забруднення атмосфери автотранспортом, а також робота проводилася відповідно з планами навчальної та науково-дослідної роботи Тернопільського національного економічного університету, зокрема в рамках теми „Методи та засоби математичного моделювання складних систем на основі теоретико-множинного та інтервального підходів” (номер державної реєстрації 0106U012529), зокрема у розробці розділу „Розробка алгоритмів опрацювання даних інформаційних систем навчальних закладів”, у якій автором розроблено моделі управління затратами часу навчання для досягнення бажаного рівня успішності, розроблено критерій та метод оцінки складності тестових тверджень.

При дослідженнях по темі «Моделювання динаміки управлінських, маркетингових, виробничих, фінансових та екологічних процесів в організаційних системах» (номер державної реєстрації 0106U012531), зокрема у розробці розділу „Методи формалізації процесу генерації тестових завдань для електронних навчальних курсів на основі семантичних класів”, автором вдосконалено алгоритми автоматичної генерації тестових завдань відкритого та закритого типів. А також у виконанні госпдогвірної науково-дослідної роботи на тему „Методи та засоби генерування тестів із застосуванням теорії онтологій” (номер державної реєстрації 0111U006279). Автором реалізовано веб-орієнтовану систему автоматичної генерації тестових завдань SAGT.

Мета і задачі дослідження. Метою дослідження є створення інформаційної технології автоматичної генерації тестових завдань з керованою складністю, яка

відрізняється від інших використанням семантичної, формальної та синтаксичної типізації структурованих текстових тверджень, формуванням завдань для перевірки засвоєння методологічних знань та задач для перевірки вмінь та навиків алгоритмізації.

Для досягнення цієї мети у дисертаційній роботі поставлено наступні завдання:

1) аналіз сучасних моделей, методів та засобів організації навчального процесу, які забезпечують його автоматичну підтримку;

2) розробка моделей підтримки семантичної, формальної та синтаксичної однорідності компонентів тестових завдань;

3) розробка методів автоматичної генерації тестових завдань для перевірки рівня засвоєння теоретичних знань, а також практичних вмінь та навичок;

4) розробка методу апріорної оцінки складності автоматично згенерованих тестових завдань;

5) розробка моделі управління навчанням з метою диференціації зусиль студента в умовах обмежених часових ресурсів;

6) на основі розроблених моделей та методів здійснити програмну реалізацію технології автоматичної генерації тестових завдань з керованою складністю.

Об'єкт досліджень: процеси контролю та оцінювання знань з використанням інформаційних технологій.

Предмет досліджень: методи та засоби інформаційної технології автоматичної генерації тестових завдань з керованою складністю.

Методи дослідження. Дослідження, виконані під час роботи над дисертацією, ґрунтуються на використанні системного аналізу, теоретико-множинного підходу, методів реляційної алгебри – для розробки засобів формалізації навчального контенту, методів автоматичної генерації тестових завдань; методах оптимізації – для реалізації методу оцінки складності тестових завдань, методу адаптивного управління структурою процесу навчання.

Наукова новизна отриманих результатів.

Вперше:

– розроблений метод генерації тестових завдань для перевірки засвоєння методологічних знань на основі формалізації типових проблемних ситуацій та їх варіативних компонентів, який на відміну від відомих методів використовує спеціальну систему відношень та предикатних операторів, що дозволило зменшити часові затрати при встановленні причинно-наслідкових зв'язків для формування тестових завдань;

– розроблений метод генерації задач для перевірки вмій та навиків, що забезпечує модифікацію структури задачі, який на відміну від відомих використовує формалізації програмованих операторів, відношень між лінгвістичними та формалізованими компонентами вхідних даних та операцій. Метод забезпечує економію затрат часу розробника на створення нових завдань на основі варіативності основних компонентів структури задачі;

– розроблений адаптивно-структурний метод оцінки складності тестових завдань на основі їх формальних характеристик із налаштуванням для конкретної аудиторії та предмету вивчення. Метод, на відміну від існуючих, дозволяє автоматизувати генерацію тестових завдань заданої складності узгодженої із успішністю їх проходження для певного предмету та аудиторії.

Вдосконалено:

– метод генерації тестових завдань закритої форми на основі формалізації структурованих текстових тверджень, який на відміну від відомих використовує семантичну, формальну та синтаксичну типізацію і забезпечує економію затрат часу розробника на створення нових тестів при забезпеченні їх якості на основі багатократного використання раніше напрацьованих компонент.

Отримав подальший розвиток:

– метод адаптивного управління структурою процесу навчання на основі функції часових затрат. Цей метод на відміну від існуючих, враховує розподіл тестових завдань по рівнях складності та забезпечує адаптацію рекомендованих часових затрат до динаміки процесу навчання, що дозволило зменшити час досягнення заданих рівнів успішності.

Практичне значення отриманих результатів полягає в тому, що на основі розроблених методів та засобів реалізовано систему автоматичної генерації тестових завдань, яка дозволяє генерувати тестові завдання різних типів, проводити управління процесом навчання для досягнення максимальних результатів при мінімальних часових затратах викладачів та студентів. На базі системи розроблено програмне забезпечення, яке використовується у навчальному процесі в Тернопільському національному економічному університеті та Івано-Франківській філії Відкритого міжнародного університету розвитку людини «Україна».

Основні практичні результати дисертаційної роботи використані:

- для автоматичної генерації тестів із достатньою варіативністю та керованим рівнем складності;
- у навчальному процесі на кафедрах комп'ютерних наук та економічної кібернетики та інформатики Тернопільського національного економічного університету при вивченні дисциплін «Проектування баз даних і знань», «Математичне забезпечення програмних систем», «Засоби програмування баз даних знань», «Технологія створення програмних та інтелектуальних систем».

Особистий внесок здобувача. Усі наукові результати, викладені у дисертаційній роботі, отримані автором особисто. У друкованих працях, опублікованих у співавторстві, автору належить: методика використання системи базових класів для побудови онтологій різних предметних областей [14]; дослідження відомих підходів до моделювання параметрів навчального процесу [16, 17]; метод мінімізації часу інтерактивного навчання для забезпечення необхідного рівня успішності на основі моделей успішності та функції часових затрат [1, 5, 12]; метод оцінки складності тестових завдань на основі показників документованості та автентичності навчального матеріалу [2, 18]; методи генерації тестових завдань закритої форми на основі формалізації структурованих текстових тверджень [3, 15]; метод генерації тестових завдань для перевірки засвоєння методологічних знань на основі формалізації типових проблемних ситуацій та їх варіативних компонентів [4, 10]; метод генерації задач для перевірки вмін та навиків із

програмованим оператором [7]; розроблення інформаційної технології автоматичної генерації тестових завдань з керованою складністю [6, 13].

Апробація результатів дисертації. Основні положення дисертації були представлені та обговорювалися на 12 міжнародних та всеукраїнських конференціях, семінарах, симпозіумах: VIII Всеукраїнській науковій конференції “Сучасні проблеми прикладної математики та інформатики” (Львів, 2006 р.); IX та X Міжнародних науково-технічних конференціях “Сучасні проблеми радіоелектроніки, телекомунікацій та комп’ютерної інженерії: TCSET’2008, 2010” (Львів, 2008, 2010 рр.); Проблемно-науковій міжгалузевій конференції “ПНМК 2009” (Бучач, 2009 р.); Міжнародній науково-технічній конференції “ISDMCI 2009” (Євпаторія, 2009 р.); IX Міжнародній науково-технічній конференції “Досвід розробки й застосування САПР в мікроелектроніці CADSM’2009” (Львів, 2009 р.); Міжнародній науково-практичній конференції “Інформаційні технології та комп’ютерна інженерія ІТКІ 2010” (Вінниця, 2010 р.); II науково-практичної конференції “Інноваційні комп’ютерні технології у вищій школі” (Львів, 2010р.); II Міжнародній науково-практичній конференції “Інформаційні технології та комп’ютерна інженерія”, 26–27 травня 2011 р., ХНЕУ, Харків; I Всеукраїнській школі-семінарі молодих вчених і студентів «Сучасні комп’ютерні інформаційні технології», 21-22 травня 2011 р., ТНЕУ, Тернопіль; Школі-семінарі для молодих науковців «Індуктивне моделювання: теорія і застосування» (с. Жукин, Київська обл., 2011); VI Міжнародному науково-технічному симпозіумі “Інтелектуальні засоби збору даних і сучасні обчислювальні системи: розробка та застосування IDAACS’2011” (Чехія, Прага, 2011 р.);

Публікації. Основні результати дисертаційної роботи висвітлено в 18 друкованих працях, загальним обсягом 82 сторінки, у тому числі 8 – у фахових виданнях, 10 – у матеріалах конференцій.

Структура та обсяг роботи. Дисертація складається зі вступу, чотирьох розділів, висновків, списку використаних джерел та додатків. Робота містить 138 сторінок основного тексту. Загальний об’єм дисертації – 179 сторінок, 4 таблиці, 62 рисунки, 8 додатків, 122 найменування використаних джерел.

РОЗДІЛ 1

ТЕСТУВАННЯ ЯК ЗАСІБ КОНТРОЛЮ ЗАСВОЄННЯ ЗНАНЬ В КОМП'ЮТЕРИЗОВАНИХ НАВЧАЛЬНИХ СИСТЕМАХ

1.1 Комп'ютеризація контролю знань в процесі навчання

Процес навчання - це цілеспрямований процес послідовного розвитку взаємодії викладача і студента, в ході якого вирішуються завдання освіти, виховання і загального розвитку [24]. Структуру навчального процесу складають мета, викладач, студент, зміст навчання [77], а метою виступає обсяг і якість знань, якими повинен оволодіти студент. В процесі навчання визначальна роль відводиться викладачу, який визначає мету навчального процесу, зміст навчального матеріалу, структуру занять і методи навчальної діяльності. Студент - особа зацікавлена в навчанні, яка прагне до співпраці з викладачем. Ця співпраця полягає в співпадінні цілей викладача та студента, в рівноправному їх партнерстві, коли студент стає не пасивним об'єктом, а як і викладач, активним суб'єктом навчання. Зміст навчання одночасно виступає засобом навчання, виховання і розвитку студентів.

Процес навчання є циклічним рухом, що включає послідовно повторювані етапи навчального процесу: первинна діагностика та актуалізація набутих знань студентів; постановка викладачем навчальної мети і усвідомлення студентами пізнавальних завдань; сприйняття вивчення студентами нового матеріалу із використанням різних методів і прийомів викладу; закріплення сприйнятої інформації в процесі формування нових вмінь і навичок в її застосуванні; контроль якості засвоєних знань та вмінь, що є способом зворотного зв'язку та діагностики на завершальному етапі процесу навчання. Окрім згаданих етапів варто відзначити корекцію за результатами діагностики, при відхиленнях від наміченого результату пізнавальної діяльності; узагальнення, яке завершує попередні етапи і передбачає засвоєння і усвідомлення студентами причинно-

наслідкових зв'язків у явищах навколишнього світу, фрагменти якого вони вивчають, і встановлює внутрішньопродметні та міжпродметні зв'язки.

Контроль знань студентів є складовою частиною процесу навчання, що являє собою співставлення досягнутих результатів із запланованими цілями цього процесу. Перевірка знань студентів повинна давати відомості не тільки про правильність чи неправильність кінцевого результату виконаної діяльності, але і про неї саму: чи відповідає форма дій даному етапу засвоєння.

На сьогоднішній день в освітніх установах контроль знань здійснюється як традиційними методами (усне опитування, письмові роботи), так і з використанням комп'ютерних технологій (електронне тестування). Теоретичною основою тестів послужила біхевіористична концепція навчання, розроблена американськими дослідниками Торндайком, Скіннером. Ця концепція розглядає навчання як процес, що реалізується по етапах: стимул → реакція → підкріплення або $S \rightarrow R \rightarrow P$ [80].

Процес використання тестів породжує як своїх прихильників, так і супротивників. Перші розглядають тести як засіб радикального перетворення навчального процесу в бік його комп'ютеризації, зниження трудомісткості та зниження суб'єктивності оцінювання. Інші вважають тести засобами надмірної формалізації процесу навчання, фрагментації знань, зменшення творчої ролі викладача в навчальному процесі [4, 52]. На наш погляд небезпека надмірної формалізації навчального процесу при використанні тестів існує. Однак вона може бути знівельована за рахунок високої варіативності форм тестових завдань, коли контролюється реакція не на форму, а на зміст подачі матеріалу. Економія зусиль викладача на здійснення рутинних операцій в контролі знань дає змогу направити більше уваги на покращення змісту навчального матеріалу, що дозволяє розглядати тести, при належному їх використанні, як засіб підвищення ефективності навчального процесу.

За своїм призначенням тест є системою тестових завдань зростаючої складності, що дозволяє ефективно виміряти рівень і якісно оцінити структуру підготовленості суб'єктів тестування. Аналізуючи основні принципи побудови

тестових завдань, що забезпечують їх належну якість [2, 4, 15, 27], відзначимо необхідність перевірки знань на найновішому навчальному та контрольному матеріалі, що породжує високу варіативність тестів для прикладних дисциплін [17].

Перші наукові праці з теорії тестів з'явилася на початку ХХ століття, на стику психології, соціології, педагогіки та інших так званих поведінкових наук (Behavioral Sciences), коли тестування починає розвиватися як самостійна наука – тестологія. Тестологія (від англ. Test - проба, грец. Logos - знання) - міждисциплінарна наука про створення якісних і науково обгрунтованих вимірювальних діагностичних методик [63].

Ключовими поняттями тестології є моделі тестування, зміст і форма тестових завдань, їх складність, надійність і валідність результатів вимірювання. Проаналізуємо основні підходи до оцінювання складності тестових завдань і моделей тестування, а також запропонуємо деякі формальні критерії їх ефективності.

Практично усі сучасні методи оцінки складності тестових завдань базуються на статистичних оцінках або структурному аналізі (рисунок 1.1).

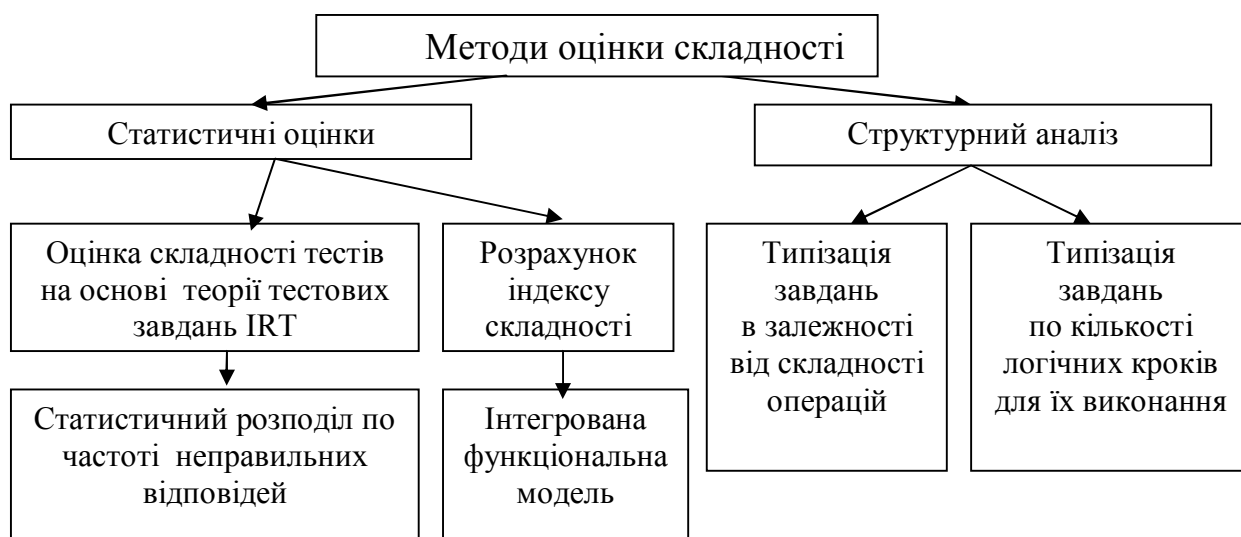


Рисунок 1.1 – Основні методи оцінки складності тестових завдань

В теорії IRT (Item Response Theory) залежно від кількості оцінюваних параметрів розрізняють кілька видів математичних моделей, використовуваних для аналізу якості тестів [68]. Однопараметрична модель Раша встановлює залежність між рівнем підготовленості θ_i випробуваного i та трудностю β_j завдання j . Ймовірність правильної відповіді P на завдання тут виражається за допомогою логістичних функцій (рисунок 1.2, 1.3)

$$P_j(\theta) = \frac{e^{1,7(\theta - \beta_j)}}{1 + e^{1,7(\theta - \beta_j)}}, \quad j = 1, \dots, M, \quad (1.1)$$

$$P_i(\beta) = \frac{e^{1,7(\theta - \beta)}}{1 + e^{1,7(\theta - \beta)}}, \quad i = 1, \dots, N, \quad (1.2)$$

де θ та β - незалежні змінні; M - кількість завдань у тесті; N - кількість тестованих.

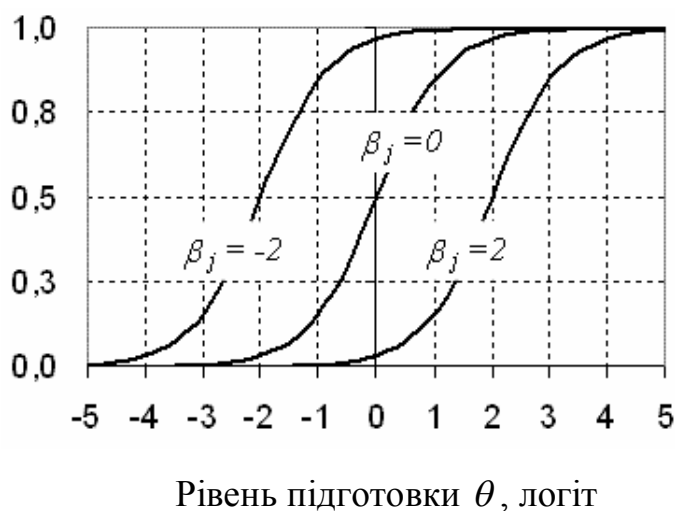


Рисунок 1.2 – Ймовірність вірної відповіді в залежності від рівня підготовки для різних рівнів складності завдань β_j

Раш припустив, що рівень підготовленості випробуваного θ_i і рівень складності завдання β_j розміщені на одній шкалі і вимірюються в одних і тих

же одиницях - логітах. Аргументом функції успішності випробуваного є різниця $\theta_i - \beta_j$.

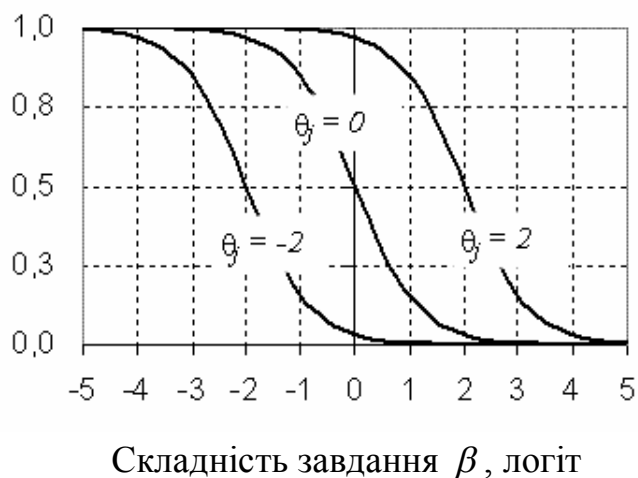


Рисунок 1.3 – Ймовірність вірної відповіді в залежності від рівня складності завдань для різних рівнів підготовки θ_j

Двопараметрична модель Бірнбаума (A. Birnbaum) дозволяє додати для аналізу ще один параметр – диференційовану здатність, що дозволяє оцінити величини структурованості знань. При цьому функцій ймовірності правильної відповіді набувають наступного вигляду (рис. 1.4):

$$P_j(\theta) = \frac{e^{1,7a_j(\theta - \beta_j)}}{1 + e^{1,7a_j(\theta - \beta_j)}}, \quad (1.3)$$

$$P_i(\beta) = \frac{e^{1,7a_i(\theta - \beta)}}{1 + e^{1,7a_i(\theta - \beta)}}, \quad (1.4)$$

де a_j характеризує диференційовану здатність завдань при зміні різних значень θ , а параметр a_i вказує на величину структурованості знань особи.

Багато авторів, зокрема Пол Клайн, відзначають, що IRT має численні недоліки. Для того щоб отримати надійну і не залежну від випробуваних шкалу властивостей, потрібно провести тестування великої вибірки (не менше 1000

випробовувань). Тестування досягнень показує, що існують значні розбіжності між прогнозами моделі і емпіричними даними.

Головний же недолік IRT - ігнорування проблеми валідності. У психологічній практиці не спостерігається випадків, коли відповіді на завдання тесту були б обумовлені лише одним фактором. Моделі IRT доцільно розглядати для коротких тестів з валідними завданнями (факторно простих тестів) [22].

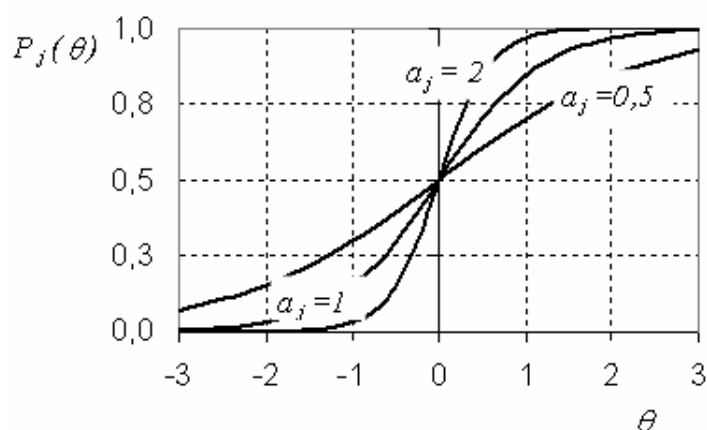


Рисунок 1.4 – Ймовірність вірної відповіді в залежності від рівнів диференційної здатності завдань a_j

Статистичний метод оцінки складності тестових завдань оцінює рівні складності в залежності від частоти помилкових відповідей під час проведення тестового контролю. Недоліки – необхідна велика статистична вибірка для адекватної оцінки параметрів розподілу згаданої частоти. Розвитком статистичного методу є метод індексу складності, який оцінює ступінь легкості завдання тесту з врахуванням усіх правильних відповідей на нього, поданих екзаменованими сильною та слабкою груп. Інтегрована функціональна модель ґрунтується на розподілі тестових завдань по рівнях складності виходячи з оцінки рівня знань, отриманого на попередньому тестуванні із врахування типів тестових завдань [13].

В структурному методі типізації завдань в залежності від складності операцій розглядають п'ять рівнів [79]: інформаційний (розпізнавання відомої

інформації); репродуктивний (основними операціями якого є відтворення інформації та її перетворення алгоритмічного характеру); базовий (розуміння важливих сторін навчальної інформації, володіння загальними принципами пошуку алгоритмів); підвищений рівень складності тестового завдання (вміння перетворювати алгоритми до умов, які відмінні від стандартних, вміння вести евристичний пошук); творчий (вміння розв'язувати нестандартні завдання). Основним недоліком даного методу є те, що оцінка складності йде по параметру документованості навчального матеріалу без врахування відповідності альтернатив.

В методі типізації завдань по кількості логічних кроків за критерій складності приймається кількість операцій, які необхідні для виконання того чи іншого завдання [84]. Кожна із операцій оцінюється за наступною шкалою:

- а) перший рівень складності (відновлення інформації – 0,5 балів);
- б) типове завдання по дисципліні (1 бал);
- в) застосування в нестандартних умовах (1.5 бали).

Застосування даного методу супроводжується суб'єктивними оцінками при формуванні завдань.

Підсумовуючи проведений вище аналіз необхідно відзначити базування усіх перелічених вище методів на статистичних (розподіл правильних та неправильних відповідей) або структурних оцінках. Статистичні методи вимагають великої вибірки, а методи на основі структурного аналізу – супроводжуються суб'єктивністю. Окрім того аналізовані теорії розглядають загальну оцінку складності тестового завдання без врахування характеристик його елементів. Ці недоліки можуть бути усунуті за допомогою адаптивно-структурного методу оцінки складності тестових завдань в якому слід поєднати переваги структурного аналізу та статистичних методів. Такий метод створено в рамках проведених досліджень і описано в підрозділі 3.2.

Реалізація методів оцінки складності завдань значною мірою залежить від форм їх представлення. Останні можуть бути достатньо різноманітними, зокрема, в роботі [64] виділяється 24 види таких форм, узагальнена

класифікація яких наведена на рисунку 1.5. Всі завдання розбиваються на дві великі групи - завдання відкритого та закритого типу. В основу класифікації покладено наявність або відсутність введення додаткової інформації випробуваним. Якщо додаткова інформація потрібна, то це завдання відкритого типу, а якщо ні – то закритого типу. Завдання відкритого типу поділяються на завдання з доповненням і у вигляді вільного викладу. У першому випадку випробуваному необхідно доповнити зміст завдання одним-двома словами. В результаті завдання повинне перетворитися на логічний вираз.



Рисунок 1.5 – Класифікація тестових завдань [29]

У тестуванні найчастіше використовуються завдання закритого типу [29]. Ці завдання характерні тим, що містять в собі і основу (питання, твердження) і відповіді (елементи відповідей), з яких випробуваний повинен вибрати або скласти правильну.

Завдання з градуйованими відповідями містять відповіді, які можливо є правильними в тій чи іншій мірі. Відповіді мають градацію за ступенем

правильності. Завдання викладача полягає в тому, щоб знайти і застосувати ознаку, що дозволяє здійснити таку градацію. Максимальну кількість балів випробуваний отримує, якщо його градація відповідей повністю збігається з градацією експерта, наприклад, розробника завдання.

Завдання на заповнення прогалін у тексті (завдання закритої форми) - вимагає заповнення таких прогалін у пропонованому запитанні. Завдання на встановлення відповідності вимагають від випробуваного знайти відповідність між елементами двох множин. Відповідність встановлюється на підставі логічних висновків або використанні смислових асоціацій.

У завданнях на встановлення правильної послідовності випробуваному необхідно не просто вибрати відповідні елементи відповіді, але і розташувати їх у потрібній послідовності. Завданнями такого типу добре перевіряти знання алгоритмів дій, технологічних прийомів, логіки міркувань і т.п. За допомогою цих завдань зручно перевіряти знання і розуміння формулювань визначень, понять, термінів, шляхом конструювання з окремим слів, пропозицій, символів, графічних елементів.

Форми тестових завдань у поєднанні з оцінкою їх складності є ключовими параметрами моделей тестування, огляд яких наведено на рисунку 1.6. Класична модель тестування передбачає наявність n завдань з яких випадковим чином вибирається k завдань, $k \leq n$, які пропонуються особі, яка тестується. Результатом тестування є процент правильних відповідей. Недоліки: через випадковості вибірки не можна заздалегідь визначити, які завдання за складністю отримаються, у підсумку один студент може отримати k легких завдань, а інший k складних, оцінка залежить тільки від кількості правильних відповідей і не враховується складність отриманих завдань.

Модель із врахуванням складності завдань передбачає оцінку складності кожного завдання T_i , $i=1, \dots, n$. З цієї множини завдань випадковим чином вибирається k завдань, $k \leq n$. При підрахунку результату тестування враховується складність питань, на які дано правильну відповідь. Недолік:

через випадковість вибірки не можна заздалегідь визначити, які завдання за складністю отримуються.

У моделі із зростаючою складністю виділяються m рівнів. У тесті мають бути присутніми завдання всіх рівнів. З множини завдань випадковим чином вибирається k завдань, $k \leq n$. Вибрані завдання сортуються за зростанням складності, кількість завдань по кожному рівню має бути однаковою. Дана модель забезпечує паралельність тестів за складністю, тобто надійність результатів тестування вища, ніж у попередніх моделях.

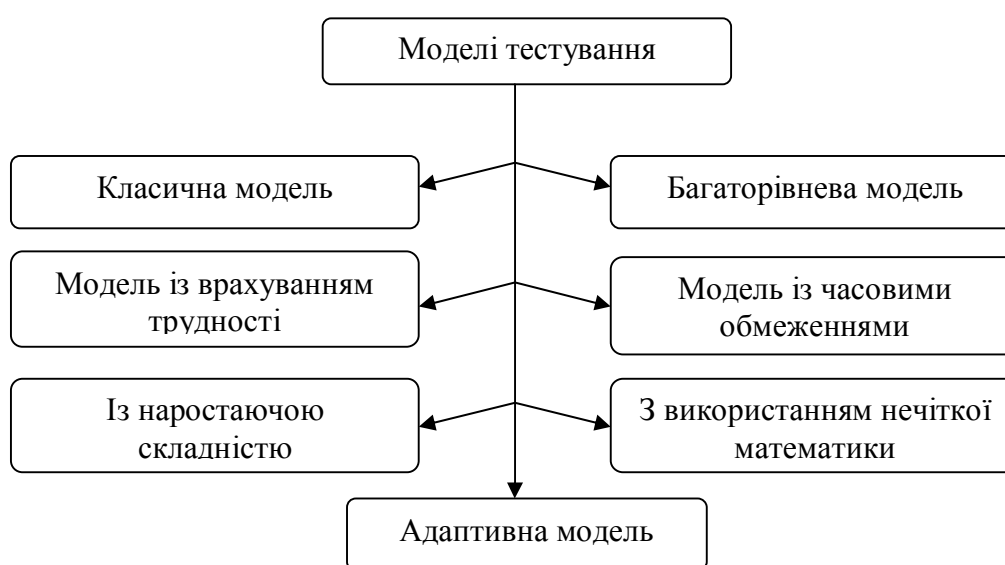


Рисунок 1.6 – Моделі тестування

Модель з розподілом завдань по рівнях засвоєння [89,90,91,92] передбачає п'ять рівнів засвоєння навчального матеріалу: розуміння, розпізнавання при повторному сприйнятті, відтворення засвоєних раніше знань, застосування знань в типових ситуаціях, генерування раніше невідомої інформації (розв'язання нетипових задач). Рівень засвоєння позначається a і може змінюватися від 0 до 4. Завдання складаються для кожного з п'яти рівнів. Тестування проводять послідовно по рівнях. Перед переходом з рівня на рівень обчислюється ступінь володіння навчальним матеріалом на даному рівні і визначається можливість переходу на наступний рівень. В моделях із часовими

обмеженнями на завдання або тест, контролюється рівень попередньої підготовки студента та швидкість аналізу в процесі формування [58].

Модель тестування із нечіткою математикою може застосовуватися в будь-якому із попередніх підходів, коли замість чітких характеристик тестових завдань і відповідей використовуються їх нечіткі аналоги. Наприклад: складність завдання («легке», «середнє», «вище середнього», «складне»); правильність відповіді («правильно», «частково правильно», «скоріше неправильно», «неправильно»); час відповіді («короткий», «середній», «тривалий», «дуже тривалий») [16, 23].

Адаптивним називається тест, в якому складність завдань змінюється залежно від правильності відповідей випробуваного. Якщо студент правильно відповідає на тестові завдання, складність наступних завдань підвищується, якщо неправильно - знижується. Також є можливість внесення додаткових питань з тем, які особа знає не дуже добре для більш тонкого з'ясування рівня знань у цих галузях. Дана модель застосовується для тестування студентів за допомогою комп'ютерних засобів, тому що «вручну» неможливо заздалегідь розмістити стільки питань і в тому порядку, скільки і в якому вони повинні бути пред'явлені особі, яка тестується [23].

Підсумовуючи наведену інформацію можна виділити адаптивні моделі тестування як перспективний напрямок у створенні автоматизованих систем навчання за критеріями ефективності навчання та складності програмної реалізації. В рамках даного напрямку розроблено метод адаптивного управління структурою процесу навчання, який наведено в підрозділі 3.3.

Автоматизація процесу навчання передбачає зменшення витрат праці викладача не тільки в проведенні самого тестування, але і при формуванні самих тестових завдань. Варто відзначити, що задачі автоматизації формування банку завдань досліджені недостатньо. Поряд із цим новітні інформаційні технології дають можливість суттєво зменшити трудові витрати на створення самих тестових завдань із можливістю їх постійного оновлення, що формує актуальний напрямок наукових досліджень.

В наступному підрозділі проводиться аналіз моделей подання знань, а також методів їх використання в автоматизації процесу генерування тестових завдань.

1.2 Технології подання знань та їх використання при автоматичній генерації тестових завдань

1.2.1 Моделі подання знань

При пошуку найбільш зручних, раціональних засобів і форм інформаційного наповнення часто виникає проблема компактного, однозначного та достатньо повного подання знань. Під знаннями розуміється система понять і відношень які включають інформацію про властивості об'єктів, закономірності процесів і явищ, а також правила використання цієї інформації для ухвалення рішень. Можна умовно класифікувати знання в предметній області на понятійні, конструктивні, процедурні, фактографічні та методологічні [4].

Понятійні знання – набір понять, які використовуються для розв'язання конкретної задачі, наприклад у фундаментальних науках і теоретичних областях наук, тобто понятійний апарат науки. Конструктивні знання – набори структур, підсистем системи і взаємозв'язків між їх елементами. Процедурні знання – методи, процедури (алгоритми) їх реалізації та ідентифікації. Фактографічні знання – кількісні та якісні характеристики об'єктів та явищ. Методологічні знання – це знання про методи перетворення дійсності, наукові знання про організацію ефективної діяльності. В трактуванні В.С. Аванесова останній вид знань віднесено до знань найвищого рівня.

Представлення знань – це процес, метою якого є відображення семантичного змісту, значення у вигляді синтаксичних форм: фраз усної мови,

письмових фраз, сторінок книг, понять довідника, об'єктів географічних карт тощо.

На даний момент розроблено множину моделей подання знань, які зображено на рисунку 1.7.

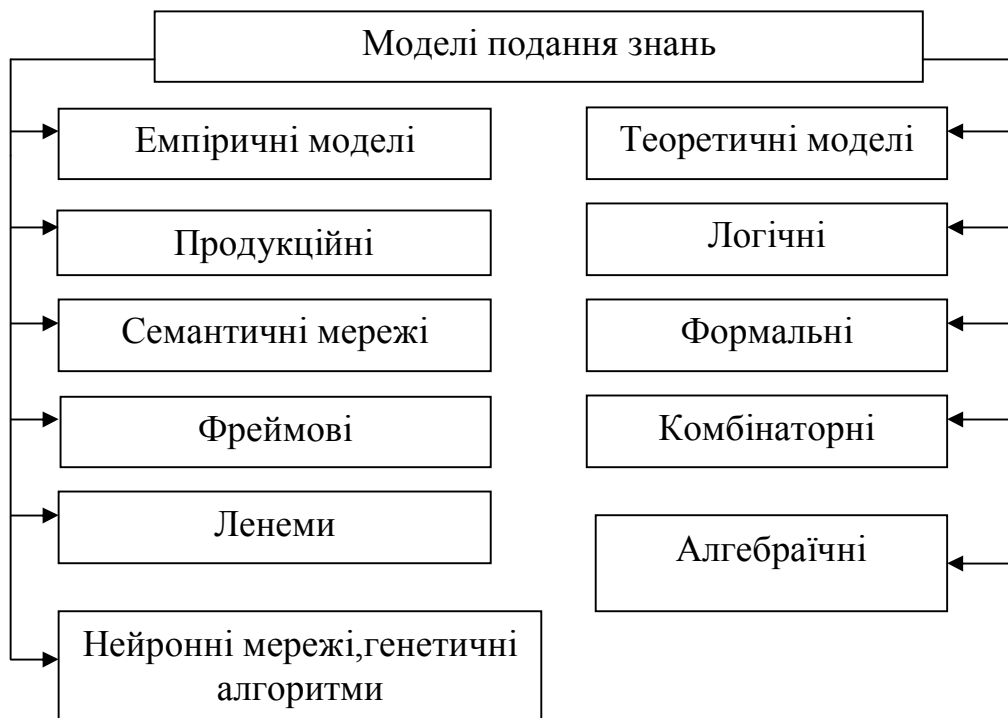


Рисунок 1.7 – Моделі подання знань

Із множини наведених моделей проаналізуємо лише найвідоміші з них, тобто, продукційні, фреймові моделі, а також семантичні мережі. Продукційна модель ґрунтується на правилах, які дозволяють представити знання у вигляді відношень типу: «якщо умова, то дія». Вона є надто жорсткою конструкцією і тому при накопичені великої кількості продукцій вони можуть суперечити одна одній. В інженерії знань під семантичною мережею розуміється граф, який відображає зміст цілісного образу. Вузли графу відповідають поняттям і об'єктам, а дуги – відношенням між об'єктами. Недоліком такого підходу є громіздкість представлення можливих зв'язків в мережі. Фреймова модель будується на основі фреймів, тобто структур, призначених для представлення деякого концептуального об'єкта. Інформація фрейму міститься у його слотах. Слоти можуть бути термінальними або бути також фреймами, утворюючи

ієрархічну мережу [47,83].

Одним із засобів представлення формалізованих знань є реляційна модель даних. Принципи реляційної моделі були сформульовані в 1969-1970 роках Е. Ф. Коддом (EF Codd) [67]. Переваги реляційної моделі: простота і доступність для розуміння користувачем, оскільки використовується єдина інформаційна конструкція - таблиця; правила проектування, що базуються на математичному апараті; повна незалежність даних. Зміни в прикладній програмі при зміні реляційної БД мінімальні; для організації запитів та написання прикладного програмного забезпечення немає необхідності знати конкретну організацію БД у зовнішній пам'яті.

Недоліки реляційної моделі: не завжди предметна область може бути представлена у вигляді "таблиці"; в результаті логічного проектування з'являється множина "таблиць". Це призводить до труднощів розуміння структури даних; БД займає відносно багато зовнішньої пам'яті; відносно низька швидкість доступу до даних [65]. Алгебра відношень, яка використовується у РМД, подано в додатку А.

1.2.2 Аналіз відомих методів автоматичної генерації тестових завдань

Серед відомих засобів автоматичної генерації тестових завдань необхідно відзначити методи параметризованих тестів, використання семантичних мереж а також використання понятійно-тезисної моделі [32].

Одним із перспективних і порівняно нескладним у реалізації є метод параметризованих тестів [81, 82]. Суть підходу полягає в поданні різним студентам шаблонного завдання, яке відрізнятиметься певними параметрами, що генеруються автоматично. Таким чином, кожен студент отримує індивідуальне завдання, а система за певною формулою чи алгоритмом, підставляючи параметри, отримує правильну відповідь для подальшої перевірки відповіді, введеної студентом. На рисунку 1.8. наведено приклад

побудови тестового завдання на основі методу параметризованих тестів. Недоліком підходу є його вузька предметна спрямованість. Так, параметризовані тести добре підходять для організації контролю практичних навичок у точних науках і програмуванні [112], проте не можуть використовуватись для перевірки теоретичних знань, а також контролю в гуманітарних науках [81].

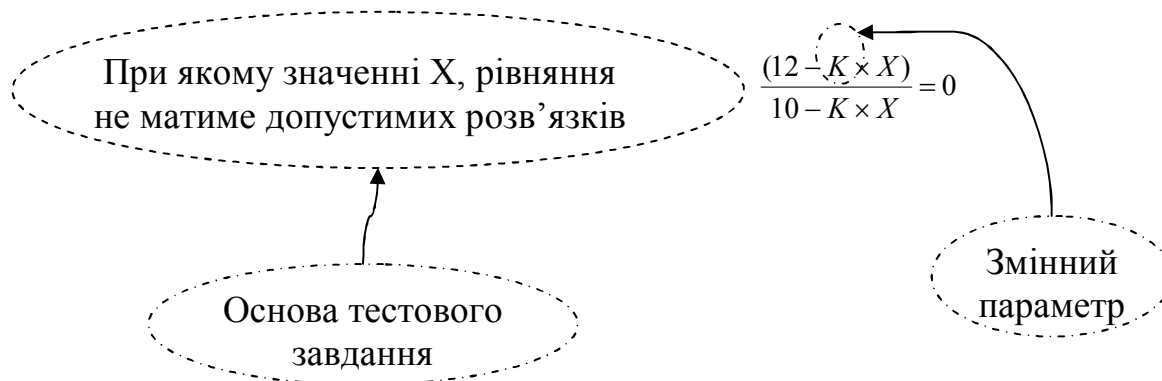


Рисунок 1.8 – Тестове завдання, згенероване на основі методу параметризованих тестів

Певного поширення в дослідженнях автоматизації контролю знань отримав підхід застосування семантичних мереж для автоматизації побудови тестів [25, 118]. Приклад застосування даного методу представлено на рисунку 1.9.

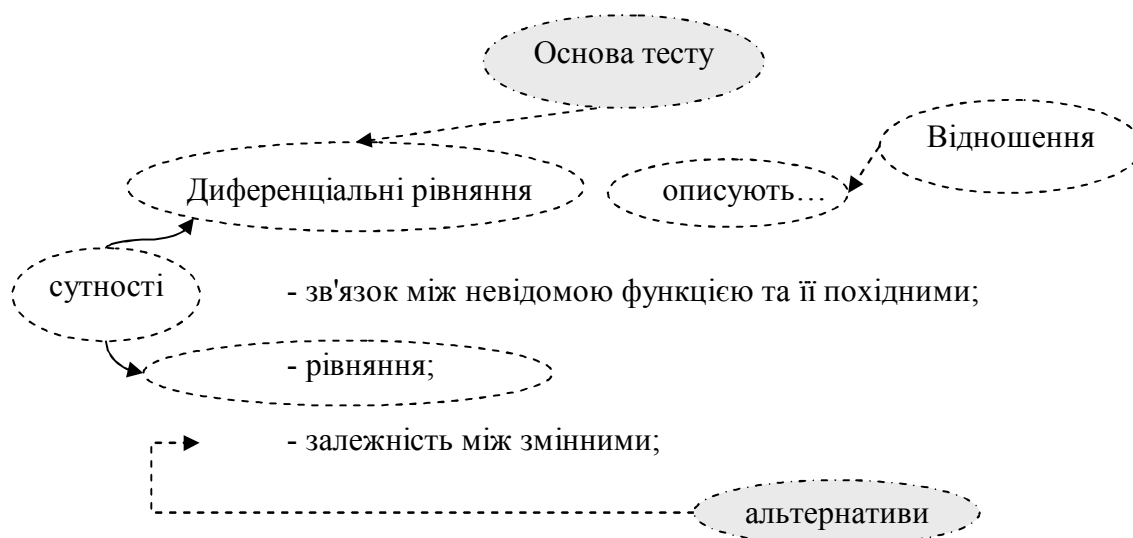


Рисунок 1.9 – Тестове завдання, згенероване на основі методу семантичних мереж

Недолік дадого підходу полягає у великих часових витратах при формалізації елементів семантичної мережі, які б коректно відображали досліджувану предметну область. Ще одним недоліком підходу є лінгвістична незрозумілість і, часом, недоцільність завдань, що генеруються. Так, на основі методу семантичних мереж часто ставляться запитання про такі особливості об'єктів предметної області, що не мають у даному навчальному контексті педагогічної цінності [81].

Одним з сучасних підходів є метод понятійно-тезисної моделі [38, 81, 82]. Центральною структурною ідеєю для формування тестових завдань є зв'язок “навчальний матеріал”—“тези”—“поняття”. Генерація тестів відбувається на основі створених тестових шаблонів (рис. 1.10.).

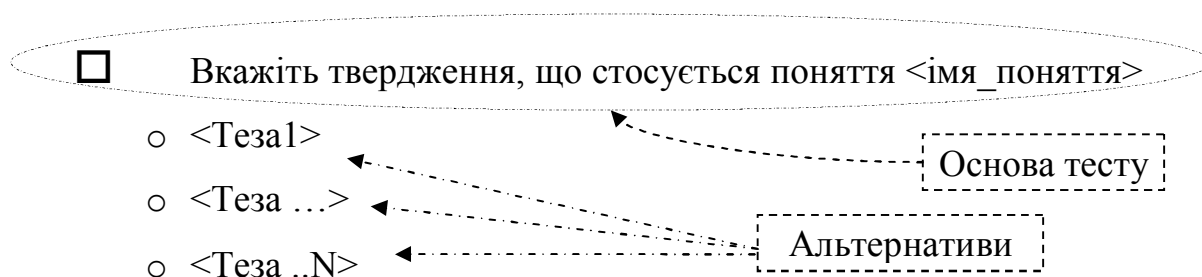


Рисунок 1.10 – Тестове завдання, яке згенероване на основі понятійно–тезисної моделі

Понятійно-тезисна модель передбачає можливість розширення множини класів користувачькими класами тез, що дозволяє розширювати модель і налаштовувати її для різних предметних областей. До переваг даного методу слід віднести кращу, ніж в семантичних мережах лексичну зрозумілість, витрати часу на створення бази понять та тез є меншими, ніж для створення великої бази завдань вручну, а також опису повної семантичної мережі. До недоліків методу можна віднести: надмірним є обмеження структури тесту рамками понять та тезисів, оскільки в практиці формування тестових завдань часто зустрічаються „тезисно-тезисні” конструкції доволі складної структури; неструктурованість представлення тез в даному методі обмежує складність тестового завдання,

оскільки не проводиться аналіз окремих частин самої тези; запропонована авторами структура класів не є обґрунтованою, що породжує невизначеність в їх наповненні і може спричинити формування альтернатив з низькою якістю.

На основі проведеного аналізу в якості перспективних підходів можна виділити фреймові моделі подання знань при їх реалізації за допомогою реляційних відношень. Таке подання знань є основою методів автоматичної генерації тестових завдань для перевірки теоретичних та методологічних знань, а також вмінь та навичок, які описані в підрозділах 2.2, 2.3, 3.1. Кожен із відомих підходів до автоматичної генерації тестових завдань має свої переваги, які доцільно використати при розробці нових технологій. Зокрема підхід параметризованих тестів узагальнено при формуванні завдань для перевірки вмінь та навичок, ідею підходу семантичних мереж розвинуто при побудові методологічних тестів, а ідею понятійно-тезисної моделі розвинуто при побудові завдань для перевірки теоретичних знань.

Методи автоматичної генерації тестових завдань особливо ефективні в реалізації підходів дистанційного навчання (англ. Distant learning), що дозволяє навчатись на відстані без безпосереднього, особистого контакту між викладачем і студентом [20]. Важливою характеристикою такого навчання є оптимізація його структури з метою економії часових витрат для досягнення бажаного рівня освоєння предмету. В наступному підрозділі проведено аналіз сучасних систем дистанційного навчання та підходів до управління часовими витратами процесу навчання.

1.3 Методи та засоби організації дистанційного навчання із управлінням часовими ресурсами

При використанні систем дистанційного навчання застосовуються поняття платформи дистанційного навчання та засобів створення електронних курсів. Платформа дистанційного навчання, представляє собою програмне

забезпечення, що дозволяє не тільки розміщувати матеріали, спілкуватися та контролювати знання студентів, але й здійснювати управління курсом та процесами навчання. Засоби розробки електронних курсів є спеціалізованими програмними середовищами, що дозволяють інтегрувати та обробляти різні формати медіа-файлів, підтримують міжнародні стандарти електронного навчання, мають засоби підтримки різних платформ дистанційного навчання, надають можливість використовувати шаблони та отримувати якісний навчальний курс [14, 71].

На даний час кількість розроблених платформ дистанційного навчання наближається до двохсот. Згідно звіту про стан розвитку дистанційного навчання в Україні [75], розробляють власну платформу дистанційного навчання невеликий відсоток вишів, більшість використовує платформи дистанційного навчання визнаних виробників. У додатку Б наведено відомі системи дистанційного навчання [83].

До найбільш використовуваних належать Moodle, eLearning Server, Blackboard, WebCT Campus Edition, WebCT Vista, IBM Lotus LearningSpace, WebTutor, Sakai, Доцент, Прометей, Орокс (рис. 1.11). Спільним для цих платформ є те, що вони відповідають основним та загальноприйнятим у світі вимогам та стандартам організації дистанційного навчання. Тобто вони доступні, персоніфіковані, модульні, прості у використанні, інтерактивні, адаптовані, відповідають вимогам комп'ютерної безпеки тощо. Серед представлених систем є системи з відкритим кодом (Moodle, Sakai тощо) та комерційні (Blackboard, WebCT, eLearning Server, WebTutor, Прометей, ДОЦЕНТ). Системи з відкритим кодом мають ті ж самі вбудовані функції, що й комерційні, але вони дозволяють пристосовувати інструменти до певних вимог користувача.

Moodle, тобто модульне динамічне об'єктно-орієнтоване середовище для навчання, - це вільна система управління навчанням, орієнтована на організацію взаємодії між викладачем та студентами [48]. Також дана система використовується для організації традиційних дистанційних курсів.

Середовище Moodle розроблена на PHP, з використанням SQL-бази, має модульну архітектуру, що дозволяє легко розгалужувати його можливості. Також Moodle має продуману систему безпеки, а функції адміністратора дозволяють налаштовувати зовнішній вигляд системи та її функціональність, тобто включати або виключати вбудовані модулі. Привертає до себе увагу гнучкість системи, тобто викладач, наприклад, може власноруч контролювати доступ до своїх курсів, використовувати часові обмеження, створювати власні системи оцінки знань, контролювати запізнення студентів при виконанні завдань, дозволяти або забороняти перездачу тощо. Система підтримує показ будь-якого електронного формату документів, що є корисним при створенні курсів.

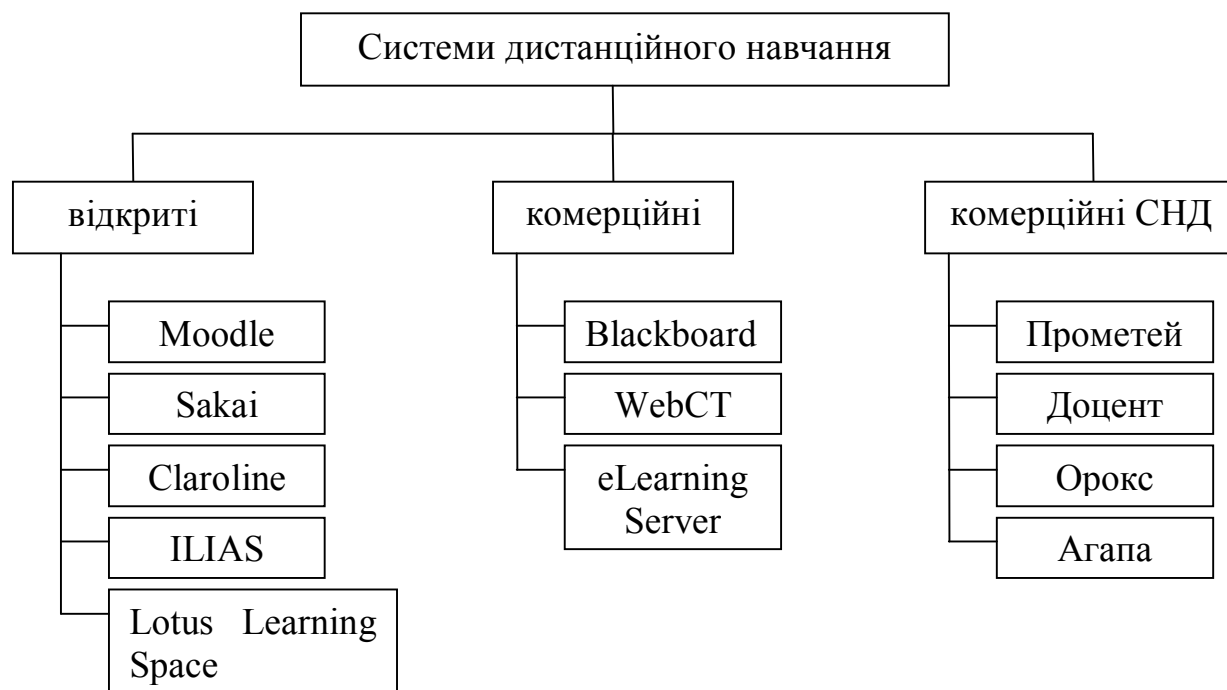


Рисунок 1.11 – Найбільш використовувані системи дистанційного навчання

Для організації взаємодії між учасниками навчального процесу існують чати та форуми з можливістю використання графічної інформації, а також інструменти проведення онлайн-класів та надсилання відгуків студентам. Контроль знань здійснюється в системі за допомогою окремого модуля, який представляє багато видів тестів, можливість перетестування з дозволу викладача, можливість захисту від списування шляхом рендомізації питань та

встановлення бази даних питань, для використання у тестах. До переваг даної системи можна віднести і підтримку багатьох мов, в тому числі й української, а також можливість зацікавлення студентів використанням кросвордів та взаємної конструктивної критики. Але для даної системи не вистачає вбудованих інструментів розробки контенту курсів та їх формалізації.

Декілька найкрупніших вищих навчальних закладів світу розробляють та використовують платформу Sakai, як зазначено у [49, 111], ця платформа призначена для створення систем підтримки дистанційного навчання, для організації групової роботи та створення портфоліо окремих користувачів. Система дозволяє впроваджувати особисто-орієнтований підхід у навчанні, включає функції налаштування курсу для потреб конкретного користувача тощо.

Платформа для електронного навчання Claroline [94, 95] дозволяє створювати ефективні онлайн-курси та управляти процесом навчання та сумісними діями через Інтернет. Ця платформа з відкритим кодом та підтримкою 35 мов, серед яких є російська. Простір кожного курсу вміщує інструменти, що дозволяють розробляти шляхи навчання, об'єднувати студентів у групи, готувати онлайн-вправи, керувати термінами виконання.

Система Lotus LearningSpace, повний опис якої представлений у [103], є розробкою IBM у сфері підтримки дистанційного навчання. Оболонка керування дистанційними курсами з інтеграцією асинхронних і синхронних режимів навчання при загальній підтримці реляційного середовища керування базами даних (IBM DB2 чи MS SQL чи ORACLE). Система має досить потужну систему тестування як в режимі реального часу, так і в режимі «офф-лайн». Недоліками системи Lotus LearningSpace є висока трудомісткість створення нових спеціалізованих засобів.

Досить поширена система дистанційного навчання є Blackboard. Дана платформа є комерційною розробкою і представлена на офіційному сайті [91] розробниками тільки в якості реклами.

Ще на сьогодні досить популярними платформами є комерційні WebCT Campus Edition (CE) та WebCT Vista, але так само як і платформа BlackBoard мають досить високу вартість та обмежений доступ до документації. За джерелами [26] обидві системи розроблені за підтримкою системи управління реляційними базами даних, але доступ до опису архітектури цих систем відсутній. WebCT CE орієнтована на підтримку дистанційного навчання в окремому університеті або іншій освітній організації.

Система eLearning Server, опис якої представлений у [96, 97], є програмним забезпеченням, що дозволяє створювати навчальні центри в мережі. Ця система може виконувати досить широкі завдання з організації дистанційного навчання, а саме реєстрація та створення навчальних курсів, ведення бази даних викладачів та студентів, публікація навчальних матеріалів, тестів та завдань в різній формі, аналіз результатів навчання, самоконтроль тих, що навчаються, формування та ведення синхронізованого розкладу, ведення статистики навчання, виконання адміністративних функцій та інше.

Одними з досить популярних на території СНД є системи дистанційного навчання «Прометей», «ДОЦЕНТ» ОРОКС, «Агапа». Ці системи об'єднують модульну архітектуру, що дозволяє їх розширювати, модернізувати та масштабувати, а також автори надають можливість імпорту курсів, створених у міжнародних стандартах [54, 69].

Усі розглянуті системи є переважно системами управління навчальним процесом і головна їх мета - організація доступу до навчальних матеріалів [14]. Аналізуючи ці системи можна підсумувати, що практично відсутні системи, які б інтелектуалізували роботу викладача (автоматична побудова навчальних курсів, автоматична генерація тестових завдань, моделювання процесу навчання, побудова інтелектуальних навчальних середовищ із елементами штучного інтелекту).

Проаналізуємо інформаційні технології управління знаннями в комп'ютерному навчанні. Разом із розквітом експертних систем, бази знань (БЗ) яких будувалися як продукційні системи на основі продукційних правил,

робочої пам'яті та механізму логічного виведення, були започатковані численні дослідження по застосуванню таких систем для комп'ютерного навчання (зокрема системи GUIDON, MYCIN) [93]. Не зважаючи на окремі успіхи, практика використання виявила принциповий недолік даного підходу: орієнтація на побудову розв'язку проблеми без пояснення принципів такої побудови [55, 105].

Інтерактивні середовища вивчення (ICB, англ. Interactive Learning Environments - ILS) та їх представники мікросвіти (Microworlds) виникають разом із розвитком комп'ютерних технологій як альтернативний підхід ІНС. В той час як ІНС мають тенденцію повністю контролювати навчальний процес, ICB сповідують спосіб навчання, головним ініціатором в якому є студент, який вчиться за допомогою методу дослідження. За допомогою спроектованих у ICB об'єктивних властивостей світу, студенти мають змогу вивчати його, досліджуючи ці властивості і відношення [105, 116].

ICB мають ряд недоліків. По-перше, це метод навчання, що реалізується лише через дослідження, що із його скромним дидактичним впливом має ризик не досягти цілей навчання. Також ICB переживають труднощі у визначенні і оцінюванні навчальних результатів. Крім усього іншого такі системи є дуже складними для програмної реалізації, а коли мова іде про Інтернет-освіту, їх застосування виявляється особливо ускладненим через саму архітектуру і обмеження мережі Інтернет.

Абсолютно інший підхід являє собою ряд технологій, пов'язаних з аналізом інформації і інформаційним пошуком, що активно застосовуються в Інтернет. В даному випадку знання безпосередньо не формалізуються, а зберігаються в текстовому вигляді. Натомість основний акцент робиться на механізмах видобуття і пошуку необхідної (релевантної) навчальної інформації. Як зазначалося вище, технології інтелектуального аналізу інформації, інформаційного пошуку та інформаційної фільтрації, будучи дуже популярними в області інформаційних систем, не використовувалися раніше у навчальному контексті. Об'єм навчального вмісту був порівняно невеликим, і

потреба спрямовувати користувача до найбільш підходящого матеріалу з легкістю підтримувалася адаптивним плануванням курсу навчання і адаптивним гіпермедіа. Однак потреба побудови великих інформаційних освітніх систем, що обслуговуватимуть широкий спектр навчальних дисциплін, та Інтернет з його великою кількістю неіндексованих відкритих освітніх ресурсів зробив ці технології дуже привабливими для розробників сучасних систем освіти. MLTutor [107] представляє один перших цікавих прикладів застосування фільтрації інформації на основі вмісту у навчанні. Навчальний приклад сумісницької адаптивної фільтрації можна знайти у WebCOBALT [55, 88].

Спеціально для підтримки досліджень у сфері Семантичної Павутини для Інтернет-освіти створена спільнота SW-El (Semantic Web for E-Learning) – Семантична Павутина для Е-навчання (СПЕН) (<http://www.win.tue.nl/SW-EL/>). Вона займається питаннями створення так званих онтологій для освіти та визначенням концепцій для освітніх Інтернет-систем у рамках досліджень СП.

Серед найбільш ефективних напрямків розвитку даних технологій можна виділити інтелектуальний лінгвістичний аналіз, який дозволить ширше використовувати можливості природної мови для побудови пошукових запитів, а також застосування адаптивності у методах пошуку і фільтрації, що включає побудову моделей користувачів [92]. Майбутні сценарії пошуку і фільтрації повинні включати інструменти, здатні «розуміти», чого бажає користувач, щоб більш точно оцінювати релевантність документу його потребам, і щоб «слухати» і «навчатись» від користувача, коли він надає корисний зворотний зв'язок під час взаємодії [45, 55].

Серед переваг підходу аналізу тексту і інформаційного пошуку відносна простота процесу додавання в систему нових знань – фактичне додавання текстових документів. В даному випадку персонал освітньої системи загалом не бере безпосередньої участі у детальній формалізації навчальної інформації на відміну від експертних систем і семантичних мереж. Такі системи не володіють типовим недоліком навчальних систем ШІ, який можна охарактеризувати як

вимогу «всеосвіченості» системи з предметної області. Такі методи можна застосовувати для пошуку специфічної навчальної інформації, що відповідає навчальним цілям учня, а також для адаптивного подання навчального матеріалу і фільтрації елементів, які не відповідають навчальним потребам учня або групи учнів [55]. Недоліком застосування таких підходів є їх природна обмеженість, пов'язана із слабкою формалізацією і структуризацією навчальної інформації.

В процесі аналізу відомих рішень в сфері управління процесом навчання необхідно виділити такий напрям як оптимальне управління часом навчання для максимального швидкого засвоєння знань та здобуття практичних навичок. Дослідження у цьому напрямку є актуальними в процесі динамічного розвитку інформаційного суспільства. Аналізуючи основні підходи до вирішення зазначених проблем необхідно відзначити такі основні методи:

- традиційні методи, суть яких полягає у формальній регламентації часу навчання для невеликих груп студентів з постійним супроводом експертів з предметної галузі. Основними недоліками даних методів є суб'єктивність вибору параметрів навчального процесу та складність їх застосування в автоматизованих навчальних системах;

- логіко-математична модель управління навчанням [10]. В даному підході розглядаються індивідуальні характеристики студентів, різноманітні параметри навчальних матеріалів, які оцінюються за допомогою експертних оцінок, але фактор впливу часу навчання розглядається в рамках традиційного підходу;

- метод формування індивідуальної траєкторії навчання студентів [51]. До переваг даного підходу необхідно віднести широкий спектр параметрів навчального процесу, основними з яких є навчальна завантаженість, успішність, час, відсутність відволікаючих чинників, знання та практичні навички. До недоліків даного методу можна віднести суб'єктивні побудови індивідуальних траєкторій навчання, застосування в комп'ютеризованих навчальних системах супроводжується складністю ідентифікації параметрів навчання та постійною присутністю висококваліфікованих експертів.

Аналізуючи відомі рішення актуальної задачі управління процесом навчання з точки зору мінімізації часових витрат для досягнення бажаного рівня засвоєння знань та практичних навичок, необхідно відзначити складність вибору правильних параметрів навчання та їх ідентифікації, постійний або частковий супровід експертів з предметної області, який послаблює можливість їх застосування в автоматизованих навчальних системах. Виходячи з цього виникає необхідність створення методу, який вирішуватиме зазначені проблеми, а саме – при невеликій кількості параметрів управління забезпечуватиме досягнення бажаного рівня успішності з мінімальними часовими витратами при вивченні дисциплін. Такий метод реалізовано в рамках проведених досліджень, а його особливості описані в підрозділі 3.3.

Проблема зниження часових витрат викладачів при організації якісного тестового контролю включає два основних аспекти: автоматизація генерації тестових завдань; забезпечення якості самопідготовки студентів та проведення контрольних замірів знань. Тому технологія автоматичної генерації тестових завдань з керованою складністю включає дві основні частини: автоматична генерація тестів для перевірки засвоєння теоретичних знань, методології, практичних вмінь та навичок; формування тестів заданої складності, цільове подання тестових завдань в процесі формуючого тестування з метою мінімізації часових витрат на досягнення бажаної успішності. Існуючі системи автоматизованого навчання лише частково відповідають сформульованим вимогам. Це породжує необхідність проведення спеціального дослідження з метою розробки якісно нової технології. Для оцінки її параметрів та порівняння з відомими аналогами доцільно розробити систему кількісних показників, які дозволяють формалізувати постановку задачі дослідження.

1.4 Постановка задачі дослідження

Для оцінки відмінностей розроблюваної технології автоматичної генерації тестових завдань з керованою складністю від існуючих, побудована система

кількісних показників ефективності, яка володіє диференціюючими властивостями. Оскільки вона включає інтегральний усереднюючий показник, то її складові нормуються [44].

Як зазначається у [4] процес оцінювання знань повинен охоплювати їх вичерпну структуру, яка складається з наступних елементів [4, 99]:

- знання, які дозволяють відтворювати факти, тобто які базуються на запам'ятовуванні;
- розуміння знань, які відтворюються;
- застосування знань в новій ситуації;
- вміння аналізувати та синтезувати ознаки;
- вміння дати оцінку, зробити загальний висновок.

Всі перераховані компоненти є складовими таксономії Блюма [4, 99], яка, як показано в [44], є однією із найповніших, а інші відомі спроби побудови нових конструкцій не виводять за її межі. Однак, на наш погляд, таксономія Блюма повинна бути узагальнена хоча б одним додатковим показником. Цей показник пов'язаний із масовим застосуванням обчислювальної техніки в процесах обробки інформації і може бути сформульований як вміння алгоритмізувати процедуру розв'язання задачі, тобто побудувати і відлагодити програму такого розв'язання.

Оскільки формування ефективних технологій автоматичної генерації тестових завдань повинно включати завдання для перевірки знань усіх рівнів, то оцінку даного фактору можна здійснити на основі показника повноти контролю набутих знань (по узагальненій шкалі Блюма) [44]:

$$KB = \frac{KB_r}{KB_f}, \quad (1.5)$$

де KB - критерій повноти контролю набутих знань; KB_r - число реалізованих елементів узагальненої шкали Блюма; KB_f - загальне число елементів узагальненої шкали Блюма.

Для прикладу розглянемо оцінку KB для різних форм реалізації тестового контролю. Зокрема, використання тестових завдань закритого типу в системі тестування забезпечує наступне значення критерію повноти контролю:

$$KB = \frac{KB_r}{KB_f} = \frac{|\{a, b\}|}{|\{a, b, c, d, e, f, g\}|} \approx 0.3,$$

де $||$ - оператор потужності множини; a - запам'ятовування знань; b - осмислення знань; c - застосування вивчених законів та теорій в конкретних ситуаціях; d - виділення базових структур та причинно-наслідкових зв'язків (аналіз); e - комбінування елементів для отримання нового цілого (синтез); f - обґрунтованість висновків за наявними даними (оцінка матеріалу); g - вміння алгоритмізувати процедуру розв'язання задачі.

Оскільки тестові завдання для перевірки вмінь та навичок передбачають комбінування елементів для отримання нового цілого, то їх використання забезпечує наступне значення критерію повноти

$$KB = \frac{KB_r}{KB_f} = \frac{|\{c, e, g\}|}{|\{a, b, c, d, e, f, g\}|} \approx 0.4.$$

Таким чином використання традиційних тестових завдань закритого та відкритого типів не забезпечує максимальної величини критерію повноти. В той же час В.С. Аванесовим у [4] виділено вид знань методологічного рівня, який характеризується вміннями застосовувати вивчені закони та теорії в конкретних ситуаціях, виділяти причинно-наслідкові зв'язки, та будувати обґрунтовані оцінки структури конкретних ситуацій. Зокрема, на методологічних тестах критерій повноти приймає наступні значення

$$KB = \frac{KB_r}{KB_f} = \frac{|\{d, f\}|}{|\{a, b, c, d, e, f, g\}|} \approx 0.3.$$

Проведений аналіз свідчить, що інформаційні технології, які підтримують побудову завдань закритого типу для перевірки теоретичних знань, завдань для перевірки вмій та навичок та завдань для перевірки методологічного типу знань відповідає максимальне значення критерію повноти. Функції компонентів цієї сукупності наочно представлені на рисунку 1.12 за допомогою методу опису процесів технології ARIS [28].

Одне із основних завдань тестування – звільнення викладача від виконання рутинних робіт [9]. Окрім економії часу на проведення тестового контролю виникає потреба зменшення завантаженості викладача при створенні тестових завдань. Оскільки на сьогоднішній день найчастіше зустрічаються тестові завдання закритого типу, а технологіями побудови таких завдань слугують методи «ручної» генерації, тому оцінку економії часу на генерування тестів доцільно здійснювати відносно продуктивності згаданого підходу. Це положення формалізуємо на основі показника ΔEZ - приведеної економії часу на генерацію тестів.

При генерації закритих тестів їхня варіативність зростає за рахунок побудови нових основ, оскільки варіативність дистракторів забезпечує лише відділення змісту тестового завдання від його форми. Основи можна генерувати за допомогою варіативної декомпозиції формалізованих тверджень. В традиційному підході до формування тестового завдання його унікальність забезпечується унікальністю базового твердження, яке розбивається на два фрагменти – основу тесту та множину вірних альтернатив. Якщо базове твердження формалізується як структура пов'язаних фрагментів, число яких більше ніж два, то це дозволяє генерувати більшу кількість основ тестового завдання в порівнянні з традиційним підходом. У випадку коли формалізація забезпечує автоматичну генерацію нових видів тестових завдань, їх варіативність зростає без додаткових часових витрат розробника.

Технологію автоматичної генерації тестових завдань з керованою складністю можна використовувати як для автоматичного так і ручного генерування, хоча останній підхід містить значно більші часові витрати.

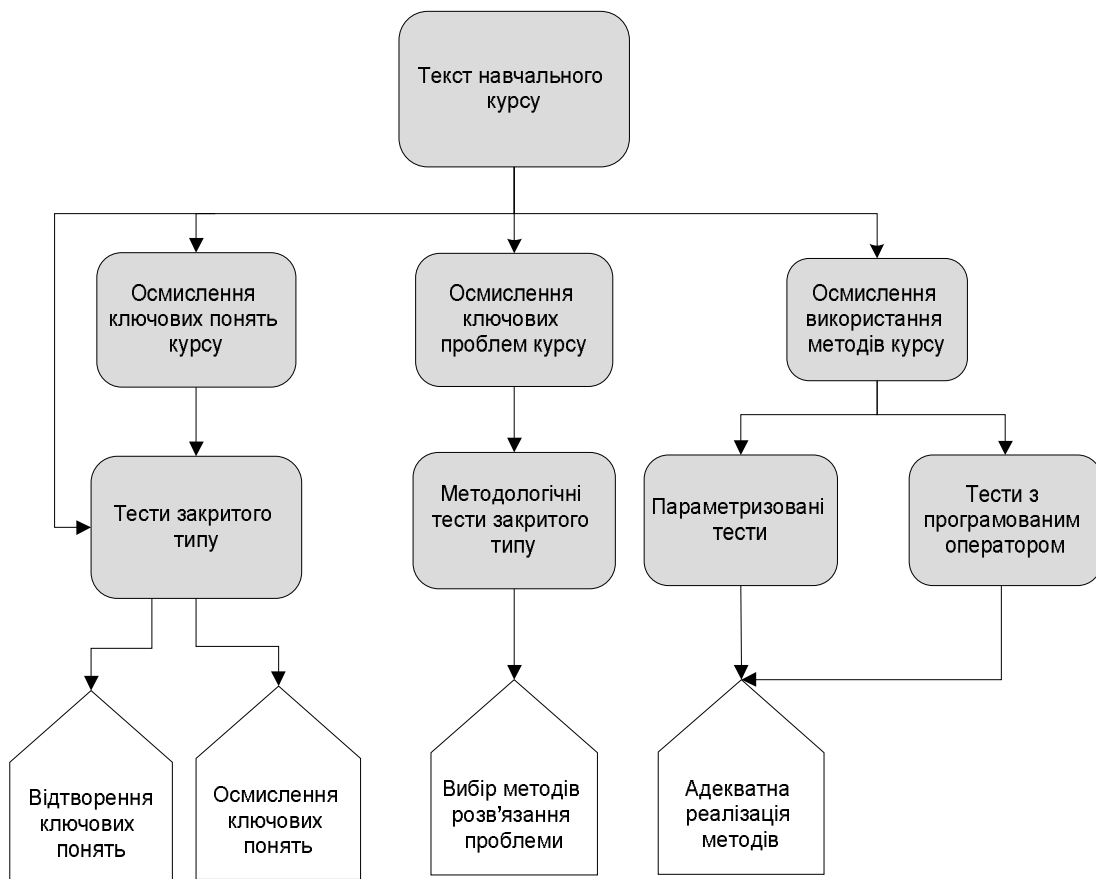


Рисунок 1.12 – Функції компонентів системи повного контролю знань

Наведені положення формалізуємо за допомогою наступної формули:

$$\Delta EZ = \begin{cases} 0 - \text{при "ручній генерації"} \\ \frac{(M(N_{FR}) - 1) \cdot nv}{(\text{Max}(N_{FR}) - 1) \cdot \text{Max}(nv)} - \text{при "автоматичній генерації"} \end{cases} \quad (1.6)$$

де $M(N_{FR})$ - математичне сподівання кількості фрагментів формалізованих базових тверджень; nv - кількість реалізованих видів тестових завдань із наведених на рисунку 1.5.

Поряд з економією часових витрат на генерацію тестів важливе місце займає забезпечення їхньої якості. З теорії тестів IRT відомо, що якість тестів залежить від однорідності дистракторів [68].

Таку оцінку проведемо на основі показника однорідності альтернатив, яка враховує їх змістовні та формальні характеристики. В якості формальних характеристик використовуємо формальні класи та синтаксичну узгодженість, а в якості змістовних характеристик – семантичні класи. Найвища однорідність забезпечується при повному семантичному контролі, який може здійснювати людина або системи штучного інтелекту. Однорідність на основі формальних критеріїв може досягати значно нижчого рівня (не перевищує 50% максимального). Наведені положення формалізуємо у вигляді наступного співвідношення:

$$KO = \begin{cases} \frac{K_r}{K_f} \cdot 0.5 & \text{при } i_{ps} = 0, \\ 1 & \text{при } i_{ps} = 1 \end{cases}, \quad (1.7)$$

де KO - коефіцієнт однорідності; K_r - кількість реалізованих рівнів однорідності; K_f - загальна кількість рівнів однорідності, i_{ps} - індекс реалізації повного семантичного контролю однорідності.

У відповідності з класичною теорією тестів, основним показником якості тестових завдань є складність [68]. Побудову тестів із заданою складністю можна здійснити на основі показника об'єктивності індивідуального оцінювання Eg - для контрольних тестів (еквівалентність наборів тестових завдань по складності). Достовірність в оцінюванні складності тестових завдань здійснюємо із врахуванням впливу відносної похибки такого оцінювання:

$$Eg = \max(1 - er, 0), \quad (1.8)$$

де er - відносна похибка оцінки складності набору тестових завдань.

Диференціювання складності завдань дозволяє не тільки керувати нею, але й управляти тривалістю підготовки по тестах для досягнення реально

досяжного максимального рівня успішності по сукупності предметів при обмеженому ресурсі часу відведеного на підготовку [41]. Ефективність такого управління пропонується оцінювати на основі середньої відносної економії часу EG на досягнення бажаного рівня успішності відносно послідовного проходження тестів без диференціювання їх складності:

$$EG = \frac{T_n - T_a}{T_n}, \quad (1.9)$$

де T_n - середні витрати часу на вивчення матеріалу без виділення рівня складності; T_a - середні витрати часу на вивчення матеріалу з адаптацією до складності завдань.

На основі запропонованих показників отримуємо інтегральний показник ефективності технологій IEf автоматизованого навчання, який можна представити за допомогою наступної формули

$$IEf = \frac{(KB + \Delta EZ + KO + Eq + EG)}{n}, \quad (1.10)$$

де n - загальна кількість показників.

Підсумовуючи проведений аналіз, відзначимо, що контроль знань студентів є складовою частиною процесу навчання, який на сьогоднішній день здійснюється як традиційними методами, так і з використанням комп'ютерних технологій у формі електронного тестування.

Організація процесу тестування включає інтелектуальні операції із різною насиченістю творчої компоненти: від авторського бачення структури завдань із врахуванням складності до рутинної перевірки правильності їх розв'язання. Виділення та формалізація нетворчих компонентів процесу навчання із наступною їх автоматизацією дозволяє вивільнити час викладача на посилення творчої компоненти, що покращує розвиток творчих задатків студентів.

Жодна із аналізованих технологій автоматизованого тестування не забезпечує генерацію повного спектру форм контролю знань, економію витрат часу викладача, управління складністю тестів при контрольних замірах, а також управління складністю завдань при організації навчального тестування. Тому основними задачами, що виникають при розробці інформаційної технології автоматичної генерації тестових завдань з керованою складністю є наступні:

1. Розробка методів автоматичної генерації тестових завдань для перевірки теоретичних знань із забезпеченням варіативності на основі формалізації навчальних матеріалів в інформаційних структурах.

2. Розробка методів автоматичної генерації тестових завдань для перевірки вмінь та навичок з використанням програмних засобів для динамічної побудови правильних розв'язків.

3. Розробка методу апіорної оцінки складності тестових завдань із адаптацією до особливостей їх реалізації та використання.

4. Розробка методу адаптивного управління структурою процесу навчання з метою оптимізації часових витрат студентів для досягнення баженого рівня успішності.

5. Реалізація інформаційної технології автоматичної генерації тестових завдань з керованою складністю, яка б забезпечила суттєве підвищення ефективності систем автоматизованого навчання.

Висновки до розділу 1

1. Проаналізовано структуру процесу навчання та ролі тестів в реалізації функції контролю знань. Розглянуто принципи побудови тестових завдань, проаналізовано їх основні види та типи. Описано основні моделі тестування та вказано переваги адаптивних підходів. Проведено порівняльний аналіз методів оцінки складності тестових завдань. Виділено їх основні переваги та недоліки.

2. Розглянуто моделі представлення знань в інформаційних системах, що використовуються для формалізації змісту дисциплін в системах автоматизованого навчання. Виділено переваги фреймової моделі представлення знань при її реалізації за допомогою реляційних моделей даних. Досліджено відомі підходи до автоматичної генерації тестових завдань. Встановлено основні їх недоліки і намічено напрямки їх подолання шляхом врахування семантичних характеристик компонентів тестових завдань та їх синтаксичної узгодженості.

3. Проаналізовані особливості відомих систем дистанційного навчання. Відзначено необхідність автоматизації процесу наповнення банку тестових завдань, що дозволить знизити часові витрати. Проаналізовано відомі інформаційні технології подання знань в комп'ютерному навчанні. Виділено методи інтелектуально–лінгвістичного аналізу, як перспективні методи в розробці інформаційних навчальних систем.

4. Проаналізовано відомі підходи до управління процесом навчання з метою оптимізації часових витрат. Оскільки вони ґрунтуються на регламентаційному підході та експертних оцінках, які важко ідентифікуються, відзначається необхідність створення методів, які при невеликій кількості параметрів управління забезпечуватимуть досягнення бажаного рівня успішності з мінімальними часовими витратами при вивченні конкретних дисциплін.

5. Сформовано систему кількісних показників для оцінки якості технологій автоматизованого навчання, яка дозволяє сформулювати напрямки підвищення їх ефективності. На основі введеної системи показників формалізовано постановку задачі, яка дозволяє управляти процесом розробки методів та засобів технології автоматичної генерації тестів з керованою складністю.

6. На основі проведеного аналізу методів та засобів тестового контролю в комп'ютеризованих навчальних системах виділено основні завдання, що виникають при розробці інформаційної технології автоматичної генерації

тестових завдань з керованою складністю. Здійснено постановку задачі дослідження, яка включає необхідність розробки методів автоматичної генерації тестових завдань для перевірки теоретичних і методологічних знань, а також вмінь та навичок, методів оцінки складності тестових завдань та адаптивного управління структурою процесу навчання.

РОЗДІЛ 2

ТЕХНОЛОГІЯ ФОРМАЛІЗАЦІЇ НАВЧАЛЬНИХ МАТЕРІАЛІВ ЗАСОБАМИ РЕЛЯЦІЙНОЇ АЛГЕБРИ

2.1 Формалізація навчального матеріалу за допомогою деревовидних структур

Сучасні завдання інноваційного розвитку цивілізації можна ефективно вирішувати на основі інтеграції, міждисциплінарного синтезу знань, на основі системного підходу до аналізу взаємозв'язку явищ [1]. Освіта, яка закладає основи професійної підготовки, повинна давати не просту суму знань, але й розуміння їх сутності, творче оволодіння базовими вміннями. Перевірка якості здобутих знань є необхідною частиною процесу навчання, тому вона повинна базуватися на системному підході, містити творчу складову. Тестування є важливою компонентою перевірки якості здобутих знань, її об'єктивність досягається шляхом стандартизації процедури проведення та перевірки показників якості завдань. Воно також є ефективним з економічної точки зору, оскільки основні витрати при тестуванні припадають на розробку якісного інструментарію, тобто мають разовий характер. При всіх позитивних характеристиках тестування, воно залишається лише однією із багатьох форм контролю знань. На даний час, за допомогою тестів не можна перевіряти рівень доведення теорем, рівень виконання творчих завдань, тощо [23].

У той час, як багато досліджень у галузі комп'ютерного контролю знань зосереджені на питаннях валідності і надійності тестів [81], питання формування самого банку завдань у більшості випадків залишається виключно прерогативою викладача, який працює без використання інтелектуальних засобів автоматизації даного процесу. Насичення тестових завдань елементами творчої складової дозволить усунути адаптацію студента до структури тестових завдань за рахунок зростання їх варіативності. Це призводить до подальшого

зростання завантаженості викладачів. Цьому можна запобігти шляхом раціонального використання інформації про предметну область формалізованої викладачем. Тоді за рахунок автоматичної генерації тестів економляться зусилля людини на забезпечення варіативності тестів.

Такі результати досягаються за допомогою аналізу формалізованого представлення навчальної інформації, що представлена в текстовій формі. Розуміння природної мови вважають AI-повною задачею, оскільки розпізнавання природної мови потребує величезних знань системи про навколишній світ і можливостей взаємодії з ним [7]. Тому раціонально зосередитися на формалізації тексту без створення його повної смислової моделі. Останнє завдання може бути розв'язане в межах спеціалізованих інформаційних технологій, в яких текст розглядається як система пов'язаних елементарних фрагментів.

Тестові завдання для перевірки теоретичних знань представляються простою ієрархічною структурою, кореневим елементом якої є основа тестового завдання, а його альтернативи є дочірніми елементами. На основі групи однорідних текстових тверджень, формалізованої за допомогою багаторівневих дерев, можна легко будувати тестові завдання, відносячи до основи тестового завдання частину гілки одного із дерев, а до альтернатив – як пов'язані з основою гілки її піддерева, так і подібні гілки інших дерев. Варіативність вибору основи та альтернатив забезпечує багатоваріантність тестових завдань.

Таким чином множини ієрархічних структур можна поповнювати за рахунок комбінування їх частин в нові об'єкти, що при виконанні певних обмежень зберігають семантичну цінність. Цей підхід буде використано при генерації тестів для перевірки теоретичних та методологічних знань. Інший підхід економії зусиль на поповнення множини ієрархічних структур полягає в формалізації лише окремих їх нових елементів, що використано для побудови задач із варіативною структурою для перевірки вмінь та навичок.

Розглянемо принципи використання деревовидних структур для формалізації навчальних матеріалів. Кожне текстове твердження розбиваємо на компоненти, які описують процеси, поняття або їх характеристики. Компоненти поєднуються в речення сполучниками або сполучними словами, які можна представити наступним чином:

$$Df = \langle IdE, CS, CNJ, TCNJ, Pr_IdE, IdSc, IdFc \rangle, \quad (2.1)$$

де Df - текстове твердження, яке формалізує певну сутність конкретного семантичного класу; IdE - ідентифікатор компоненти твердження; CS - зміст компоненти; CNJ - представлення зв'язку між компонентами тверджень; $TCNJ$ - тип зв'язку між компонентами; Pr_IdE - посилання на батьківський елемент дерева (ідентифікатор відповідної компоненти); $IdSc$ - ідентифікатор семантичного класу; $IdFc$ - ідентифікатор формального класу.

Для забезпечення формування дистракторів, які не будуть завідомо невірними необхідно включити в деревовидні структури атрибути, що ідентифікують їх однорідність із вірними альтернативами. Хоча основа тестового завдання представляється в текстовій формі, вірні альтернативи можуть виступати у формах чисел, схем, алгоритмів, рисунків, програмного коду. Якщо форма дистрактора буде відмінною, то його невідповідність буде очевидною. Тому найпростішою ознакою однорідності виступає форма представлення альтернатив.

Відомо, що тест складається з множини тестових завдань. Структура тестового завдання включає основу (одне або декілька додаткових висловлювань, які пов'язані за змістом з правильною альтернативою) та множини альтернатив (вірні альтернативи, які можуть бути відсутніми і множини дистракторів) [76]. Тому множини тестів можна представити структурою:

$$Ts = \langle IdTs, IdTTt, IdSc, IdFc, IdCh, CTt, CSa, IdAu \rangle, \quad (2.2)$$

де Ts – тест, який складається з тестових завдань; $IdTs$ – ідентифікатор тесту; $IdTTt$ – ідентифікатор виду тестового завдання (множинний вибір, альтернативний, встановлення відповідності), $IdSc$ – ідентифікатор семантичного класу; $IdFc$ – ідентифікатор формального класу; CTt – кількість тестових завдань вибраної структури; CSa – кількість альтернатив в тестовому завданні; $IdCh$ – ідентифікатор розділу; $IdAu$ – ідентифікатор автора навчальних матеріалів.

Вид тестового завдання TTt описується своєю назвою $NmTTt$ та доповнюється інструкцією $NmIns$ для формування його умови

$$TTt = \langle IdTTt, NmTTt, NmIns \rangle, \quad (2.3)$$

Для формалізації процесу генерації тестових завдань виділяємо певний кортеж Ts_C , що відповідає конкретному тесту

$$Ts_C = \left\langle IdTs_C, TTt_C, IdSc_C, IdFc_C, IdCh_C, CTt_C, CSa_C, IdAu_C \right\rangle. \quad (2.4)$$

Тестові завдання Tt представимо наступним чином

$$Tt = \langle BTt, Sa \rangle, \quad (2.5)$$

де BTt – основа тестового завдання; Sa – множина альтернатив тестового завдання.

Зростання варіативності тестових наборів забезпечується за рахунок розширення множин основ тестів та їх дистракторів. Це досягається за рахунок варіативної декомпозиції та повторного використання пов'язаних елементів дерева інформаційного представлення. Динамічна генерація тестових завдань характеризується відсутністю ключів правильних відповідей, при цьому

деревовидні структури використовуються для ідентифікації вірної альтернативи.

Формальні класи ідентифікуються лише їхніми назвами за допомогою двійки:

$$Fc = \langle IdFc, NmFc \rangle, \quad (2.6)$$

де $NmFc$ – назва формального класу.

Ця ознака слугує для збереження в базі інформації про лексичний характер того чи іншого фрагменту навчального контенту [35, 38, 57]. Виходячи з даних особливостей можна виділити наступні основні формальні класи, які зображено на рисунку 2.1: клас текстового представлення, який базується на відображенні знань у формі звичайних текстових повідомлень; клас числових представлень, що подає основні характеристики вибраного об'єкта у цифровій формі; клас формульних представлень, що формалізують процеси та зв'язки між їх параметрами; алгоритмічний клас, що включає набір конструкцій, які визначають порядок дій для досягнення результату; клас інструкцій, які представляються у вигляді фрагменту програмного коду; клас ілюстративних представлень, коли знання подаються у формі рисунків.

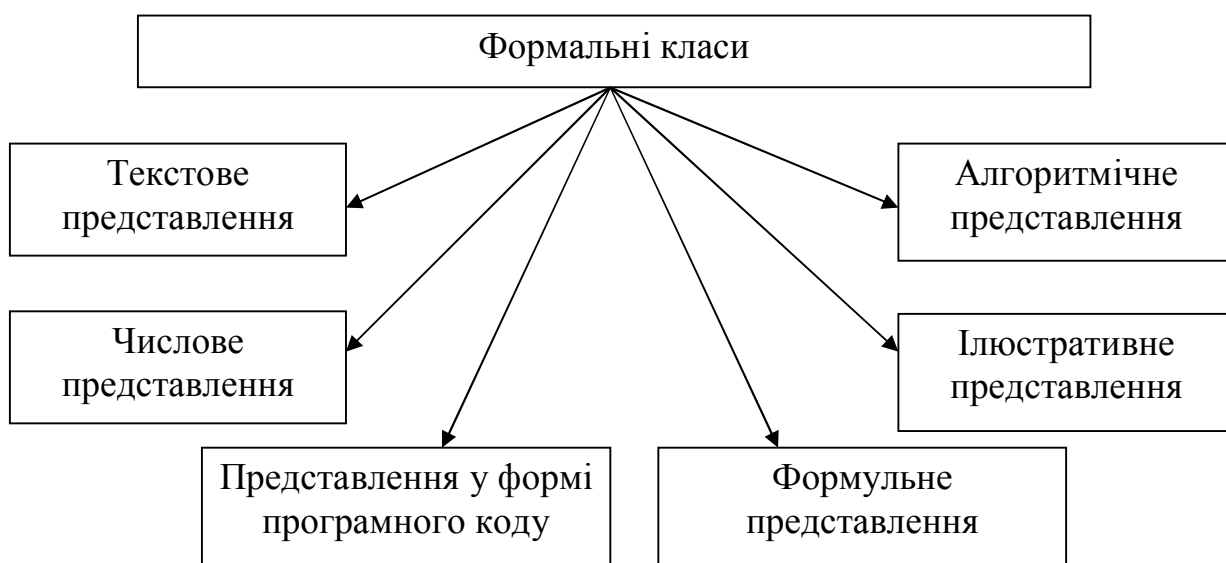


Рисунок 2.1 – Базові формальні класи

Необхідно відзначити, що перелік формальних класів може розширюватися в залежності від вимог, які ставляться перед описовими представленнями знань, перш за все за рахунок комбінаційного поєднання формальних класів. Атрибути формальних класів повинні враховуватися при побудові усіх форм тестових завдань, що дозволить уникати побудови незрозумілих, та завідомо невірних дистракторів при їх генеруванні.

Однак формальна однорідність повинна підсилюватися і змістовною однорідністю альтернатив. Для її забезпечення будуємо семантичні класи, які забезпечують змістовну однорідність тверджень на основі когнітивної функції мови. Сутністю останньої є розв'язання задач [61].

Постановка задачі передбачає формулювання її умови та мети дослідження. Формулювання задачі повинно бути компактним і тому використовує спеціальну термінологію, яка будується на основі певних означень. Розв'язання задач здійснюється за допомогою спеціальних інструментів – методів, де під методом розуміється систематизована сукупність кроків або дій, які необхідно здійснити для досягнення певної мети [45].

Складовими елементами опису методу можна вважати характеристики задачі, яку він вирішує, форми представлення вхідної інформації, перелік допоміжних методів, основну ідею методу, побудову додаткових структур, сукупність кроків методу. Тут під кроком розуміється сукупність відомих дій або методів. Одну і ту ж задачу, як правило, можна розв'язувати різними методами. Виходячи із структури методу, його схему можна представити структурою, яка описує послідовність обов'язкових та необов'язкових операцій в ході реалізації мети

$$Sm = \left\langle \begin{array}{l} IdSm, Cp, CS, Pr_IdSm, \\ SN, IdSc, IdFc, Lg_Int \end{array} \right\rangle, \quad (2.7)$$

де Sm – схема методу; $IdSm$ – ідентифікатор компоненти схеми методу; Cp – ідентифікатор обов'язковості елемента; CS – компонента схеми; Pr_IdSm – посилання на батьківський елемент; SN – назва методу; $IdSc$ – ідентифікатор

семантичного класу; $IdFc$ – ідентифікатор формального класу; Lg_Int – лінгвістична інтерпретація операцій [33,35,43].

Обґрунтований вибір методу можна здійснити на основі опису його ефективності, тобто доцільності використання методу в певних умовах. Формальний опис методу допускає його неоднозначне трактування, а його повний опис породжує значну надлишковість представлення. Тому для забезпечення вірного використання методу використовують приклади його конкретної реалізації [38, 106].

Виходячи з наведеного аналізу для повної характеристики пізнання деякої предметної області необхідно передбачити наступні абстрактні семантичні класи: означення, задачі, методи, ефективність методів, приклади реалізації методів [106]. Структура класів повинна забезпечити їх ідентифікацію в процесі аналізу довільного текстового масиву (рис. 2.2), а їх представлення описується наступним чином

$$Sc = \langle IdSc, NmSc \rangle, \quad (2.8)$$

де $NmSc$ – назва семантичного класу.

Для кожної предметної області формуємо відповідні екземпляри семантичних класів, які ідентифікуються на конкретних текстах. В процесі аналізу нових текстів структура предметної області може модифікуватися.

Набір найзагальніших проблемних когнітивних об'єктів буде деталізувати структуру предметної області, яка є базою для формування тестових завдань. Загальні компоненти структури предметної області (розділи, підрозділи) використовуються для класифікації тестових завдань. Таким чином, запропоновані семантичні класи дозволяють структурувати розділ (підрозділ) тексту певної предметної області на сукупності семантично близьких тверджень.

Однак семантична близькість альтернатив не повинна нівелюватися їх синтаксичною неузгодженістю між компонентами тестового завдання.

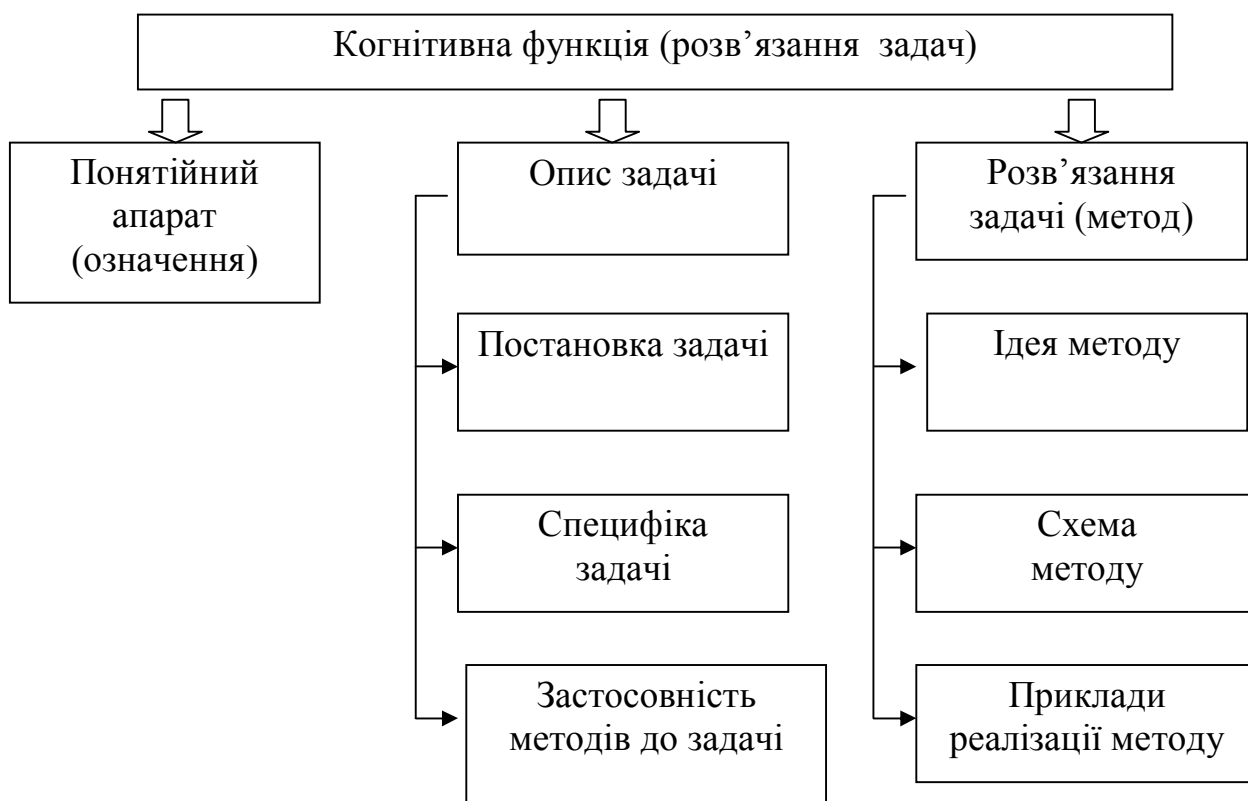


Рисунок 2.2 – Базові семантичні класи [38]

Таке синтаксичне узгодження забезпечується однорідністю зв'язків компонент за типами та числами. Забезпечення формальної, семантичної та синтаксичної однорідності дозволяє за допомогою маніпуляцій будувати дистрактори, які не є завідомо невірними, що забезпечує варіативність тестових завдань (рис. 2.3).

Важливою компонентою в перевірці засвоєння знань є оцінка набутих вмінь та навичок. Схема цього процесу наведена на рисунку 2.4. Умовно професійну діяльність поділяємо на аналітичну та оперативну форми. Засобами прискорення перевірки ефективності набутих навичок у цих видах діяльності виступають тренажери або програмні інтерпретатори, які імітують операції або виконують їх формалізоване представлення. Залежно від предметної області інтерпретаторами можуть виступати реалізовані програмні середовища або спеціально створені засоби.

Згідно з концепції, наведеної в [21] задачі для перевірки вмінь та навичок складаються з умови, вхідних даних, та змінних, значення яких необхідно знайти. При використанні програмних інтерпретаторів варіативність завдань

забезпечується лише модифікацією умови. Економія зусиль при такій модифікації досягається шляхом виділення невеликих структурних елементів, варіація яких забезпечує зміну структури умови задачі.

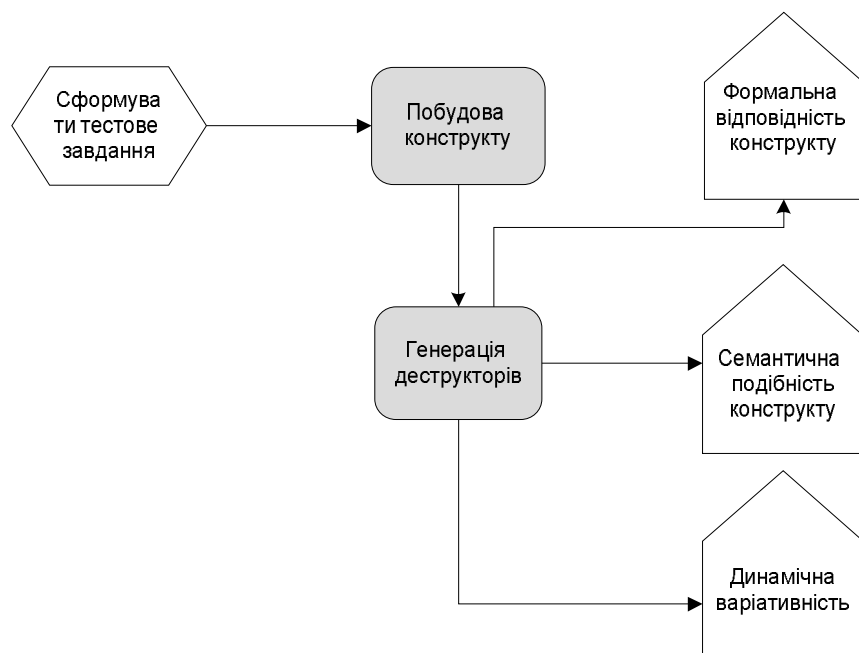


Рисунок 2.3 – Схема формування тестового завдання

Подані положення дозволяють реалізувати генерацію варіативних тестових завдань для перевірки теоретичних і методологічних знань, практичних вмінь та навичок за допомогою інформаційної технології, структура якої розглядається в наступних підрозділах.



Рисунок 2.4 – Процес перевірки вмінь та навичок

2.2 Методи автоматичної генерації тестових завдань для перевірки теоретичних знань

Виходячи із формалізації інформативних матеріалів, яка описана структурами (2.1), (2.7) пропонуються методи автоматичної генерації тестових завдань множинного вибору, альтернативного виду, на заповнення прогалін у тексті, на встановлення відповідності, на перестановку. Основи даних підходів описані в роботах [32,35,38,106].

Метод автоматичної генерації тестових завдань виду «множинний вибір»

Завдання з множинним вибором припускають наявність варіативності у виборі. Особа, яка тестується, повинна здійснити вибір з можливих альтернативних варіантів, серед яких один або декілька є вірними.

В процесі генерації завдань закритого типу необхідно побудувати основу BTt , множини вірних альтернатив StA та дистракторів SfA . При цьому потрібно забезпечити динамічну перевірку вибраних альтернатив і побудову рекомендації щодо вивчення тем, по яких були допущені помилки. Основа тесту (процедура P_BTt) будується за допомогою формалізованого твердження, вибраного випадковим чином. Цей вибір ідентифікує кореневий елемент формалізованого твердження, що є базою першої частини основи тесту. Друга частина є продовженням першої, має варіативну довжину і непусте продовження у формалізованому твердженні.

Множину правильних альтернатив (процедура P_StA) формуємо за допомогою множини дочірніх елементів кінцевого вузла другої компоненти основи тестового завдання. Множину дистракторів (процедура P_SfA) будуємо із множини формалізованих тверджень узгоджених по семантичних, синтаксичних та формальних критеріях із кінцевим вузлом другої компоненти

основи тестового завдання. Перевірка правильності вибраних альтернатив (процедура P_CtrAns) полягає в перевірці їх належності до множини правильних альтернатив StA . Якщо тестове завдання формується на основі кількох розділів, то по результатах тестування будуються рекомендації (процедура P_Rec) щодо їх поглибленого вивчення. Видається перелік розділів із яких були вибрані хибні альтернативи в порядку спадання частоти їх вибору [33,35].

Розглянемо процедуру генерації основи тестового завдання P_BTt .

Основа тестового завдання формується із формалізованих тверджень, які представлені у деревовидному вигляді Df . Як зазначалося вище, вона складається з двох частин: перша обов'язково включає батьківський елемент, тобто $Pr_IdE = 0$, а друга частина формує альтернативне представлення тестового завдання. Першу частину основи випадковим чином вибираємо із Df , тобто виконуємо операцію $\tau_{RANDLIMIT(1)}$, але при умові що вибрана компонента CS та її зв'язок з іншими компонентами CNJ належать розділу, який вибрано в процесі тестування ($IdCh = IdCh_C$). Також необхідно врахувати вибрані ідентифікатори семантичних ($IdSc = IdSc_C$) та формальних ($IdFc = IdFc_C$) класів, і провести перевірку чи не було вибрано дану компоненту твердження для попереднього тесту ($IdE \notin Side$), тобто забезпечуємо унікальність генерованих тестових завдань в процесі тестування. Ці операції формалізуємо наступним чином.

Генеруємо BTt_1 першу частину основи тестового завдання $BTt = \langle BTt_1, BTt_2 \rangle$:

$$BTt_{1_C} = \pi_{IdE,CS,CNJ} (\sigma_{\substack{IdCh=IdCh_C \wedge IdSc=IdSc_C \wedge IdFc= \\ =IdFc_C \wedge IdE \notin Side \wedge Pr_IdE=0}} (\tau_{RANDLIMIT(1)}(Df))), \quad (2.9)$$

де π – оператор проєкції реляційної алгебри, що забезпечує вибір атрибутів відповідного відношення; σ – оператор селекції, що забезпечує виконання вказаних умов відповідних атрибутів вибраного відношення; τ –

оператор сортування реляційної алгебри; *RAND LIMIT* – функція випадкового вибору заданої кількості елементів множини; $SidE = \langle IdE \rangle$ - множина ідентифікаторів вибраних основ тестових завдань; $SidE = \{\emptyset\}$ - перед початком процедури генерації тестових завдань, на кожному наступному кроці цю множину поповнюємо ідентифікаторами вибраних основ на попередньому кроці:

$$SidE = SidE \cup \pi_{IdE}(BTt_1 - C). \quad (2.10)$$

Друга частина основи тестового завдання BTt_2 є одним з можливих продовжень першої частини в даному формалізованому тверджені, тобто батьківській елемент BTt_2 повинен дорівнювати ідентифікатору побудованої першої частини основи $Pr IdE = \pi_{IdE}(BTt_1 - C)$. За цією умовою вибираємо один із дочірніх елементів першої частини основи як кореневий елемент другої частини. Будуємо єдину гілку піддерева вибраного елемента за допомогою об'єднання операторів алгебри відношень. Перший із них визначає кореневий елемент, а другий рекурсивно під'єднує до попереднього елемента його дочірний, виходячи із псевдонімів відношень ($AS(Df, DF1)$), $AS(Df, DF2)$ та їх з'єднання по відповідних атрибутах $\triangleright \triangleleft_{Df1(Pr_{IdE})=Df2(IdE)}$. В процесі рекурсивної побудови гілки запам'ятовуємо рівень його кожної компоненти, тобто $\gamma_{COUNT(IdE)ASlc}$. Як і для першої частини основи тестового завдання здійснюємо семантичний та формальний контроль по відповідних ідентифікаторах. Формально цей крок можна описати наступною формулою

$$Br = \pi_{IdE,CS,CNJ,Pr_{IdE}} \gamma_{AS(COUNT(IdE),lc)} \left(\sigma_{\substack{IdCh=IdCh_C \wedge IdSc=IdSc_C \wedge IdFc=IdFc_C \wedge Pr_{IdE}= \\ = \pi_{IdE}(BTt_1 - C)}} \right) \cup \pi_{IdE,CS,CNJ,Pr_{IdE}} \gamma_{AS(COUNT(IdE),lc)} \left(\tau_{RANDLIMIT(1)}(Df) \right) \cup \pi_{IdE,CS,CNJ,Pr_{IdE}} \gamma_{AS(COUNT(IdE),lc)} \left(\tau_{RANDLIMIT(1)}(AS(Df, DF1)) \right), \quad (2.11)$$

$$\triangleright \triangleleft_{Df1(Pr_{IdE})=Df2(IdE)} \sigma_{Pr_{IdE}=\pi_{IdE}(BTt_1 - C)}(AS(Df, DF2))$$

де γ - оператор групування реляційної алгебри; $\triangleright\triangleleft$ – оператор з'єднання двох відношень за умови рівності атрибутів з однаковими іменами; AS - оператор, який використовується для задання псевдонімів в рамках використовуваного оператора; lc - рівень елемента Br , $COUNT$ – оператор підрахунку кількості значень вибраного атрибуту.

Знаходимо висоту побудованої гілки дерева вибраного формалізованого твердження, тобто знаходимо максимальний рівень $\gamma_{MAX(lc)}(Br)$ вибраних компонент елемента Br

$$Brh = \gamma_{MAX(lc)}(Br). \quad (2.12)$$

Для забезпечення варіативності завдання випадковим чином визначаємо висоту другої частини основи тесту BTt_2h :

$$BTt_2h = Rand(Brh - 1), \quad (2.13)$$

де $Rand(Rn)$ – функція генерації випадкового цілого числа з інтервалу $[0; Rn]$.

Знаходимо ідентифікатор $IdBk$ кінцевого вузла компоненти BTt_2_C :

$$IdBk = \pi_{IdE}(\sigma_{lc=BTt_2h}(Br)). \quad (2.14)$$

Генеруємо BTt_2 другу частину основи тестового завдання:

$$BTt_2_C = \pi_{IdE,CS,CNJ}(\sigma_{IdE \leq IdBk}(Br)), \quad (2.15)$$

тоді основа тестового завдання:

$$BTt_C = BTt_1_C \cup BTt_2_C. \quad (2.16)$$

Розглянемо процедуру генерації множини правильних альтернатив P_StA .

Будуємо множину $DBTt_2$ дочірніх елементів кінцевого вузла компоненти BTt_2_C :

$$DBTt_2 = \pi_{IdE,CS,CNJ}(\sigma_{Pr_IdE=Id_Bk}(Df)). \quad (2.17)$$

Знаходимо кількість Dc дочірніх елементів кінцевого вузла компоненти BTt_2_C :

$$Dc = \gamma_{COUNT(IdE)}(DBT_2). \quad (2.18)$$

Визначаємо кількість правильних альтернатив CtA тестового завдання:

$$CtA = Rand(Dc). \quad (2.19)$$

Будуємо початкові вузли правильних альтернатив StA_0 :

$$StA_0 = \pi_{IdE} \tau_{RANDLIMIT(Dc)}(DBTt_2), \quad (2.20)$$

та відповідно саму множину правильних альтернатив

$$StA = \pi_{IdE,CS,CNJ}(\sigma_{IdE=StA_0}(Df)) \cup (\pi_{IdE,CS,CNJ}(\sigma_{IdE=\pi_{IdE}(StA_0) \vee Pr_IdE=\pi_{IdE}(StA_0)}(Df))). \quad (2.21)$$

Розглянемо процедуру генерації хибних альтернатив P_SfA .

Встановлюємо кількість хибних альтернатив:

$$CfA = CSa - CtA, \quad (2.22)$$

Генерування хибних альтернатив відбувається на основі семантичної ($IdSc = IdSc_C$), формальної ($IdFc = IdFc_C$), та синтаксичної $CNJ = CNJ_C$ узгодженості із основою тестового завдання:

$$SfA = \pi_{IdE,CS,CNJ,Pr_IdE} \left(\sigma_{\substack{IdCh=IdCh_C \wedge IdSc=IdSc_C \wedge IdFc=IdFc_C \wedge CNJ= \\ CNJ_C \wedge Pr_IdE \neq \pi_{IdE}(BT_{\bar{1}}_C)}} \right) \\ (\tau_{RANDLIMIT(1)}(Df))) \cup \pi_{IdE,CS,CNJ,Pr_IdE} (\tau_{RANDLIMIT(1)}(AS(Df, DF1))) \quad (2.23) \\ \triangleright \triangleleft \\ \sigma_{Pr_IdE=\pi_{IdE}(BT_{\bar{1}}_C)} (AS(Df, DF2))$$

Синтаксична узгодженість забезпечується рівністю типів зв'язків між вибраною компонентою та компонентою основи тестового завдання.

Випадковим перемішуванням отримуємо множину SgA згенерованих альтернатив в завданні:

$$SgA = \tau_{RAND} (StA \cup SfA). \quad (2.24)$$

Розглянемо процедуру перевірки правильності вибраних альтернатив в процесі проведення тестового контролю P_CtrAns .

Перевіряємо які альтернативи були вибрані, як відповідь

$$IdN = \pi_{IdE}(SgA), \quad (2.25)$$

де IdN - ідентифікатор відзначеної альтернативи тестового завдання.

Проводимо перевірку кожної вибраної альтернативи на правильність, використовуючи наступне правило:

$$\text{якщо } IdN \subset \pi_{IdE}(StA), \quad (2.26)$$

то відповідь на тестове завдання правильна;

$$\text{якщо } (IdN \cap \pi_{IdE}(StA) \neq \emptyset) \wedge (IdN \cap \pi_{IdE}(SfA) \neq \emptyset), \quad (2.27)$$

то відповідь на тестове завдання правильна частково;

$$\text{якщо } IdN \cap \pi_{IdE}(StA) = \emptyset, \quad (2.28)$$

то відповідь на тестове завдання не правильна.

Розглянемо процедуру формування рекомендацій щодо вивчення навчальних тем, по яких були допущені помилки в процесі тестування P_Rec .

Із множини усіх вибраних хибних альтернатив $CSaf_C$ знаходимо розділи Chf , по яких були допущені помилки:

$$CSaf_C = IdN \cap \pi_{IdE}(SfA), \quad (2.29)$$

$$Chf_C = \pi_{IdCh, Ch(NmCh)} \gamma_{COUNT(IdCh) AS E_h} (\sigma_{IdE=\pi_{IdE}(CSaf_C) \wedge IdCh=Ch(Id)}(Df, Ch)), \quad (2.30)$$

де Ch - структура навчальних розділів, яка представлена як

$$Ch = \langle IdCh, NmCh, Pr_IdCh \rangle, \quad (2.31)$$

де $IdCh$ – ідентифікатор розділу навчальної теми; $NmCh$ – назва розділу; Pr_IdCh – ідентифікатор батьківського розділу.

Знаходимо розділи та відсортовуємо їх по найбільшій кількості помилок:

$$Chf = \tau_{E_h}(Chf_C). \quad (2.32)$$

На основі виконання умов (2.29)-(2.32) можна управляти структурою процесу навчання, яка орієнтована на подолання найбільш проблемних ділянок при вивченні навчальних дисциплін.

На рисунку 2.5 схематично зображено процес генерації тестового завдання множинного типу. На даній схемі представлено тестове завдання, яке складається з трьох альтернативних варіантів відповідей.

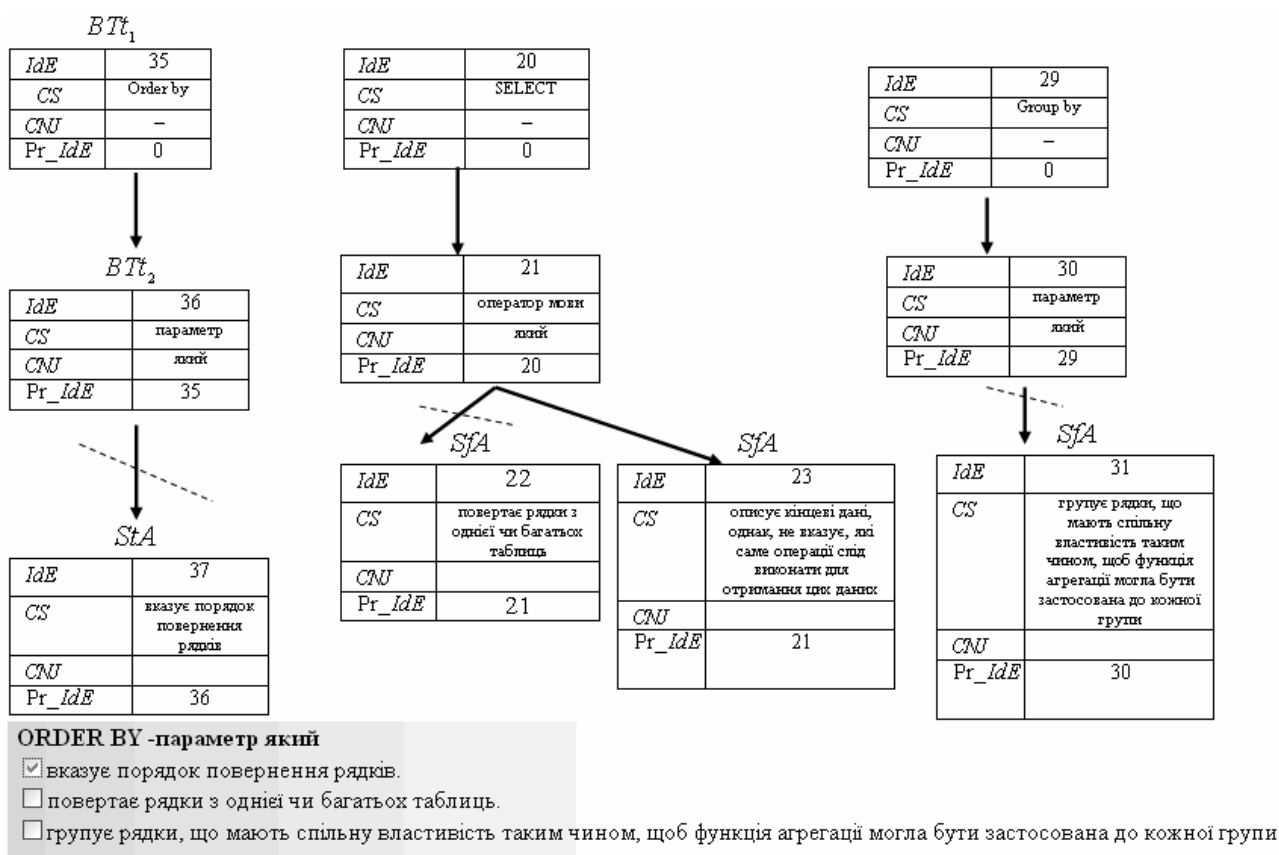


Рисунок 2.5 – Схема генерації тестового завдання [32]

Основа та альтернативи генеруються на основі трьох формалізованих тверджень:

Order by – параметр, який вказує порядок повернення рядків;

SELECT – оператор мови, який повертає рядки з однієї чи багатьох таблиць; описує кінцеві дані, однак, не вказує, які саме операції слід виконати для отримання цих даних;

Group by – параметр, який групує рядки, що мають спільну властивість таким чином, щоб функція агрегації могла бути застосована до кожної групи.

Як видно з даного рисунка, основа тестового завдання BTt складається з двох компонент твердження $BTt_1 = \{\text{Order by, -}\}$ і $BTt_2 = \{\text{параметр, який}\}$, а альтернативні варіанти StA та Sfa формуються шляхом маніпулювання

відповідними компонентами інших тверджень, узгоджених по типах зв'язків між ними $CNJ = \{\text{який}\}$ та характеристиками формальних атрибутів. Точка розриву між компонентами показана штриховою лінією.

Метод автоматичної генерації тестових завдань виду «альтернативний вибір»

Для кожного завдання альтернативного типу даються тільки два варіанти відповіді. Особа, яка тестується повинна вибрати один з варіантів – «так - ні», «правильно – неправильно». Особливістю методу автоматичної генерації завдань альтернативного виду є стандартна множина альтернатив $Sa = \{\text{'Так'}, \text{'Ні'}\}$. Варіативність умови завдання закладається в його основі. Тому вона доповнюється третьою частиною, яка випадковим чином вибирається із множини фрагментів формалізованих тверджень, узгоджених по формальних, семантичних та синтаксичних ознаках із основою. Розглянемо даний процес покроково [32].

1. Генеруємо BTt_1, BTt_2 першу та другу частину основи тестового завдання $BTt = \langle BTt_1, BTt_2, BTt_3 \rangle$, де BTt_3 - третя частина основи тестового завдання. Це здійснюємо аналогічно до процедури P_BTt попереднього методу, формули (2.9) – (2.16).

2. Генеруємо $BTt_3 = \langle BTtt, BTtf \rangle$, де $BTtt$ - вірні доповнення; $BTtf$ - хибні доповнення:

2.1 Генеруємо множин $BTtt$ для BTt_3 .

2.1.1 Будуємо множину $DBTt_2$ відповідно до (2.17);

2.1.2 Знаходимо кількість Dc дочірніх елементів кінцевого вузла компоненти BTt_2_C за допомогою формули (2.18);

2.1.3 Визначаємо кількість правильних доповнень CtA для формування BTt_3 тестового завдання (2.19);

2.1.4 Будуємо початкові вузли правильних доповнень BTt_3 (2.20);

2.1.5 Будуємо вірні BTt_3 аналогічно до (2.21);

2.2. Генерація хибних доповнень $BTtf$ відповідно до (2.22)-(2.23);

3. Випадковий вибір одного елемента BTt_3 з множини згенерованих вірних та хибних доповнень:

$$BTt_3 = Rand(BTtt, BTtf) \quad (2.33)$$

4. Перевірка вибраної альтернативи в процесу тестування відбувається на основі наступного правила:

$$\text{якщо } Sa_C = 'Так' \wedge BTt_3 \in BTtt, \quad (2.34)$$

або

$$\text{якщо } Sa_C = 'Ні' \wedge BTt_3 \in BTtf, \quad (2.35)$$

то відповідь на тестове завдання є вірною, в протилежному випадку вибрана альтернатива являється хибною.

Метод автоматичної генерації тестових завдань типу «на встановлення відповідності»

Завдання на встановлення відповідності передбачає встановлення зв'язків між поняттями, фігурами, конструкціями, згрупованими у два списки [30]. При генерації тестових завдань на встановлення відповідності елементами першого списку слугують дві компоненти формалізованих тверджень, однорідних по формальних та синтаксичних ознаках. Елементами другого списку – треті компоненти основ, які випадковим чином вибираються із множини фрагментів формалізованих тверджень, узгоджених по формальних, семантичних та

синтаксичних ознаках із відповідною основою [32].

Деталізуємо опис даного підходу. Множину елементів лівого списку тестового завдання представимо як LTt , а множина елементів правого списку - RTt . Процес генерації тестового завдання можна описати наступними кроками:

1. Генеруємо задану кількість елементів $CLTt$ множини LTt . Для цього скористаємося формулами (2.9)–(2.16), в результаті виконання яких отримаємо один елемент множини LTt , цей процес виконуємо поки не виконається умова:

$$CLTt = \gamma_{COUNT(IdE)}(LTt). \quad (2.36)$$

2. Генеруємо елементи множини RTt , при умові що

$$CLTt = \gamma_{COUNT(IdE)}(RTt). \quad (2.37)$$

Побудова елементів множини RTt відбувається відповідно до (2.20)–(2.21), які виконуються поелементно із врахуванням результату отриманої множини елементів LTt .

3. На цьому етапі проводимо перемішування елементів отриманих множин LTt та RTt , тобто

$$LTt = \tau_{RAND}(LTt), \quad (2.38)$$

$$RTt = \tau_{RAND}(RTt). \quad (2.39)$$

4. Перевірка правильності вибраних альтернатив в процесі проведення тестового контролю проводиться за допомогою наступного правила:

4.1 Якщо вибраний елемент множини LTt лівого стовпчика, а саме його кінцевий елемент відповідає вибраному початковому елементу множини RTt правого стовпчика, тобто ідентифікатор кінцевого елемента LTt відповідає ідентифікатору батьківського елемента в початковому вузлі вибрано елемента із множини RTt :

$$\pi_{IdE}(LTt_C) = \pi_{Pr_IdE}(RTt_C), \quad (2.40)$$

у протилежному випадку встановлена відповідність вважається хибною. Підсумковий бал за тестове завдання розраховується як відношення кількості правильно встановлених відповідностей до загальної кількості елементів множини LTt .

Метод автоматичної генерації тестових завдань типу «на відновлення послідовності»

Завдання на відновлення послідовності можна розглядати як варіант завдань на встановлення відповідності, коли одним із рядів є час, відстань або інший конструкт [30]. У цих завдання потрібно впорядкувати елементи списку в потрібній послідовності, наприклад, розставити події в хронологічному порядку.

Процес генерації завдань на відновлення послідовності схожий до попереднього описаного методу, але з деякими особливостями:

1) На основі формул (2.9)-(2.16) відтворюємо CSa тверджень, кінцеві компоненти яких представлені у формі чисел або дати

$$Id_Fc \in Fc('числовий', 'дата'), \quad (2.41)$$

2) На основі кінцевих компонент відтворених тверджень формуємо множину для впорядкування альтернатив

$$StA = (\pi_{IdE,CS,CNJ}(\sigma_{IdE=\pi_{IdE}(StA_0) \vee Pr_IdE=\pi_{IdE}(StA_0)}(Df))). \quad (2.42)$$

3) За допомогою відтворених тверджень з вилученою кінцевою компонентою формуємо множину тверджень, послідовність яких потрібно

впорядкувати, які записуємо в представлення тестового завдання в довільному порядку

4) Перевірка правильності встановлюється виходячи із умови завдання (відновлення в зростаючому, спадаючому порядку) та порівняння послідовності встановленої особою, яка тестується із послідовністю, що задається формулами:

$$\tau_{(IdE)DESC}(StA) , \text{ якщо умова «>»}, \quad (2.43)$$

або

$$\tau_{(IdE)ASC}(StA) , \text{ якщо умова «<»}. \quad (2.44)$$

Метод автоматичної генерації тестових завдань типу «заповнення прогалин у твердженні».

Відповіддю на це питання є заповнення прогалин у самому запитанні. Суть даного підходу можна описати за допомогою наступних кроків:

1) на основі екземпляру формуємо випадкове твердження, цю процедуру можна здійснити відповідно до формул (2.9) – (2.16);

2) випадковим чином вибираємо компоненту Tc , яку замінюємо «прогалиною», тобто

$$Tc_C = \tau_{RAND}(BTt_1_C \cup BTt_2_C \cup STa). \quad (2.45)$$

3) вибрану компоненту переносимо в множину альтернатив:

$$Sa = Tc_C . \quad (2.46)$$

4) інші альтернативи вибираємо з компонент того самого рівня інших екземплярів семантичного класу, узгоджених по типу зв'язку із сусідніми елементами основи тестового завдання.

5) Перевірка правильності проводиться на основі наступного правила:

$$\pi_{IdE}(BTt_1_C) = \pi_{Pr_IdE}(Tc_C), \text{ якщо } Tc_C = BTt_2_C, \quad (2.47)$$

$$\pi_{IdE}(BTt_2_C) = \pi_{Pr_IdE}(Tc_C), \text{ якщо } Tc_C = STa, \quad (2.48)$$

$$\pi_{IdE}(Tc_C) = \pi_{Pr_IdE}(BTt_2_C), \text{ якщо } Tc_C = BTt_1_C. \quad (2.49)$$

Запропоновані методи дозволяють автоматично генерувати більшість основних завдань, які використовуються у комп'ютеризованих тестувальних системах. У підрозділі 4.2 наведено приклади автоматично згенерованих тестових завдань різних видів та для різних дисциплін.

2.3 Метод генерації проблемних ситуацій для тестування методологічних знань

За допомогою тестових завдань для перевірки теоретичних знань, описаних в попередньому підрозділі, можна верифікувати лише рівень відтворення знань і вміння їх осмислювати. Обмеження такими завданнями може призвести до розвитку шаблонного мислення. Отже тестування повинно включати перевірку розуміння знань, у тому числі рівень логічного мислення. Логічне мислення передбачає не лише здатність аналізу окремих аспектів дійсності та розв'язання різних типів практичних завдань, а і можливість застосування набутих знань та навичок в конкретних нестандартних ситуаціях.

Для цього необхідні вміння будувати обґрунтовані оцінки структури конкретних ситуацій, вивчати і аналізувати певні системи і підсистеми, вміти визначати в них залежні (змінні) і незалежні (постійні) параметри, виділяти причинно-наслідкові зв'язки між ними і, нарешті, здобувати для реалізації описаних аналізів необхідну інформацію [1]. Такі вимоги задіюють повний спектр вмінь кваліфікованого спеціаліста, взаємодію яких наведено на рис. 2.6.

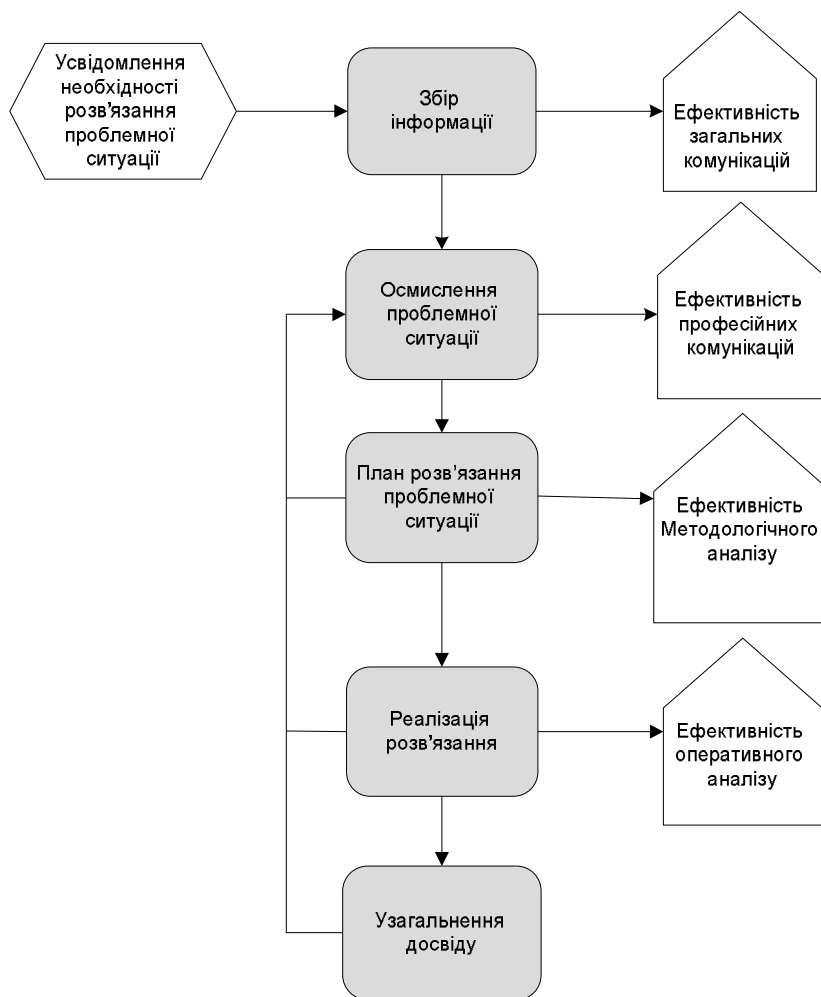


Рисунок 2.6 – Процес вирішення проблемної ситуації

Зокрема, збір потрібної інформації для розв'язання проблем передбачає вміння будувати ефективні комунікації. Вони формуються в процесі вивчення загальнотеоретичних дисциплін. Осмислення проблеми передбачає використання теоретичних положень прикладних дисциплін з метою професійного аналізу проблеми та представлення його результатів. Формування відповідних навичок контролюється тестами перевірки теоретичних знань. Їх наявність свідчить про ефективність професійних комунікацій. Результатом осмислення проблеми повинен стати план її розв'язання за допомогою відомих методів. Здатність будувати подібні плани контролюється за допомогою тестів для перевірки методологічних знань [37]. Ефективність методологічного аналізу повинна бути підтверджена успішним застосуванням методів

розв'язання поставлених задач. Вміння застосовувати такі методи контролюється за допомогою тестів для перевірки вмінь та навичок, що будуть розглянуті в наступному підрозділі.

Таким чином в аналізі нестандартних ситуацій на передній план виходять вміння формулювати задачі, розв'язання яких забезпечує досягнення бажаного результату. Синтез формулювання задач та методів їх розв'язання пропонується методиках ТРИЗ, розроблених Г.С.Альтшуллером.

Однією із таких ефективних методик є адаптивний алгоритм розв'язку творчих задач, побудований на основі АРИЗ-85В в рамках загальної теорії сильного мислення [59]. Розглянемо детальніше даний алгоритм.

Алгоритм РТЗ (розв'язку творчих задач)

- РТЗ.1. Початок. Попередній опис проблемної ситуації.
- РТЗ.2. Виділення конкретної задачі із проблемної ситуації.
- РТЗ.3. Побудова абстрактної моделі конкретної задачі, формулювання протиріччя.
- РТЗ.4. Побудова абстрактної моделі розв'язку задачі, представлення вірного кінцевого результату.
- РТЗ.5. Вплив факторів і вихід на конкретний результат.
- РТЗ.6. Якщо рішення зрозуміле лише як концептуальне, то необхідно сформулювати його підзадачі. Перехід на РТЗ.3 для розв'язання сформульованих підзадач та конкретизації результату.
- РТЗ.7. Оцінка отриманого результату
- РТЗ.8. Кінець.

Аналіз наведеного алгоритму показує, що в силу своєї загальності він передбачає побудову багаторівневої структури задач, що підлягають розв'язку (циклічний перехід від кроку РТЗ.6 до РТЗ.3). В умовах генерації тестового завдання така багаторівневність буде надмірною. Тому обмежимося випадком, коли проблемна ситуація породжує конкретну задачу, яка може бути розв'язана відомими методами. Вибір методів залежить від факторів ситуації (РТЗ.5.).

На основі аналізу структури проблемної ситуації та завдань перевірки засвоєння методологічних знань, проблемну ситуацію можна формалізувати за допомогою наступних структур [37]:

$$Ps = \langle Pa, Pi, Po, Attr, Pr, Pc, Pmt \rangle, \quad (2.50)$$

де Pa - предметна область аналізу проблемної ситуації; Pi - описи проблемної ситуації; Po - множина об'єктів виникнення проблеми; $Attr$ - множина атрибутів; Pr - множина, яка описує результат вирішення проблемної ситуації; Pc - множина характеристик перебігу проблемної ситуації; Pmt - множина методів, які направлені на вирішення проблемної ситуації.

$$Pa = \langle IdPa, NmPa \rangle, \quad (2.51)$$

де $IdPa$ - ідентифікатор предметної області; $NmPa$ - предметна область.

$$Pi = \langle IdPi, NmPi, IdPa \rangle, \quad (2.52)$$

де $IdPi$ - ідентифікатор твердження; $NmPi$ - текстові твердження для опису проблемної ситуації;

$$Po = \langle IdPo, NmPo, IdPa, IdPi \rangle, \quad (2.53)$$

де $IdPo$ - ідентифікатор об'єкта; $NmPo$ - інформація, що описує структуру об'єкта виникнення проблемної ситуації.

$$Attr = \langle IdAttr, AT, AV, IdPa, IdPi, IdPo \rangle, \quad (2.54)$$

де $IdAttr$ - ідентифікатор атрибуту; AT - тип атрибуту; AV - значення атрибутів проблемної ситуації;

$$Pr = \langle IdPr, RNm, IdPa, IdPi, IdPo \rangle, \quad (2.55)$$

де $IdPr$ - ідентифікатор результату; RNm твердження, які описують результат.

$$Pc = \langle IdPc, PA, Dsc, IdPa, NA, IdPo, IdPi, IdAttr \rangle, \quad (2.56)$$

$$SuMth = \langle IdPmt, IdPc \rangle, \quad (2.57)$$

де $SuMth$ - метод, який вирішує задану проблемну ситуацію; $IdPc$ - ідентифікатор ознак, які впливають на умови перебігу проблемної ситуації; PA - позитивна ознака; NA - альтернативна ознака, Dsc - твердження, яке описує умови виникнення та перебігу проблемної ситуації.

$$Pmt = \langle IdPmt, IdMth, IdPo, IdPa, IdPi, IdAttr \rangle, \quad (2.58)$$

$$Mth = \langle IdMth, NmMth, Ac \rangle, \quad (2.59)$$

де $IdMth$ - ідентифікатор методу; $NmMth$ - метод розв'язання проблемної ситуації; Ac - множина тверджень, яка визначає метод.

Виходячи із формалізації проблемних ситуації та алгоритму розв'язку творчих задач пропонується метод автоматичної генерації тестових завдань для перевірки засвоєння методологічних знань [37], схематичне представлення якого наведено на рисунку 2.7.

Формування тестового завдання відбувається на основі вибраної предметної області, в якій випадковим чином вибирається проблемна ситуація. Для забезпечення варіативності формалізованої проблемної ситуації випадковим чином вибираємо її атрибути та бажаний результат її розв'язання. Для відображення невизначеності проблемної ситуації визначаємо множину потенційних методів, що можуть бути використані для її розв'язання.

Серед них випадковим чином вибираємо єдиний метод, який виявиться реальним засобом розв'язання цієї ситуації. Реальність цього методу

забезпечуємо за допомогою умов, що ідентифікують його застосовність. Для моделювання перебігу практичної ситуації ці умови поповнюємо такими, що не підтримуються атрибутами даної задачі. Умови, що реалізують даний метод можуть допускати застосування ще й інших методів. Тому генеруються додаткові умови, що забезпечують унікальність реального методу. Взаємодія атрибутів, умов перебігу та методів проблемної ситуації супроводжується інформаційними потоками, що схематично зображено потовщеними стрілками.

Детальне представлення методу опишемо за допомогою наступної послідовності кроків [37]:

1) Фіксуємо $IdPa_C$ предметної області, яка вибирається користувачем в ході тестування.

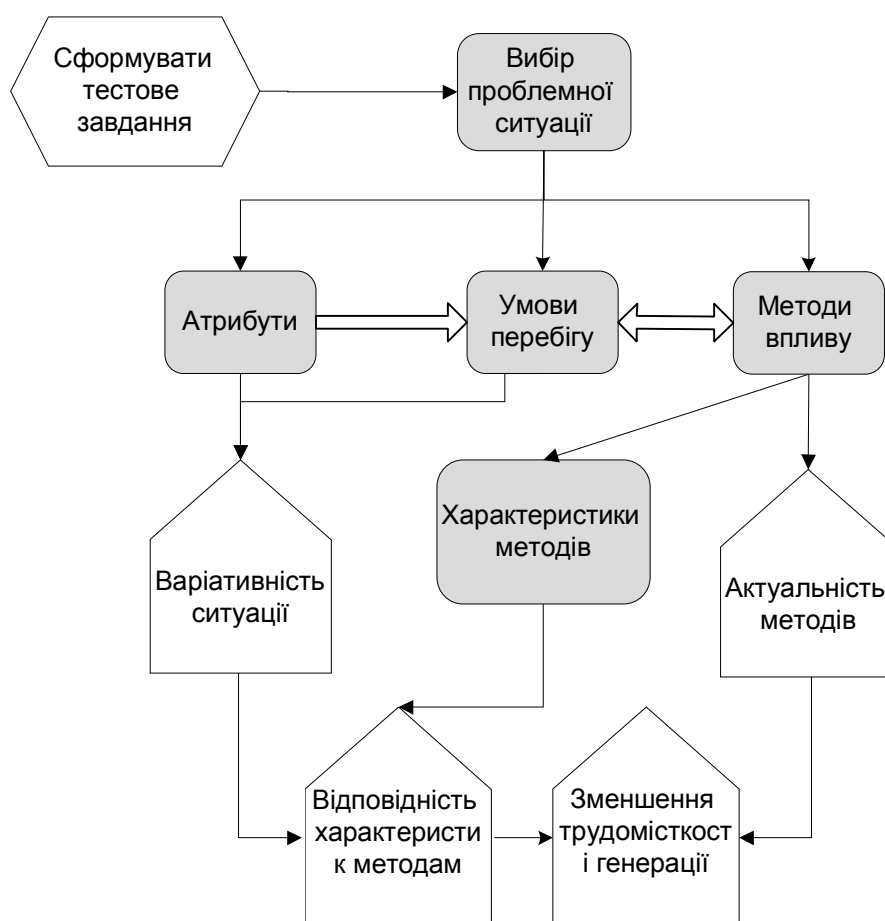


Рисунок 2.7 – Схема генерації тестових завдань для перевірки засвоєння методологічних знань

2) Генерація проблемної ситуації

$$Pi_C = \pi_{IdPi, NmPi} (\sigma_{Pi(IdPa)=IdPa_C} (\tau_{RANDLIMIT(1)} (Pi))). \quad (2.60)$$

3) Генеруємо об'єкт проблемної ситуації

$$Po_C = \pi_{IdPo, NmPo} (\sigma_{Po(IdPa)=IdPa_C \wedge Po(IdPi)=IdPi_C} (\tau_{RANDLIMIT(1)} (Po))). \quad (2.61)$$

4) Генеруємо атрибути проблемної ситуації:

$$Attr_C = \pi_{IdAttr, AV} (\sigma_{Attr(IdPa)=IdPa_C \wedge Attr(IdPi)=IdPi_C \wedge Attr(IdPo)=IdPo_C} (\tau_{RANDLIMIT(1)} (Attr))). \quad (2.62)$$

5) Для сформованих значень Pi_C та Po_C генеруємо бажаний результат вирішення проблемної ситуації

$$Pr_C = \pi_{IdPr, RNm} (\sigma_{Pr(IdPa)=IdPa_C \wedge Pr(IdPi)=IdPi_C \wedge Pr(IdPo)=IdPo_C} (\tau_{RANDLIMIT(1)} (Pr))). \quad (2.63)$$

6) Формуємо множину потенційних методів розв'язання сформованої проблемної ситуації:

$$Pmt_C = \pi_{IdPmt, IdMth} (\sigma_{Pmt(IdPa)=IdPa_C \wedge Pmt(IdPi)=IdPi_C \wedge Pmt(IdPo)=IdPo_C \wedge Pmt(IdAttr)=Attr_C} (Pmt)) \quad (2.64)$$

7) Генеруємо реальний метод $PmtR$ розв'язання проблемної ситуації:

$$PmtR = \pi_{IdPmt, IdMth} (\tau_{RANDLIMIT(1)} (Pmt_C)) \quad (2.65)$$

8) Формуємо умови перебігу проблемної ситуації:

$$Pc_C = \pi_{IdPc, Dsc, IdAttr} (\sigma_{Pc(IdPa)=IdPa_C \wedge Pc(IdPi)=IdPi_C \wedge Pc(IdPo)=IdPo_C} (Pc)). \quad (2.66)$$

9) Формуємо умови PcR відповідні до реального методу:

$$PcR = \pi_{IdPc, Dsc} (\sigma_{SuMth(IdMth)=Pmtr(IdPmt) \wedge Pc(IdPc)=SuMth(IdPc)} (Pc_C, SuMth)). \quad (2.67)$$

10) Формуємо умови PcF_1 , що не підтримуюся вибраними атрибутами:

$$PcF_1 = \pi_{IdPc, Dsc} (\sigma_{Pc_C(IdAttr) \neq Attr_C(IdAttr)} (Pc_C, Attr_C)). \quad (2.68)$$

11) Вибираємо умови, що порушують унікальність реального методу і доповнимо їх альтернативними ознаками:

$$PcF = (Pc_C \setminus PcR) \setminus PcF_1. \quad (2.69)$$

12) Представляємо згенероване завдання TPs :

$$TPs = \{Pi_C(NmPi), Po_C(NmPo), Attr_C(AV), Pr_C(RNm), PMt_C(NmMth), PcF(NA, Dsc), PcF_1(PA, Dsc), PcR(PA, Dsc)\}. \quad (2.70)$$

13) Для зменшення ймовірності вгадування правильної альтернативи, генеруємо твердження, які будуть характеризувати представлені методи, причому хоча б одне характеризуватиме правильний метод, генерування здійснюємо на основі формул (2.9) – (2.24).

Виконання кроків (1)-(13) дозволяє згенерувати завдання для перевірки засвоєння методологічних знань. Використання запропонованого методу дозволяє економити часові ресурси викладачів при створенні проблемних ситуації, а також при їх перевірці, тобто на етапі встановлення причинно-наслідкових зв'язків перебігу проблемної ситуації. Завдання даного виду

дозволяють також проводити оцінку ступенів розуміння проблеми, можливість застосування конкретних методів їх розв'язання в залежності від умов перебігу проблеми та факторів, які на неї впливають.

Висновки до розділу 2

1. Показано доцільність використання деревовидних інформаційних структур для формалізації інформативних матеріалів, яка дозволяє сформувати базові структури для автоматичної генерації тестових завдань. Запропоновано формальні та семантичні класи, які дозволяють враховувати основні характеристики інформативних представлень для уникнення генерації завідомо невірних альтернатив.

2. Розроблено методи автоматичної генерації тестових завдань для перевірки теоретичних знань, який на відміну від існуючих дозволяє генерувати тестові завдання із врахуванням формальної, семантичної та синтаксичної узгодженості. Метод розповсюджено на формування відомих видів тестових завдань закритого типу.

3. Побудовано інформаційні структури для формалізації процесу розв'язання проблемних ситуацій, що дозволило сформувати структуру тестових завдань для перевірки методологічних знань.

4. Розроблено метод автоматичної генерації тестових завдань для перевірки засвоєння методологічних знань, який дозволяє автоматизувати процедуру генерації проблемних ситуацій.

РОЗДІЛ 3

ТЕХНОЛОГІЯ ОРГАНІЗАЦІЇ ПРОЦЕСУ НАВЧАННЯ НА ОСНОВІ АДАПТИВНИХ ГЕНЕРАТОРІВ ТЕСТОВИХ ЗАВДАНЬ

3.1 Метод генерації завдань для перевірки вмій та навичок

Осміслене володіння понятійним апаратом предмету та вмінням вибирати вірний метод отримання бажаного результату в умовах невизначеної ситуації – це лише перші етапи в осмисленні та професійному розв’язанні проблемної ситуації. В цьому процесі надзвичайно важливим є етап використання відомих методів для побудови розв’язку в конкретних, нетипових умовах. Вміння такого типу формуються та шліфуються в процесі розв’язання задач із відповідної предметної області. Під задачею будемо розуміти формулювання умови із вказанням необхідних даних для побудови її розв’язку або пояснення причин його відсутності.

Під час розв’язання практичних задач отримуються нові знання, оскільки виникає необхідність знайомства з новою ситуацією, яка описана в задачі, можливості застосування різноманітних теорій та методів до її розв’язання. При оволодінні новим методом розв’язання задач у людини формується вміння знімати проблемні ситуації, а при достатньому тренуванні формуються практичні навички, які прискорюють процес розв’язку. Застосування задач в навчальному процесі дозволяє адаптувати суб’єкта навчання до практичної діяльності в майбутньому [87].

Складність передавання знань і практичних навичок під час розв’язання задач є однією із основних проблем процесу навчання. В процесі побудови, використання та перевірки розв’язків задач виникають великі часові та інтелектуальні витрати викладачів, що часто спонукають до скорочення варіативності під час перевірки засвоєння та використання знань і навичок. Виникає проблема формалізації практичних задач для їх автоматичного

формування, пошуку розв'язків з метою отримання достатньої множини унікальних завдань, які будуть максимально близькими до практичних ситуацій. Їх використання в прикладних програмних середовищах дозволить суттєво економити викладацькі ресурси, що сприятиме підвищенню ефективності навчання.

Зазвичай формулювання задачі складається з умови і даних, які необхідно знайти. Між умовою та даними існують внутрішні відношення (зв'язки, залежності), характер яких визначає структуру задачі [21]. Структуру задачі можна формалізувати за допомогою наступної структури:

$$Task = \langle Condition, Inp_Value, Res_Value \rangle, \quad (3.1)$$

де *Condition* – умова задачі; *Inp_Value* – вхідні дані; *Res_Value* – результуючі змінні;

Окрім неї, для перевірки правильності дій студента, опис задачі повинен містити еталонний виклад ходу її розв'язання, або хоча б вірну відповідь. Такі задачі і складають основу тестів для перевірки вмінь та навичок, які можуть подаватися як у відкритій так і закритій формах.

Процеси побудови згаданих задач та їх розв'язання достатньо громіздкі. З метою підвищення ефективності цього процесу широко застосовуються параметризовані тести, умови та розв'язки яких залежать від певних параметрів. При такому підході одним тестом охоплюється множина задач, чим економляться зусилля викладачів. При цьому студент будує розв'язок усно або шляхом ручних обчислень та перетворень.

В той же час масове використання програмованих обчислювальних засобів робить актуальним перевірку вміння алгоритмізувати процедуру розв'язання тестових задач. При цьому з'являється можливість будувати програмовані процедури розв'язання широкого класу задач та моніторити кроки розв'язку в багатокроковому процесі. Тут зміна вхідних змінних вже не змінюватиме самої задачі, бо програмна реалізація при цьому змінюватися не буде. Вхідні змінні

застосовуватимуться лише для тестування працездатності запропонованого процедурного розв'язання. В той же час в даному підході параметризований тест реалізується як частковий випадок. Обмеженням даного підходу виступає наявність програмного інтерпретатора для реалізації розв'язку задачі за його формалізованим алгоритмом.

Процес розв'язання такого роду тестових завдань можна реалізувати за допомогою бібліотеки загальних та спеціально розроблених процедур в певному програмному середовищі. Задіюючи інші програмні компоненти або генеруючи нові, за рахунок незначних зусиль можна суттєво міняти структуру пропонованої задачі та модифікувати процедуру її розв'язку. Тому даний підхід отримав назву метод генерації тестових завдань з програмованим оператором. Його загальну схему подано на рисунку 3.1.

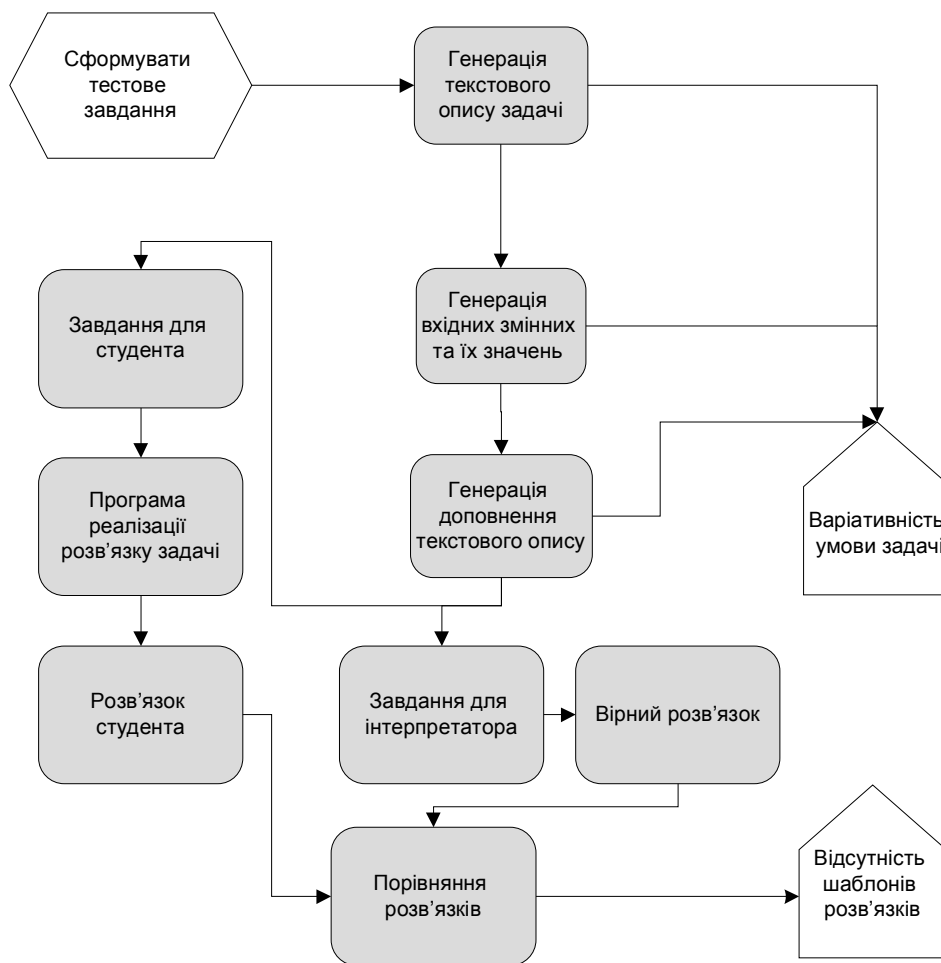


Рисунок 3.1 – Автоматична генерація тестових завдань з програмованим оператором

Процес генерації тестового завдання даного типу включає вибір загального текстового опису задачі із заготованих шаблонів. На основі типу змінних задачі, які задані в її описі, випадковим чином генеруються їх значення. Аналогічно, відповідно до обраної задачі, генерується її уточнення, що дозволяє з мінімальними зусиллями суттєво модифікувати структуру першої. За допомогою наповнення відповідних відношень паралельно генеруються текстова постановка задачі та процедура побудови розв'язку у відповідному програмному середовищі. На основі отриманої задачі студент будує та відлагоджує програму її розв'язання в тому ж програмному середовищі. Результати, отримані програмою студента та еталонною процедурою на спеціальному наборі контрольних даних, порівнюються між собою. Такий підхід дозволяє підтримувати значну варіативність задач та уникати шаблонності в підходах до їх розв'язання. Якщо студенту подається лише умова задачі без можливості програмної реалізації процедури її розв'язання, то система працює в режимі параметричного тесту із розширеною варіативністю за рахунок модифікації структури задачі.

Виходячи із поставлених завдань структура задач на перевірку вмінь та навичок тесту формалізована наступним чином [34, 35]:

$$Condition = \langle Ft, Tf, Tfil \rangle, \quad (3.2)$$

де Ft – множина представлень умови задачі; Tf – множина допустимих типів значень змінних; $Tfil$ – множина текстових представлень для уточнення умови задачі.

Множина представлень умови задачі та її уточнення визначається так:

$$Ft = \langle IdFt, Nmf, Lg_Int, IdCh, Tcv, IdTf, IdVtv \rangle, \quad (3.3)$$

$$Tfil = \langle IdTfil, Nmfil, Lg_Int, Tcv, IdTf, IdVtv \rangle, \quad (3.4)$$

де $IdFt$ – ідентифікатор варіанту представлення; Nmf , $Nmfil$ – функції, які реалізовані в середовищі програмування; Lg_Int – лінгвістична інтерпретація Nmf ; Tcv – кількість вхідних змінних; $IdTf$ – ідентифікатор типу змінних; $IdVtv$ – ідентифікатор виду змінних.

Множина допустимих типів значень змінних та їх видів визначається структурами

$$Tf = \langle IdTf, Type \rangle, \quad (3.5)$$

$$Vtv = \langle IdVtv, TypeVtv \rangle, \quad (3.6)$$

де $Type$ – тип значень вхідних та результуючих змінних, наприклад `string`, `integer`, $TypeVtv$ – вид вхідних та результуючих змінних, наприклад масив, число, тощо.

Вхідні та результуючі змінні подаємо наступним чином

$$Inp_Value = \langle IdV, Value, IdTf, IdVtv \rangle, \quad (3.7)$$

$$Res_Value = \langle Id, Value, IdTf \rangle, \quad (3.8)$$

що дозволяє побудувати результуюче представлення задачі у вигляді

$$Task = \langle Ft, Tf, Tfil, Inp_Value, Res_Value \rangle. \quad (3.9)$$

Таким чином при створенні нової структури для генерації тестових завдань викладач повинен внести текстовий опис задачі та її уточнення, а також їх процедурні відповідники. Набори вхідних даних генеруються автоматично на основі їх опису. Для модифікації структури завдання необхідно поповнити лише множину уточнень в текстовому та процедурному представленні.

Метод автоматичної генерації задач з програмованим оператором

Метод автоматичної генерації множини задач з програмованим оператором передбачає генерацію задач на природній мові для студента (3.11 – 3.14), та відповідне завдання для інтерпретатора (3.15 – 3.17). Суть даного підходу запропонована у [34] та розвинена в [35]. При порівнянні результатів виконання програм згенерованих системою та створених студентом встановлюється адекватність розв'язку. Розглянемо покроково процедуру генерації задач для перевірки вмінь та навичок.

1. Генерація умови задачі $Task$:

$$Task = \langle TaskS, TaskI \rangle, \quad (3.10)$$

де $TaskS$ – умова задачі для студента; $TaskI$ – умова задачі для інтерпретатора.

2. Генерація задачі для студента $TaskS$:

2.1 формуємо умову задачі для студента на природній мові:

$$TaskS_C = \pi_{IdFl, IdCh, Lg_Int, Tcv, IdTf, IdVtv} (\sigma_{IdCh=IdCh_C} (\tau_{RANDLIMIT(1)} (Ft))) \quad (3.11)$$

2.2 доповнюємо умову задачі вхідними змінними $TaskIv$:

$$TaskIv = \pi_{IdV, Value} (\sigma_{Inp_Value(IdTf)=TaskS_C(IdTf) \wedge Inp_Value(IdVtv)=TaskS_C(IdVtv)} (\tau_{RANDLIMIT(TaskS_C(Tcv))} (TaskS_C, Inp_Value))) \quad (3.12)$$

2.3 уточнюємо умову задачі $TaskSp$:

$$TaskSp = \pi_{IdTfil, Nmfil, Lg_Int, Tcv, IdTf, IdVtv} (\sigma_{TaskSp(IdTf)=TaskS_C(IdTf) \wedge TaskSp(IdVtv)=TaskS_C(IdVtv)} (\tau_{RANDLIMIT(1)} (Tfil, TaskS_C))) \quad (3.13)$$

2.4 Виводимо завдання для студента:

$$TaskS = CONCAT(TaskS_C(Lg_Int), \\ ,TaskIv(Value),TaskSp(Lg_Int)) \quad (3.14)$$

3. Генеруємо завдання для інтерпретатора $TaskI$:

$$TaskI_C = \pi_{IdFi,IdCh,Nmf,Tcv,IdTf,IdVtv} (\sigma_{Fi(IdFi)=TaskS_C(IdFi)} (Ft,TaskS_C)) \quad (3.15)$$

3.1 доповнюємо умову задачі контрольними вхідними змінними $TaskCIv$:

$$TaskCIv = \pi_{IdV,Value} (\sigma_{\substack{Inp_Value(IdTf)=TaskS_C(IdTf) \wedge Inp_Value(IdVtv)= \\ =TaskS_C(IdVtv) \wedge TaskIv(IdV) \neq Inp_Value(IdV)}}} \\ (\tau_{RAND\ LIMIT(TaskS_C(Tcv))} (TaskS_C, Inp_Value, TaskIv))) \quad (3.16)$$

3.2 формуємо завдання для інтерпретатора:

$$TaskI = CONCAT(TaskI_C(Lg_Int), \\ TaskCIv(Value),TaskSp(Nmfil)) \quad (3.17)$$

3.3 фіксуємо результат його виконання:

$$ResT = Interp(TaskI) \quad (3.18)$$

4. Фіксуємо текст програми $ProgS$ для інтерпретатора, сформовану студентом.

5. Фіксуємо результат виконання програми студента на контрольних даних.

$$ResS = Interp(\{ProgS,TaskCIv(Value)\}) \quad (3.19)$$

6. Якщо $ResS == ResT$ – то задача розв’язана вірно.

Алгоритм представлення, автоматичного контролю та перевірки згенерованих задач в програмному середовищі.

Після реалізації формалізованого представлення задачі (3.1) - (3.9) та описаного вище методу генерації множини задач виникає необхідність розробки алгоритму, який дозволяє проводити автоматизований контроль розв'язання згенерованої задачі в процесі оцінки вмінь та навичок суб'єкту навчання. Запропонований алгоритм представлено на рисунку 3.2.

На першому кроці реалізуємо алгоритм генерації умови задачі. На другому кроці проводимо уточнення умови задачі, з метою ускладнення її структури. На третьому кроці проводимо ідентифікацію структури вхідної інформації, кількості вхідних змінних задачі, а також їх типу. На четвертому кроці необхідно представити вхідну інформацію, яка буде використовуватися для автоматичного опрацювання, це може бути для прикладу - масив з даними, таблиця з даними в базі даних, різноманітні структури. Вибір представлення вхідної інформації залежить від інтерпретатора, який буде її опрацьовувати.

На п'ятому кроці автоматично генеруємо завдання для інтерпретатора на основі раніше згенерованої умови. В якості інтерпретатора може виступати компілятор мови програмування, SQL- інтерпретатор, або створений власно інтерпретатор, який здатний виконувати послідовність певних службових операцій. На шостому кроці генеруємо завдання для студента на природній мові, шляхом комбінування відповідно до формалізованих структур задачі та заміни службових конструкцій команд їх аналогами на природній мові. На цьому кроці проводиться порівняння результатів, які повертає інтерпретатор на запит студента та автоматично згенерований запит із встановленням правильності відповіді.

Даний підхід дозволяє автоматично генерувати завдання практичного спрямування із варіативною структурою. В процесі його використання отримується суттєва економія викладацьких ресурсів, які доцільно направити на формалізацію нових типів завдань та удосконалення навчального курсу.

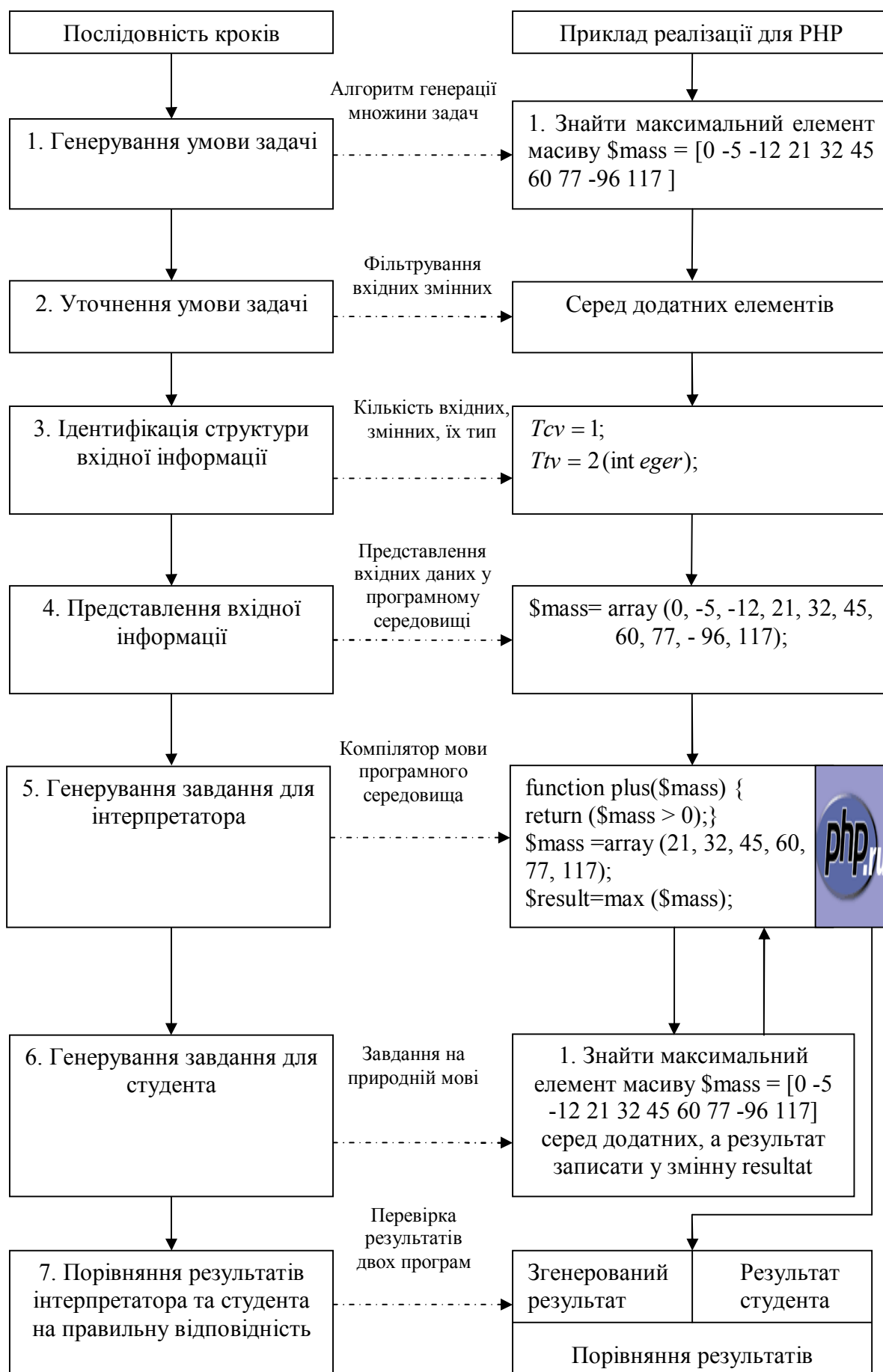


Рисунок 3.2 – Алгоритм представлення, автоматичного контролю та перевірки згенерованих задач в програмному середовищі

3.2 Адаптивно-структурний метод оцінки складності тестових завдань

Тестування, незважаючи на ряд недоліків, на сьогодні є одним із основних засобів для вимірювання рівня знань. Складність тестового завдання вимірюється невизначеністю виявлення вірної альтернативи. Чим багатша структура знань і чим вищий рівень їх засвоєння, тим більше вірогідність руху по найефективнішому розв'язанні актуальної проблеми, тобто по шляху, який коротший по кількості операцій і який добре контрольований в плані виявлення і корекції помилок. Таким чином зменшується вірогідність інших шляхів руху, помилкових і трудомістких, і знижується невизначеність відповіді. Якщо обсяг знань великий, але вони не підтримані відповідними практичними навичками їх використання, то всі шляхи, і правильні і помилкові майже рівно ймовірні і невизначеність вибору максимальна. Якщо обсяг знань незначний, то невизначеність відповіді теж невелика, оскільки найбільш вірогідний певний, як правило помилковий шлях розв'язку. Для цього студента завдання теж здаватиметься нескладним з точки зору вибору рішення. Складність залежить також і від кількості виконуваних операцій. При цьому визначальним є вплив лише істотно нових операцій. Під істотно новою операцією розуміємо операцію, виконання і перевірка якої ще не автоматизована суб'єктом, тобто правильне виконання якої потребує значних часових витрат. Таким чином, бачимо, що складність завдання формується дією різних факторів, як об'єктивних, так і суб'єктивних [41,50,108].

Особливо актуальним є завдання оцінки складності завдань в автоматизованих системах тестування, оскільки навіть порівняно складні моделі для цих систем не дадуть помітної втрати ефективності в формуванні наборів тестів. Аналізуючи сучасні методи оцінки складності тестових завдань, необхідно відзначити, що практично усі вони базуються на статистичних оцінках або структурному аналізі. Статистичні методи вимагають великої вибірки, а методи на основі структурного аналізу – супроводжуються суб'єктивністю. Тому

виникла задача створення адаптивно-структурного методу оцінки складності, що поєднував би переваги структурного аналізу та статистичних методів, та дозволяв проводити апріорну оцінку складності автоматично згенерованих тестових завдань.

Контроль знань за допомогою тестових завдань полягає в представленні сукупності альтернативних тверджень, серед яких необхідно вибрати вірні. З класичної теорії тестів та теорії тестування IRT відомо, що складність тестового завдання залежить від правдоподібності неправильних відповідей, тобто чим краще підібрані дистрактори (неправильні відповіді) тим складніше завдання. Встановлено, що вища частка вибору неправильних відповідей в процесі тестування, свідчить про вищу якість формування тесту.

Нехай тестові твердження представляються наступним чином:

$$O_i \rightarrow \{A_{i1}, \dots, A_{iN}\}, \quad (3.20)$$

де O_i – основна частина тестового твердження; A_{ij} – альтернативні специфікації тестового твердження; N – кількість альтернатив.

Оскільки у більшості автоматизованих систем тестування використовуються тестові завдання закритого типу (багато альтернативні та одноальтернативні) [41], то для побудови оцінок складності тестових завдань розглянемо саме цей випадок.

Будемо оцінювати реальну складність RCS_i розв'язання тестових завдань на основі відносної частоти невірних відповідей на них. Тому значення функція складності повинна приймати із інтервалу $[0, 1]$. В теорії оцінки параметрів тестування IRT моделі будуються на основі неспостережуваних (латентних) змінних. Оскільки ми хочемо контролювати складність завдань, то параметри моделі складності повинні бути легко спостережувані.

В результаті попереднього аналізу проведених експериментів було встановлено, що складність тестового завдання залежить від кількості N альтернатив в ньому та від близькості його альтернатив до вірної. Міру такої

близькості пропонується оцінювати за допомогою показників формальної F та композиційної D спорідненості альтернатив із основою.

Формальна спорідненість встановлюється при генерації тестового завдання і приймає максимальне значення коли альтернатива вірна, проміжні значення при наявності спільних фрагментів із вірною альтернативою та мінімальні значення, при відсутності будь-яких спільних фрагментів із вірною альтернативою. Показник композиційної спорідненості приймає максимальне значення при належності альтернативи до того ж підрозділу, що і основа тестового завдання, проміжне значення, при належності окремого фрагмента альтернативи до підрозділу, що включає основу тестового завдання та мінімальне значення при належності альтернативи до підрозділу, що не включає основу тестового завдання.

Дані показники не є абсолютно незалежними, хоча і не зводяться один до одного. Так, якщо $D = D_{\min}$, то $F = F_{\min}$ і якщо $F = F_{\max}$, то $D = D_{\max}$. Однак проміжні значення цих параметрів не містять таких однозначних залежностей. Зокрема, значенню $D = D_{\text{avg}}$ можуть відповідати значення як $F = F_{\min}$, так і $F = F_{\text{avg}}$, чим забезпечується незалежність показників. Тут індексами $\min, \text{avg}, \text{max}$ позначаються мінімальні, середні та максимальні значення відповідних показників.

На основі цих початкових міркувань спробуємо встановити загальний вид функції складності тестового завдання, який будемо поетапно уточнювати. Перш за все забезпечимо належність значень цієї функції одиничному інтервалу. З цією метою введемо поняття ненормованої складності CS_i тестового завдання, за допомогою якої визначимо реальну складність

$$RCS_i(\vec{F}_i, \vec{D}_i, N) = \frac{CS_i(\vec{F}_i, \vec{D}_i, N) - MnCS(N)}{MCS(N)}, \quad (3.21)$$

$$\text{де } MCS(N) = \max_{(i=1 \dots N)} (CS_i(\vec{F}_i, \vec{D}_i, N)), \quad (3.22)$$

$$MnCS(N) = \min_{(i=1...N)} \left(CS_i(\vec{F}_i, \vec{D}_i, N) \right), \quad (3.23)$$

В цьому випадку значення функції RCS_i задовольнятиме обмеженню:

$$0 \leq RCS_i(\vec{F}_i, \vec{D}_i, N) \leq 1. \quad (3.24)$$

Після нормалізації перейдемо до деталізації структури ненормованої функції складності тестового завдання. Перш за все змоделюємо вплив кількості альтернатив на складність тестового завдання. Очевидно, що необмежене збільшення кількості альтернатив приведе до максимальної складності тестового завдання:

$$\lim_{N \rightarrow \infty} CS_i(\vec{F}_i, \vec{D}_i, N) = \infty. \quad (3.25)$$

Реалізувати даний постулат можна за допомогою компонування складності альтернатив в складність завдання за адитивним принципом:

$$CS_i(\vec{F}_i, \vec{D}_i, N) = \sum_{j=1}^N CS_{ij}^*(F_{ij}, D_{ij}) + INV(F_{\max}), \quad (3.26)$$

де $CS_{ij}^*(F_{ij}, D_{ij})$ - функція складності j -ої альтернативи; INV – поправка на складність у врахуванні негативного представлення основної частини тестового завдання, яке прирівнюється до появи ще однієї вірної альтернативи.

Змоделюємо поведінку функції складності альтернативи тестового завдання, що допоможе встановити її структуру. Як вже відзначалося, приналежність альтернативи до іншого підрозділу із основою одразу сигналізує про її хибність, що призводить до мінімальної складності її оцінки

$$CS_{ij}^*(F_{ij}, D_{\min}) \equiv CS_{ij}^*(F_{\min}, D_{\min}) = CS_{\min}. \quad (3.27)$$

В той же час, навіть максимальне значення композиційно близькості не приводить до високої складності при умові мінімальних значень формальної спорідненості

$$CS_{ij}^*(F_{\min}, D_{\max}) \ll CS_{\max}. \quad (3.28)$$

Лише максимальні значення по обох показниках приводять до максимізації значення функції складності альтернативи

$$CS_{ij}^*(F_{\max}, D_{\max}) = CS_{\max}. \quad (3.29)$$

Оскільки взаємодія факторів не носить лінійного характеру, то моделюємо її однією із найпростіших нелінійних залежностей – мультиплікативним механізмом:

$$CS_{ij}^*(F_{ij}, D_{ij}) = D_{ij}(\beta_1) \cdot F_{ij}(\beta_2), \quad (3.30)$$

де фактори композиційної та формальної спорідненості вимірюються на рівномірних шкалах із відповідними градаціями,

$$D_{ij}(\beta_1) = \begin{cases} 1 & \text{— якщо } \pi_{IdCh}(StA_i) \notin Ch_{i,j} \\ 1 + \beta_1 & \text{— якщо } \pi_{IdCh}(StA_i) \in Ch_{i,j} \wedge Ch_{i,j} \setminus \pi_{IdCh}(StA_i) \neq \emptyset, \\ 1 + 2\beta_1 & \text{— якщо } Ch_{i,j} \setminus \pi_{IdCh}(StA_i) = \emptyset \end{cases} \quad (3.31)$$

де $Ch_{i,j} = \pi_{IdCh}(\sigma_{IdE=Id_j}(SgA_i))$; SgA_i – множина згенерованих альтернатив;
 StA_i – множина вірних альтернатив.

$$F_{ij}(\beta_2) = \begin{cases} 1 - \text{якщо } \pi_{Pr_IdE}(\sigma_{IdE=IdA_j}(SgA_i)) \neq IdBk_i, \\ 1 + \beta_2 - \text{якщо } \forall k \pi_{IdE}(StA_{i,k}) \setminus \pi_{IdE}(\sigma_{IdE=Id_j}(SgA_i)) \neq \emptyset \wedge \\ \quad \wedge \pi_{Pr_IdE}(\sigma_{IdE=IdA_j}(SgA_i)) \neq IdBk_i, \\ 1 + 2\beta_2 - \text{якщо } \exists k \pi_{IdE}(StA_k) = \pi_{IdE}(\sigma_{IdE=Id_j}(SgA_i)) \end{cases}, \quad (3.32)$$

де IdA_j - ідентифікатор кореневого елемента j -ої альтернативи; $IdBk_i$ - ідентифікатор кінцевого вузла основи i -го тестового завдання; $StA_{i,k}$ - k -та вірна альтернатива.

Сумуючи всі наведені співвідношення, отримуємо представлення реальної складності RCS як функції градацій композиційної та формальної спорідненості альтернатив

$$RCS_i(\vec{\beta}) = \frac{\sum_{j=1}^N D_{ij}(\beta_1) F_{ij}(\beta_2) - MnCS(N)}{MCS(N)}. \quad (3.33)$$

Відносній складності повинна відповідати відносна частота неправильних відповідей:

$$REA_i = \frac{EA_i}{MEA}, \quad (3.34)$$

$$MEA = \max_{(i=1..CTt)} (EA_i), \quad (3.35)$$

де REA_i - відносна частота помилкових відповідей; EA_i - частота помилкових відповідей на i -е тестове завдання; MEA - максимальна частота помилкових відповідей; CTt - кількість тестових завдань.

Процедуру ідентифікації здійснимо на основі співвідношення

$$\sum_{i=1}^{CTt} (RCS_i(\vec{\beta}) - REA_i)^2 \xrightarrow{\vec{\beta}} \min \quad (3.36)$$

$$\beta_1, \beta_2 \geq 0$$

Ідентифікація реалізується на основі алгоритму Левенберга-Марквардта із використанням процедури *nlinfit* в програмному середовищі моделювання MatLab. При цьому застосовується метод дзеркальних відображень, що запропонований в [56], який полягає у використанні $|\beta_i|$ замість β_i в представленні (3.36).

Орієнтація методу на структурні особливості тестів дозволяє використовувати його для генерації тестів заданої складності в автоматизованих системах тестування.

3.3 Метод адаптивного управління структурою процесу навчання

Застосування тестів дозволяє мінімізувати витрати часу викладача на процес контролю правильності знань та активізувати процес її самоконтролю, які є цільовими компонентами структури процесу автоматизованого навчання (рис. 3.4). Якість самопідготовки студентів зростатиме якщо тестові завдання впорядковувати по наростанню складності. Питання оцінки складності завдань досліджувалося в роботах [13,41,108], та обґрунтовано в попередньому підрозділі даної роботи, де було запропоновано ефективний підхід до оцінки складності тестових завдань на основі мало-параметричної регресійної моделі, параметри якої ідентифікуються по невеликій вибірці тестів для конкретних студентських аудиторій та типів тестових завдань. Даний підхід дозволяє генерувати рівноцінні набори тестових завдань, а також забезпечувати адаптацію студентів до наростання їх складності.

Однак диференціювання складності завдань дозволяє не тільки нарощувати складність завдань, але й управляти тривалістю підготовки по тестах для отримання реально досяжного рівня успішності по сукупності предметів при обмеженому ресурсі часу відведеного на підготовку.

Актуальність цієї задачі впливає із потреби в забезпеченні вузької спеціалізації при достатньо високому загальному рівні підготовки та недостатньому їх висвітленню в наукових публікаціях.

Серед близьких робіт до даної тематики варто відзначити роботи [10, 51]. Однак абстрактний класифікаційний підхід [10] не дозволяє врахувати індивідуальні особливості суб'єктів навчального процесу, а формування індивідуальної траєкторії навчання в [51] базується на врахуванні великої кількості суб'єктивних оцінок важко спостережуваних факторів.

В кредино-модульній системі процес контрольного тестування рівня знань студента носить дискретний характер. Підсумкова успішність за курс є зваженою оцінкою успішності по залікових модулях, основна маса яких може бути отримана на основі тестового контролю. Залікові модулі значною мірою прив'язані до змістовних модулів, які розбивають навчальний семестр на періоди по 5-9 тижнів. Якщо тестування здійснюється по окремих темах, то часовий інтервал на їх вивчення ще більше скорочується. На протязі такого періоду раціонально можна отримати не більше 3-5 проміжних результатів тестувань. Модель залежності успішності студента від часових витрат на такій незначній вибірці може містити лише мінімальну кількість параметрів.

Нехай матеріал в навчальній системі згруповано по темах tm . При ознайомленні із матеріалом кожної теми студент s виконує навчальні тести рівнів складності k , що стимулюють процес засвоєння навчального матеріалу. Після освоєння матеріалу розділу студент складає контрольні тести по темі [36, 42]. Результат контрольного тестування можна спрогнозувати по результатах навчального тестування. Якщо цей прогноз студента не задовольняє, він інтенсифікує свої зусилля для досягнення вищої успішності в навчальному тестуванні. Зусилля студента направляються на виявлення слабкостей в засвоєнні матеріалу та їх переосмисленні. Із ростом успішності питома частка слабо засвоєного матеріалу зменшується. Однак витрати часу на надійну ідентифікацію слабких місць зростають. Крім того, для переосмислення студент повинен аналізувати частини матеріалу вищої складності. Отже із

ростом успішності витрати часу зростають. Це означає, що для моделювання залежностей успішності від часових витрат необхідно використати монотонно зростаючі залежності: степеневу, експоненціальну, логарифмічну, дробово-раціональну.

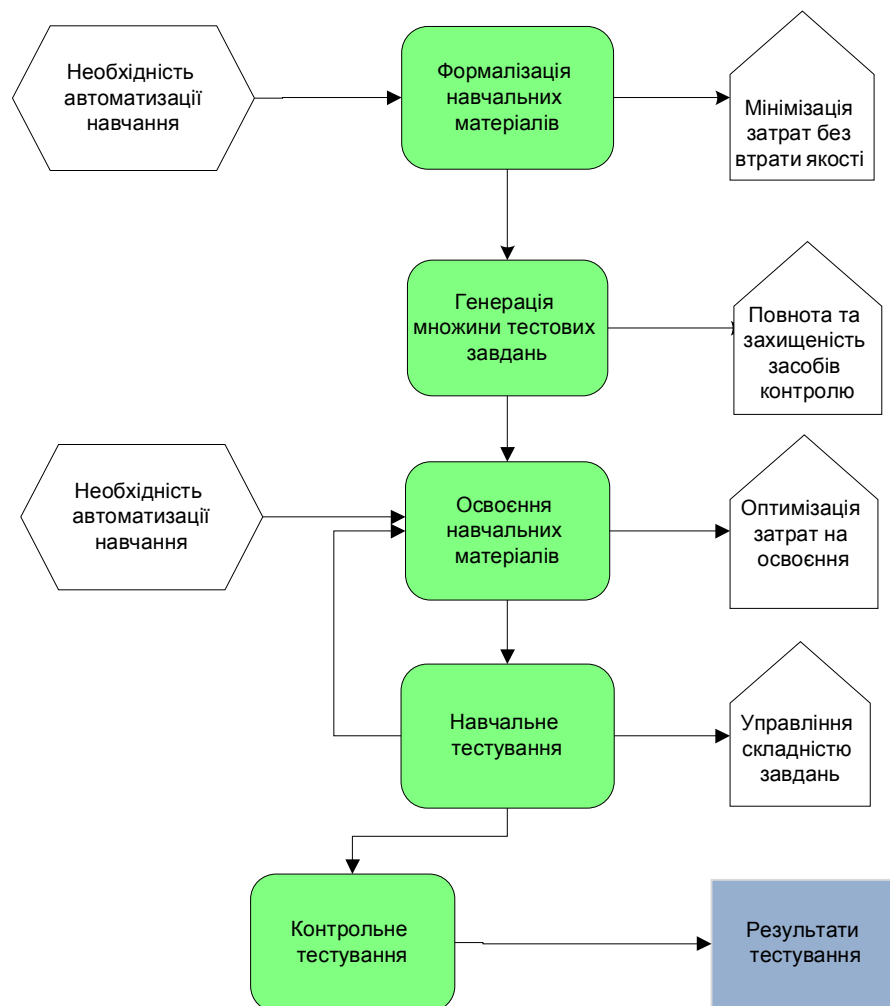


Рисунок 3.4 – Цільові компоненти процесу автоматизованого навчання

Для вибору найбільш придатної регресійної функції для моделювання залежності успішності від часових витрат були проведені експериментальні дослідження в умовах підготовки студентів до здачі модульного контролю. В таблиці 3.1 наведені показники успішності типового студента при завершенні кожного лабораторного заняття та результати її наближення відповідними функціональними залежностями. Експериментальні дані свідчать, що початковий рівень знань студента дозволив досягнути успішності на рівні 0.47, яка кожного заняття зростала і перед модульним контролем склала 0.89.

Графічне представлення побудованих залежностей наведено на рисунку 3.5.

У наведеному експерименті максимальна відносна похибка при використанні степеневій залежності майже вдвічі менша аналогічного показника для наступної за точністю залежності. Подібна картина спостерігалася при проведенні інших експериментів, тому для наближення залежності успішності від часових витрат вибрано степеневу залежність.

Таблиця 3.1 – Порівняння похибок регресійних наближень успішності студента в залежності від витрат часу

Витрати часу, у год.	Успішність	Похибки регресійних наближень			
		$U = aT^\gamma$	$U = ae^{\gamma T}$	$U = \gamma \ln(aT + 1)$	$U = \gamma \left(1 - \frac{1}{aT + 1}\right)$
4	0.47	0.0014	0.0616	-0.0155	-0.0363
6	0.55	-0.0113	0.0059	-0.0117	-0.0151
8	0.61	0.0069	-0.0021	0.0140	0.0203
10	0.67	0.0112	-0.0124	0.0210	0.0314
12	0.72	0.0064	-0.0232	0.0158	0.0265
14	0.76	-0.0060	-0.0344	0.0010	0.0093
16	0.81	-0.0125	-0.0335	-0.0092	-0.0050
18	0.83	-0.0032	-0.0109	-0.0046	-0.0056
20	0.86	-0.0029	0.0080	-0.0096	-0.0167
22	0.89	0.0100	0.0448	-0.0025	-0.0162
Максимальна відносна похибка, у %		1.4	6.9	2.4	4.1

Оскільки процес навчання носить динамічний характер, його раціонально організувати на матеріалах із наростаючою складністю. Ідентифікація може бути здійснена на основі адаптивно-структурного методу [41]. Також важливо встановити моменти часу для підвищення рівня складності пропонованих тестів, які б дозволили мінімізувати загальний час досягнення бажаної успішності. При побудові залежності, що оцінює часові витрати для досягнення заданої успішності на основі експериментальних даних можуть отримуватися співвідношення, що не є функціональними у випадку, коли із ростом часових

витрат успішність не тільки не покращується, а й погіршується. Зокрема, в цьому випадку одному значенню аргументу не відповідає єдине значення функції. Цей недолік невластивий функції прогнозованої успішності при заданих часових витратах. Тому функцію часових витрат слід будувати як обернену до функції успішності, представлені за допомогою степеневі функції.

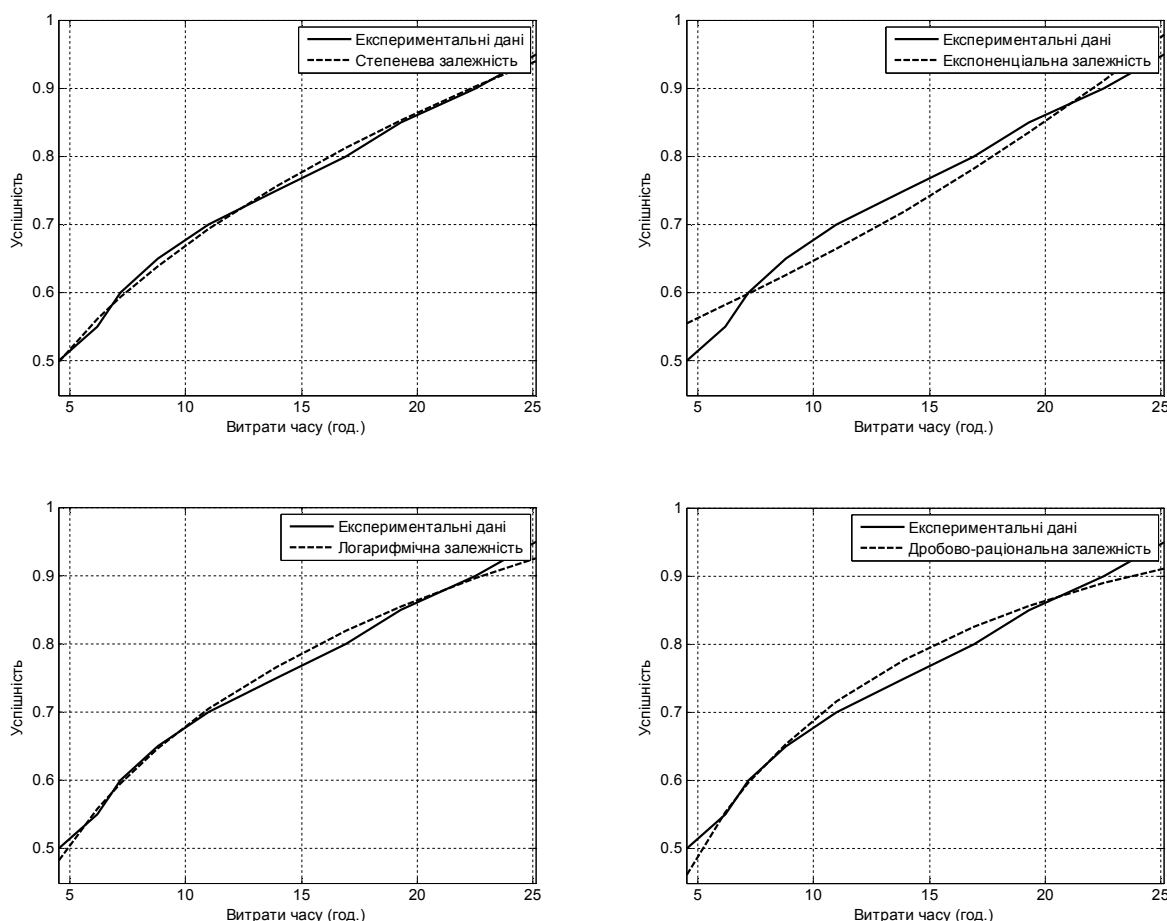


Рисунок 3.5 – Наближення функції успішності регресійними моделями

Задачу управління ефективністю навчання розглядаємо як мінімізацію часу на досягнення бажаного рівня успішності по окремій темі [36,108]. При початковому тестуванні студентів пропонуються тести всіх рівнів складності. Отримані значення успішності дозволяють оцінити її динаміку лінійною функцією по кожному рівню. При отриманні наступної точки спостереження по вибраному рівні складності переходимо до двопараметричної степеневі апроксимації:

$$U_k^{s,tm,nt}(T) = \begin{cases} \frac{U_k^{s,tm,1}}{t_k^{s,tm,nt}} T, & \text{коли } nt = 1 \\ \min \left\{ A_k^{s,tm,nt} T^{\gamma_k^{s,tm,nt}}, 1 \right\}, & \text{коли } nt > 1 \end{cases}, \quad (3.37)$$

де $U_k^{s,tm,nt}$ - функція залежності успішності від часових витрат; t - поточний час ознайомлення із навчальними матеріалами в системі; T - часові витрати; $A_k^{s,tm,nt}$, $\gamma_k^{s,tm,nt}$ - параметри регресійної функції; nt - номер навчального тестового контролю.

На основі формули (3.37) будемо функцію часових витрат, як обернену до функції успішності

$$T_k^{s,tm,nt}(U) = \begin{cases} \frac{t_k^{s,tm,nt}}{U_k^{s,tm,1}} U, & \text{коли } nt = 1 \\ \left(\frac{U}{A_k^{s,tm,nt}} \right)^{1/\gamma_k^{s,tm,nt}}, & \text{коли } nt > 1 \end{cases}, \quad 0 \leq U \leq 1, \quad (3.38)$$

де $T_k^{s,tm,nt}$ - функція часових витрат, U - бажаний рівень успішності.

Виходячи із побудови функції успішності та функції часових витрат, пропонується метод мінімізації часу навчання для забезпечення необхідного рівня успішності, що ґрунтується на формулах (3.37) та (3.38) [36,40,42,108,109,110]. Оптимізаційна задача управління ефективністю навчання, окрім критерію, включає обмеження на очікуваний рівень успішності із врахуванням часток тестових завдань відповідної складності в контрольному тесті. Вона також включає природне обмеження на рекомендовані успішності по рівнях складності тестів для конкретного студента та теми навчального матеріалу.

Складність цієї оптимізаційної задачі полягає у високій мінливості функції часових витрат в процесі освоєння знань та навчального тестування [36]. Тому для її розв'язання запропоновано покрокову процедуру, яка включає поетапну

ідентифікацію функції часових витрат при отриманні нових результатів навчального тестування. Кожен етап навчального тестування містить тести одного рівня складності. Вибір цього рівня забезпечує використання рекомендацій щодо обсягів часу навчання, побудованих на попередньому кроці. На 1-му кроці етапу прогноуються успішності $U_k^{s,tm,nt}$ по рівнях складності, які б забезпечили бажаний рівень очікуваної успішності із мінімальними витратами часу. При цьому компоненти цільової функції задають рекомендовані витрати часу на освоєння тестів по кожному рівню складності.

$$\text{Крок 1.} \begin{cases} z = \sum_k T_k^{s,tm,nt}(U_k^{s,tm,nt}) \rightarrow \min, & (3.39) \\ \sum_k c_k^{tm} \cdot U_k^{s,tm,nt} \geq U_{\min}^s, & (3.40) \\ 0 \leq U_k^{s,tm,nt} \leq 1 - U^*, & (3.41) \end{cases}$$

де c_k^{tm} - частка тестових завдань k -го рівня складності в тесті, що побудований для теми tm ; U_{\min}^s - успішність, якої прагне досягнути студент в процесі навчання при проходженні тестового контролю; U^* - рівень значущості (максимально можливу успішність понижуємо на рівень значущості, оскільки максимальної успішності при належному рівні складності тестів практично досягнути складно).

На другому кроці вибирається рівень складності k^* пропонованих тестів, на якому рекомендовані часові витрати є найменш освоєними із проведенням додаткового тестування на цьому рівні.

$$\text{Крок 2. } k^* = \underset{k}{\text{ArgMax}} \left(T_k^{s,tm,nt}(U_k^{s,tm,nt}) - t_k^{s,tm,nt} \right); \quad (3.42)$$

$$nt(k^*) := nt(k^*) + 1; \quad (3.43)$$

На третьому кроці перевіряється умова завершення навчання при поточних часових витратах.

$$\text{Крок 3. Якщо } \sum_k c_k^{im} U_k^{s,tm,nt}(t_k^{s,tm,nt}) < U_{\min}^s \text{ то Крок 1.} \quad (3.44)$$

В результаті застосування розробленої моделі встановлено, що адаптивне управління структурою навчального процесу при автоматичній генерації тестів із врахуванням їх складності забезпечує суттєву економію часових витрат без втрати ефективності засвоєння навчальних матеріалів, яка дозволяє будувати покращені оперативні навчальні стратегії в межах локальних тем або модулів.

Висновки до розділу 3

1. Формалізовано структуру задачі, в якій виділено такі основні елементи як умова задачі, вхідні змінні, результуючі змінні, які характеризуються певною множиною атрибутів, що дозволило сформувати структуру тестових завдань для перевірки практичних вмінь та навичок.

2. Розроблено метод автоматичної генерації завдань з програмованим оператором, який використовується для оцінки засвоєння вмінь та навичок. На відміну від відомих рішень запропонований метод дозволяє генерувати не тільки вхідні параметри задачі, але і забезпечує варіативність структурних елементів її умови.

3. Запропоновано критерій та метод оцінки складності тестових завдань, який дозволяє оцінити складність тестів під час автоматичної генерації. Даний метод володіє перевагами статистичних методів і методів структурного аналізу.

4. Розроблено метод адаптивного управління структурою процесу навчання, яка дозволяє ефективно управляти часом навчання для досягнення бажаного рівня засвоєння знань, виходячи з мінімізації показника часових витрат.

РОЗДІЛ 4

ІНФОРМАЦІЙНА ТЕХНОЛОГІЯ АВТОМАТИЧНОЇ ГЕНЕРАЦІЇ ТЕСТОВИХ
ЗАВДАНЬ З КЕРОВАНОЮ СКЛАДНІСТЮ

4.1 Система SAGT автоматичної генерації тестових завдань

Процес автоматичної генерації тестових завдань покликаний зменшити трудомісткість наповнення банку тестів при забезпеченні їх достатньої якості та варіативності. Оскільки технології повного семантичного аналізу навчальних матеріалів на сьогоднішній день не реалізовані, викладач повинен здійснити формалізацію положень курсу, які підлягають перевірці в рамках засобів інформаційної системи. Однак на основі цієї формалізації генерація конкретних тестових завдань здійснюється автоматично, із забезпеченням їх динамічної варіативності по змісту та формі. Динамічний характер процесу генерації передбачає спеціальну процедуру верифікації правильності відповідей студентів, що виключає можливість несанкціонованого використання ключів тестових завдань.

Повнота контролю процесу засвоєння знань включає як засоби перевірки засвоєння теоретичних основ навчального курсу, так і навичок аналізу проблемних ситуацій предметної області, а також практичних вмінь та навичок розв'язання прикладних задач. Перевірка навичок аналізу проблемних ситуацій передбачає формалізацію допустимих характеристик їх параметрів. На основі цієї формалізації відбувається динамічна генерація екземпляру проблемної ситуації із допустимої множини. Складність структури моделі проблемної ситуації дозволяє зменшувати часові витрати як на побудову варіанту конкретного завдання, так і на перевірку правильності відповіді студента, що полягає у встановленні адекватності причинно-наслідкових зв'язків між специфікацією проблемної ситуації та методами її розв'язання.

Процес перевірки рівня набутих практичних вмінь та навичок включає формалізацію певного типу задач та елементів конкретизації його варіантів в рамках конкретних програмних середовищ. На основі цієї формалізації можна здійснити автоматичну генерацію структурних елементів типової задачі, що включає специфікацію типової умови та типу, виду та кількості вхідних змінних, що не може бути реалізоване в рамках методу побудови параметризованих тестів. Окрім того дана формалізація дає принципово нову можливість перевірки вміння алгоритмізувати та запрограмувати розв'язання типових задач із динамічно генерованою структурою у відповідному програмному середовищі. При цьому правильність формування рішення розв'язку задачі здійснюється автоматично.

Організація процесу якісного навчання передбачає управління складністю тестових завдань. Це дає змогу будувати завдання із наростаючою складністю в процесі освоєння курсу, а також генерувати еквівалентні по складності тести для підсумкового контролю рівня знань в групі студентів. Запропонований в роботі адаптивно-структурний метод оцінки складності тестових завдань дозволяє реалізувати таке управління на основі формальних характеристик елементів тестового завдання із адаптацією до особливостей студентської аудиторії та стилю формалізації тестових завдань на основі невеликого обсягу попереднього тестового контролю.

Управління складністю тестових завдань дає змогу оптимізувати часові витрати студентів для досягнення бажаних рівнів успішності з конкретних дисциплін. Ця можливість може бути реалізована за допомогою методу адаптивного управління структурою процесу навчання, який запропоновано в даній роботі. Це дає змогу диференціювати зусилля студента по формуванню профілю власної підготовки, адекватного його майбутній професійній діяльності.

Усі запропоновані підходи доцільно покласти в основу нової інформаційної технології автоматичної генерації тестових завдань з керованою складністю, яка дозволяє мінімізувати часові витрати як викладачів так і

студентів в процесі як традиційного, так і дистанційного навчання. Цей підхід реалізовано в рамках системи SAGT (System of the automatic generation of tests), яка створена засобами технологій PHP та MySQL з елементами JavaScript [35,43].

На рисунку 4.1 зображено головна сторінка системи, яка включає функції представлення доступних навчальних матеріалів, їх пошуку, перехід на сторінку організації тестового контролю, формалізації пропозиції щодо формування контенту тестових завдань. Також реалізовано функції зворотного зв'язку, перегляду змісту навчальної дисципліни, авторизації користувачів системи.

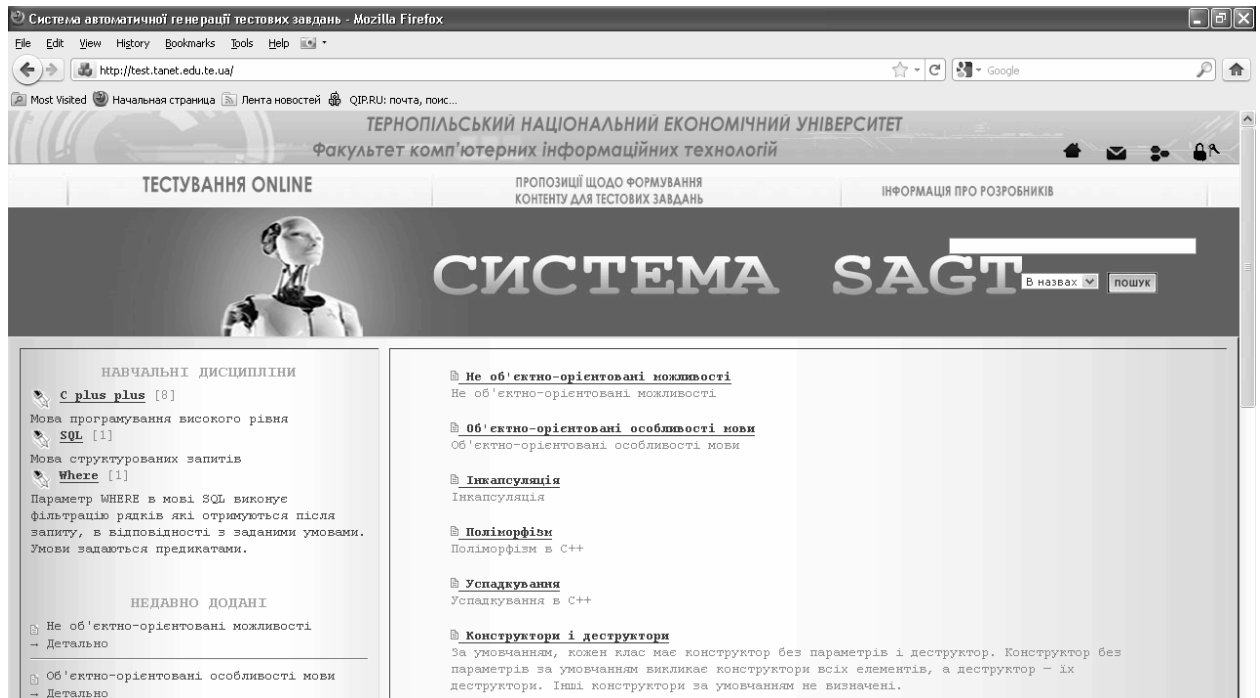


Рисунок 4.1 – Головна сторінка системи SAGT

На рисунку 4.2 представлено схему бази даних системи SAGT. Розглянемо детальніше основні відношення, які безпосередньо використовуються для зберегання інформації, що застосовується для генерації тестових завдань.

Відношення “definition”, “conjunctive”, “conjunctive_type” містять інформацію про структуру текстових тверджень, зв'язки та типи зв'язків між окремими компонентами, ідентифікатори семантичних та формальних класів, належність твердження до конкретної теми, розділу. Відношення “method”, “envmethod”,

“sрesmethod” містять інформацію про структуру методів, які використовуються при автоматичній генерації задач, назву методу, операції та їх програмні інтерпретації. Відношення “problem”, “prbarea”, “atrbprb”, “fmngenprob” містять інформацію про структуру проблемної ситуації, середовище її виникнення, атрибути та умови перебігу. Відношення “articles”, “articles_cat” містять інформацію про теми навчальних матеріалів, їх належність до конкретних дисциплін, дату створення та редагування. Також у відповідних відношеннях бази даних зберігається інформація про осіб, які навчаються, тестуються, а також результати їх роботи.

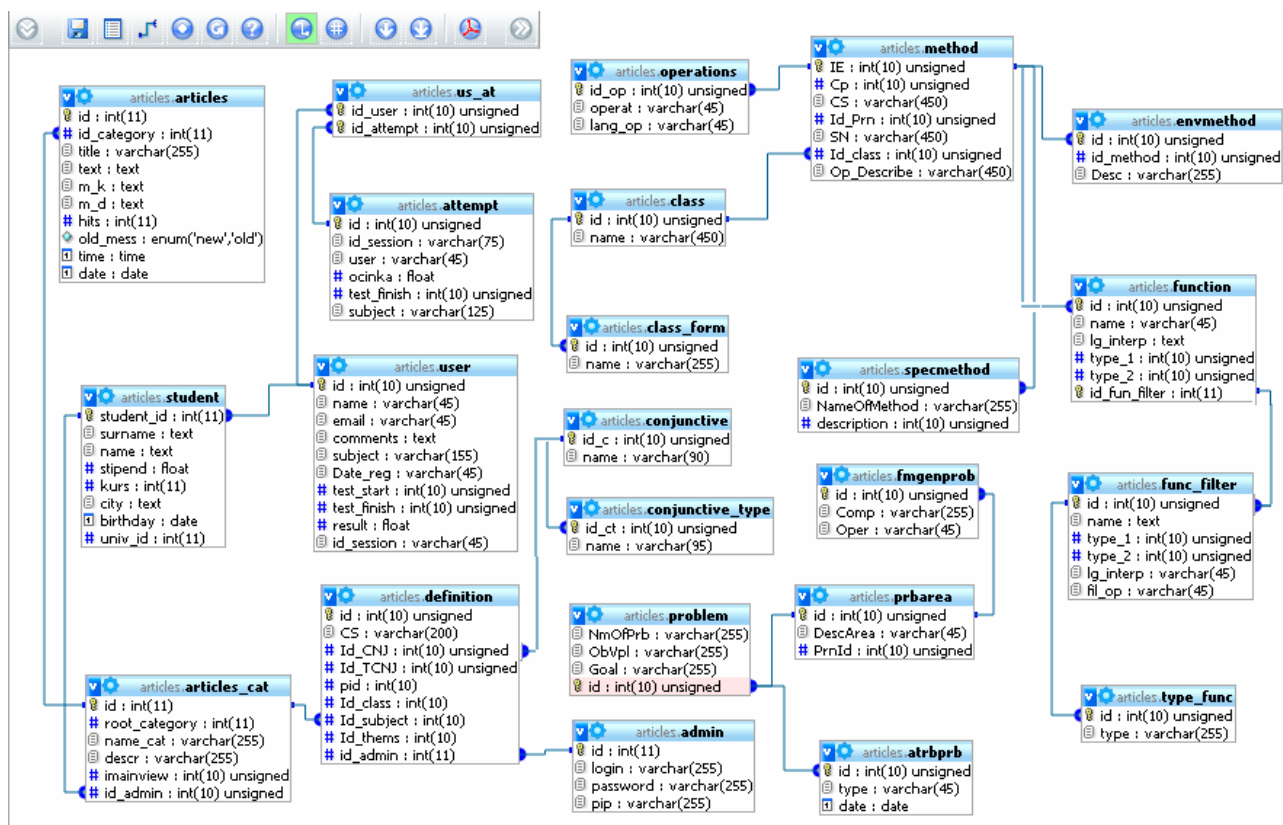


Рисунок 4.2 – Схема бази даних системи SAGT

Програмне забезпечення складається з наступних модулів (рис. 4.3): модуль головної сторінки системи, модуль представлення навчальних матеріалів, модуль зворотного зв’язку, модуль проведення тестового контролю, модуль генерації тестових завдань закритого типу, модуль генерації завдань для

перевірки методологічних знань, модуль генерації задач з програмованим оператором, модуль авторизації, модуль формалізації навчальних матеріалів, модуль дерева формалізованих матеріалів, модуля управління навчальним контентом.

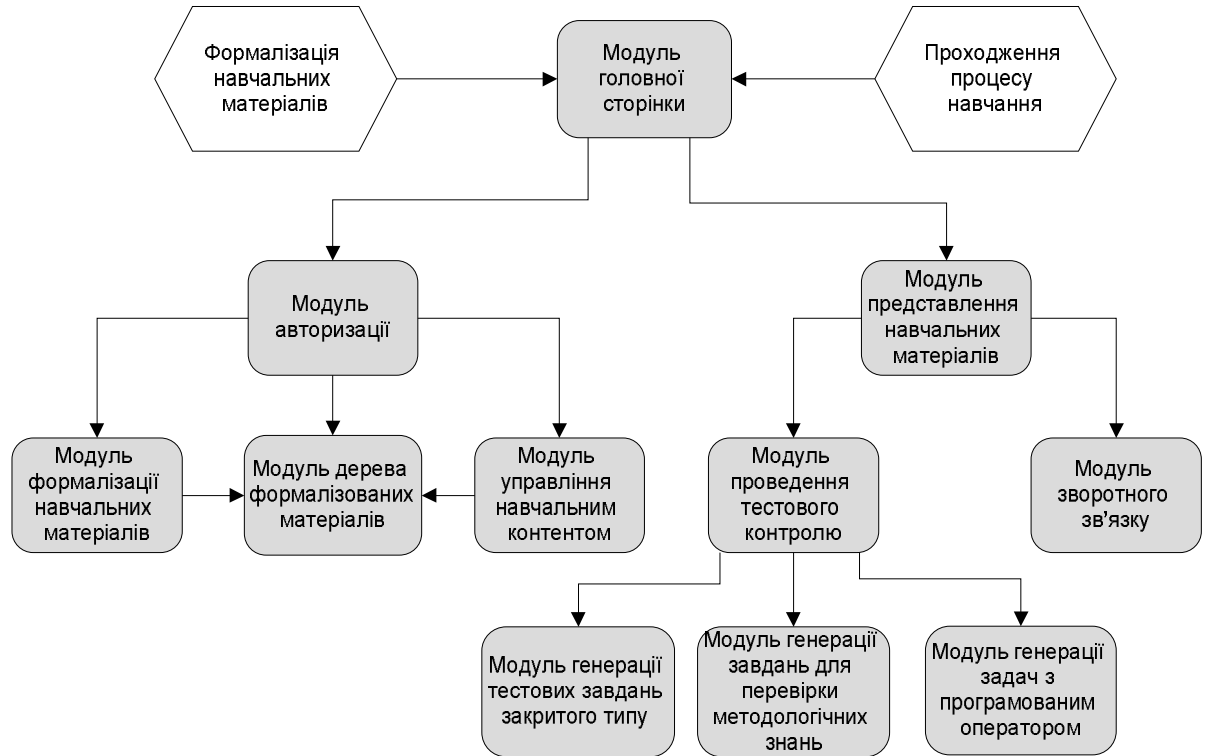


Рисунок 4.3 – Структура модулів системи SAGT

Модуль головної сторінки системи дозволяє організувати навігацію по структурі інформативних матеріалів з метою ознайомлення з навчальним контентом. За допомогою даного модуля також організовується взаємодія з іншими модулями системи. Модуль представлення навчальних матеріалів надає можливості представлення доступних навчальних матеріалів, їх пошуку. Модуль зворотного зв'язку виконує функції взаємодії студентів із викладачами, дозволяє вносити пропозиції щодо формування структури навчальних контентів, а також організувати процес надання online-консультацій.

Модуль проведення тестового контролю реалізовує основні функції процесу тестування. В ньому є можливості по формуванню структури тестових завдань, виходячи із їх кількості, типів та видів, управління складністю завдань,

а також рекомендації щодо вивчення тем на основі результатів тестування. Модуль автоматичної генерації тестових завдань закритого типу дозволяє генерувати завдання із забезпеченням семантичної, формальної та синтаксичної узгодженості (див. рис. 4.4.). Він забезпечує автоматичну побудову завдань п'яти видів: множинний вибір, альтернативні, заповнення прогалін у тексті, на встановлення відповідності, завдання перестановки (відновлення послідовності). Лістинг основних процедур даного модуля наведено в додатку В.

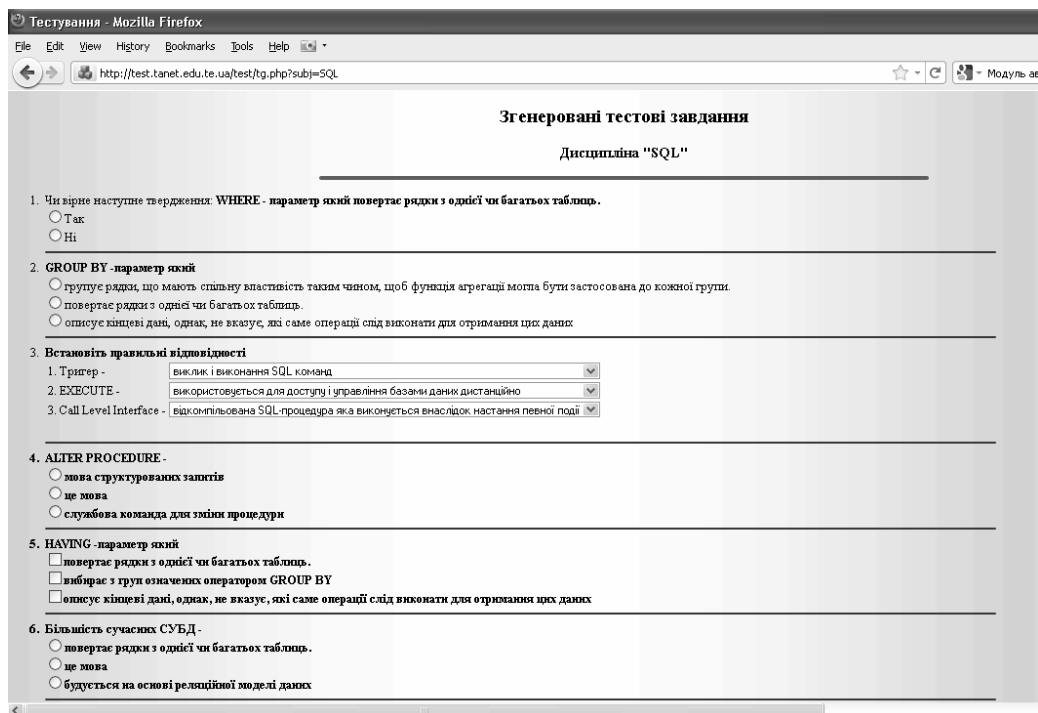


Рисунок 4.4 – Автоматична генерація тестових завдань закритого типу

Модуль автоматичної генерації тестових завдань для перевірки методологічних знань – реалізовує основні процедури методу, описаного у підрозділі 2.3. Модуль генерації задач з програмованим оператором дозволяє автоматично генерувати задачі, проводити їх автоматичне опрацювання та перевірку на основі взаємодії на рівні інтерпретації реалізованих команд з програмним інтерпретатором.

Модуль авторизації виконує функції ідентифікації користувачів у системі, з метою розмежування їх функціональних ролей, доступу до власних інформаційних ресурсів, управління процесом тестування по дисциплінах.

Модуль управління навчальним контентом використовується для додавання нових дисциплін, тем, підтем, формування їх структури. Тут реалізовано функцію відображення або приховування матеріалів для студентів в процесі навчання (див. додаток Г).

Модуль формалізації навчальних матеріалів слугує засобом побудови інформаційної бази для автоматичної генерації тестових завдань. Тут реалізовано усі властивості функціональних атрибутів, які описані в підрозділах 2.2, 2.3, 3.1. На рисунку 4.5. відображено графічний інтерфейс модуля формалізації навчальних матеріалів, яка включає вибір навчальних дисциплін, їх відповідних тем, автоматичне розбиття конспекту по реченнях, а також форму для формалізації відповідних тверджень. В межах даного модуля також реалізовано функцію автоматичного опрацювання текстових матеріалів навчальних дисциплін, яка дозволяє значно спростити процес формалізації навчальних матеріалів, що сприяє суттєвому зменшенню витрат часу на побудову тестових завдань (див. додаток Д).

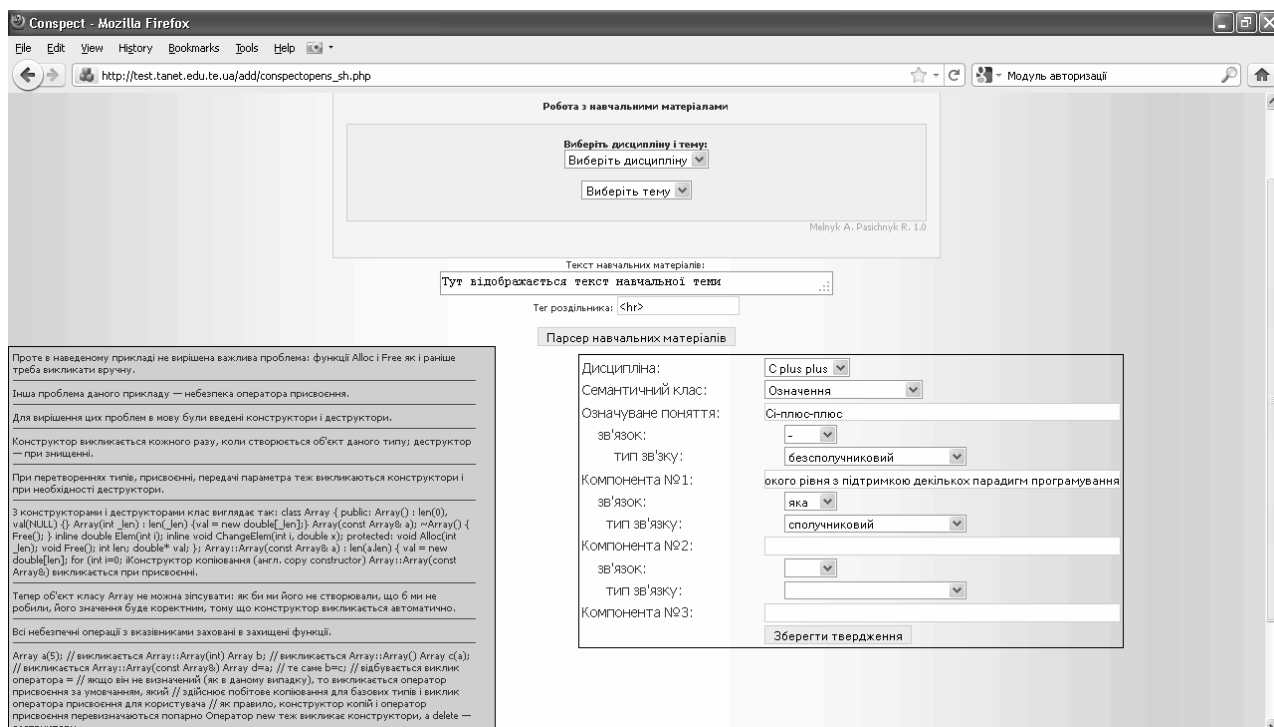


Рисунок 4.5 – Графічний інтерфейс модуля формалізації навчальних матеріалів

Модуль дерева формалізованих матеріалів розроблений з метою відображення інформаційного дерева, дозволяє переглянути структуру формалізованих представлень структур навчального конспекту з вибраної дисципліни. В ньому реалізовано функції, що підтримують тиражування формалізованих тверджень при модифікації структури навчальної дисципліни, або створення нових дисциплін з використанням елементів раніше напрацьованих формалізацій (рис. 4.6).

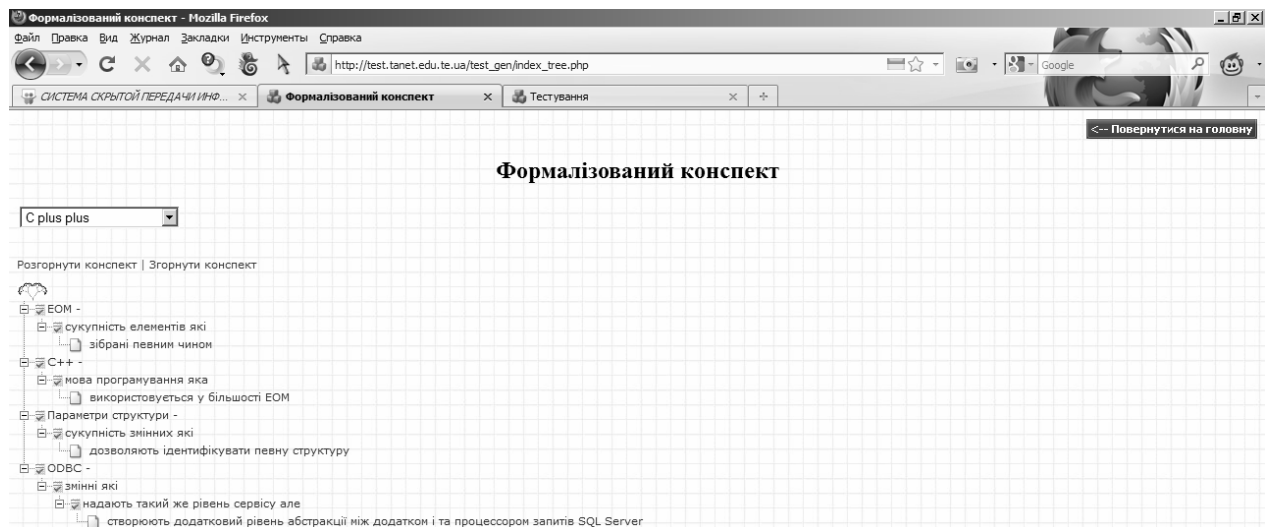


Рисунок 4.6 – Графічний інтерфейс модуля дерева формалізованих матеріалів

На рисунку 4.7 зображено графічний інтерфейс функції опрацювання результатів тестування та рекомендації щодо готовності до тестового контролю. Він включає форму представлення результатів тестування з деталізацією по кожному тестовому завданні. Наводиться загальна тривалість тестування, загальна кількість отриманих балів та загальні рекомендації щодо готовності до проходження підсумкового тестового контролю.

Таким чином система SAGT дозволяє реалізувати основні етапи процесу навчання, починаючи від формування структури навчального контенту, завершуючи процедурою тестового контролю. У системі чітко розділено функціональні можливості викладача та студента, що забезпечує

конфіденційність внесеної інформації. Студентові надається можливість отримання консультацій у будь-якого зареєстрованого викладача. Система дозволяє автоматично генерувати тестові завдання для перевірки теоретичних та методологічних знань, а також задачі для контролю вмінь і навичок.

Результати тестування	
З дисципліни SQL	
Тестове питання	Результат
1	хибна відповідь
2	хибна відповідь
3	відповідь правильна частково
4	хибна відповідь
5	правильна відповідь
6	хибна відповідь
7	правильна відповідь
8	хибна відповідь
Тривалість тестування	41 хв. 20 сек.
Загальний результат	25

Загальні рекомендації: Ви не готові до тестового контролю

Рисунок 4.7 – Сторінка відображення результатів тестування

4.2 Експериментальні дослідження методів автоматичної генерації тестових завдань закритого типу

На основі розроблених методів [38, 39], які описані у попередніх підрозділах, та програмної реалізації системи автоматичної генерації тестових завдань розглянемо згенеровані приклади завдань різних видів та проаналізуємо їх особливості.

На рисунках 4.8-4.9 представлено приклади автоматично згенерованих тестових завдань виду «множинний вибір». На першому рисунку альтернативи побудовані на основі текстових представлень, а на другому у формі програмного коду. На рисунку 4.8 штрих-пунктирною лінією виділено змінні компоненти тверджень та відповідні зв'язки між ними за допомогою яких

проводиться формування альтернатив, що формально та синтаксично узгоджені між собою.

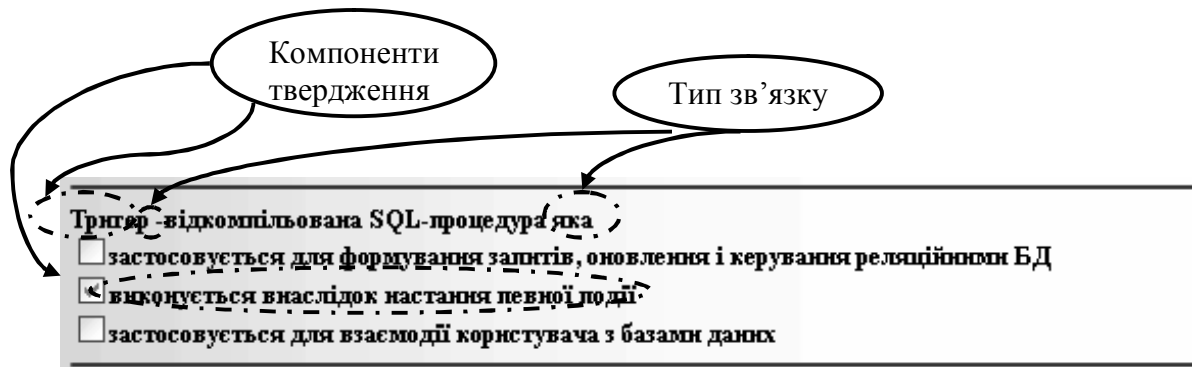


Рисунок 4.8 – Завдання множинного виду з дисципліни «Основи SQL»

На рисунку 4.9 альтернативи завдання будуються виходячи із умови приналежності компонент до формального класу, який реалізовується на основі представлень у формі програмного коду. В цьому випадку базове твердження включало текстовий опис, який доповнений фрагментом програмного коду.

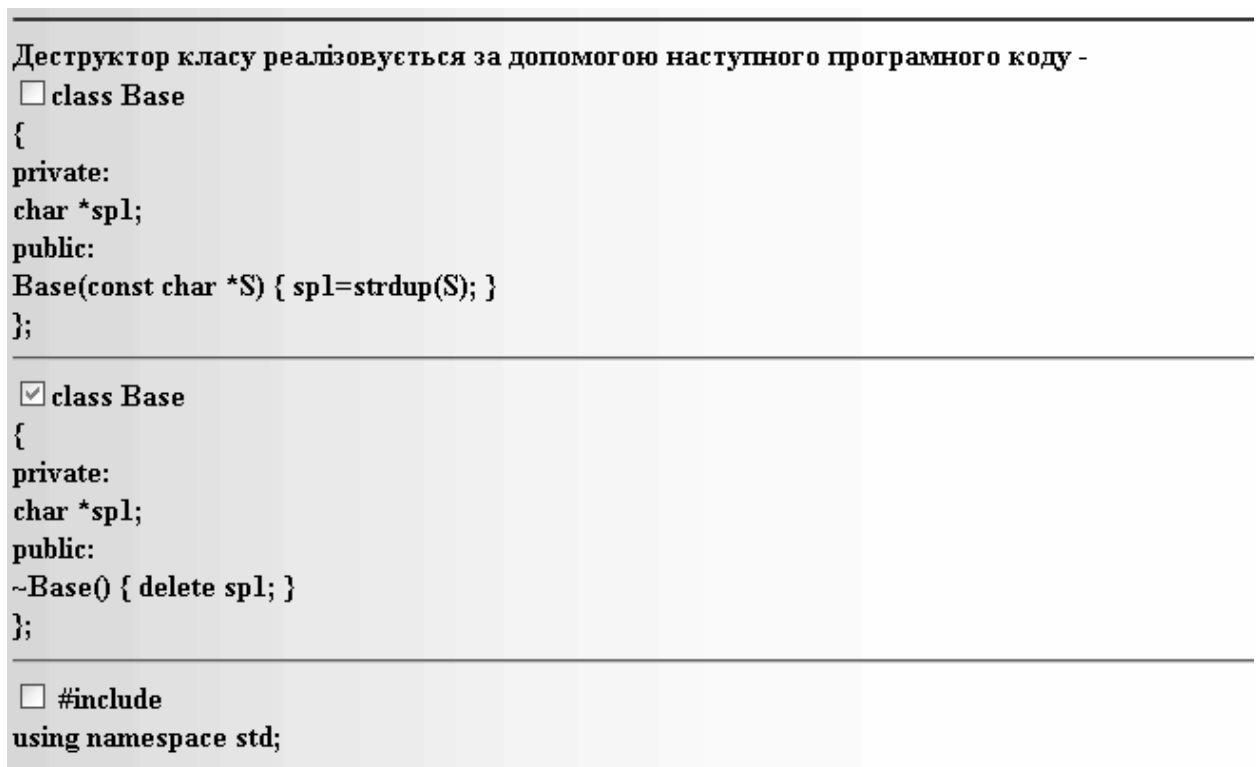


Рисунок 4.9 – Завдання множинного виду з дисципліни «Основи C++» із компонентою представлення у формі програмного коду

Така структура формалізації передбачає його декомпозицію на основу та альтернативу на межі зміни формальних типів. Генерації хибних альтернатив здійснюється на основі спорідненості формальних класів з вірною альтернативою. Це дає змогу урізноманітнити представлення завдань, а також проводити перевірку засвоєння та вміння розпізнавати реалізацію конкретних об'єктів відповідними програмними інструкціями.

На рисунках 4.10–4.11 показано експериментальні приклади завдань альтернативного виду з дисципліни «Основи С++». На рисунку 4.9 показано завдання, де всі компоненти тверджень сформовані у формі текстового представлення, а на рисунку 4.11 є змінна компонента у формі числового представлення. Побудова тестових завдань даного виду відбувається на основі опрацювання одного формалізованого твердження. Якщо його модифікація не міняє його істинності, то вірною відповіддю вважається ствердна альтернатива, в іншому випадку – заперечна альтернатива. Як і у випадку множинного вибору істинність модифікованого твердження встановлюється автоматично. Наявність численних компонентів формалізованого твердження забезпечує достатню варіативність тестових завдань, що можуть ним породжуватися.

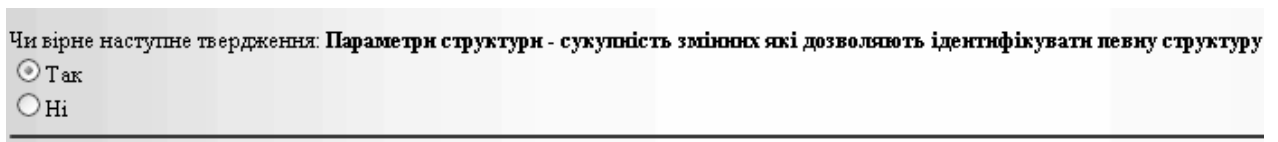


Рисунок 4.10– Завдання альтернативного виду з дисципліни «Основи С++»

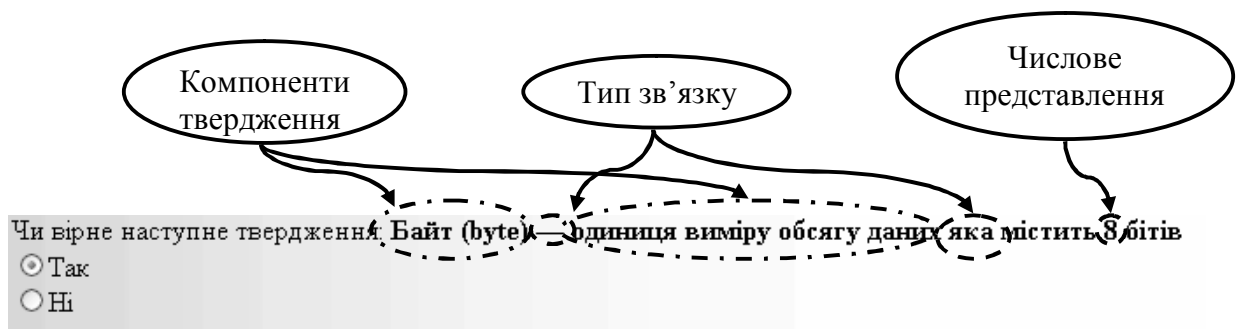


Рисунок 4.11 – Завдання альтернативного виду з дисципліни «Основи С++» із компонентою числового типу

Особливим видом тестових завдань є «заповнення прогалин у твердженні». Складність автоматичної генерації такого завдання полягає в необхідності формальної, синтаксичної та семантичної узгодженості сусідніх компонент до «прогалини», при умові наявності їх зв'язку із нею. При відсутності синтаксичних зв'язків, узгодженість контролюється лише по семантичних та формальних критеріях, що дещо спрощує процедуру генерації. Цей випадок представлено на рисунку 4.12, де подано завдання з дисципліни «Основи SQL». Тут «прогалина» належить до семантичного класу «Схема методу» та формального класу «Програмний код». Однорідність альтернатив забезпечується врахуванням цих ознак.

Відновіть твердження: `SELECT * FROM TABLE ... id=pid;`
 GROUP BY
 ORDER BY
 WHERE

Рисунок 4.12 – Завдання виду «заповнення прогалин у тексті» з дисципліни «Основи SQL» із компонентою представлення у формі схеми методу

На рисунку 4.13 завдання, де всі компоненти описуються текстовою формою представлення. Генерація завдання для даного прикладу відбувається із врахуванням всіх типів узгодженості, що призводить до часткового обмеження варіативності тестового завдання.

Відновіть твердження: **Вказівник - ... , який використовується в комп'ютерних мовах програмування**
 елемент масиву
 тип даних
 оператор присвоєння

Рисунок 4.13 – Завдання виду «заповнення прогалин у тексті» з дисципліни «Основи C++» із компонентою представлення у формі тексту

Завдання на встановлення правильної відповідності будуються на основі методу, який описаний в підрозділі 2.2. На рисунках 4.14 та 4.15 згенеровані

завдання відповідного виду, рисунок 4.15 – відображає завдання, де альтернативи описані у формі компоненти типу дата, які дозволяють проводити перевірку знань конкретних подій. Текстове представлення альтернатив вимагає множинної узгодженості всіх компонент лівого списку із всіма компонентами правого списку із врахуванням усіх критеріїв узгодженості. Якщо альтернативи не належать до текстового класу, то необхідність у синтаксичній узгодженості відпадає (рис. 4.15).

Встановіть правильні відповідності

1. Параметри структури -	сукупність елементів які зібрані певним чином
2. EOM -	мова програмування яка використовується у більшості EOM
3. C++ -	сукупність змінних які дозволяють ідентифікувати певну структуру

Рисунок 4.14 – Завдання на встановлення відповідності з дисципліни «Основи C++»

Встановіть правильні відповідності

1. Страуструп перейменував мову у C++ у	1983
2. Стандарт ISO C++ ратифіковано у	1998
3. Перший випуск Borland C++ відбувся у	1990

Рисунок 4.15 – Завдання на встановлення відповідності з дисципліни «Основи C++» із компонентою типу дата

Наведені експериментальні дослідження дають можливість зробити висновок, що автоматично згенеровані завдання на основі запропонованих у роботі методів, мають достатню варіативність, семантично, формально, та синтаксично узгоджені, а також дозволяють оцінювати знання конкретних понять, фактів, їх хронологічних представлень.

4.3 Експериментальні дослідження методів автоматичної генерації тестових завдань для перевірки засвоєння методологічних знань, вмінь та навичок

В даному підрозділі розглядаються засоби контролю навичок аналізу проблемних ситуацій предметної області, а також практичних вмінь та навичок розв'язання прикладних задач. Методи контролю засвоєння методологічних знань є новим видом тестування, що має суттєвий потенціал розвитку. Реальні проблемні ситуації як правило породжують багатокрокові процедури їх розв'язання. Запропонований в роботі метод розглядає лише однокрокові процедури, хоча може бути узагальнений на багатокроковий випадок [37]. Цей підхід було практично реалізовано окремим модулем в системі SAGT.

На рисунку 4.16 представлено автоматично згенерований тест для представлення і розв'язання проблемних ситуацій в сфері «Операційні системи».

Проблемна ситуація : Збій системи в результаті суттєвих помилок, який трапився на Windows XP Professional 21.12.2010	
Необхідний результат вирішення проблеми: Відновлення системи без перестановлення	
Виберіть найефективніший метод розв'язання описаної вище проблеми та ідентифікуйте твердження, які стосуються цього методу.	
Проблема супроводжується наступними умовами:	Перелік методів розв'язання проблеми:
- відсутній установочний диск Windows XP Professional.	<input type="checkbox"/> 1. Перевірка цілісності системних файлів Твердження: <input type="checkbox"/> 1 <input type="checkbox"/> 2 <input type="checkbox"/> 3 <input type="checkbox"/> 4 <input type="checkbox"/> 5
- немає доступу адміністратора.	<input type="checkbox"/> 2. Відновлення за допомогою механізму System Restore Твердження: <input type="checkbox"/> 1 <input type="checkbox"/> 2 <input type="checkbox"/> 3 <input type="checkbox"/> 4 <input type="checkbox"/> 5
- створений образ операційної системи є збережений на жорсткому диску.	<input type="checkbox"/> 3. Метод останньої відомої конфігурації Твердження: <input type="checkbox"/> 1 <input type="checkbox"/> 2 <input type="checkbox"/> 3 <input type="checkbox"/> 4 <input type="checkbox"/> 5
- відсутня можливість вибору параметрів загрузки ОС.	<input type="checkbox"/> 4. Використання Recovery Console Твердження: <input type="checkbox"/> 1 <input type="checkbox"/> 2 <input type="checkbox"/> 3 <input type="checkbox"/> 4 <input type="checkbox"/> 5
- є загрузочний диск Norton Ghost.	<input type="checkbox"/> 5. Установка системи Windows XP Professional в режимі відновлення Твердження: <input type="checkbox"/> 1 <input type="checkbox"/> 2 <input type="checkbox"/> 3 <input type="checkbox"/> 4 <input type="checkbox"/> 5
	<input checked="" type="checkbox"/> 6. Відновлення із раніше створеного образу Твердження: <input type="checkbox"/> 1 <input type="checkbox"/> 2 <input checked="" type="checkbox"/> 3 <input type="checkbox"/> 4 <input checked="" type="checkbox"/> 5
Перелік тверджень:	
1. Консоль відновлення дозволяє вимкнути і відновити служби, форматувати диски, читати і записувати дані на пом'якшій диск.	
2. sfc /scannow здійснюється запуском із командної стрічки	
3. Acronis True Image - програма для створення образів дисків.	
4. Acronis True Image - програма для перевірки системи на віруси.	
5. Створення повністю налаштованого образу операційної системи з встановленими додатками часто буває необхідно в корпоративному середовищі.	
<input type="button" value="Завершити тестування"/>	

Рисунок 4.16 – Приклад згенерованого тестового завдання для розв'язання проблемної ситуації «Збій системи в результаті суттєвих помилок, який трапився на Windows XP Professional 21.12.2010»

Розглянемо детальніше даний приклад. Тут проблемна ситуація описується твердженням $P_i =$ "Збій системи Windows XP Professional в результаті суттєвих помилок". Змінним атрибутом для даного прикладу є *дата*, тобто $Attr = \{дата, 21.12.2010\}$. Результат вирішення проблеми $Pr = \{Відновлення системи без переустановлення\}$.

Загальна множина методів, які задаються в тестовому завданні, як альтернативні варіанти $Pmt = \{1. Перевірка цілісності системних файлів; 2. Відновлення за допомогою механізму System Restore; ... 6. Відновлення із раніше створеного образу\}$ відповідно до проблемної ситуації генеруються автоматично. Із цієї множини випадковим чином вибирається ефективний метод розв'язку, у даному випадку це буде метод – 6. Відновлення із раніше створеного образу (рис. 4.17).

Перелік методів розв'язання проблеми:	
<input type="checkbox"/> 1. Перевірка цілісності системних файлів Твердження: <input type="checkbox"/> 1 <input type="checkbox"/> 2 <input type="checkbox"/> 3 <input type="checkbox"/> 4 <input type="checkbox"/> 5	
<input type="checkbox"/> 2. Відновлення за допомогою механізму System Restore Твердження: <input type="checkbox"/> 1 <input type="checkbox"/> 2 <input type="checkbox"/> 3 <input type="checkbox"/> 4 <input type="checkbox"/> 5	
<input type="checkbox"/> 3. Метод останньої вдалої конфігурації Твердження: <input type="checkbox"/> 1 <input type="checkbox"/> 2 <input type="checkbox"/> 3 <input type="checkbox"/> 4 <input type="checkbox"/> 5	
<input type="checkbox"/> 4. Використання Recovery Console Твердження: <input type="checkbox"/> 1 <input type="checkbox"/> 2 <input type="checkbox"/> 3 <input type="checkbox"/> 4 <input type="checkbox"/> 5	
<input type="checkbox"/> 5. Установка системи Windows XP Professional в режимі відновлення Твердження: <input type="checkbox"/> 1 <input type="checkbox"/> 2 <input type="checkbox"/> 3 <input type="checkbox"/> 4 <input type="checkbox"/> 5	
<input checked="" type="checkbox"/> 6. Відновлення із раніше створеного образу Твердження: <input type="checkbox"/> 1 <input type="checkbox"/> 2 <input checked="" type="checkbox"/> 3 <input type="checkbox"/> 4 <input checked="" type="checkbox"/> 5	

Рисунок 4.17 – Приклад переліку методів розв'язання проблеми

Ефективність вибраного методу забезпечується за допомогою умов, що ідентифікують його застосовність. Інші методи не можуть бути застосовані, оскільки не виконуються відповідні умови. Така конфігурація середовища генерується автоматично в умовах їх перебігу за допомогою модифікації їх предикатних компонентів. В даному випадку середовище проблемної ситуацій,

сформовано в наступній конфігурації умов її встановлення та перебігу $Pc = \{ \text{відсутній установочний диск Windows XP Professional}; \text{ немає доступу адміністратора}; \text{ створений образ операційної системи є збережений на жорсткому диску}; \text{ відсутня можливість вибору параметрів загрузки ОС}; \text{ є загрузочний диск Norton Ghost} \}$, де предикатні компоненти, які виділені підкреслюючою лінією згенеровані автоматично. Умова «відсутній установочний диск Windows XP Professional» виключає методи 2 і 5, умова «немає доступу адміністратора» виключає методи 1 та 4, умова «відсутня можливість вибору параметрів загрузки ОС» виключає методи 3 та 4. Таким чином лише метод 6 не виключається згаданими умовами. В той же час він підтримується умовами «створений образ операційної системи є збережений на жорсткому диску» та «є загрузочний диск Norton Ghost», що ідентифікує його як єдиний ефективний метод.

Перед студентом ставиться задача вибору методу розв'язання даної проблемної ситуації за допомогою самостійного аналізу умов задачі по аналогії з вище наведеною схемою. Для уникнення вгадування правильної альтернативи необхідно крім встановлення ефективного методу вибрати твердження, які його однозначно ідентифікують або доповнюють. Для даного прикладу із множини тверджень методу $Pmt = \{ 1. \text{ Консоль відновлення дозволяє включати і відключати служби, формувати диски, читати і записувати дані на локальний диск}; \dots 5. \text{ Створення повністю налаштованого образу операційної системи з встановленими додатками часто буває необхідно в корпоративному середовищі.} \}$ вибираємо твердженнями, які однозначно визначають вірні твердження 3 та 5. Генерація хибних тверджень може відбуватися на основі методів автоматичної генерації тестових завдань альтернативного виду (рис. 4.18).

Складністю тестових завдань даного виду можна керувати за допомогою кількості умов середовища проблемної ситуації, кількості методів розв'язання проблеми, та врахування складності оцінки істинності тверджень, що доповнюють ефективний метод.

Перелік тверджень:
1. Консоль відновлення дозволяє вимкнути і відключити служби, форматувати диски, читати і записувати дані на локальний диск.
2. <code>sfc /scannow</code> здійснюється запуском із командної стрічки
3. Acronis True Image - програма для створення образів дисків.
4. Acronis True Image - програма для перевірки системи на віруси.
5. Створення повністю налаштованого образу операційної системи з встановленням додатками часто буває необхідно в корпоративному середовищі.

Рисунок 4.18 – Приклад переліку методів розв’язання проблеми

Як видно з наведеного прикладу така форма тестового завдання дозволяє оцінювати вміння знаходити правильне розв’язання проблемної ситуації, виходячи із умов її перебігу. Автоматично згенеровані параметри такого тестового шаблону дозволяють одержувати необхідну кількість наборів проблемних ситуацій без рутинної процедури знаходження конфігурації середовища проблемної ситуації, що підтримує ефективність методу її розв’язання.

Поряд із апробацією тестових завдань для перевірки методологічних знань було проведено ряд експериментів, пов’язаних із автоматичною генерацією задач з програмованим оператором для перевірки вмінь та навичок. На рисунку 4.19 продемонстровано згенероване практичне завдання з дисципліни «Основи Web-програмування» в середовищі системи програмування PHP версії 5.3.1 та метод її розв’язання, який пропонує студент у ході вирішення поставленого завдання.

Задано наступне завдання, необхідно написати програму для його вирішення

1. Знайти кількість елементів масиву `$mass = [0 -1 0 3 8 15 24 35 48 63 80 99 120 143 168]` серед додатніх а результат записати у змінну `resultat`

Вікно для написання програми:

Рисунок 4.19 – Приклад згенерованої задачі для PHP мови програмування

У даному прикладі умова задачі складається із множини представлень її основного опису, тобто «Задано наступне завдання, необхідно написати програму для його вирішення: Знайти кількість елементів масиву». Вхідні дані для умови задачі представлено масивом цілих чисел, які генеруються випадковим чином із множини визначеної розмірності $\$mass = [0-103815243548638099120143168]$. Уточнення умови задачі, виходячи із основної умови та типу і значень множини вхідних змінних, генеруються автоматично на основі формалізації специфікації типової задачі: «серед додатних».

Після генерування умови задачі, студентів пропонується її розв'язати, використовуючи інструкції програмного інтерпретатора. Відзначимо, що контроль основних синтаксичних помилок, які можуть виникати в процесі побудови програми-розв'язку, проводиться системою на основі взаємодії з програмним інтерпретатором. Після виправлення синтаксичних помилок, система дає запит на відправку готового рішення для перевірки і після того студентів повертається результат перевірки його відповіді. На рисунку 4.20 наведено процедуру розв'язання задачі згенерованої системою.

Вікно для написання програми:

```
<?php
$c=' $var >0';
// функція фільтрування
// позитивних елементів масиву
function plus($var)
{
    return ('$c');
}
// ініціалізація масиву
$mass = array (0, -1, 0, 3, 8, 15, 24, 35, 48, 63, 80, 99, 120, 143, 168);
// побудова масиву додатніх елементів
$mass_res =array_filter($mass, "plus");
// знаходження кількості додатніх елементів масиву
$result=count($mass_res);
echo "Кількість додатніх елементів мавису складає ", $result;
?>
```

Перевірити результат

Рисунок 4.20 – Процедура розв'язання задачі згенерованої системою

Ця процедура формується поетапно. На першому етапі в спеціальний системний файл записується процедура (function plus) , що відповідає уточненню умови задачі. На другому етапі система генерує контрольний масив для перевірки правильності програми студента (\$mass) і дописує його в системний файл. На третьому етапі цей масив обробляється згідно уточнення умови задачі. На наступному етапі отриманий масив опрацьовується типовою процедурою (count) розв’язання задачі, а результат записується в змінну \$result.

На рисунку 4.21 показано згенеровану задачу з дисципліни SQL, де в ролі інтерпретатора виступає SQL інтерпретатор, який функціонально реалізований в СКБД MySQL 5.1.41. В даному прикладі умова задачі та її доповнення генеруються автоматично і направлені на побудову запиту на вибірку із бази даних. Множиною вхідних змінних виступає таблиця бази даних, розмірність полів та її наповнення динамічно змінюються в процесі побудови нових завдань.

Задано наступну інформацію, де сірим кольором виділено назви полів таблиці student

Прізвище	Ім'я	Курс	Стипендія
surname	name	kurs	stipend
Котик	Борис	2	0
Микитюк	Іван	1	150
Козак	Петро	3	200
Киба	Павло	5	150
Подольська	Ольга	2	2502
Болтик	Вадим	5	250
Климчук	Артем	3	200
Петрик	Антон	4	200
Павлик	Андрій	3	0
Федоришин	Вадим	4	150

Необхідно знайти: Прізвище, ім'я студентів, за умови що стипендія більше або рівно 157

Вікно побудови запиту:

```
SELECT surname,name FROM student WHERE stipend>=157
```

Згенерувати відповідь

Рисунок 4.21 – Приклад згенерованої задачі для SQL мови та розв’язку її студентом в конкретному середовищі

Після виконання даного завдання генерується вірний результат, який отримується системою та проводиться його порівняння із результатом, сформованим в процесі виконання запиту, що побудований студентом.

Перевагою запропонованих методів над відомими рішеннями є те, що крім автоматичної генерації множини вхідних змінних, автоматично генерується умова задачі із її уточненнями. Це дозволяє отримувати достатньо варіативні

набори задач для практичної перевірки вмінь та навичок на основі поєднання базових типових процедур та порівняно невеликих процедур уточнення умови задачі.

4.4 Експериментальні дослідження адаптивно-структурного методу оцінки складності тестових завдань

В даному підрозділі проводиться дослідження адаптивно-структурного методу оцінки складності тестових завдань [41] з метою підтвердження його ефективності.

Для проведення експериментів використано результати проведення тестового контролю з дисципліни «Засоби програмування баз даних і знань» для студентів четвертого курсу спеціальності «Програмне забезпечення систем». У додатку Е представлено перелік використовуваних тестових завдань. Частоти помилкових відповідей наведено у таблиці 4.1.

Таблиця 4.1 – Розподіл частот помилкових відповідей по тестових завданнях

№ тесту	Частота помилок	№ тесту	Частота помилок	№ тесту	Частота помилок
1	0,06	11	0,00	21	0,19
2	0,63	12	0,13	22	0,38
3	0,06	13	0,56	23	0,13
4	0,19	14	1,00	24	0,31
5	0,38	15	0,25	25	0,13
6	0,88	16	0,44	26	0,19
7	0,25	17	0,00	27	0,19
8	0,19	18	0,13	28	0,06
9	0,69	19	0,06	29	0,19
10	0,38	20	0,25	30	0,06

На рисунку 4.22 представлено результати апроксимації відносних частот функцією складності. В ході експериментів встановлено, що параметр оцінки сили впливу кількості задалегідь невірних альтернатив α_0 доцільно покласти рівним 1. При цьому вектор коефіцієнтів функції складності для формул (3.30)–(3.32) отримав представлення $\vec{\beta} = [0.16 \ 1.28]$.

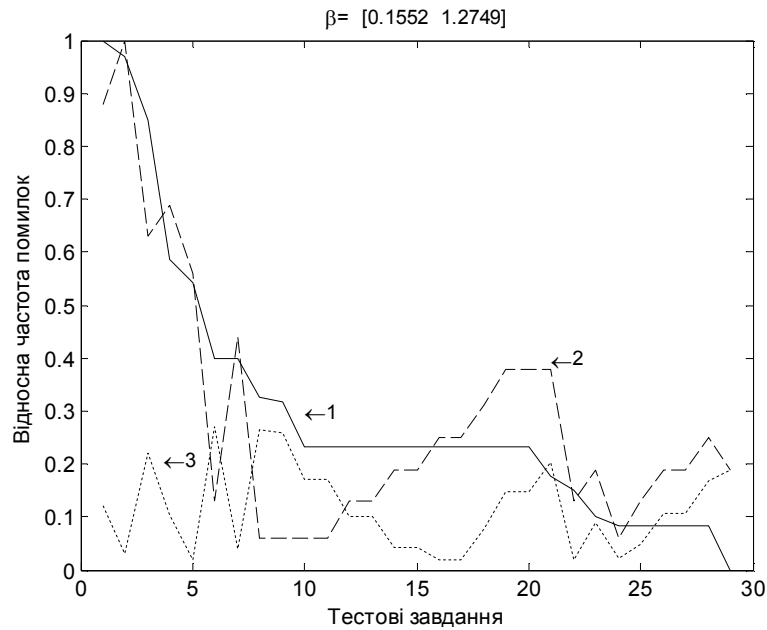


Рисунок 4.22 – Апроксимація відносних частот функції складності (модельована відносна складність тесту (1), відносна частота помилок (2), відхилення прогнозованої складності від частоти (3))

Максимальне відхилення прогнозованої складності від частоти помилок становило 27%. Однак оскільки для контролю знань на практиці використовуються набори тестів, то інформативною оцінкою відхилення функції складності від частоти слід вважати її середнє значення. Для цього випробовування воно склало 11 %, що дозволяє використовувати побудовану модель для формування тестових наборів заданої складності із достатньою для практики точністю.

Проведено дослідження методу на стійкість до обсягу вибірки. З цією метою із вибірки випадковим чином вилучалося 52% її обсягу. Встановлено, що

для забезпечення стійкості моделі на порівняно невеликій вибірці серед вилучених не повинно міститися спостережень, складність яких вище 50%. На рисунку 4.23 наведено типовий приклад результатів проведення одного із таких експериментів.

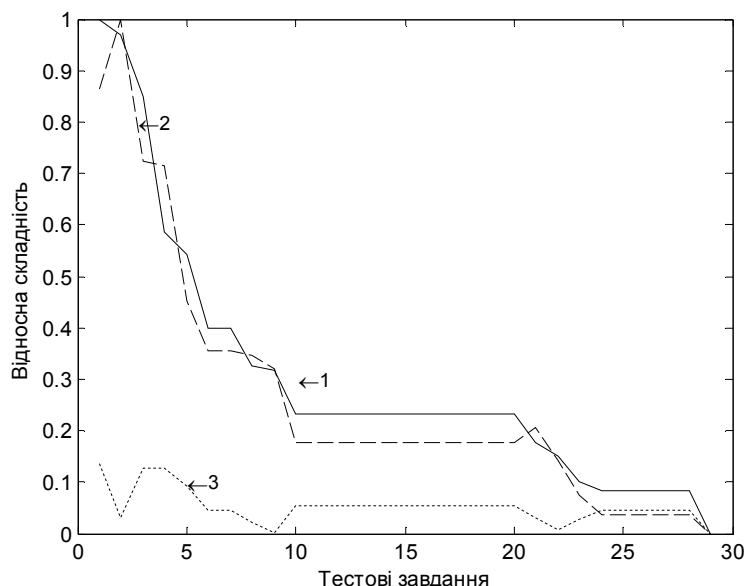


Рисунок 4.23 – Порівняння функції складності для різних експериментів (модель складності по повній вибірці (1), модель складності по редукованій вибірці (2), відхилення прогнозованої складності (3))

Оцінка відхилення прогнозованої складності для різних варіантів вибірок не перевищувала 6.3%, що підтверджує можливість застосування даного методу для оцінки складності з невеликою інформаційною вибіркою.

У таблиці 4.2 наведено ранжування тестових завдань за рівнями складності для різних видів експериментів. У 1 експерименті даний метод застосовувався для повної вибірки тестових завдань, тобто складність обчислювалася по повній вибірці. Для другого експерименту складність для всіх тестів розраховувалася за умови випадкового відкидання більше 52 % тестових завдань. Для третього експерименту випадково відкидалися 52 % тестових завдань, серед яких 10 % обов'язково мають найвищу складність.

Таблиця 4.2 – Ранжування тестових завдань за рівнями складності для різних експериментів

Ранг	Номери тестових завдань по експериментах			Ранг	Номери тестових завдань по експериментах		
	№1	№2	№3		№1	№2	№3
1	14	14	14	16	27	18	9
2	6	6	2	17	21	27	17
3	2	2	6	18	20	21	28
4	9	9	8	19	15	20	1
5	13	13	10	20	24	15	23
6	12	19	3	21	22	24	18
7	16	12	13	22	5	22	27
8	19	16	5	23	8	8	21
9	11	11	19	24	26	26	20
10	3	3	26	25	30	10	15
11	17	5	30	26	25	30	24
12	28	17	25	27	29	25	22
13	1	28	29	28	4	29	12
14	23	1	4	29	7	4	16
15	18	23	7	30	10	7	11

На рисунку 4.24 зображено порівняння функції складності для різних експериментів, що дозволяє відслідковувати відхилення в рангах.

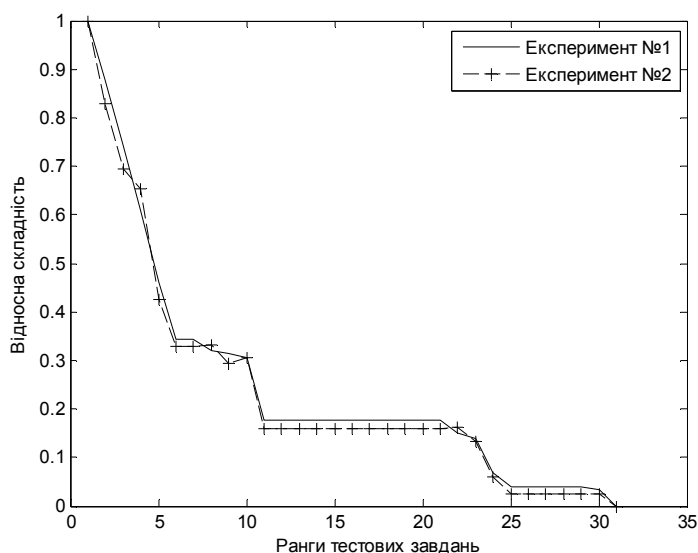


Рисунок 4.24 – Порівняння рангів функції складності для експерименту №1 та експерименту №2

Як видно з даного рисунка похибка експерименту невелика, що дозволяє застосовувати даний метод для оцінки складності з невеликою інформаційною вибіркою.

На рисунку 4.25 проілюстровано порівняння функції складності для експерименту №1 та експерименту №3.

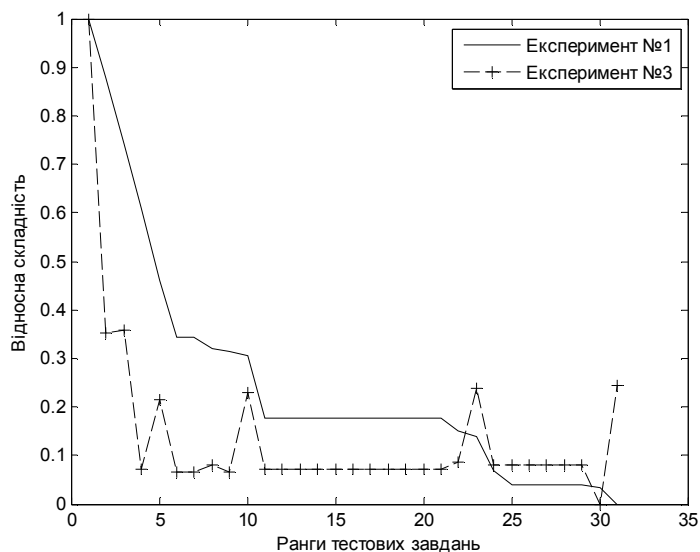


Рисунок 4.25 – Порівняння рангів функції складності для експерименту №1 та експерименту №3

На рисунку 4.26 відображено апроксимацію відносних частот функції складності, при якій вектор коефіцієнтів $\vec{\beta} = [0.189 \ 1.006]$, що це демонструє функціональну залежність між показником композиційної та формальної близькості, який у даному випадку складає $\beta_2 = 5 \cdot \beta_1$.

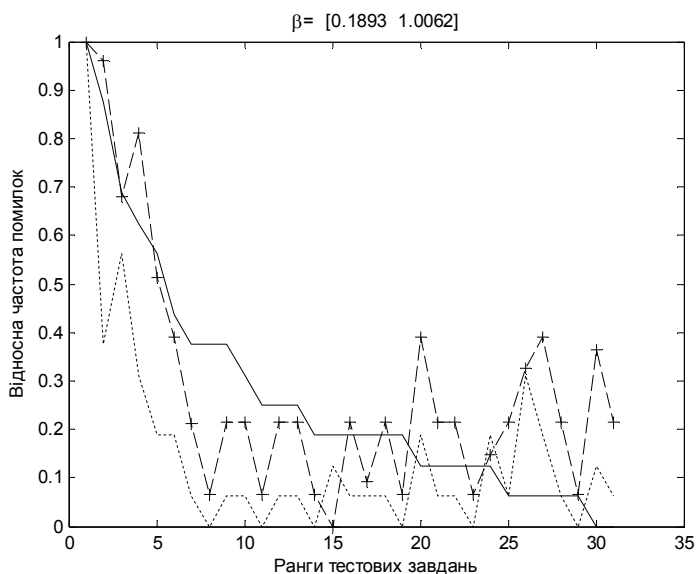


Рисунок 4.26 – Апроксимація відносних частот функції складності

Похибки експерименту оцінені за допомогою критеріїв максимальності та мінімальності відносних похибок. Кожен експеримент проводився наступним чином – випадково формувалася тестова вибірка, яка включала більше 50 % всієї сукупності тестових завдань, і тоді проводилося порівняння з оцінками, які одержані для всієї сукупності тестів.

Максимальні відносні похибки оцінювання параметрів для п'яти проведених експериментів прийняли значення: 6,45; 9.67; 6,15; 3,23; 7.24. Враховуючи результати експериментальних досліджень максимальна похибка склала 9.67.

У таблиці 4.3 проведено порівняльний аналіз відомих методів оцінки складності тестових завдань (див. рис. 1.1) із запропонованим адаптивно-структурним методом. У порівнянні з відомими методами, створений метод для налаштування моделі вимагає суттєво меншої вибірки. Однак суттєвою перевагою запропонованого методу в порівнянні з іншими є те, що після налаштування моделі можна оцінювати складність наново розроблених тестів для аналогічних навчальних курсів та студентських аудиторій без навчального тестування.

Адаптивно-структурний метод дає можливість автоматичної оцінки складності згенерованих завдань за формальними критеріями, а також може слугувати основою для апріорної оцінки складності автоматично згенерованих тестових завдань, тобто дозволить генерувати тестові завдання заданої складності.

На рисунку 4.27 наведено приклад автоматично згенерованого тестового завдання множинного виду з дисципліни «Основи С++» високого рівня складності, а на рисунку 4.28 автоматично згенероване завдання з низьким рівнем складності. З аналізу даних прикладів можна зробити висновки про можливість застосування адаптивно-структурного методу для генерації завдань із заданою складністю, а також його обмеження.

Таблиця 4.3 – Порівняльна характеристика різних методів оцінки складності тестових завдань

№ п/п	Методи оцінки	Критерії порівняння			Основні недоліки
		Розмірність статистичної вибірки	Кількість факторів, які впливають на складність	Можливість застосування для генерації тестів заданої складності	
1	Складність на основі типізації завдань	-	1	+	Суб'єктивність типізації
2	Метод оцінки на основі стохастичної теорії тестів IRT	>1000	1	-	Велика вибірка, 1 фактор впливу на складність, висока розбіжність між емпіричними та модельними даними
3	Інтегрована функціональна модель	>100	2	-	Необхідність попереднього тестування, та залежність від його результатів
4	Метод на основі системи логічних кроків	-	1	+	Суб'єктивність, складність логічного розподілу
5	Статистичний метод	>100	1	-	Велика вибірка, залежність тільки від одного фактора
6	Індекс складності	>100	1	-	Велика вибірка, залежність тільки від одного фактора
7	Адаптивно-структурний метод	>20	2	+	Складність початкового налаштування параметрів моделі

5. Конструктор класу -

- спеціальний блок інструкцій, який викликається при створенні об'єкта
- спеціальний блок інструкцій, який ідентифікує параметри функцій
- спеціальний блок інструкцій, який викликається при знищенні об'єкта

Рисунок 4.27 – Згенероване тестове завдання третього рівня складності

5. Конструктор класу -

- мова програмування, яка використовується у більшості ЕОМ
- запит, який використовується для оновлення даних
- спеціальний блок інструкцій, який викликається при створенні об'єкта

Рисунок 4.28 – Згенероване тестове завдання першого рівня складності

Ці обмеження полягають в способі генерації хибних альтернатив на основі маніпуляції фрагментами формалізації компонент тверджень. В той же час викладацька практика може виявляти семантично близькі хибні альтернативи,

що не мають спільних текстових фрагментів із вірною альтернативою, однак викликають великі труднощі в їх ідентифікації студентами. Для усунення вказаного обмеження в системі передбачено спеціальні відношення для документування окремих семантично близьких альтернатив до можливих базових тверджень. Таким альтернативам присвоюється найвищий рівень складності при оцінці загальної складності тестового завдання.

4.5 Експериментальні дослідження методу адаптивного управління структурою процесу навчання

На основі запропонованого у [36,40,42] та обґрунтованого у підрозділі 3.3 методу адаптивного управління структурою процесу навчання було проведено ряд чисельних експериментів.

Для проведення експериментів використано результати навчального тестування в ході підготовки до задачі модуля з дисципліни «Засоби програмування баз даних і знань». Аналіз застосовано до представників категорій студентів, що характеризуються невисоким, середнім та високим рівнями базової підготовки, який визначається на основі попереднього аналізу їх успішності.

Суть експериментальних досліджень зводилася до оцінки витрат часу для досягнення бажаного рівня успішності по трьох рівнях складності навчальних матеріалів для студентів із різними ступенями базової підготовки. Для кожного студента фіксувалися його показники успішності при освоєнні завдань різного рівня складності шляхом розподілу затраченого часу пропорційно кількості розв'язаних завдань кожного рівня.

Функція успішності студентів в освоєнні навчального матеріалу міняється після кожного сеансу навчання і остаточно проявляється лише на завершальних

його етапах. Показники успішності для типових представників кожної із груп студентів наведені в додатку Ж.

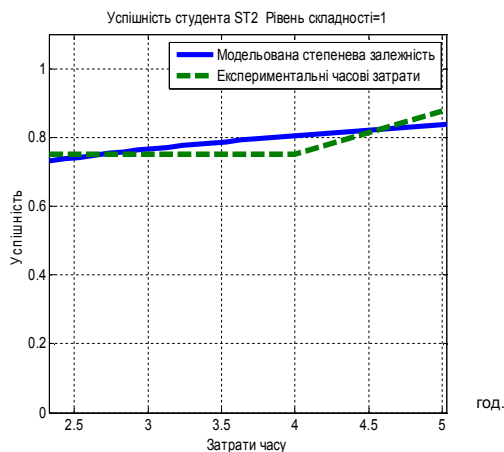
Оскільки відомі аналізовані підходи, зокрема логіко-математичний метод та метод формування індивідуальної траєкторії навчання через абстрактність та суб'єктивізм застосовуваних показників не можуть бути застосовані в умовах експериментів щодо засвоєння матеріалу конкретних модулів, виграш в ефективності адаптивного методу порівняємо із найпростішим недиференційованим підходом.

Мета цього підходу полягає в досягненні бажаної успішності по всіх рівнях складності. Ріст ефективності, забезпечуваний адаптивним методом оцінювався відношенням зекономленого часу відносно недиференційованого підходу до загальних витрат часу останнього. Цей процес можна представити за допомогою наступної формули:

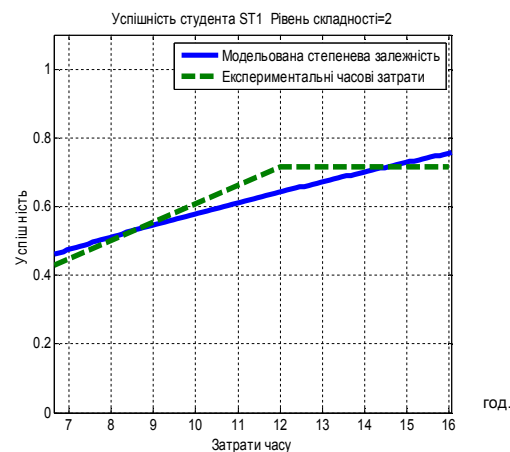
$$EG = \frac{\sum_k T_k^{s,tm,nt}(U_{\min}^s) - \sum_k T_k^{s,tm,nt}(U_k^{s,tm,nt})}{\sum_k T_k^{s,tm,nt}(U_{\min}^s)} \times 100\%, \quad (4.1)$$

де EG - приріст ефективності.

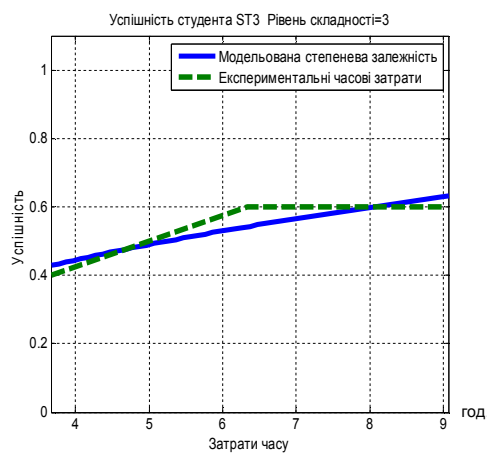
На рисунку 4.29 наведено графічне представлення результатів представлення залежності успішності від часових витрат для різних студентів при диференціації рівнів складності. Варто відзначити, що при проведенні даних експериментів очевидна вимога неспадного характеру успішності навчання із ростом часових витрат не завжди виконувалася. Це можна пояснити нерівномірним освоєнням студентами різних аспектів навчального матеріалу навіть при формуванні тестових завдань однакового рівня складності. В цьому випадку експериментальні дані підлягають корекції. Зокрема при отриманні спадної ділянки в експериментальних даних вона заміняється на усередненні постійні значення.



а) рівень складності 1;



б) рівень складності 2;



в) рівень складності 3;

Рисунок 4.29 – Представлення функцій успішності в залежності від часових витрат для різних рівнів складності а), б), в).

Як видно з представлених графіків, запропонована у підрозділі 3.3 функція дозволяє проводити якісний прогноз успішності при заданих часових витратах. Після застосування методу адаптивного управління структурою процесу навчання було отримано розподіл часових ресурсів по рівнях складності, а також з'явилася можливість оцінити приріст ефективності у порівнянні з недиференційованим підходом.

На рисунку 4.30 наведено результати управління витратами часу для студента із невисоким рівнем базової підготовки для досягнення успішності 0.75. У порівнянні з недиференційованим підходом ефективність склала 26.6 %.

Якщо розглянути розподіл часу по рівнях складності, то тут варто відзначити суттєву економію для вивчення навчальних розділів, які формують найскладніший рівень тестового розподілу.

Розподіл тестових завдань по рівнях складності та адаптивне управління дозволяє оцінити можливості кожного студента (часові, індивідуальну підготовку, кінцеву мету навчання) та сформувавши вірну навчальну стратегію в межах локальних тем або модулів.

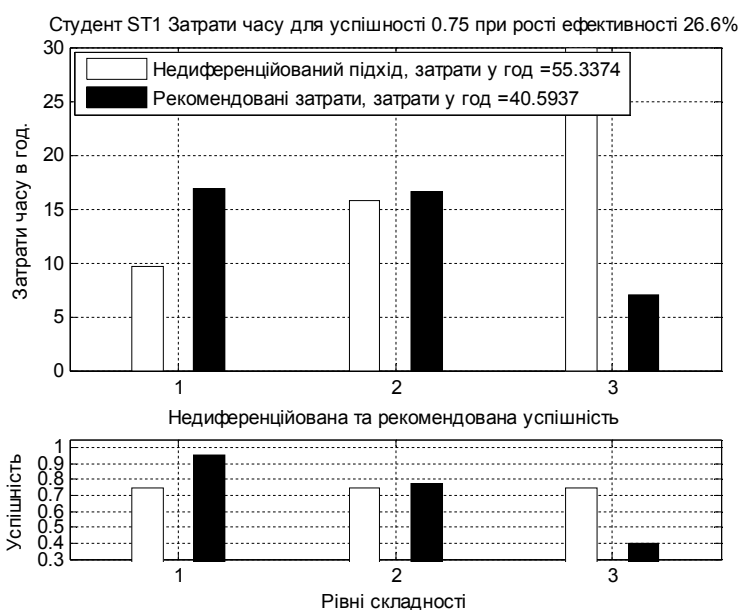


Рисунок 4.30 – Управління витратами часу навчання для студента 1 для досягнення успішності 0.75 з ефективністю 26.6 %

Як видно з рисунка 4.30 по завданнях 1 рівня складності студентіві необхідно орієнтовно набрати успішність на рівні 0.92, і відповідно для 2 та 3 рівнів 0.78 та 0.4, що дозволить зекономити час в межах 14.74 год. для досягнення успішності 0.75.

Розглянемо випадок коли цьому ж студентіві необхідно досягнути найнижчий позитивний рівень успішності 0.6. На рисунку 4.31 наведено результати моделювання для даного прикладу. Ефективність в даному випадку склала 23%, що дорівнює економії часу на рівні 7.84 годин. В результаті

застосування даного методу для досягнення невисоких рівнів успішності середня ефективність знаходилася в межах 20%.

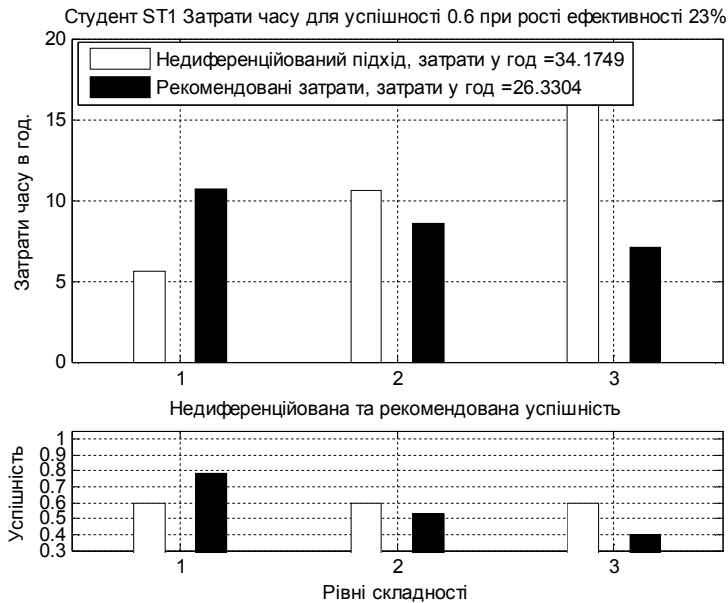


Рисунок 4.31 – Управління витратами часу навчання студента 1 для досягнення успішності 0.6 при ефективності 23.0 %

Проаналізуємо дію методу для студентів із високим рівнем базової підготовки і якому необхідно досягнути рівня успішності, який дорівнює «відмінній» оцінці. На рисунку 4.32 представлено результати такого моделювання, ефективність якого склала 4%. Адаптація студента до тестових завданнях по рівнях складності середнього та високого рівнів дозволяє йому сформулювати стратегію, яка буде включати вивчення навчальних матеріалів згаданих рівнів складності на рівні успішності 0.92-0.95.

На рисунку 4.33 зображено результати для цього ж студента, але у випадку, коли йому необхідно досягнути оцінки 0.75. Ефективність для цього прикладу склала 0.6 %, що підтвердило теоретичні припущення щодо рівномірності розподілу часових ресурсів студента з високим рівнем підготовки для досягнення середньої успішності, тобто виграш даного методу в цій стратегії є мінімальним.

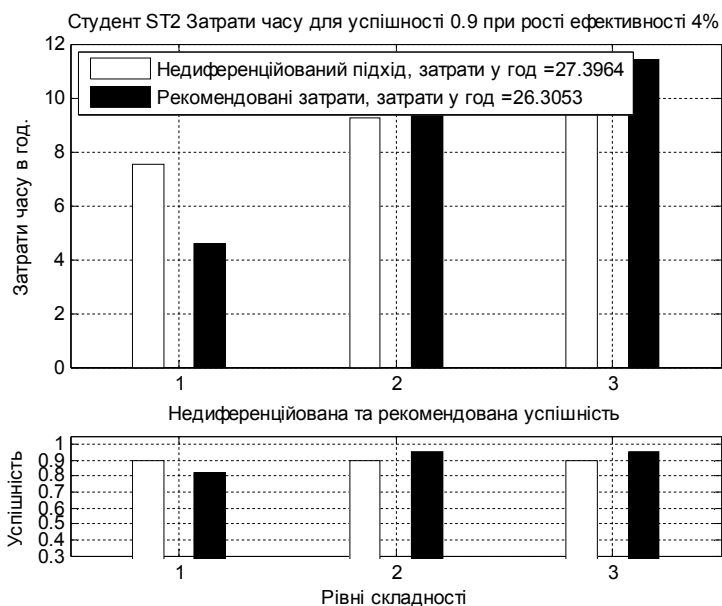


Рисунок 4.32 – Управління витратами часу навчання для студента 2 для досягнення успішності 0.9 при ефективності 4.0 %

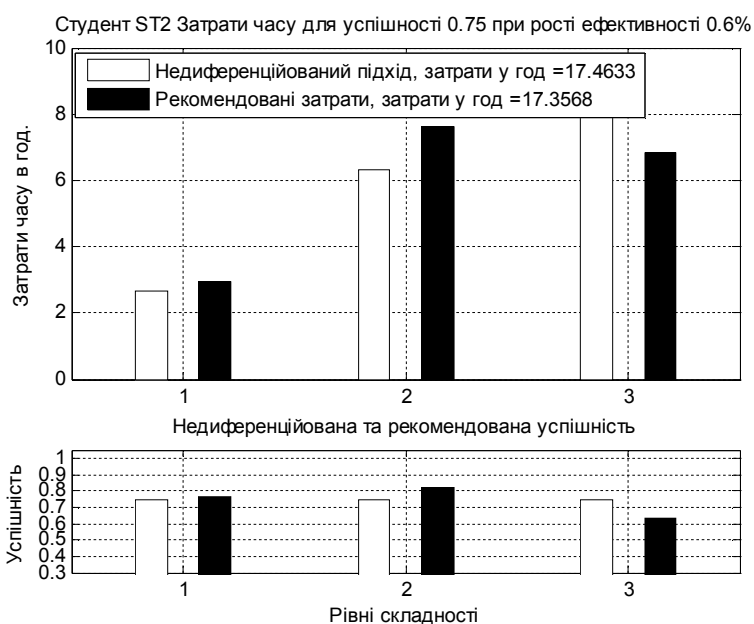


Рисунок 4.33 – Управління витратами часу навчання для студента 2 для досягнення успішності 0.75 при ефективності 0.6%

Дослідимо можливості методу адаптивного управління структурою процесу навчання для студента із середнім рівнем базової підготовки, оскільки їх частка в загальній вибірці є найбільшою. Нехай студентові необхідно суттєво підняти свій бал успішності, до рівня 0.85, то в результаті моделювання ефективність склала 17.2% або в часовому еквіваленті економія 6.11 год. у

порівнянні з недиференційованим підходом, що зображено на рисунку 4.34. Даному студентові необхідно зосередити свою увагу на 1 та 2 рівнях складності, оскільки саме в цьому випадку економія часу буде досягати максимальної величини.

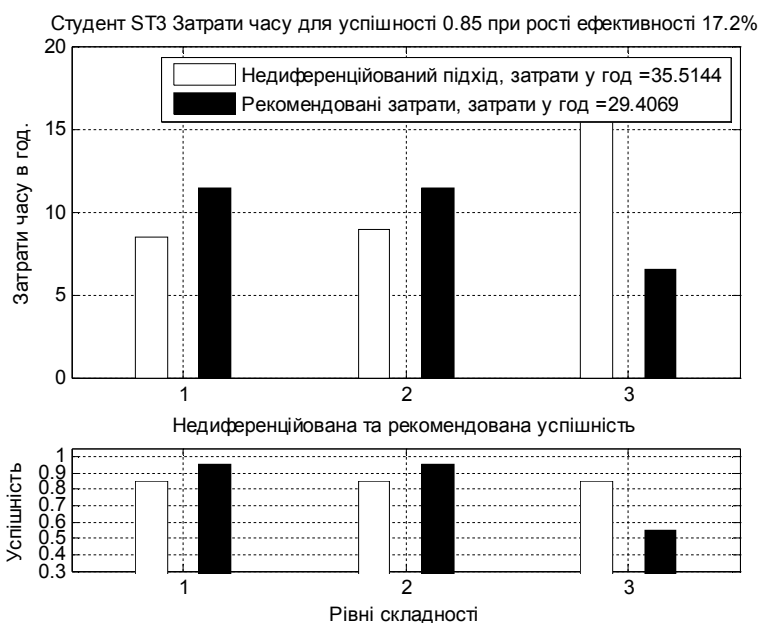


Рисунок 4.34 – Управління витратами часу навчання для студента 3 для досягнення успішності 0.85 при ефективності 17.2 %

Якщо студентові 3 необхідно просто підтвердити свій рівень, при цьому розподіливши свої часові ресурси за принципом мінімізації, то ефективність методу в даному випадку знаходилася на рівні 18.9%, тобто економія часу 4.97. Даний експеримент представлено на рисунку 4.35. Основну увагу при формуванні навчальної стратегії необхідно звернути на завдання середнього рівня складності, успішність по якому повинна знаходитися в межах 0.85 – 0.87. Витрати часу на вивчення навчальних матеріалів, на основі яких формуються завдання найскладнішого рівня можна мінімізувати, що дозволить максимально оптимізувати процес навчання.

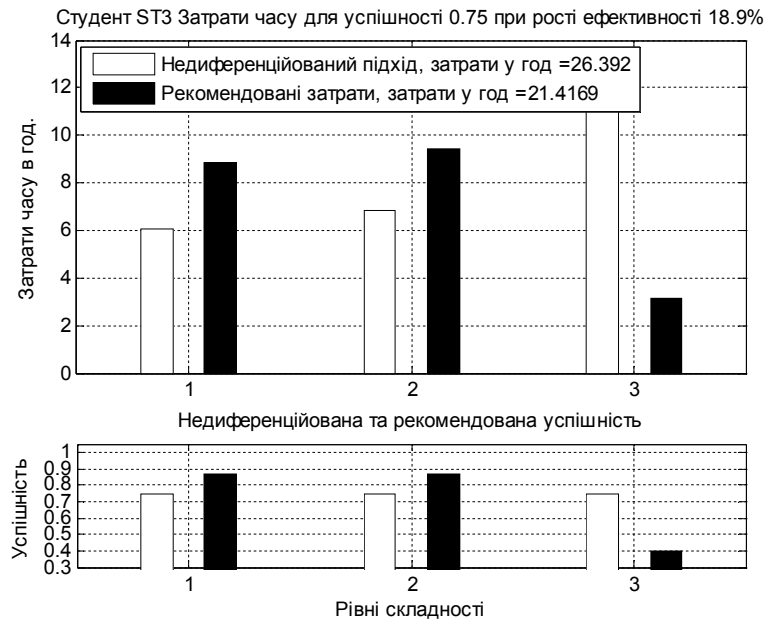


Рисунок 4.35 – Управління витратами часу навчання для студента 3 для досягнення успішності 0.75 при ефективності 18.9%

Із наведених рисунків можна ідентифікувати рівні студентів *ST1* як слабкий, *ST2* як високий, а *ST3* як середній. Відповідно до їх рівнів оцінювалися найбільш реальні альтернативні бажані рівні їх успішності. У всіх випадках адаптивний метод забезпечив ріст ефективності. Як і очіувалося, в більшості випадків рекомендується досягнення вищої успішності на нижніх рівнях складності та зменшення успішності на вищих рівнях в порівнянні із недиференційованим підходом. Однак для студентів із високим рівнем підготовки для досягнення високої успішності рекомендовано збільшення часових витрат для освоєння завдань із високим рівнем складності.

Найбільші прирости ефективності метод демонструє для студентів із слабкою та середньою рівнями підготовки. На основі проведених досліджень можна стверджувати, що адаптивне управління структурою навчального процесу при автоматичній генерації тестів із врахуванням їх складності забезпечує суттєву економію часових витрат без втрати ефективності засвоєння навчальних матеріалів, яка дозволяє будувати покращені оперативні навчальні стратегії в межах локальних тем або модулів.

У таблиці 4.4 наведено порівняльний аналіз ефективності методу адаптивного управління структурою процесу навчання з відомими рішеннями (див. підрозділ 1.3), які максимально близькі до даного, виходячи із специфіка особливостей функціонування предмету дослідження.

Таблиця 4.4 – Порівняння методів управління процесом навчання

Метод	Критерії оцінки			
	Кількість параметрів	Час навчання	Складність застосування	Роль експертів
Традиційний підхід	Індивідуальний вибір	Формальна регламентація	Застосування лише для невеликих множин студентів	Повний супровід процесу навчання (постійна присутність викладача)
Логіко-математична модель управління навчанням	Понад 3 (індивідуальні характеристики студентів, параметри навчальних матеріалів...)	Не розглядається	Складність побудови логічних міжпредметних зв'язків	Повний супровід процесу навчання (експертні оцінки)
Метод формування індивідуальної траєкторії навчання студента	Понад 7 (навчальна завантаженість, успішність, відсутність відволікаючих чинників...)	Застосування в межах нормативних часових витрат	Складність та неоднозначність вибору критеріїв оцінки навчання	Частковий супровід процесу навчання
Метод адаптивного управління структурою процесу навчання	Час та успішність	Мінімізація часу	Автоматизована ідентифікація в процесі тестування	Частковий супровід процесу навчання

У порівнянні з відомими методами розроблений метод дозволяє отримувати ефективні результати при побудові навчальних стратегій, використовуючи при цьому мінімальну кількість параметрів для побудови моделі, не вимагає повного супроводу процесу навчання експертами та сприяє оптимальному розподілу часових ресурсів при досягненні бажаних результатів процесу навчання.

4.6 Порівняння інформаційних технологій автоматизованого навчання

Після аналізу особливостей програмної реалізації запропонованої технології автоматичної генерації тестових завдань з керованою складністю (AGT) в системі SAGT доцільно розглянути питання оцінки її ефективності. Особливого значення набуває порівняння ефективності розробленої технології із основними представниками технології автоматизації процесу тестування по системі показників, яка володіє диференціюючими властивостями на аналізованій множині технологій. Оскільки згадана система оцінки включає п'ять показників, які визнаються рівноправними, кожен із показників зручно представляти його вкладом в інтегральну оцінку. При цьому його максимальне значення складатиме 20% [44].

В даному підрозділі на основі системи показників [44], описаної в підрозділі 1.4, проведено дослідження ефективності розробленої технології із основними відомими технологіями генерації тестових завдань, в результаті якого встановлено переваги розробленої інформаційної технології над відомими рішеннями. В перелік досліджуваних технологій були включені:

- технологія на основі ПТМ (понятійно–тезисної моделі), оскільки в ній застосовуються методи автоматичної генерації тестів (див. підрозділ 1.2);
- технологія на основі ПТ (параметризованих тестів) – один з перших підходів до автоматичної генерації тестових завдань (див. підрозділ 1.2);
- технології, які реалізують автоматизацію процесу тестування на основі завдань, які будуються на принципах «ручної» генерації (Moodle, Агапа). Ця технологія вибрана з метою оцінки використання сучасних систем дистанційного навчання у порівнянні з можливостями використання технологій автоматичної генерації тестових завдань (див. підрозділ 1.3);
- технології закритого та комерційного характеру. Ефективність отримання деталізованої інформації про їх параметри оцінити неможливо. Однак їх варто тримати в полі зору, як можливі носій принципово нових підходів розвитку систем даного класу (див. підрозділи 1.2, 1.3);

На першому етапі дослідження оцінимо ефективність технології AGT по системі показників, наведених в підрозділі 1.4. Оскільки ця технологія передбачає використання тестових завдань для перевірки засвоєння теоретичного апарату, методологічних знань та вмінь і навичок розв'язання практичних завдань при контролі адекватності його алгоритмізації то за показником повноти контролю знань (узагальнена шкала Блюма), вона отримує максимальну оцінку. Оскільки в технології AGT реалізовані всі відомі види закритих тестів а також використовуються всі можливі фрагменти формалізованих базових тверджень, вона отримає максимальну оцінку за критерієм приведеної економії часу на генерацію тестів. Об'єктивність індивідуального оцінювання даної технології детермінується максимальною похибкою її моделі оцінки складності, що склала 11%.

Показник адаптації до складності завдань отримується на основі усередненого значення відносної економії часу на досягнення бажаної успішності із використанням методу адаптивного управління структурою процесу навчання, яка склала 15%. Показник однорідності альтернатив визначається відсутністю повного семантичного контролю та наявністю контролю по семантичній, синтаксичній та формальній однорідності альтернатив і склала 50% максимального значення. Отримані значення показників технології AGT представлені на рисунку 4.36.

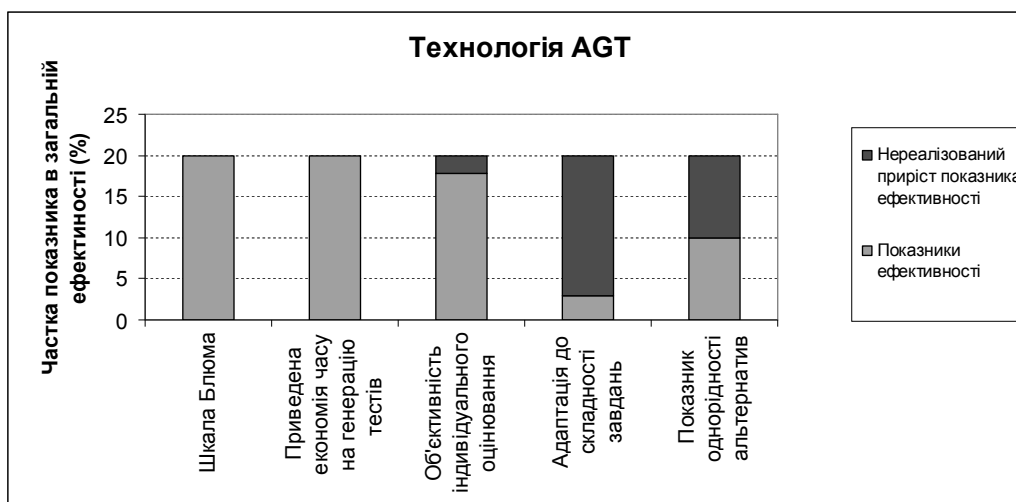


Рисунок 4.36 – Система показників оцінки технології AGT

На другому етапі дослідження оцінимо ефективність технології ПТМ. Оскільки ця технологія передбачає використання тестових завдань для перевірки засвоєння теоретичного апарату то за показником узагальненої шкали Блюма вона отримує 30% від максимальної оцінки. Оскільки в технології ПТМ реалізовані 2/3 відомих видів закритих тестів а також використовуються по два фрагменти формалізованих базових тверджень, вона отримає 60 % максимальної оцінки за критерієм приведеної економії часу на генерацію тестів. Оскільки в даній технології не реалізовані оцінка складності та можливість адаптації до складності завдань, то по наступних двох показниках вона отримає нульові оцінки. Показник однорідності альтернатив визначається відсутністю повного семантичного контролю та наявністю контролю лише по синтаксичній та формальній однорідності альтернатив і склала 33% максимального значення. Отримані результати представлено на рисунку 4.37.

На третьому етапі дослідження оцінимо ефективність технології ПТ. Оскільки ця технологія передбачає використання тестових завдань для перевірки засвоєння практичних навичок то за показником узагальненої шкали Блюма вона отримує 30% від максимальної оцінки. Оскільки в технології ПТ реалізовані 1/3 відомих видів закритих тестів а також використовуються по два фрагменти формалізованих базових тверджень, вона отримає 40% максимальної оцінки за критерієм приведеної економії часу на генерацію тестів.

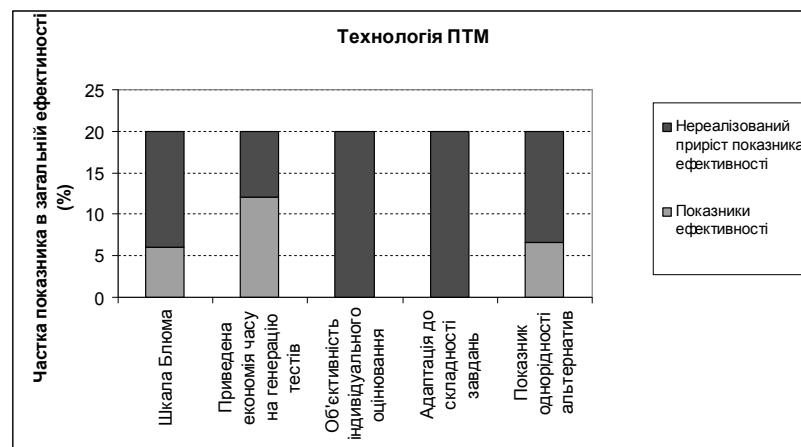


Рисунок 4.37 – Система показників оцінки технології ПТМ

Об'єктивність індивідуального оцінювання даної технології детермінується максимальною похибкою її моделі оцінки складності, що при достатньому обсязі статистичної вибірки може бути доведена до 10%. Оскільки в даній технології не реалізована можливість адаптації до складності завдань, то по наступному показнику вона отримує нульову оцінку. Показник однорідності альтернатив визначається наявністю повного семантичного контролю і складає максимальне значення. Отримані значення показників технології ПТ представлені на рисунку 4.38.

На четвертому етапі дослідження оцінимо ефективність технології тестування на базі «ручної» генерації. Оскільки ця технологія передбачає використання тестових завдань для перевірки засвоєння теоретичного апарату та практичних навичок то за показником узагальненої шкали Блюма вона отримує 60% від максимальної оцінки. В цій технології банк завдань будується на основі «ручної» генерації, тому вона отримує нульову оцінку за критерієм приведеної економії часу на генерацію тестів.

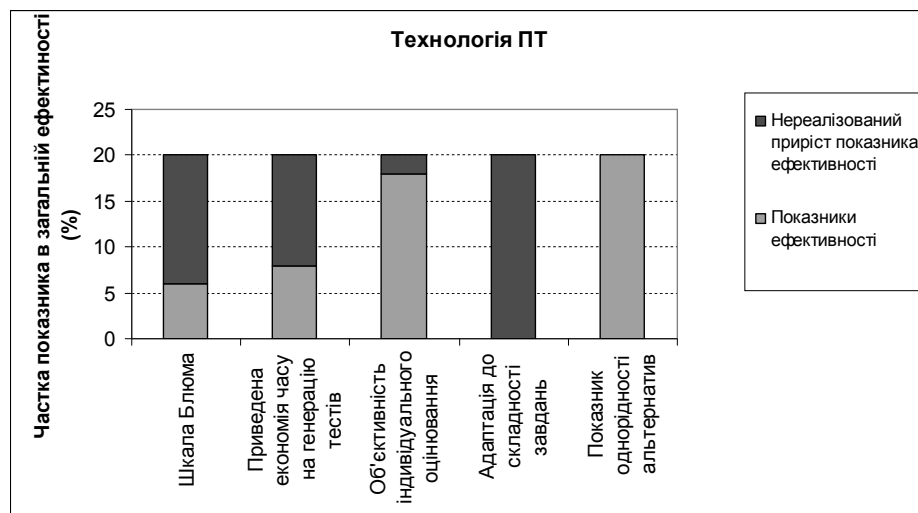


Рисунок 4.38 – Система показників оцінки технології ПТ

Об'єктивність індивідуального оцінювання детермінується максимальною похибкою моделі оцінки складності, що при достатньому обсязі статистичної вибірки може бути доведена до 10% як і для попередньої технології. Не реалізовано можливість адаптації до складності завдань, то по наступному показнику отримується нульова оцінка. Показник однорідності альтернатив

визначається наявністю повного семантичного контролю і складає максимальне значення. Отримані значення показників технології на базі «ручної» генерації ображено на рисунку 4.39.



Рисунок 4.39 – Система показників оцінки технології на базі «ручної» генерації

Виходячи з аналізу кількісних показників оцінок технологій автоматизованого навчання, проведено розрахунок інтегрального показника ефективності. Технологія AGT у порівнянні з відомими рішеннями забезпечує приріст показника ефективності на 18.8%, що складає 36% відносного приросту щодо ефективності технології ПТ. Результати представлено на рисунку 4.40.

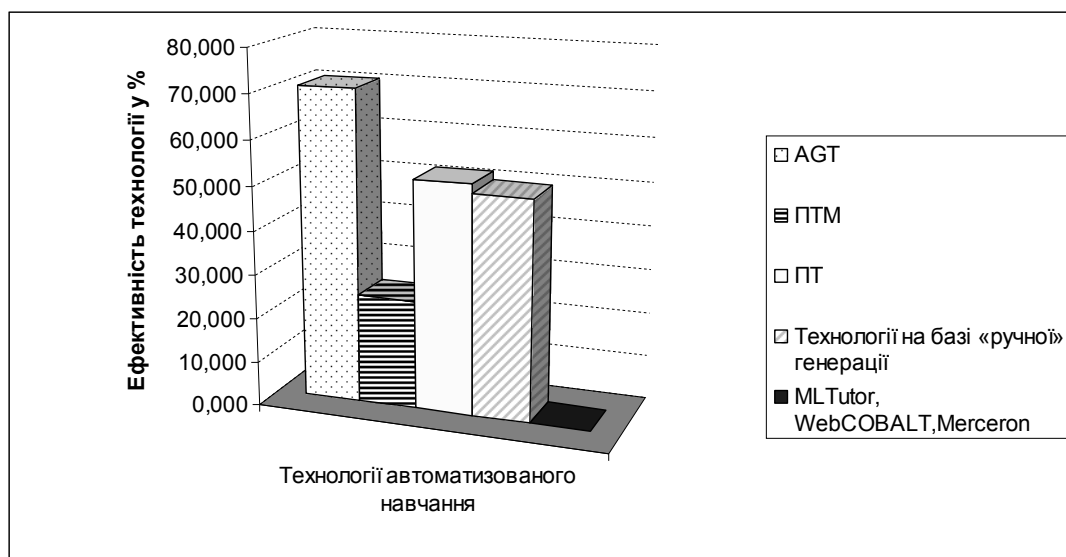


Рисунок 4.40 – Порівняння технологій автоматизованого навчання на основі інтегрального показника ефективності

Отже технологія AGT має ряд переваг у порівнянні з відомими рішеннями, що підтверджено експериментальними дослідженнями.

Висновки до розділу 4

1. На основі запропонованих методів та алгоритмів реалізовано веб-орієнтовану систему SAGT автоматичної генерації тестових завдань, яка включає засоби управління навчальним контентом, модуль формалізації навчальних матеріалів, модулі генерації тестових завдань різних типів.

2. Проведено експериментальні дослідження реалізації тестових завдань різних видів для закритого типу. Вони показали достатній рівень варіативності та однорідності альтернатив в процесі автоматичної генерації тестових завдань, що забезпечує ефективне їх використання для проведення тестових контролів з метою оцінки засвоєння теоретичних знань.

3. Проведено експериментальні дослідження методу генерування завдань для перевірки методологічних знань, які підтвердили економію часових витрат на побудову моделей проблемних ситуацій та перевірку правильності їх розв'язань студентами.

4. Досліджено можливості автоматичної генерації тестових завдань з програмованим оператором для перевірки використання вмінь та навичок які підтвердили економію часових витрат за рахунок суттєвого збільшення варіативності генерованих варіантів типових задач та їх автоматичної перевірки системою. Окрім того, даний підхід дозволяє контролювати вміння студентів алгоритмізувати та програмувати розв'язання практичних задач проблемної області.

5. Отримано чисельні результати експериментів із оцінки складності генерованих тестових задач та оптимізації часових витрат для досягнення бажаної успішності. Їх аналіз засвідчив про досягнення допустимих рівнів

похибки в оцінці складності тестових завдань, максимальне значення якої склало 11%. Середня економія часових витрат при використанні методу адаптивного управління структурою процесу навчання склала 15%.

6. Здійснено оцінку ефективності розробленої технології АГТ в порівнянні із відомими підходами на основі розробленої системи показників, що володіють диференційною здатністю. Аналіз отриманих результатів засвідчив суттєве підвищення ефективності процесу автоматизації тестового контролю, абсолютне значення якої склало 19%.

ВИСНОВКИ

В дисертаційній роботі розв'язана актуальна наукова задача розроблення інформаційної технології автоматичної генерації тестових завдань з керованою складністю з метою мінімізації часових витрат на формування тестових завдань та досягнення бажаного рівня успішності в процесі навчання при забезпеченні достатньої однорідності альтернатив та повноти контролю знань.

Основні наукові та практичні результати роботи полягають у наступному.

1. Проведено аналіз структури процесу навчання та роль тестів в реалізації функції контролю знань. Досліджено відомі підходи до автоматичної генерації тестових завдань, встановлено їх недоліки і намічено напрями їх подолання шляхом врахування семантичних характеристик компонентів тестових завдань та їх синтаксичної узгодженості. Виділено роль показника складності тестових завдань та проаналізовано основні підходи до її оцінки. Сформовано систему кількісних показників для оцінки якості технологій автоматизованого навчання, що дозволило сформулювати вимоги до параметрів розроблюваної технології.

2. Запропоновано методи автоматичної генерації тестових завдань для перевірки теоретичних знань, який на відміну від відомих дозволяє генерувати тестові завдання із врахуванням формальної, семантичної та синтаксичної узгодженості. Запропоновано системи формальних та семантичних класів, які дозволяють враховувати основні характеристики інформативних представлень для уникнення генерації завідомо невірних альтернатив.

3. Розроблено метод автоматичної генерації тестових завдань для перевірки засвоєння методологічних знань, який дозволяє автоматизувати процедуру генерації проблемних ситуацій, та економити часові ресурси в процесі встановлення причинно-наслідкових зв'язків при пошуку методів розв'язання проблем.

4. Побудовано метод автоматичної генерації завдань з програмованим оператором, який використовується для оцінки засвоєння вмінь та навиків. На

відміну від відомих рішень пропонується метод дозволяє генерувати не тільки вхідні параметри задачі, але і забезпечує варіативність структурних елементів її умови.

5. Запропоновано та обґрунтовано метод оцінки складності тестових завдань під час їх автоматичної генерації. Він володіє перевагами статистичних методів та методів структурного аналізу. Розроблено метод адаптивного управління структурою процесу навчання, яка дозволяє ефективно управляти часом навчання для досягнення бажаного рівня засвоєння знань, виходячи з мінімізації показника часових затрат. Середня економія часових затрат при використанні методу адаптивного управління структурою процесу навчання склала 15%.

6. Реалізовано веб-орієнтовану систему SAGT автоматичної генерації тестових завдань, ріст ефективності якої в порівнянні із відомими підходами склав 19%.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Абовский Н. П. Развитие системного мышления в обучении и тестировании / Н. П. Абовский, В. И. Палагушкин // *Alma mater.* – 2009. – №9. – С. 32-39.
2. Аванесов В. С. Вопросы объективизации оценки результатов обучения. / В. С. Аванесов НИИ проблем высшей школы. Обзорная информация. Серия: Обучение и коммунистическое воспитание в высших и средних специальных учебных заведениях – М.: 1976. – 66 с.
3. Аванесов В. С. Основы научной организации педагогического контроля в высшей школе / В. С. Аванесов – М. : 1989. – 167 с.
4. Аванесов В. С. Теория и методика педагогических измерений: [Электронный ресурс] / В. С. Аванесов // Материалы публикаций в открытых источниках и Интернет. – 2005. – С. 44-47. – Режим доступа: <http://viperson.ru/data/200812/jbjejbjxjklmjje.pdf>. – Назва з титул. екрану.
5. Аванесов В. С. Композиция тестовых заданий / В. С. Аванесов – М.: АДЕПТ, 1998.
6. Аванесов В.С. Трудность теста и тестовых заданий / В. С. Аванесов // "Управление школой". – 1999. – №40.
7. AI-полная задача [Электронный ресурс] // Материалы із Вікіпедії. – Режим доступа: http://ru.wikipedia.org/wiki/AI-полная_задача. – Назва з титул. екрану.
8. Альшуллер Г. С. О психологии изобретательского творчества / Г. С. Альшуллер, Р. Б. Шапиро // *Вопросы психологии.* — 1956. — № 6. — С. 37-49.
9. Андруховський А. Б. Застосування xml-сервісу для побудови системи педагогічного тестування / А. Б. Андруховський // Матеріали XVI Всеукраїнської наукової конференції «Сучасні проблеми прикладної математики та інформатики». – Львів, 2009. – С. 15-16.

10. Антоник М. С. Інформаційна технологія побудови автоматизованої системи управління навчальним процесом: Автореф. дис. канд. техн. наук: спец. 05.13.06 «Інформаційні технології» / М. С. Антоник. – Львів, 2005. – 20 с.
11. Берко А.Ю. Системи баз даних та знань. Книга 1. Організація баз даних та знань / А. Ю. Берко, О. М. Верес, В. В. Пасічник – Львів: «Магнолія 2006», 2008. – 456 с.
12. Беспалько В. П., Татур Ю. Г. Системно-методическое обеспечение учебно-воспитательного процесса подготовки специалистов / В. П. Беспалько, Ю. Г. Татур. – М.: “Высшая школа”, 1989.
13. Бондаренко М. Технология оценивания тестов в зависимости от типа и уровня сложности тестовых заданий на основе интегрированной модели / Бондаренко М., Семенец В., Білоус Н., Борисенко В., Куцевич И., Білоус И., Мележик О. // Proceedings of the Fourth International Conference "Modern (e-) Learning" MeL 2009. – Varna, Bulgaria, June-July 2009. – P. 55-62.
14. Борисовська Ю. О. Аналіз сучасних платформ дистанційного навчання / Ю. О. Борисовська, О. С. Козлова, О. А. Лисенко // Вісник Херсонського державного технічного університету. – 2010. – № 2 (38). – С. 491–496.
15. Гегель Г. Энциклопедия философских наук / Г. Гегель // 1929. – Т. 1.
16. Глова В. И. Мягкие вычисления (soft computing) и их приложения / В. И. Глова, И. В. Аникин, М. А. Аджели – Казань, 2000.
17. Давыдов В. В. Виды обобщения в обучении / В. В. Давыдов – М.: Педагогика, 1972. – 424 с.
18. Дахин А. Н. Актуальные проблемы оптимального управления образовательным процессом / А. Н. Дахин // Журнал «Педагог». – 1999. – №7.
19. Джонсон У. Д. Суд над системой образования: стратегия на будущее. / У. Д. Джонсон. – М.: Педагогика, 1991. – 264 с.
20. Дистанційне_навчання [Електронний ресурс] // Матеріали із Вікіпедії. –

- Режим доступу: http://uk.wikipedia.org/wiki/Дистанційне_навчання. – Назва з титул. екрану.
21. Дмитриев В. М. Формализованное представление задач для компьютерного моделирования / В. М. Дмитриев, А. Ю. Филиппов, О. Н. Шарова // Вестник Московского городского педагогического университета. – 2004. – №3. – С. 53-59.
 22. Дружинин В. Н. Экспериментальная психология : Учебник для вузов– 2-е изд., доп. / В. Н. Дружинин. – СПб.: Питер, 2003. – 319 с: ил. – (Серия “Учебник для вузов”).
 23. Дуплик С. В. Модели педагогического тестирования / С. В. Дуплик // Казанский Государственный Технический университет им. А.Н. Туполева модели педагогического тестирования
 24. Евреинов Э. В. Информатика и дистанционное образование / Э. В. Евреинов, В. А. Каймин– М.: ВАК, 1998.
 25. Елизаренко Г. Н. Проектирование компьютерных курсов обучения: концепция, язык, структура / Г. Н. Елизаренко. — К.: НТУУ “КПИ”, 2001.
 26. Звіт за навчальний рік 2008-2009 [Електронний ресурс] // Український інститут інформаційних технологій в освіті. - К.: НТТУ «КПІ», 2009. – Режим доступу до файлу: http://www.udec.ntu-kpi.kiev.ua/ua/about-iiite/public/singlerecord.html?tx_wfqbe_pil%5Bid%5D=16. – Назва з титул. екрану.
 27. Зорина Л. Я. Единство двух культур в содержании непрерывного образования / Л. Я. Зорина // Педагогика. – 1998. – № 5. – с. 22-28.
 28. Инструментарий ARIS Версия 4.1 [Електронний ресурс] // Офіційний сайт. – Режим доступу: <http://www.softwareag.com>. – Назва з титул. екрану.
 29. Ким В. С Тестирование учебных достижений / В. С. Ким– Монография. – Уссурийск: Издательство УГПИ, 2007. – 214 с.
 30. Майоров А. Н. Теория и практика создания тестов для систем образования / А. Н. Майоров. – М.: «Интеллект–центр», 2001. – 296 с.

31. Маслак А. А. Измерение латентных переменных в социально-экономических системах / А. А. Маслак // Монография : Славянск-на-Кубани: Изд.центр СГПИ, 2006, – 333 с.
32. Мельник А. М. Автоматична генерація тестових завдань різних типів / А. М. Мельник // Вісник Хмельницького національного університету. – 2010. – № 4. – С. 124–129.
33. Мельник А. М. Автоматична генерація тестових завдань як засіб підвищення ефективності процесу навчання / А. М. Мельник, Р. М. Пасічник, Р. П. Шевчук // Тези доповідей II міжнародної науково-практичної конференції «Інформаційні технології та комп'ютерна інженерія». – Харків, 2011. – С. 57-59.
34. Мельник А. М. Алгоритми генерації множини задач та їх автоматичне розв'язання в прикладному програмному середовищі / А. М. Мельник, Р. М. Пасічник // Вісник Тернопільського національного технічного університету. – 2011. – Т. 16, № 2. – С. 157–163.
35. Мельник А. М. Інформаційна технологія автоматичної генерації тестових завдань з керованою складністю / А. М. Мельник, Р. М. Пасічник, Р. П. Шевчук // Системи обробки інформації. Харків, 2011. – № 3 (93). – С. 57–61.
36. Мельник А. М. Метод адаптивного управління структурою процесу навчання / А. М. Мельник, Р. М. Пасічник // Вісник Хмельницького національного університету. – 2011. – № 1. – С. 132–137.
37. Мельник А. М. Метод генерації проблемних ситуацій для тестування методологічних знань / А. М. Мельник, Р. М. Пасічник // Науковий вісник Чернівецького університету: Комп'ютерні системи та компоненти. – 2010. – Т. 1, Вип. 2. – С. 67 – 72.
38. Мельник А. М. Метод генерації тестових завдань на основі системи семантичних класів / А. М. Мельник, Р. М. Пасічник // Вісник Тернопільського державного технічного університету. – 2010. – Т. 15, № 1. – С. 187–193.

39. Мельник А. М. Методи та засоби автоматичної генерації тестових завдань різних форм / А. М. Мельник // Матеріали 2-ї науково-практичної конференції [”Інноваційні комп’ютерні технології у вищій школі”], (Львів, 23-25 листопада 2010 р.) / М-во освіти і науки України, Національний університет «Львівська політехніка». – Львів: Видавництво Львівської політехніки, 2010. – С. 147-152.
40. Мельник А. М. Моделювання параметрів навчального процесу в умовах невизначеності / А. М. Мельник, Р. М. Пасічник // Поступ в науку: зб. наук. праць за матеріалами проблемно-наукової міжгалузевої конференції [”Інформаційні проблеми комп’ютерних систем, юриспруденції, економіки та моделювання”, (ПНМК–2009)], (Бучач, Скоморохи, Карпати, 19-22 травня 2009 р.) / НАН України, Академія правових наук України [та ін.]. – Бучач: Бучацький інститут менеджменту і аудиту, 2009. – Т. 1., № 5.– С. 216-219.
41. Мельник А. М. Модель оцінки складності тестових завдань / А. М. Мельник, Р. М. Пасічник // Науковий вісник Чернівецького університету: Комп’ютерні системи та компоненти. – 2009. – № 479. – С. 108 – 113.
42. Мельник А. М. Моделювання результативності навчання в інтелектуальних адаптивних навчальних системах / А. М. Мельник, Р. М. Пасічник // Міжнародний науково-технічний журнал "Інформаційні технології та комп’ютерна інженерія". — 2009. — № 3. — С. 107-116.
43. Мельник А. М. Система автоматичної генерації тестових завдань / А. М. Мельник, Р. М. Пасічник // Тези доповідей міжнародної науково-практичної конференції [”Інформаційні технології та комп’ютерна інженерія ”, (ІТКІ–2010)], (Вінниця, Україна, 19-21 травня 2010 р.) / М-во освіти і науки України, Вінницький національний технічний університет [та ін.]. – Вінниця: Вінницький національний технічний університет, 2010. – С. 408-409.
44. Мельник А. М. Система кількісних показників оцінки ефективності

- технологій автоматизованого навчання / А. М. Мельник // Матеріали І Всеукраїнської школи семінару молодих вчених і студентів «Сучасні комп'ютерні інформаційні технології». – Тернопіль, 2011. – С. 128-129.
45. Метод [Електронний ресурс] // Матеріали із Вікіпедии. – Режим доступу: <http://ru.wikipedia.org/wiki/Метод>. – Назва з титул. екрану.
46. Методическая теория: уточнение понятий. / Рыжова Н. И. // Проблемы и перспективы развития методики обучения математике. Сб. науч. работ. Под ред. В.В. Орлова. СПб: Изд-во РГПУ им. А.И. Герцена, 1999. – С.3-9.
47. Модели представления знаний [Електронний ресурс] // Матеріал из Портала искусственного интеллекта. – Режим доступу: <http://www.aiportal.ru/articles/knowledge-models/knowledge-odels.html>. – Назва з титул. екрану.
48. Модульне динамічне об'єктно-орієнтоване середовище для навчання [Електронний ресурс] // Система дистанційної освіти при Києво-Могилянській Академії. – Режим доступу до файлу: <http://moodle.ukma.kiev.ua/mod/resource/view.php?inpopup=trae&id=72>. – Назва з титул. екрану.
49. Науковий звіт E-Learning платформи підтримки дистанційного навчання (аналіз і порівняльна оцінка) [Електронний ресурс] // Український інститут інформаційних технологій в освіті. – К.: Національний Технічний Університет України «КПІ», 2004. – Режим доступу: http://www.udec.ntu-kpi.kiev.ua/ua/aboutuiite/public/single_record.html?tx_wfqbe_pil%5Bid%5D=5. – Назва з титул. екрану.
50. Нехаев И. Н. Постановка задачи эффективного адаптивного тестирования уровня знаний / И. Н. Нехаев // Вестник Московского городского педагогического университета, серия “Информатика и информатизация образования”. – 2008. – № 15. – С. 124–127.
51. Носов П. С. Інтелектуальне формування індивідуальної траєкторії навчання студента : Автореф. дис. канд. техн. наук: спец. 05.13.23 «Системи та засоби штучного інтелекту» / П. С. Носов. – Одеса, 2007. –

- 17 с.
52. Образование в документах: Информационный бюлетень. – 1997. – №1 (40). – С. 36.
53. Оксамитна Л. П. Методи та засоби самоорганізації моделі знань в автоматизованих системах контролю знань та навчання: Автореф. дис. канд. техн. наук: 05.13.06 / Черкаський державний технологічний університет. – Черкаси, 2003. – 18 с.
54. ОРОКС [Електронний ресурс] // Московский областной центр новых информационных технологий. – Режим доступа: <http://mosnit.ra/mosnit/oroks.html>. – Назва з титул. екрану.
55. Освітні інтернет-системи та моделювання знань [Електронний ресурс] / С. В. Титенко // Лабораторія СЕТ. – 2006. – Режим доступа : http://www.setlab.net/?view=AIED_Overview. – Назва з титул. екрану.
56. Пасічник Р. М. Ідентифікація моделі броварного бродіння в умовах невизначеності / Пасічник Р. М., Піговський Ю. Р. // Вісник Вінницького політехнічного інституту. — Вінниця, 2007. — № 1. — С. 10–15.
57. Пасічник Р. М. Формалізація процесу побудови онтологій на основі базових класів / Р. М. Пасічник, А. С. Саченко, А. М. Мельник // Збірник наукових праць за матеріалами XIII всеукраїнської науково конференції [”Сучасні проблеми прикладної математики та інформатики ”], (Львів, 3-5 жовтня 2006 р.) / М-во освіти і науки України, Львівський національний університет імені Івана Франка. – Львів: Видавничий центр ЛНУ ім. Івана Франка, 2006. – С. 162-163.
58. Попов Д. И. Способ оценки знаний в дистанционном обучении на основе нечетких отношений / Д. И. Попов // Дистанционное образование. – 2000. – № 6.
59. Применение алгоритма анализа творческих задач для решения проблем преподавания курса на основе ОТСМ-ТРИЗ [Електронний ресурс] // Корзун А. В. – Режим доступа: <http://www.trizminsk.org/e/prs/232049.htm>

– Назва з титул. екрану.

60. Про порядок розробки складових нормативного та навчально-методичного забезпечення підготовки фахівців з вищою освітою. Наказ міністерства освіти України №285 від 31.07.1998., – С.10-11.
61. Проблема [Електронний ресурс] // Матеріали із Вікіпедії. – Режим доступу: [http://ru.wikipedia.org/wiki/ Проблема](http://ru.wikipedia.org/wiki/Проблема). – Назва з титул. екрану.
62. Проблемная ситуация [Електронний ресурс] // Матеріали Центра Креативных Технологий. – Режим доступу: <http://ru.wikipedia.org/wiki/http://www.inventech.ru/lib/glossary/problemsit/>. – Назва з титул. екрану.
63. Психологический словарь [Електронний ресурс]. – Режим доступу: <http://azps.ru/handbook/>. – Назва з титул. екрану.
64. Распопов В. М. Программирование и организация самостоятельной работы учащихся / В. М. Распопов. – М. : Высшая школа, 1965.
65. Реляционная модель данных [Електронний ресурс] // Матеріали із Вікіпедії. – Режим доступу: http://wiki.mvтом.ru/index.php/Реляционная_модель_данных. – Назва з титул. екрану.
66. Реляционная модель данных [Електронний ресурс] // Матеріали із Вікіпедії. – Режим доступу: http://ru.wikipedia.org/wiki/Реляционная_модель_данных. – Назва з титул. екрану.
67. Реляционная модель данных для больших совместно используемых банков данных [Електронний ресурс] / Кодд Е. Ф. – Режим доступу: <http://citforum.ru/database/classics/codd/>. – Назва з титул. екрану.
68. Самылкина Н. Н. Современные средства оценивания результатов обучения / Н. Н. Самылкина. — М.: БИНОМ. Лаборатория знаний, 2007. — 172 с.
69. СДО «ДОЦЕНТ» [Електронний ресурс] // СДО. UniaR. – Режим доступу: <http://проху.uniar.ru/www/dt-docent.html>. – Назва з титул. екрану.
70. Система дистанционного обучения «Прометей 4.2» [Електронний ресурс] // Прометей. – Режим доступу: <http://www.prometeus.ru>. – Назва з титул. екрану.

71. Системы поддержки обучения [Электронный ресурс] // Центр внедрения систем электронного обучения Киевского университета им. Тараса Шевченко. – Режим доступа: <http://www.dl.com.ua/ras/techno/lms.shtm>. – Назва з титул. екрану.
72. Снижко Е. А. Методика применения экспертных систем для корректировки процесса обучения и оценки эффективности ППС: Дис. канд. пед. наук: 13.00.02 / Е. С. Снижко. – Санкт-Петербург, 1997. – 161 с.
73. Снижко Е. А. Методика применения экспертных систем для корректировки процесса обучения и оценки эффективности ППС: Дис. канд. пед. наук: 13.00.02 . – Санкт-Петербург, 1997. – 161 с.
74. Соловов А. В. Проектирование компьютерных систем учебного назначения: учебное пособие / Соловов А. В. – Самара: СГАУ, 1995.
75. Стан розвитку дистанційного навчання в Україні (станом на 11.02.2008) [Електронний ресурс] // Український інститут інформаційних технологій в освіті. – К. : Національний Технічний Університет України «КПІ», 2008. – Режим доступа: http://www.udec.ntu-kpi.kiev.ua/ua/aboutiite/public/singlerecord.html?tx_wfqbe_pil%5Bid%5D=14. – Назва з титул. екрану.
76. Структура тестового задания [Электронный ресурс] // Жуланов С.Г. – Режим доступа: http://www.psu.ru/pub/xxi/2_1_4.rtf. – Назва з титул. екрану.
77. Сущность и структура процесса обучения. Этапы учебного процесса [Электронный ресурс] // Педагогика. – Режим доступа: <http://paidagogos.com/>. – Назва з титул. екрану.
78. Теория решения изобретательских_задач [Электронный ресурс] // Материалы із Википедии. – Режим доступа: http://ru.wikipedia.org/wiki/Теория_решения_изобретательских_задач. – Назва з титул. екрану.
79. Тесленко В. И., Сосновский В. И Методика составления пробного педагогического теста / В. И. Тесленко, В. И Сосновский // Сибирский обозревательный журнал. – 2002. – № 2.

80. Тесты: общая характеристика и дидактический аспект [Электронный ресурс] // Педагогика. – Режим доступа: <http://paidagogos.com/>. – Назва з титул. екрану.
81. Титенко С. В. Генерація тестових завдань у системі дистанційного навчання на основі моделі формалізації дидактичного тексту / С. В. Титенко // Наукові вісті Національного технічного університету України Київський політехнічний інститут". – 2009. – № 1(63). – С. 47–57.
82. Титенко, С. В. Моделювання області знань в системі безперервного навчання на основі інтеграції моделі контенту Tree-Net і понятійно-тезисної моделі / С. В. Титенко, О. О. Гагарін // VIII міжнародная конференція «Інтелектуальний аналіз інформації ІАІ-2008», Київ, 14-17 мая 2008г. : Сб. тр./ Ред. кол. : С.В. Сирота (гл.ред.) и др. – К.: Просвіта, 2008. – С. 475–484.
83. Хоа Тат Тханг Сравнительный анализ систем дистанционного обучения / Хоа Тат Тханг // Общие проблемы образования. – 2009. – №2. – С. 9-13.
84. Чеботарева Н. Е. Универсальный комплекс тестовых заданий различных уровней сложности как эффективное средство систематической оценки качества знаний студентов по физике / Н. Е. Чеботарева, В. А. Федорихин, В. М. Симонов, А. В. Шильников, Л. В. Жога // Физическое образование в вузах. – 2003. – Т.9, №2. – С.45-53.
85. Чельшкова М. Б. Теория и практика конструирования педагогических тестов: Учебное пособие / Чельшкова М. Б. – М. : Логос, 2002. – 432 с.
86. Чельшкова М.Б. Теория и практика конструирования педагогических тестов: Учебное пособие / М. Б. Чельшкова.– М.: Логос, 2002. – 432 с.
87. Электронная хрестоматия по методике преподавания математики [Электронный ресурс]. – Режим доступа: <http://fmi.asf.ru/Library/Book/Mpm/9b.html>. – Назва з титул. екрану.
88. An adaptive Web-based learning system with a free-hyperlink environment / Mitsuvara H., Ochi Y., Kanenishi K., Yano Y. // Proceedings of Workshop on Adaptive Systems for Web-Based Education at the 2nd International

- Conference on Adaptive Hypermedia and Adaptive Web-Based Systems, AH'2002. – Málaga, Spain, 2002. – P. 81-91.
89. Berners-Lee T. Spinning the Semantic Web / Berners-Lee T. // Bringing the World Wide Web to Its Full Potential — The MIT Press, 2005.
 90. Berners-Lee T. The Semantic Web // T. Berners-Lee, J. Hendler, O. Lassila // Scientific American. – 2001. – № 284, 5. – P. 34-43.
 91. Blackboard+learn [Электронный ресурс] // Learn platform. – Режим доступа: <http://www.blackboard.com/Teaching-Learning/Overview.aspx>. – Назва з титул. екрану.
 92. Brusilovsky P. Preface to special issue on user modeling for Web information retrieval / P. Brusilovsky, C. Tasso // User Modeling and User Adapted Interaction. – 2003. – № 14 (2-3). – P. 147-157.
 93. Clancey W. J. Knowledge-based tutoring: The GUIDON program / Clancey W. J. – Cambridge, MA: The MIT Press, 1987.
 94. Claroline LMS [Электронный ресурс] // Офіційний сайт. – Режим доступа: <http://claroline-lms.ru>. – Назва з титул. екрану.
 95. Claroline.net [Электронный ресурс] // Офіційний сайт. – Режим доступа: <http://claroline.net>. – Назва з титул. екрану.
 96. eLearning Server: Платформа для организации дистанционного и смешанного обучения [Электронный ресурс] // Hyper method IBS. – Режим доступа: <http://learnware.m/static.php?id=3010>. – Назва з титул. екрану.
 97. eLearning Server v2.1 [Электронный ресурс] // eLEARN.RU - Учебный центр. – Режим доступа: http://www.elearn.ru/help/help_index.php4?rub=index. – Назва з титул. екрану.
 98. Gulliksen H. Theory of mental tests / H. Gulliksen. – N-Y. : Willey, 1950. – 486 pp.
 99. Handbook on Formative and Summative Evaluation of Student Learning / Bloom B.S. – N - Y., McGraw-Hill, 1971. – P. 232.
 100. Herbert A. Department of Psychology, Carnegie Mellon University / Herbert

- A. Simon // Speech presented at the Frontiers in Education Conference. – 1997.
101. ILIAS 4 e-Learning [Электронный ресурс] // ILIAS Learning Management. – Режим доступа: <http://www.ilias.de>. – Назва з титул. екрану.
102. Lord F. M., Novick M. Statistical Theories of Mental Test Scores / F. M. Lord, M. Novick. – Addison-Westley Publ. Co. Reading, Mass., 1968. – 560 pp.
103. Lotus Learning Management System [Электронный ресурс] // Сибирский центр информационных технологий ISIB. – Режим доступа: <http://www.isib.ru/erp/lotus/lms.php>. – Назва з титул. екрану.
104. McArthur D. The roles of artificial intelligence in education: Current progress and future prospects / D. McArthur, M. Lewis, M. Bishay. – RAND DRU-472-NSF, 1993.
105. McArthur D. The roles of artificial intelligence in education: Current progress and future prospects / D. McArthur, M. Lewis, M. Bishay. – RAND DRU-472-NSF, 1993.
106. Melnyk A. System of semantic classes for test's generation. / A. Melnyk, R. Pasichnyk // Modern Problems of Radio Engineering, Telecommunications and Computer Science. Proceedings of the International Conference. TCSET'2010. – Lviv-Slavsko, Ukraine, 2010. – P. 206-207
107. MLTutor: An Application of Machine Learning Algorithms for an Adaptive Web-based Information System / Smith A. S. G., Blandford, A. // International Journal of Artificial Intelligence in Education. – 2003. – № 13. – P. 233-260. Available online at http://www.cogs.susx.ac.uk/ijaied/abstracts/Vol_13/smith.html. – Назва з титул. екрану.
108. Pasichnyk R. Method of Adaptive Control Structure Learning Based on Model of Test's Complexity. / R. Pasichnyk, A. Melnyk, N. Pasichnyk, I. Turchenko // Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications Proceedings of the 6th IEEE International Conference IDAACS' 2011, Volume 2 (Prague, Czech Republic, 15-17 September 2011) / IEEE Ukraine I&MCI Joint Societies chapter, Research Institute for

- Intelligent Computer Systems [and other]., 2011. – P. 692-695.
109. Pasichnyk R. Modeling of Cognitive Processes for Bio-Technical Systems. / R. Pasichnyk, A. Melnyk // Modern Problems of Radio Engineering, Telecommunications and Computer Science. Proceedings of the International Conference. TCSET'2008. – Lviv-Slavsko, Ukraine, 2008. – P. 27-28
 110. Pasichnyk R. Modeling of effective studies in Adaptive Educational Systems. / R. Pasichnyk, A. Melnyk // The Experience of Designing and Application of CAD Systems in Microelectronics. Proceedings of the International Conference. CADSM '2009. – Lviv-Polyana, Ukraine, 2009. – P. 248-250.
 111. Powerful. Flexible. Open. [Електронний ресурс] // Офіційний сайт Sakai. – Режим доступу: <http://sakaiproject.org>. – Назва з титул. екрану.
 112. Quizzes for Parameterized Assessment of C Knowledge [Електронний ресурс] // QuizPACK. – Режим доступу: <http://www.sis.pitt.edu/~taler/QuizPACK.html>. – Назва з титул. екрану.
 113. Rash G. Probabilistic Model for Some Intelligence and Attainment Tests / Rash G. – Chicago: Univ. of Chicago Press, 1980.
 114. Rash G. Probabilistic Models for Some Intelligence and Attainment Tests / G. Rash. – Copenhagen, Denmark: Danish Institute for Educational Research, 1960.
 115. Shapiro Encyclopedia of Artificial Intelligence. 2nd ed. / Shapiro, Stuart C. – New York: John Wiley & Sons. – 1992. – P. 434.
 116. Shute V. Large-scale evaluation of an intelligent discovery world: SMITHTOWN / V. Shute, R. Glaser // Interactive Learning Environments. – 1990. – № 1. – P. 51 - 77.
 117. Spearman C. Correlation calculated from faulty data / C. Spearman // British Journal of Psychology. – 1910. – Vol. 3, № 2. – P. 271-295.
 118. Stankov S. TEx-Sys model for building intelligent tutoring / S. Stankov, B. Žitko, A. Grubišić // Computers & Education. Volume 51 Issue 3, 2008. – P. 1017 – 1036.
 119. Stankov S. Ontology as a Foundation for Knowledge Evaluation in Intelligent

- E-learning Systems / S. Stankov, B. Žitko, Grubišić A. // International Workshop on Applications of Semantic Web Technologies for E-Learning (SW-EL'05) in conjunction with 12th International Conference on Artificial Intelligence in Education (AI-ED 2005). – Amsterdam, Netherlands, 2005. – P. 81 - 84.
120. Wilson M. Constructing Measures: An Item Response Modeling Approach / M. Wilson. – Mahwah, New Jersey: Lawrence Erlbaum associates, 2005. – 228 p.
121. Wright B. D. Best Test Design / B. D. Wright, M. N. Stone. – Chicago: MESA Press, 1979.
122. Wright B. D. Rating scale analysis. Rasch measurements / B. D. Wright, M. N. Stone. Best Test Design. – Chicago: MESA Press, 1982.

Додаток А

Основні операції класичної алгебри відношень (реляційної алгебри)

№п/п	Назва операції	Запис реляційної операції
1.	Вибірка σ	$R_1 = \sigma_C(R_2)$, де R_1, R_2 – відповідні відношення, C – це умова, яка включає атрибути R_2 ;
2.	Проекція π	$R_1 = \pi_L(R_2)$, де L – це список атрибутів відношення R_2 ;
3.	Добуток \times	$R = R_1 \times R_2$, де кожний кортеж відношення R_1 з'єднується з кожним кортежем відношення R_2 і ці кортежі додаються у відношення R ;
4.	З'єднання θ	$R = R_1 \triangleright_C \triangleleft R_2$, ця операція еквівалентна $R_1 = \sigma_C(R_1 \times R_2)$;
5.	Природне з'єднання $\triangleright \triangleleft$	$R = R_1 \triangleright \triangleleft R_2$, виконується θ - з'єднання відношень R_1 та R_2 з умовою рівності атрибутів з однаковими іменами. В результат буде включено один стовпчик для кожної пари однакових атрибутів;
6.	Перейменування ρ	$\rho_{S(A_1 \dots A_n)}(R)$, створюється відношення ідентичне до відношення R , але з іменем S і з атрибутами, порядок та імена яких визначені A_1, \dots, A_n ;
7.	Оператор усунення дублікатів δ	$\delta(R)$, відношення, в якому є тільки один кортеж з декількох однакових, які є в R ;
8.	Сортування τ	$\tau_L(R)$, список кортежів R , відсортованих по значеннях атрибутів із списку L ;
9.	Розширена проекція π	в якості стовця в проекції допускається функція, яка має в якості аргумента один або декілька стовпців відношення R
10.	Групування γ	$\lambda_L(R)$, де L – список елементів, кожний з яких: а) групує атрибут; б) $\theta(A)$, де θ – оператор агрегування, а A – атрибут, в якому оператор застосовується
11.	Зовнішнє з'єднання $\triangleright^{\circ} \triangleleft$	Використовується для уникнення втрати кортежів в результаті виконання природнього з'єднання.

Додаток Б

Системи дистанційного навчання

1	LRN	24	Educator	47	LON-CAPA 1.0
2	ANGEL 5.6	25	EduSystem	48	LON-CAPA1.1
3	ANGEL 6.0	26	Eledge 1.2	49	Manhattan Virtual Classroom 2.1
4	Anion 4.1	27	Eledge3.1	50	MimerDesk 1.5.3.1
5	ATutor 1.3	28	Embanet hosting ANGEL	51	MimerDesk 2.0.1
6	ATutor 1.4	29	Embanet hosting BlackBoard	52	Moodle 2.0.2
7	Avilar WebMentor 4.0	30	Embanet hosting FirstClass	53	Teknical Virtual Campus
8	Bazaar 7	31	Embanet hosting IntraLearn	54	TeleTop
9	BlackBoard 5.5	32	Embanet hosting WebCT	55	The Learning Manager 3.2
10	BlackBoard 6	33	ETUDES	56	Unicon Academus
11	Bodington	34	FirstClass 7.0	57	Virtual-U 2.5
12	BSCW 4.0.6	35	Fle3	58	WebCT 3.6 Standard Edition
13	CentraOne 6.0	36	Groove Workspace 2.5	59	WebCT 3.7 Campus Edition
14	CHEF	37	HTMLeZ	60	WebCT 3.8 Campus Edition
15	Claroline 1.2.0	38	ILIAS	61	WebCT 4.0 Campus Edition
16	Claroline 1.4	39	Internet Course Assistant 2.0	62	WebCT 4.1 Campus Edition
17	ClassWeb 2.0	40	IntraLearn SME 3.1.2	63	WebCT Vista 1.2
18	Click2learn Aspen 2.0	41	Janison Toolbox 5.81	64	WebCT Vista 2.1
19	Colloquia 1.3.2	42	Janison Toolbox 6.2	65	Whiteboard 1.0.2
20	COSE 2.051	43	Jenzabar Internet Campus Solution 1.0	66	NauLearning
21	Coursemanager	44	Jones e-education V2004	67	eCollege AU+
22	CourseWork	45	KEWL 1.2	68	Learnwise
23	Desire2Learn	46	KnowEdge eLearning Suite	69	«Віртуальний університет»

Додаток В

Лістинг модуля автоматичної генерації тестових завдань різних видів закритого типу

```

<?php
session_start();
include("mysql.php");
include("config.php");
    // Дисципліна, яка вибрана при реєстрації

    $subj = $_GET['subj'];
    IF ($subj=='all subjects')
    {
        $subjid=0;
    }

    else
    {
        $subjid = mysql_query("SELECT id FROM articles_cat WHERE name_cat ='$subj'");
        $subjid = mysql_result ($subjid,0);
    }
    $_SESSION['qweSubj'] =$subj;
// встановлення обмежень для тестування по вибраній темі
IF ($subjid<>0)
{
    $subj_test_str='AND definition.Id_subject=';
    $subj_test=$subj_test_str. " " . $subjid;
    $subj_test_str_d='AND t2.Id_subject=';
    $subj_test_d=$subj_test_str_d. " " . $subjid;

    //$subj_test="";
}
else
{
    $subj_test="";
}
// кінець встановлення обмежень для тестування по вибраній темі
$max = mysql_query("SELECT id FROM definition WHERE pid=0 $subj_test ORDER BY RAND()
LIMIT 1" );
    $max = mysql_result ($max,0);
    $test = mysql_query ("SELECT * FROM definition WHERE definition.pid=0 AND id=$max
$subj_test");
//echo "ID SES= ", session_id());
?>
<html>
<title>Тестування</title>
<body background="/images/fon.jpg">
<center>
<h2>Згенеровані тестові завдання</h2>
<?php
        echo "<h3> Дисципліна '", $subj,'"</h3> ";
?>
<hr color=green WIDTH="50%" SIZE="5">
</center>
<form action="testresult.php" method="POST" target="_blank">
<ol>
<li> Чи вірне наступне твердження:

```



```

<strong>
<?php
// 7 тест
// альтернативний тест №1

        $max7err1 = mysql_query("SELECT id FROM definition WHERE pid=0 $subj_test ORDER BY
RAND() LIMIT 1");
        $max7err1= mysql_result ($max7err1,0);
        $selem1=0;
// перевірка кількості елементів дерева
$countelem1 = mysql_query("SELECT COUNT(id) AS count FROM definition WHERE (id=$max7err1 OR
pid=$max7err1 ) $subj_test UNION SELECT count(t1.id) FROM definition t1 JOIN definition t2 ON t1.pid = t2.id
WHERE t2.pid = $max7err1 ");
//$countelem = mysql_result ($countelem1,1);
while ($stes7count1 = mysql_fetch_array($countelem1))
    {
        $selem1=$selem1+$stes7count1['count'];
    }
while ($selem1<3)
    {
        $max7err1 = mysql_query("SELECT id FROM definition WHERE pid=0 $subj_test ORDER BY RAND() LIMIT 1");
        $max7err1 = mysql_result ($max7err1,0);
        $selem1=0;
        $countelem1 = mysql_query("SELECT COUNT(id) AS count FROM definition WHERE (id=$max7err1 OR
pid=$max7err1 ) $subj_test UNION SELECT count(t1.id) FROM definition t1 JOIN definition t2 ON t1.pid = t2.id
WHERE t2.pid = $max7err1 ");
        // $countelem = mysql_result ($countelem1,1);
        while ($stes7count1 = mysql_fetch_array($countelem1))
            {
                $selem1=$selem1+$stes7count1['count'];
            }
    }
// кінець перевірка кількості елементів дерева
//echo "KILKIST=", $selem;
$max7oper1= array('<>','=' );
$max7oper1 = $max7oper1[array_rand($max7oper1)];
//setcookie($FirstOper,$max7oper);
$_SESSION['qwe1'] =$max7oper1;
//echo "OPERACIYA=", $max7oper;
$stest7err1= mysql_query ("SELECT * FROM definition WHERE (id=$max7err1 OR pid=$max7err1) $subj_test");
while ($stes7err1 = mysql_fetch_array($stest7err1))
    {
        echo $stes7err1['CS'], " ", $stes7err1['CNJ'], " " ;
        IF ($stes7err1['id']>$stes7err1['id'-1])
            {
                $stesmax7err1=$stes7err1['id'];
            }
        $stc7err1=$stes7err1['CNJ'];
    }

//echo "br=", $br;
$stcc7err1=$stc7err1; $smc7err1 = mysql_query("SELECT cs FROM definition WHERE pid=0.1 $subj_test UNION
SELECT t1.cs
FROM definition t1 JOIN definition t2 ON t1.pid = t2.id WHERE t1.pid $max7oper1 $stesmax7err1 AND
t2.cnj='$stc7err1' $subj_test_d,")or die(mysql_error()); $stes127err1 = mysql_result ($smc7err1,0); echo $stes127err1;
// кінець альтернативний тест
?></strong><br />
<input type="radio" name="rdi[перше]" value="1">Так<br>
<input type="radio" name="rdi[перше]" value="2">Hi<br>
</li>
<hr color=red WIDTH="80%" align="left">
<li><strong><?php
$max = mysql_query("SELECT id FROM definition WHERE pid=0 $subj_test ORDER BY RAND() LIMIT 1");

```

```

$max = mysql_result ($max,0);

$selem2=0;
// перевірка кількості елементів дерева
$countelem2 = mysql_query("SELECT COUNT(id) AS count FROM definition WHERE (id=$max OR pid=$max )
$subj_test UNION SELECT count(t1.id) FROM definition t1 JOIN definition t2 ON t1.pid = t2.id WHERE t2.pid =
$max $subj_test_d" );
//$countelem = mysql_result ($countelem,1);
while ($tes7count2 = mysql_fetch_array($countelem2))
{
    $selem2=$selem2+$tes7count2['count'];
}
while ($selem2<3)
{
    $max = mysql_query("SELECT id FROM definition WHERE pid=0 $subj_test ORDER BY RAND() LIMIT 1" );
    $max = mysql_result ($max,0);
    $selem2=0;
    $countelem2 = mysql_query("SELECT COUNT(id) AS count FROM definition WHERE (id=$max OR pid=$max )
    $subj_test UNION SELECT count(t1.id) FROM definition t1 JOIN definition t2 ON t1.pid = t2.id WHERE t2.pid =
    $max $subj_test_d" );
    //$countelem = mysql_result ($countelem,1);
    while ($tes7count2 = mysql_fetch_array($countelem2))
    {
        $selem2=$selem2+$tes7count2['count'];
    }
}

$stest = mysql_query ("SELECT * FROM definition WHERE (id=$max OR pid=$max) $subj_test");
while ($stes = mysql_fetch_array($stest))
{
    echo $stes['CS']," ", $stes['CNJ'] ;
    IF ($stes['id']>$stes['id'-1])
    {
        global $stesmax;
        global $stc;
        $stesmax=$stes['id'];
    }
    $stc=$stes['CNJ'];
}
?>

</strong><br />
<input type="radio" name="rdi[дпуге]" value="1"><?php $m = mysql_query("SELECT cs FROM definition WHERE
pid=$stesmax $subj_test"); $m = mysql_result ($m,0); echo $m; ?><br>
<input type="radio" name="rdi[дпуге]" value="2"><?php $stcc=$stc; $mc = mysql_query("SELECT cs,id FROM
definition WHERE pid=0.1 UNION SELECT t1.cs,t1.id
FROM definition t1 JOIN definition t2 ON t1.pid = t2.id WHERE t1.pid<>$stesmax AND t2.cnj='$stc' $subj_test_d;")or
die(mysql_error()); $stes12 = mysql_result ($mc,0); echo $stes12 ?><br>
<input type="radio" name="rdi[дпуге]" value="3"><?php $stcc1=$stc1; $mcl = mysql_query("SELECT cs FROM
definition WHERE pid=0.1 $subj_test UNION SELECT t1.cs
FROM definition t1 JOIN definition t2 ON t1.pid = t2.id WHERE t1.pid<>$stesmax $subj_test_d AND t1.cs<>'$stes12'
AND t2.cnj='$stc;')or die(mysql_error()); $stes121 = mysql_result ($mcl,0); echo $stes121 ?><br>
</li>
<hr color=red WIDTH="80%" align="left">
<li><strong>Встановіть правильні відповідності
<table>
<tr>
<td>
<?php $max8 = mysql_query("SELECT id FROM definition WHERE pid=0 $subj_test ORDER BY RAND() LIMIT
1" );
$max8 = mysql_result ($max8,0);
$stest8 = mysql_query ("SELECT * FROM definition WHERE id=$max8 $subj_test");
while ($stes8 = mysql_fetch_array($stest8))

```

```

{
echo "1. ", $tes8['CS'], " ", $tes8['CNJ'];
    $tesmax8=$tes8['id'];
    $tc8=$tes8['CNJ'];
}
//echo $tesmax5;
?></strong>
<br />
</td>
<td>
<select name="rdi[тперѐ]" size = 1>
<option value="1"><?php
$max82 = mysql_query("SELECT id FROM definition WHERE pid=0 AND id<>$max8 $subj_test ORDER BY
RAND() LIMIT 1");
$max82 = mysql_result ($max82,0);
$tc82=$tc8; $mc82 = mysql_query("SELECT CS,CNJ FROM definition WHERE pid = $max82 $subj_test UNION
SELECT t1.CS,t1.CNJ FROM definition t1 JOIN definition t2 ON t1.pid = t2.id WHERE t2.pid = $max82
$subj_test_d ") or die(mysql_error());
while ($test812 = mysql_fetch_array($mc82))
{ echo $test812['CS'], " ", $test812['CNJ'], " "; } ?></option>
<option value="2"><?php $tc8=$tc8; $mc8 = mysql_query("SELECT CS,CNJ FROM definition WHERE pid =
$max8 $subj_test UNION SELECT t1.CS,t1.CNJ FROM definition t1 JOIN definition t2 ON t1.pid = t2.id WHERE
t2.pid = $max8 $subj_test_d") or die(mysql_error());
while ($test81 = mysql_fetch_array($mc8))
{ echo $test81['CS'], " ", $test81['CNJ'], " "; } ?></option>
<option value="3"><?php
$max83 = mysql_query("SELECT id FROM definition WHERE pid=0 AND id<>$max8 AND id<>$max82 $subj_test
ORDER BY RAND() LIMIT 1");
$max83 = mysql_result ($max83,0);
$tc83=$tc8; $mc83 = mysql_query("SELECT CS,CNJ FROM definition WHERE pid = $max83 $subj_test UNION
SELECT t1.CS,t1.CNJ FROM definition t1 JOIN definition t2 ON t1.pid = t2.id WHERE t2.pid = $max83
$subj_test_d ") or die(mysql_error());
while ($test813 = mysql_fetch_array($mc83))
{ echo $test813['CS'], " ", $test813['CNJ'], " "; } ?></option>
</select>
</td>
</tr>

<tr>
<td>
<?php
$max84 = mysql_query("SELECT id FROM definition WHERE id=$max82 $subj_test" );
$max84 = mysql_result ($max84,0);
$test8 = mysql_query ("SELECT * FROM definition WHERE id=$max84 $subj_test");
while ($tes8 = mysql_fetch_array($test8))
{
echo "2. ", $tes8['CS'], " ", $tes8['CNJ'];
    $tesmax8=$tes8['id'];
    $tc8=$tes8['CNJ'];
}
?></strong>
<br />
</td>
<td>
<select name="rdi[четверте]" size = 1>
<option value="1"><?php
$max85 = mysql_query("SELECT id FROM definition WHERE pid=0 AND id=$max83 $subj_test" );
$max85 = mysql_result ($max85,0);
$tc85=$tc8; $mc85 = mysql_query("SELECT CS,CNJ FROM definition WHERE pid = $max85 $subj_test UNION
SELECT t1.CS,t1.CNJ FROM definition t1 JOIN definition t2 ON t1.pid = t2.id WHERE t2.pid = $max85
$subj_test_d") or die(mysql_error());
while ($test8125 = mysql_fetch_array($mc85))
{ echo $test8125['CS'], " ", $test8125['CNJ'], " "; } ?></option>

```

```

<option value="2"><?php $tcc8=$tc8; $mc8 = mysql_query("SELECT CS,CNJ FROM definition WHERE pid =
$max8 $subj_test UNION SELECT t1.CS,t1.CNJ FROM definition t1 JOIN definition t2 ON t1.pid = t2.id WHERE
t2.pid = $max8 $subj_test_d") or die(mysql_error());
while ($test81 = mysql_fetch_array($mc8))
{ echo $test81['CS'], " ", $test81['CNJ'], " "; } ?></option>
<option value="3"><?php
$max83 = mysql_query("SELECT id FROM definition WHERE pid=0 AND id<>$max8 AND id=$max82 $subj_test"
);
$max83 = mysql_result ($max83,0);
$tcc83=$tc8; $mc83 = mysql_query("SELECT CS,CNJ FROM definition WHERE pid = $max83 $subj_test UNION
SELECT t1.CS,t1.CNJ FROM definition t1 JOIN definition t2 ON t1.pid = t2.id WHERE t2.pid = $max83
$subj_test_d") or die(mysql_error());
while ($test813 = mysql_fetch_array($mc83))
{ echo $test813['CS'], " ", $test813['CNJ'], " "; } ?></option>
</select>
</td>
</tr>
<tr>
<td>
<?php
$max86 = mysql_query("SELECT id FROM definition WHERE id=$max85 $subj_test" );
$max86 = mysql_result ($max86,0);
$test8 = mysql_query ("SELECT * FROM definition WHERE id=$max86 $subj_test");
while ($tes8 = mysql_fetch_array($test8))
{
echo "3. ", $tes8['CS'], " ", $tes8['CNJ'] ;
    $tesmax8=$tes8['id'];
    $tc8=$tes8['CNJ'];
}
?></strong>
<br />
</td>
<td>
<select name="rdi[пј�те]" size = 1>
<option value="1"><?php $tcc8=$tc8; $mc8 = mysql_query("SELECT CS,CNJ FROM definition WHERE pid =
$max8 $subj_test UNION SELECT t1.CS,t1.CNJ FROM definition t1 JOIN definition t2 ON t1.pid = t2.id WHERE
t2.pid = $max8 $subj_test_d") or die(mysql_error());
while ($test81 = mysql_fetch_array($mc8))
{ echo $test81['CS'], " ", $test81['CNJ'], " "; } ?></option>
<option value="2"><?php
$max88 = mysql_query("SELECT id FROM definition WHERE pid=0 AND id=$max85 $subj_test" );
$max88 = mysql_result ($max88,0);
$tcc83=$tc8; $mc83 = mysql_query("SELECT CS,CNJ FROM definition WHERE pid = $max88 $subj_test UNION
SELECT t1.CS,t1.CNJ FROM definition t1 JOIN definition t2 ON t1.pid = t2.id WHERE t2.pid = $max88
$subj_test_d") or die(mysql_error());
while ($test813 = mysql_fetch_array($mc83))
{ echo $test813['CS'], " ", $test813['CNJ'], " "; } ?></option>
<option value="3"><?php
$max87 = mysql_query("SELECT id FROM definition WHERE pid=0 AND id=$max83 $subj_test " );
$max87 = mysql_result ($max87,0);
$tcc82=$tc8; $mc82 = mysql_query("SELECT CS,CNJ FROM definition WHERE pid = $max87 $subj_test UNION
SELECT t1.CS,t1.CNJ FROM definition t1 JOIN definition t2 ON t1.pid = t2.id WHERE t2.pid = $max87
$subj_test_d ") or die(mysql_error());
while ($test812 = mysql_fetch_array($mc82))
{ echo $test812['CS'], " ", $test812['CNJ'], " "; } ?></option>
</select>
</td>
</tr>
<table>
<br>
</li>
<hr color=red WIDTH="80%" align="left">

```

```

<li><strong><?php $max2 = mysql_query("SELECT id FROM definition WHERE pid=0 $subj_test ORDER BY
RAND() LIMIT 1" );
$max2 = mysql_result ($max2,0);
$stest2 = mysql_query ("SELECT * FROM definition WHERE id=$max2 $subj_test");
while ($stes2 = mysql_fetch_array($stest2))
{
echo $stes2['CS'], " ", $stes2['CNJ'] ;
    $stesmax2=$stes2['id'];
    $stc2=$stes2['CNJ'];
}
?></strong><br />
<input type="radio" name="rdi[woocre]" value="1"><?php $stcc21=$stc21; $smc21 = mysql_query("SELECT cs FROM
definition WHERE pid=0.1 $subj_test UNION SELECT t1.cs
FROM definition t1 JOIN definition t2 ON t1.pid = t2.id WHERE t1.pid<>$stesmax2 AND t2.cnj='$stc2'
$subj_test_d;")or die(mysql_error()); $stes1221 = mysql_result ($smc21,0); echo $stes1221 ?><br>
<input type="radio" name="rdi[woocre]" value="2"><?php $stcc22=$stc22; $smc22 = mysql_query("SELECT cs FROM
definition WHERE pid=0.1 $subj_test UNION SELECT t1.cs
FROM definition t1 JOIN definition t2 ON t1.pid = t2.id WHERE t1.pid<>$stesmax2 AND t2.cnj='$stc2'
$subj_test_d;")or die(mysql_error()); $stes1222 = mysql_result ($smc22,1); echo $stes1222 ?><br>
<input type="radio" name="rdi[woocre]" value="3"><?php $m23 = mysql_query("SELECT * FROM definition
WHERE id IN (SELECT id FROM definition WHERE pid=$stesmax2 $subj_test UNION SELECT t1.id FROM
definition t1 JOIN definition t2 ON t1.pid = t2.id WHERE t2.pid=$stesmax2 $subj_test_d
)");
while ($stes23 = mysql_fetch_array($m23))
{
echo $stes23['CS'], " ", $stes23['CNJ'], " ";
}
?><br>
</li>
<hr color=red WIDTH="80%" align="left">
<li><strong>
<?php
$max4 = mysql_query("SELECT id FROM definition WHERE pid=0 $subj_test ORDER BY RAND() LIMIT 1" );
$max4 = mysql_result ($max4,0);
$selem4=0;
// перевірка кількості елементів дерева
$countelem4 = mysql_query("SELECT COUNT(id) AS count FROM definition WHERE (id=$max4 OR pid=$max4 )
$subj_test UNION SELECT count(t1.id) FROM definition t1 JOIN definition t2 ON t1.pid = t2.id WHERE t2.pid =
$max4 $subj_test_d" );
while ($stes7count4 = mysql_fetch_array($countelem4))
    {
        $selem4=$selem4+$stes7count4['count'];
    }
while ($selem4<3)
{
    $max4 = mysql_query("SELECT id FROM definition WHERE pid=0 $subj_test ORDER BY RAND() LIMIT 1" );
    $max4 = mysql_result ($max4,0);
    $selem4=0;
    $countelem4 = mysql_query("SELECT COUNT(id) AS count FROM definition WHERE (id=$max4 OR pid=$max4 )
    $subj_test UNION SELECT count(t1.id) FROM definition t1 JOIN definition t2 ON t1.pid = t2.id WHERE t2.pid =
    $max4 $subj_test_d" );
    while ($stes7count4 = mysql_fetch_array($countelem4))
        {
            $selem4=$selem4+$stes7count4['count'];
        }
}
$stest4= mysql_query ("SELECT * FROM definition WHERE (id=$max4 OR pid=$max4) $subj_test");
while ($stes4 = mysql_fetch_array($stest4))
    {
        echo $stes4['CS'], " ", $stes4['CNJ'] ;
            IF ($stes4['id']>$stes4['id'-1])
                {
                    $stesmax4=$stes4['id'];
                }
    }

```

```

        $tc4=$tes4['CNJ'];
    }
?></strong><br />
<input type="checkbox" name="rdi[сьоме]" value="1"><?php $tcc4=$tc4; $mc4 = mysql_query("SELECT cs FROM
definition WHERE pid=0.1 $subj_test UNION SELECT t1.cs
FROM definition t1 JOIN definition t2 ON t1.pid = t2.id WHERE t1.pid<>$tesmax4 AND t2.cnj='$tc4'
$subj_test_d;")or die(mysql_error()); $tes124 = mysql_result ($mc4,0); echo $tes124 ?><br>
<input type="checkbox" name="rdi[сьоме]" value="2"><?php $m4 = mysql_query("SELECT cs FROM definition
WHERE pid=$tesmax4 $subj_test");
$m4= mysql_result ($m4,0); echo $m4; ?><br>
<input type="checkbox" name="rdi[сьоме]" value="3"><?php $tcc43=$tc43; $mc43 = mysql_query("SELECT cs
FROM definition WHERE pid=0.1 $subj_test UNION SELECT t1.cs
FROM definition t1 JOIN definition t2 ON t1.pid = t2.id WHERE t1.pid<>$tesmax4 AND t2.cnj='$tc4'
$subj_test_d;")or die(mysql_error()); $tes1243 = mysql_result ($mc43,1); echo $tes1243 ?><br>
</li>
<hr color=red WIDTH="80%" align="left">
<li><strong><?php $max5 = mysql_query("SELECT id FROM definition WHERE pid=0 $subj_test ORDER BY
RAND() LIMIT 1" );
$max5 = mysql_result ($max5,0);
$test5 = mysql_query ("SELECT * FROM definition WHERE id=$max5 $subj_test");
while ($tes5 = mysql_fetch_array($test5))
{
echo $tes5['CS'], " ", $tes5['CNJ'] ;
    $tesmax5=$tes5['id'];
    $tc5=$tes5['CNJ'];
}
?></strong><br />
<input type="radio" name="rdi[восьме]" value="1"><?php $tcc51=$tc51; $mc51 = mysql_query("SELECT cs FROM
definition WHERE pid=0.1 $subj_test UNION SELECT t1.cs
FROM definition t1 JOIN definition t2 ON t1.pid = t2.id WHERE t1.pid<>$tesmax5 AND t2.cnj='$tc5'
$subj_test_d;")or die(mysql_error()); $tes1251 = mysql_result ($mc51,0); echo $tes1251 ?><br>
<input type="radio" name="rdi[восьме]" value="2"><?php $tcc52=$tc52; $mc52 = mysql_query("SELECT cs FROM
definition WHERE pid=0.1 $subj_test UNION SELECT t1.cs
FROM definition t1 JOIN definition t2 ON t1.pid = t2.id WHERE t1.pid<>$tesmax5 AND t2.cnj='$tc5'
$subj_test_d;")or die(mysql_error()); $tes1252 = mysql_result ($mc52,1); echo $tes1252 ?><br>
<input type="radio" name="rdi[восьме]" value="3"><?php $m53 = mysql_query("SELECT * FROM definition
WHERE id IN (SELECT id FROM definition d WHERE pid=$tesmax5 $subj_test UNION SELECT t1.id FROM
definition t1 JOIN definition t2 ON t1.pid = t2.id WHERE t2.pid=$tesmax5 $subj_test_d);
while ($tes53 = mysql_fetch_array($m53))
{
echo $tes53['CS'], " ", $tes53['CNJ'], " ";
}
?><br>
</li>
<hr color=red WIDTH="80%" align="left">
<li> Чи вірне наступне твердження:
<strong>
<?php
$max7err = mysql_query("SELECT id FROM definition WHERE pid=0 $subj_test ORDER BY RAND() LIMIT 1" );
$max7err = mysql_result ($max7err,0);
$selem=0;
// перевірка кількості елементів дерева
$countelem = mysql_query("SELECT COUNT(id) AS count FROM definition WHERE (id=$max7err OR
pid=$max7err ) $subj_test UNION SELECT count(t1.id) FROM definition t1 JOIN definition t2 ON t1.pid = t2.id
WHERE t2.pid = $max7err $subj_test_d" );
//$countelem = mysql_result ($countelem,1);
while ($tes7count = mysql_fetch_array($countelem))
    {
        $selem=$selem+$tes7count['count'];
    }
while ($selem<3)
    {
        $max7err = mysql_query("SELECT id FROM definition WHERE pid=0 $subj_test ORDER BY RAND() LIMIT 1" );

```

```

$max7err = mysql_result ($max7err,0);
$selem=0;
$countelem = mysql_query("SELECT COUNT(id) AS count FROM definition WHERE (id=$max7err OR
pid=$max7err ) $subj_test UNION SELECT count(t1.id) FROM definition t1 JOIN definition t2 ON t1.pid = t2.id
WHERE t2.pid = $max7err $subj_test_d" );
while ($tes7count = mysql_fetch_array($countelem))
{
    $selem=$selem+$tes7count['count'];
}
}
// кінець перевірка кількості елементів дерева
//echo "KILKIST=", $selem;
$max7oper= array('<>','=' );
$max7oper = $max7oper[array_rand($max7oper)];
$_SESSION['qwe']=$max7oper;
$stest7err= mysql_query ("SELECT * FROM definition WHERE (id=$max7err OR pid=$max7err) $subj_test");
while ($stes7err = mysql_fetch_array($stest7err))
{
    echo $stes7err['CS'], " ", $stes7err['CNJ'], " ";
    IF ($stes7err['id']>$stes7err['id'-1])
    {
        $stesmax7err=$stes7err['id'];
    }
    $stc7err=$stes7err['CNJ'];
}
$stc7err=$stc7err; $mc7err = mysql_query("SELECT cs FROM definition WHERE pid=0.1 $subj_test UNION
SELECT t1.cs
FROM definition t1 JOIN definition t2 ON t1.pid = t2.id WHERE t1.pid $max7oper $stesmax7err AND t2.cnj='$stc7err'
$subj_test_d;")or die(mysql_error()); $stes127err = mysql_result ($mc7err,0); echo $stes127err;
?></strong><br />
<input type="radio" name="rdi[дев'яте]" value="1">Так<br>
<input type="radio" name="rdi[дев'яте]" value="2">Hi<br>
</li>
<hr color=red WIDTH="80%" align="left">
<li><strong>Встановіть правильні відповідності
<table>
<tr>
<td>
<?php $max8 = mysql_query("SELECT id FROM definition WHERE pid=0 $subj_test ORDER BY RAND() LIMIT
1" );
$max8 = mysql_result ($max8,0);
$stest8 = mysql_query ("SELECT * FROM definition WHERE id=$max8 $subj_test");
while ($stes8 = mysql_fetch_array($stest8))
{
echo "1. ", $stes8['CS'], " ", $stes8['CNJ'] ;
    $stesmax8=$stes8['id'];
    $stc8=$stes8['CNJ'];
}
?></strong>
<br />
</td>
<td>
<select name="rdi[десяте]" size = 1>
<option value="1"><?php $stcc8=$stc8; $mc8 = mysql_query("SELECT CS,CNJ FROM definition WHERE pid =
$max8 $subj_test UNION SELECT t1.CS,t1.CNJ FROM definition t1 JOIN definition t2 ON t1.pid = t2.id WHERE
t2.pid = $max8 $subj_test_d") or die(mysql_error());
while ($stest81 = mysql_fetch_array($mc8))
{ echo $stest81['CS'], " ", $stest81['CNJ'], " "; } ?></option>
<option value="2"><?php
$max82 = mysql_query("SELECT id FROM definition WHERE pid=0 AND id<>$max8 $subj_test ORDER BY
RAND() LIMIT 1" );
$max82 = mysql_result ($max82,0);

```

```

$tc82=$tc8; $mc82 = mysql_query("SELECT CS,CNJ FROM definition WHERE pid = $max82 $subj_test UNION
SELECT t1.CS,t1.CNJ FROM definition t1 JOIN definition t2 ON t1.pid = t2.id WHERE t2.pid = $max82
$subj_test_d") or die(mysql_error());
while ($test812 = mysql_fetch_array($mc82))
{ echo $test812['CS'], " ", $test812['CNJ'], " "; } ?></option>
<option value="3"><?php
$max83 = mysql_query("SELECT id FROM definition WHERE pid=0 AND id<>$max8 AND id<>$max82 $subj_test
ORDER BY RAND() LIMIT 1");
$max83 = mysql_result ($max83,0);
$tc83=$tc8; $mc83 = mysql_query("SELECT CS,CNJ FROM definition WHERE pid = $max83 $subj_test UNION
SELECT t1.CS,t1.CNJ FROM definition t1 JOIN definition t2 ON t1.pid = t2.id WHERE t2.pid = $max83
$subj_test_d") or die(mysql_error());
while ($test813 = mysql_fetch_array($mc83))
{ echo $test813['CS'], " ", $test813['CNJ'], " "; } ?></option>
</select>
</td>
</tr>
<tr>
<td>
<?php
$max84 = mysql_query("SELECT id FROM definition WHERE id=$max82 $subj_test");
$max84 = mysql_result ($max84,0);
$test8 = mysql_query ("SELECT * FROM definition WHERE id=$max84 $subj_test");
while ($tes8 = mysql_fetch_array($test8))
{
echo "2. ", $tes8['CS'], " ", $tes8['CNJ'];
    $tesmax8=$tes8['id'];
    $tc8=$tes8['CNJ'];
}
?></strong>
<br />
</td>
<td>
<select name="rdi[одинадцете]" size = 1>
<option value="1"><?php $tc8=$tc8; $mc8 = mysql_query("SELECT CS,CNJ FROM definition WHERE pid =
$max8 $subj_test UNION SELECT t1.CS,t1.CNJ FROM definition t1 JOIN definition t2 ON t1.pid = t2.id WHERE
t2.pid = $max8 $subj_test_d") or die(mysql_error());
while ($test81 = mysql_fetch_array($mc8))
{ echo $test81['CS'], " ", $test81['CNJ'], " "; } ?></option>
<option value="2"><?php
$max85 = mysql_query("SELECT id FROM definition WHERE pid=0 AND id=$max83 $subj_test" );
$max85 = mysql_result ($max85,0);
$tc85=$tc8; $mc85 = mysql_query("SELECT CS,CNJ FROM definition WHERE pid = $max85 $subj_test UNION
SELECT t1.CS,t1.CNJ FROM definition t1 JOIN definition t2 ON t1.pid = t2.id WHERE t2.pid = $max85
$subj_test_d") or die(mysql_error());
while ($test8125 = mysql_fetch_array($mc85))
{ echo $test8125['CS'], " ", $test8125['CNJ'], " "; } ?></option>
<option value="3"><?php
$max83 = mysql_query("SELECT id FROM definition WHERE pid=0 AND id<>$max8 AND id=$max82 $subj_test"
);
$max83 = mysql_result ($max83,0);
$tc83=$tc8; $mc83 = mysql_query("SELECT CS,CNJ FROM definition WHERE pid = $max83 $subj_test UNION
SELECT t1.CS,t1.CNJ FROM definition t1 JOIN definition t2 ON t1.pid = t2.id WHERE t2.pid = $max83
$subj_test_d") or die(mysql_error());
while ($test813 = mysql_fetch_array($mc83))
{ echo $test813['CS'], " ", $test813['CNJ'], " "; } ?></option>
</select>
</td>
</tr>
<tr>
<td>
<?php
$max86 = mysql_query("SELECT id FROM definition WHERE id=$max85 $subj_test");

```



```

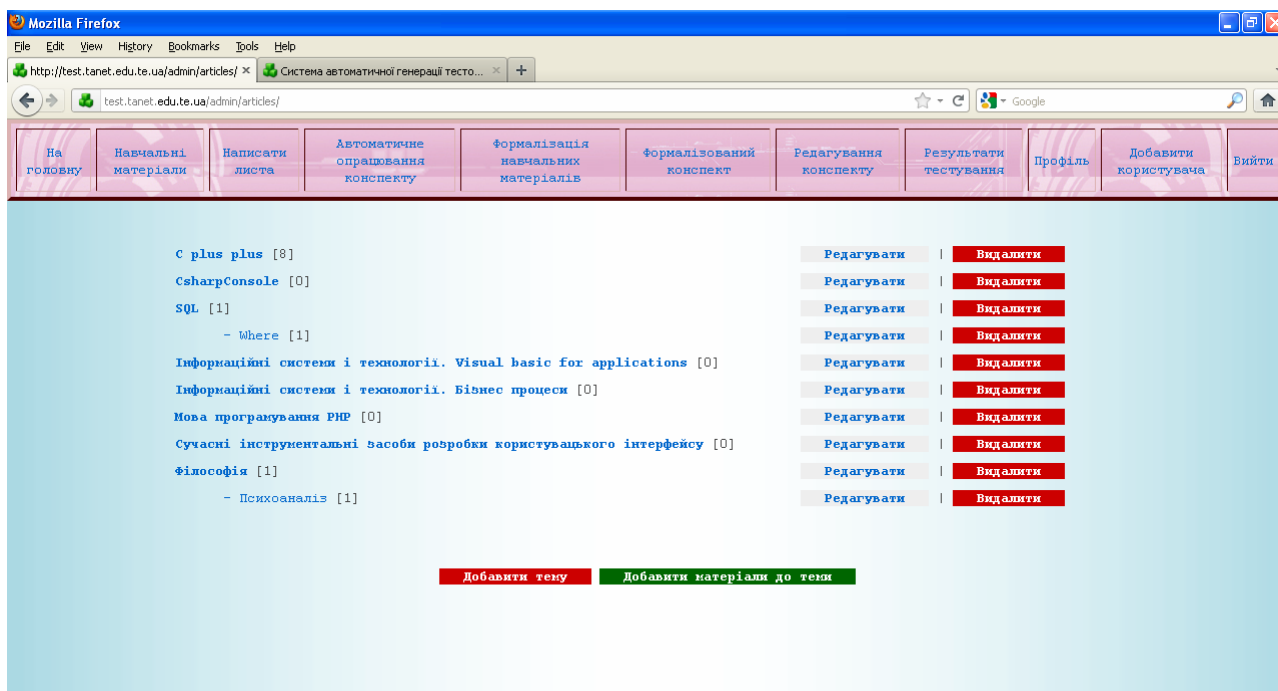
$max86 = mysql_result ($max86,0);
$stest8 = mysql_query ("SELECT * FROM definition WHERE id=$max86 $subj_test");
while ($stes8 = mysql_fetch_array($stest8))
{
echo "3. ", $stes8['CS'], " ", $stes8['CNJ'] ;
    $stesmax8=$stes8['id'];
    $stc8=$stes8['CNJ'];
}
//echo $stesmax5;
?></strong>
<br />
</td>
<td>
<select name="rdi[дванадцятe]" size = 1>
<option value="1"><?php $tcc8=$tcc8; $mcc8 = mysql_query("SELECT CS,CNJ FROM definition WHERE pid =
$max8 $subj_test UNION SELECT t1.CS,t1.CNJ FROM definition t1 JOIN definition t2 ON t1.pid = t2.id WHERE
t2.pid = $max8 $subj_test_d") or die(mysql_error());
while ($stest81 = mysql_fetch_array($mcc8))
{ echo $stest81['CS'], " ", $stest81['CNJ'], " "; } ?></option>
<option value="2"><?php
$max87 = mysql_query("SELECT id FROM definition WHERE pid=0 AND id=$max83 $subj_test" );
$max87 = mysql_result ($max87,0);
$tcc82=$tcc8; $mcc82 = mysql_query("SELECT CS,CNJ FROM definition WHERE pid = $max87 $subj_test UNION
SELECT t1.CS,t1.CNJ FROM definition t1 JOIN definition t2 ON t1.pid = t2.id WHERE t2.pid = $max87
$subj_test_d") or die(mysql_error());
while ($stest812 = mysql_fetch_array($mcc82))
{ echo $stest812['CS'], " ", $stest812['CNJ'], " "; } ?></option>
<option value="3"><?php
$max88 = mysql_query("SELECT id FROM definition WHERE pid=0 AND id=$max85 $subj_test" );
$max88 = mysql_result ($max88,0);
$tcc83=$tcc8; $mcc83 = mysql_query("SELECT CS,CNJ FROM definition WHERE pid = $max88 $subj_test UNION
SELECT t1.CS,t1.CNJ FROM definition t1 JOIN definition t2 ON t1.pid = t2.id WHERE t2.pid = $max88
$subj_test_d") or die(mysql_error());
while ($stest813 = mysql_fetch_array($mcc83))
{ echo $stest813['CS'], " ", $stest813['CNJ'], " "; } ?></option>
</select>
</td>
</tr>
<table>
<br>
</li>
<hr color=red WIDTH="80%" align="left">

</ol>
<center>
<input type="submit" align="center" style="font-size: 20px;> font-family: ariel; background: Moccasin"
value="Завершити тестування" >
</form></form>
</body>
</html>

```

Додаток Г

Графічний інтерфейс модуля управління навчальним контентом



Додаток Д

Графічний інтерфейс форми «Автоматичне опрацювання конспекту»

Автоматичне опрацювання конспекту

Виберіть дисципліну і тему:

C plus plus

Недоліки мови C++

FCIT Ternopil

Контент навчальних матеріалів:

Недоліки мови C++

* Наявність безлічі можливостей, що порушують принципи типобезпеки приводить до того, що в C++ програми може легко закрастися важковловима помилка. Замість контролю з боку компілятора розробники вимушені дотримуватися вельми нетривіальних правил кодування. По суті ці правила обмежують C++ рамками якогось безпечнішої підмови. Більшість проблем типобезпеки C++ успадкована від C, але важливу роль в цьому питанні грає і відмова автора мови від ідеї використовувати автоматичне управління пам'яттю (наприклад, збірку сміття). Так візитною картою C++ стали вразливості типу «переповнення буфера».

* Погана підтримка модульності. Підключення інтерфейсу зовнішнього модуля

Опрацювати конспект

Додаток Е

Список тестових завдань, що використано в експериментальних дослідженнях

№	Тестові завдання
1	ORDER BY -параметр який <input type="radio"/> повертає рядки з однієї чи багатьох таблиць. <input type="radio"/> вказує порядок повернення рядків <input type="radio"/> описує кінцеві дані, однак, не вказує, які саме операції слід виконати для отримання цих даних
2	SQL -це мова яка <input checked="" type="checkbox"/> використовується SQL інтерпретаторами і застосовується для формування запитів, оновлення і керування реляційними БД <input checked="" type="checkbox"/> використовується SQL інтерпретаторами і застосовується для взаємодії користувача з базами даних <input checked="" type="checkbox"/> використовується SQL інтерпретаторами і дозволяє здійснювати пошук, вставку, оновлення, і вилучення даних, використовуючи систему управління і адміністративні функції <input type="checkbox"/> використовується SQL інтерпретаторами і не дозволяє здійснювати пошук, вставку, оновлення, і вилучення даних, використовуючи систему управління і адміністративні функції
3	Встановіть правильні відповідності 1. EOM - <input type="text" value="сукупність змінних які дозволяють ідентифікувати певну структуру"/> 2. Параметри структури - <input type="text" value="мова програмування яка використовується у більшості EOM"/> 3. C++ - <input type="text" value="сукупність елементів які зібрані певним чином"/>
4	Чи вірне наступне твердження: MySQL - система управління базами даних <input checked="" type="radio"/> Так <input type="radio"/> Ні
5	Чи вірне наступне твердження: SUM - повертає суму значень елементів яка застосовується для формування запитів <input type="radio"/> Так <input checked="" type="radio"/> Ні
6	ORDER BY - застосовується у такій послідовності <input checked="" type="checkbox"/> SELECT ... WHERE ... ORDER BY <input type="checkbox"/> SELECT ... ORDER BY ... FROM <input type="checkbox"/> SELECT ... WHERE ... ORDER BY ... FROM <input checked="" type="checkbox"/> SELECT ... FROM ... WHERE ... ORDER BY
7	Чи вірне наступне твердження: C++ - мова програмування яка використовується у більшості EOM <input checked="" type="radio"/> Так <input type="radio"/> Ні
8	ToLower – <input type="radio"/> визначає логічний тип <input type="radio"/> визначає стрічковий тип <input checked="" type="radio"/> визначає перетворення рядка до нижнього регістру
9	Конструктор класу - <input checked="" type="checkbox"/> спеціальний блок інструкцій, який викликається при створенні об'єкта <input type="checkbox"/> спеціальний блок інструкцій, який ідентифікує параметри функцій <input type="checkbox"/> спеціальний блок інструкцій, який викликається при знищенні об'єкта
10	Чи вірне наступне твердження: C++ - мова програмування <input type="radio"/> Так <input checked="" type="radio"/> Ні
11	Параметри структури -сукупність змінних які <input checked="" type="checkbox"/> дозволяють ідентифікувати певну структуру <input type="checkbox"/> зібрані певним чином <input type="checkbox"/> надають такий же рівень сервісу

12	<p>Деструктор класу реалізується за допомогою наступного програмного коду -</p> <pre> <input type="checkbox"/> class Base { private: char *sp1; public: Base(const char *S) { sp1=strdup(S); } }; <input checked="" type="checkbox"/> class Base { private: char *sp1; public: ~Base() { delete sp1; } }; <input type="checkbox"/> #include using namespace std; </pre>
13	<p>Встановіть правильні відповідності</p> <p>1. float - <input type="text" value="тип даних з плаваючою комою, який займає 4 байтів"/></p> <p>2. double - <input type="text" value="тип даних з плаваючою комою, який займає 8 байтів"/></p> <p>3. long double - <input type="text" value="тип даних з плаваючою комою, який займає 16 байтів"/></p>
14	<p>Тригер - це збережена процедура особливого типу, яка</p> <p><input type="checkbox"/> не викликається користувачем безпосередньо, а застосовується для формування запитів, оновлення і керування реляційними БД</p> <p><input checked="" type="checkbox"/> не викликається користувачем безпосередньо, а виконання якої обумовлено настанням певної події</p> <p><input checked="" type="checkbox"/> не викликається користувачем безпосередньо, а використання якої нацлено на забезпечення цілісності даних і реалізації складної бізнес логіки</p> <p><input type="checkbox"/> не викликається користувачем безпосередньо, а виконання якої не обумовлено настанням певної події</p>
15	<p>Встановіть правильні відповідності</p> <p>1. Страуструп перейменував мову у C++ у <input type="text" value="1983"/></p> <p>2. Стандарт ISO C++ ратифіковано у <input type="text" value="1998"/></p> <p>3. Перший випуск Borland C++ відбувся у <input type="text" value="1990"/></p>
16	<p>SELECT - оператор вибору, який</p> <p><input checked="" type="checkbox"/> дозволяє знайти дані, які збережені у відповідній таблиці</p> <p><input type="checkbox"/> дозволяє знайти дані, які задають цілочисельний тип даних</p> <p><input checked="" type="checkbox"/> дозволяє знайти дані, які визначаються користувачем</p> <p><input checked="" type="checkbox"/> дозволяє знайти дані, які згруповані оператором Group by</p>
17	<p>ODBC - змінні які</p> <p><input type="radio"/> надають такий же рівень сервісу</p> <p><input type="radio"/> зібрані певним чином</p> <p><input type="radio"/> дозволяють ідентифікувати певну структуру</p>
18	<p>Конструктор класу -</p> <p><input type="checkbox"/> мова програмування, яка використовується у більшості ЕОМ</p> <p><input type="checkbox"/> запит, який використовується для оновлення даних</p> <p><input checked="" type="checkbox"/> спеціальний блок інструкцій, який викликається при створенні об'єкта</p>
19	<p>Командна мова - це мова яка</p> <p><input type="checkbox"/> застосовується для формування запитів, оновлення і керування реляційними БД</p> <p><input type="checkbox"/> застосовується для взаємодії користувача з базами даних</p> <p><input checked="" type="checkbox"/> дозволяє здійснювати пошук, вставку, оновлення, і вилучення даних, використовуючи систему управління і адміністративні функції</p>
20	<p>Чи вірне наступне твердження: Байт (byte) — одиниця виміру обсягу даних яка містить 8 бітів</p> <p><input checked="" type="radio"/> Так</p> <p><input type="radio"/> Ні</p>
21	<p>Чи вірне наступне твердження: UCASE - функція яка застосовується для формування запитів, оновлення і керування реляційними БД</p> <p><input type="radio"/> Так</p> <p><input checked="" type="radio"/> Ні</p>

22	Відновіть твердження: <code>SELECT * FROM TABLE ... id=pid;</code> <input type="radio"/> GROUP BY <input type="radio"/> ORDER BY <input checked="" type="radio"/> WHERE
23	Чи вірне наступне твердження: Параметри структури - сукупність змінних які зібрані певним чином <input type="radio"/> Так <input checked="" type="radio"/> Ні
24	Чи вірне наступне твердження: SUM - повертає суму значень елементів яка визначена оператором SELECT <input checked="" type="radio"/> Так <input type="radio"/> Ні
25	Чи вірне наступне твердження: C++ - це мова структурованих запитів <input type="radio"/> Так <input checked="" type="radio"/> Ні
26	CREATE PROCEDURE - службова команда, яка <input type="radio"/> застосовується для формування запитів, оновлення і керування реляційними БД <input checked="" type="radio"/> використовується для створення збережених процедур <input type="radio"/> застосовується для взаємодії користувача з базами даних
27	Чи вірне наступне твердження: UCASE - функція яка перетворює символ у верхній регістр <input checked="" type="radio"/> Так <input type="radio"/> Ні
28	Відновіть твердження: Вказівник - ..., який використовується в комп'ютерних мовах програмування <input type="radio"/> елемент масиву <input checked="" type="radio"/> тип даних <input type="radio"/> оператор присвоєння
29	Чи вірне наступне твердження: EOM - сукупність елементів які дозволяють ідентифікувати певну структуру <input type="radio"/> Так <input checked="" type="radio"/> Ні
30	Чи вірне наступне твердження: MySQL - найпростіша СУБД яка використовується у веб-технологіях <input checked="" type="radio"/> Так <input type="radio"/> Ні

Додаток Ж

Показники успішності для типових категорій студентів та розподіл тестових завдань по рівнях складності

Студент	Рівень складності; його частка	Спостереження	Часові витрати, год	Успішність
1	1; 0.4	1	4	0.5
		2	8	0.75
		3	10.67	0.75
	2; 0.35	4	3	0.43
		5	5.3	0.71
		6	6.97	0.71
	3; 0,25	7	8	0.4
		8	14.7	0.6
		9	20	0.6
2	1; 0.4	10	2.33	0.75
		11	4	0.75
		12	5	0.87
	2; 0.35	13	3	0.57
		14	5.33	0.57
		15	7	0.86
	3; 0,25	16	3.67	0.4
		17	6.67	0.6
		18	9	0.8
3	1; 0.4	19	3.3	0.75
		20	6	0.75
		21	8	0.87
	2; 0.35	22	3.7	0.57
		23	6.7	0.71
		24	9.01	0.86
	3; 0,25	25	4.3	0.4
		26	7.6	0.6
		27	10.2	0.6

Додаток И

Акти впровадження результатів дисертаційної роботи

“ЗАТВЕРДЖУЮ”
 Директор Івано-Франківської філії
 Відкритого міжнародного університету
 розвитку людини «Україна»
 І.В. Кіт
 2011 р.



АКТ

про впровадження результатів дисертаційної роботи

Мельника Андрія Миколайовича

„Інформаційна технологія автоматичної генерації тестових завдань з керованою складністю”

Ми, комісія в складі: Колибав'юк Світлани Петрівни, Мельника Тараса Зіновійовича, склали даний акт про те, що при реалізації сучасних інформаційних і комунікаційних технологій навчання, а також при їх практичному застосуванні використані наступні результати дисертаційної роботи аспіранта кафедри комп'ютерних наук Тернопільського національного економічного університету Мельника Андрія Миколайовича:

1. Метод генерації тестових завдань для перевірки засвоєння методологічних знань, який дозволяє отримувати якісні тестові набори, а визначена система змінюваних параметрів забезпечує варіативність таких наборів.
2. Методи генерації тестових завдань із застосуванням теорії онтологій. Дані методи використовують семантичну, формальну та синтаксичну типізацію інформаційних матеріалів.
3. Метод оцінки складності тестових завдань. Використання даного методу дозволяє будувати завдання заданого рівня складності.
4. Метод генерації практичних задач та алгоритм представлення, автоматичного контролю та перевірки згенерованих задач у прикладному програмному середовищі.
5. Програмне забезпечення для автоматичної генерації тестових завдань та проведення тестових контролів, яке включає засоби побудови завдань різних форм та типів.

Створене програмне забезпечення інтегроване у веб-орієнтовану систему управління процесом навчання, як підсистема автоматичної генерації тестових завдань. Використання методів генерації тестових завдань дозволило суттєво економити часові ресурси в процесі побудови тестових завдань, а також управляти їх складністю.

Заступник директора з
науково-методичної роботи

С.П. Колибав'юк

С.П. Колибав'юк

Завідувач
інформаційно-обчислювального центру

Т.З. Мельник

Т.З. Мельник

„ЗАТВЕРДЖУЮ”



Директор з науково-педагогічної роботи
Тернопільського національного
економічного університету

Шинкарик М.І.

2011 р.

про впровадження результатів дисертаційної роботи
Мельника Андрія Миколайовича

**„Інформаційна технологія автоматичної генерації тестових завдань з
керованою складністю ”**

Ми, комісія в складі: Дивака Миколи Петровича, Надвиничного Сергія Анатолійовича, Романця Ігоря Євгеновича, склали даний акт про те, що при розробці та реалізації інноваційних методів та засобів навчання з використанням сучасних інформаційних технологій, а також при їх застосуванні використані такі результати дисертаційної роботи аспіранта кафедри комп'ютерних наук Тернопільського національного економічного університету Мельника Андрія Миколайовича:

1. Методи автоматичної генерації тестових завдань закритої форми, які використовують семантичну, формальну та синтаксичну типізацію.

2. Адаптивно-структурний метод оцінки складності тестових завдань на основі їх формальних характеристик із налаштуванням для конкретної аудиторії та предмету вивчення.

3. Метод мінімізації часу інтерактивного навчання для забезпечення необхідного рівня успішності на основі моделей успішності та функції часових затрат.

4. Web-орієнтована система автоматичної генерації тестових завдань та проведення тестових контролів.

Застосування розробленого програмного забезпечення та вказаних методів автоматичної генерації тестових завдань уможливило зменшення у 3 рази витрат часу на розробку тестів за рахунок автоматичного опрацювання навчальних матеріалів та багатократного використання раніше напрацьованих компонент. Застосування методу мінімізації часу інтерактивного навчання для забезпечення необхідного рівня успішності у поєднанні із автоматичною генерацією тестів заданої складності дозволило зменшити час досягнення заданих рівнів успішності в середньому на 19%.

Декан факультету комп'ютерних
інформаційних технологій,
д.т.н., проф.

М.П. Дивак

Заступник начальника
навчального відділу,
к.е.н, доц.

С.А. Надвиничний

Директор центру інформаційних технологій

І. Є. Романець

«ЗАТВЕРДЖУЮ»



Проректор з наукової роботи
Тернопільського національного
економічного університету
д.е.н., проф. З.-М.В. Задорожний
Задорожний 7 червня 2011 р.

про використання результатів кандидатської дисертаційної роботи

Мельника Андрія Миколайовича

**«Інформаційна технологія автоматичної генерації тестових завдань з
керованою складністю»**

Комісія у складі голови – завідувача кафедри комп'ютерних наук, керівника науково-дослідної роботи, д.т.н., проф. Дивака М.П. та членів: начальника науково-дослідної частини Письменного В.І., начальника відділу прогнозування і маркетингу Лучки А.В. склали цей акт про те, що дослідження та результати дисертаційної роботи Мельника А.М. використані під час виконання науково-дослідних робіт на кафедрі комп'ютерних наук з безпосередньою участю автора, а саме:

- науково-дослідної роботи «Методи та засоби математичного моделювання складних систем на основі теоретико-множинного та інтервального підходів» (номер державної реєстрації 0106U012529), зокрема у розробці розділу „Розробка алгоритмів опрацювання даних інформаційних систем навчальних закладів”, у якій автором розроблено моделі управління затратами часу навчання для досягнення бажаного рівня успішності, розроблено критерій та метод оцінки складності тестових тверджень;

- науково-дослідної роботи на тему: «Моделювання динаміки управлінських, маркетингових, виробничих, фінансових та екологічних процесів в організаційних системах» (номер державної реєстрації 0106U012531), зокрема у розробці розділу „Методи формалізації процесу генерації тестових завдань для електронних навчальних курсів на основі семантичних класів”, автором вдосконалено алгоритми автоматичної генерації тестових завдань відкритого та закритого типів;

- науково-дослідної роботи на тему: Методи та засоби генерування тестів із застосуванням теорії онтологій” (номер державної реєстрації 0111U006279), у якій автором практично реалізовано веб-орієнтовану систему автоматичної генерації тестових завдань – SAGT.

Голова комісії

завідувач кафедри комп'ютерних наук,
керівник НДР, д.т.н., проф.

Дивак М.П.

Члени комісії:

Начальник НДЧ

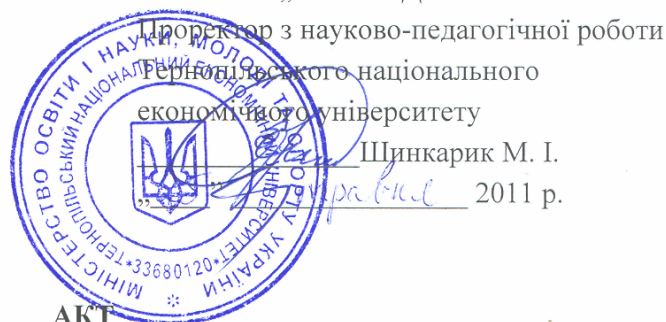
Письменний В.І.

Начальника відділу

прогнозування і маркетингу

Лучка А.В.

„ЗАТВЕРДЖУЮ”



Директор з науково-педагогічної роботи
Тернопільського національного
економічного університету
Шинкарик М. І.
2011 р.

АКТ

про впровадження в навчальний процес Тернопільського національного
економічного університету результатів дисертаційної роботи

Мельника Андрія Миколайовича

**„ Інформаційна технологія автоматичної генерації тестових завдань з
керованою складністю ”**

Даний акт складений про те, що результати дисертаційної роботи аспіранта кафедри комп'ютерних наук Мельника Андрія Миколайовича на тему „Інформаційна технологія автоматичної генерації тестових завдань з керованою складністю” використані в навчальному процесі факультету комп'ютерних інформаційних технологій Тернопільського національного економічного університету для студентів напрямів підготовки 6.050103 – „Програмна інженерія” та 6.030502 – „Економічна кібернетика”. Зокрема, при викладанні дисциплін „Проектування баз даних і знань”, „Математичне забезпечення програмних систем”, „Засоби програмування баз даних знань”, „Технологія створення програмних та інтелектуальних систем”:

- надається інформація щодо інформаційного моделювання предметної області у вигляді процедур реляційної алгебри;
- надається інформація про методи формалізації інформативних матеріалів та засобів їх опрацювання в процесі проектування інтелектуальних систем різного призначення;
- використовується у процесі виконання лабораторних робіт розроблене у дисертаційній роботі програмне забезпечення для генерації практичних задач та їх автоматизований контроль в процесі реалізації розв'язків.

Декан факультету комп'ютерних
інформаційних технологій,
зав. кафедри комп'ютерних наук,
д.т.н., проф.

М. П. Дивак

Завідувач кафедри економічної
кібернетики та інформатики
д.е.н, доц.

О. М. Ляшенко

Доцент кафедри комп'ютерних наук,
к.т.н.

М. Я. Шпінталь