

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ
ІМЕНІ АДМІРАЛА МАКАРОВА

Навчально-науковий інститут комп'ютерних наук та управління проектами
Кафедра інформаційних управляючих систем та технологій

"Допущений до захисту"

Завідувач кафедри ІУСТ

к.т.н, доц. Михелєв І.Л.

"_____" 2024 р.

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття ступеня вищої освіти "магістр"

(освітньо-кваліфікаційний рівень)

на тему **Системний аналіз процесів моніторингу знань та розробка інформаційної підсистеми індивідуального навчання**

Виконав: студент 6 курсу, групи 6145м
напряму підготовки (спеціальності)

124 – "Системний аналіз".

(шифр і назва напряму підготовки, спеціальності)

Шенер А.А.

(прізвище та ініціали)

Керівник Кроптя А.В.

(прізвище та ініціали)

м. Миколаїв – 2024 р.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ КОРАБЛЕБУДУВАННЯ
ІМЕНІ АДМІРАЛА МАКАРОВА

Інститут, факультет Навчально-науковий інститут комп'ютерних наук та управління проектами

Кафедра Інформаційних управляючих систем та технологій

Рівень вищої освіти другий (магістерський)

Спеціальність 124 – "Системний аналіз"

Освітня програма "Системний аналіз"

"ЗАТВЕРДЖУЮ"

Гарант освітньої програми

д.ф-м.н, проф. Ушкац М.В.

"___" _____ 2024 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ
на здобуття ступеня вищої освіти "магістр"

Студенту Шенер Андрію Андрійовичу

1. Тема проекту (роботи) Системний аналіз процесів моніторингу знань та розробка інформаційної підсистеми індивідуального навчання

керівник проекту (роботи) Кроптя Арсеній Володимирович,

затверджені наказом вищого навчального закладу від "08" 10 2024 року № 1055-уч.

2. Строк подання студентом проекту (роботи) 15 грудня 2024 року.

3. Вихідні дані до проекту (роботи) ДСТУ щодо обробки інформації, літературні джерела, технічна документація на існуючі аналоги систем, матеріали практики.

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

- Титульний аркуш, завдання на дипломну роботу, анотація (українською, англійською), зміст, перелік умовних означень, символів, одиниць та термінів (при необхідності).

- Вступ.

- Аналіз предметної області і постановка задачі.

- Розробка проекту автоматизованої системи.

- Реалізація проекту автоматизованої системи (технічна документація по проекту).

- Розділ з охорони праці.

- Висновки.
- Список використаної літератури.
- Додатки: загальний опис підсистеми, керівництво користувача, керівництво адміністратора, програма і методика випробувань, графічні інтерфейси, код програми.

5. Перелік графічного матеріалу

- 5.1 Обробка результатів тестування.
- 5.2 Модель Раша.
- 5.3 Деталізація рівня знань за допомогою сегментації.
- 5.4 Порівняння існуючих систем.
- 5.5 DFD – 0 рівня.
- 5.6 Логічна модель адаптивної системи навчання.
- 5.7. Екранні форми.

6. Консультанти розділів проекту (роботи)

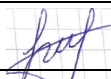
Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1.	Кроптя А.В.	05.09.2023	22.12.2023
2.	Кроптя А.В.	23.12.2023	15.06.2024
3.	Кроптя А.В.	15.06.2024	25.09.2024
4.	Кроптя А.В.	25.09.2024	14.11.2024
5.	Гурець Н.В.	25.09.2024	14.11.2024
6.	Гурець Н.В.	25.09.2024	14.11.2024

7. Дата видачі завдання 05 вересня 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту (роботи)	Примітка
1.	Вивчення і опис предметної області	22.12.2024	
2.	Розробка концепції ІС	15.03.2024	
3.	Розробка проектних рішень по ІС	15.06.2024	
4.	Реалізація проекту, розробка робочої документації	25.09.2024	
5.	Вирішення питань з охорони праці	14.11.2024	
6.	Виконання графічної частини роботи	14.11.2024	
7.	Оформлення пояснювальної записки	14.11.2024	
8.	Подання роботи на кафедральний захист	16.11.2024	
9.	Подання роботи рецензенту	20.11.2024	
10.	Подання роботи на державний захист	15.12.2024	

Студент

 Шенер А.А.

Керівник проекту (роботи)

 к.т.н., доцент Кроптя А.В.

ЗМІСТ

Вступ.....	
1. Дослідження систем адаптивного навчання та методів оцінки знань.....	
1.1 Адаптивне навчання в учбовому процесі	
1.2 Застосування інформаційних технологій як один із засобів оцінювання знань студентів	
1.3 Методи оцінки знань випробуваних та тестових завдань.....	
1.4 Статистичні методи аналізу результатів тестування.....	
1.5 Математичні моделі IRT у тестуванні навчальних досягнень.....	
1.6 Аналіз існуючих систем.....	
1.7 Розробка механізмів вдосконалення зв'язку "викладач-студент"	
2. Статистична обробка знань студентів та складності завдання	
2.1 Статистична обробка результатів тестування	
2.2 Розрахунок рівня підготовленості студентів.....	
2.3 Розрахунок складності завдання.....	
3. Розробка проекту системи адаптивного навчання	
3.1 Інформаційне забезпечення.....	
3.2 Технічне забезпечення	
3.3 Програмне забезпечення.....	
3.4 Результати розробки	
4. Розрахунок економічної ефективності впровадження ПЗ.....	
4.1 Вступ.....	
4.2 Розрахунок витрат на створення та експлуатацію інформаційної системи	
4.3 Економічна ефективність розробки і впровадження системи	
5. Охорона праці.....	
6. Охорона навколишнього середовища	
Висновки.....	
Список використаної літератури	
Додаток А – Технічне завдання.....	
Додаток Б – Код програми	
Додаток В – Інструкція користувача	

АНОТАЦІЯ

Дана магістерська робота присвячена науковому дослідженню на тему "Системний аналіз процесів моніторингу знань та розробка інформаційної підсистеми індивідуального навчання".

Метою даної роботи є дослідження методів оцінки знань для покращення якості навчального процесу та розробка системи індивідуального навчання.

Задачами дослідження є аналіз ефективності використання інформаційних технологій в навчальному процесі, актуальність використання систем індивідуального навчання, аналіз існуючих систем, аналіз методів оцінки знань випробуваних та тестових завдань.

Об'єкт дослідження – процес організації адаптивних систем навчання.

Предмет дослідження – методи аналізу результатів тестування як інструмент адаптації індивідуальної навчальної програми студента.

Науковою новизною є використання індивідуального навчання та методів оцінки знань студентів для підвищення якості освіти.

Дана дипломна робота викладена на 110 сторінках друкованого тексту, містить 29 рисунків, 14 таблиць, та має 3 додатки які містять технічне завдання, текст виконаної програми, інструкцію користувача. Список літератури з 18 найменувань.

Робота виконана на українській мові.

Ключові слова: системний аналіз; моніторинг знань; інформаційна підсистема; аналіз даних в освіті; персоналізація навчання; індивідуальне навчання; адаптивні освітні технології; оцінка знань; інтелектуальні системи; технології моніторингу та оцінки; аналітика навчальних процесів; автоматизовані системи навчання; інтерфейси користувача в освітніх системах.

ABSTRACT

This master thesis is devoted to scientific research on " System analysis of knowledge monitoring processes and development of an information subsystem for individual learning".

The purpose of this work is to study methods of knowledge assessment to improve the quality of the educational process and develop a system of individual learning.

The objectives of the study are to analyze the effectiveness of the use of information technologies in the educational process, the relevance of the use of individual learning systems, the analysis of existing systems, the analysis of methods of assessing the knowledge of test takers and test tasks.

The object of the study is the process of organizing adaptive learning systems.

The subject of the study is methods of analyzing test results as a tool for adapting a student's individual curriculum.

The scientific novelty is the use of individual learning and methods of assessing student knowledge to improve the quality of education.

This course is available at 110 pages of printed text, contains 29 drawings, 14 tables, and a 3 supplement containing the terms of reference, the text of the program, instruction manual. References in of 18 items.

Work carried out in Ukrainian.

Keywords: system analysis; knowledge monitoring; information subsystem; individual learning; data analysis in education; personalization of learning; adaptive educational technologies; knowledge assessment; intelligent systems; monitoring and evaluation technologies; analytics of educational processes; automated learning systems; user interfaces in educational systems.

ВСТУП

В наш час сучасні інформаційні технології з кожним роком все більше інтегруються в навчальному процесі викладання фахових дисциплін в вищих навчальних закладах. У сфері освіти йде пошук нового змісту та нових форм навчання, створюються нові освітні технології, розширюється використання особистісно-орієнтованих методів навчання. Але основною проблемою в освіті залишається якість навчання. З кожним роком вимоги до якості освіти зростають. Тому потрібно міняти підхід до подачі матеріалу і методи оцінки знань. Нові способи оцінювання знань та індивідуальний підхід до кожного студента, дають можливість створити зручний зворотний зв'язок між студентом і викладачем. Викладач зможе відстежувати успішність учнів і коригувати відповідний навчальний матеріал. Студент маючи свою індивідуальну програму зможе позбутися прогалин у знаннях і підтягнути свою успішність.

Метою даної роботи є дослідження методів оцінки знань для покращення якості навчального процесу та розробка системи адаптивного навчання.

Задачами дослідження є аналіз ефективності використання інформаційних технологій в навчальному процесі, актуальність використання систем адаптивного навчання, аналіз існуючих систем, аналіз методів оцінки знань випробуваних та тестових завдань.

Об'єкт дослідження – процес організації адаптивних систем навчання.

Предмет дослідження – методи аналізу результатів тестування як інструмент адаптації навчальної програми студента.

Адаптивне навчання – це спосіб організації навчального процесу з урахуванням індивідуального рівня підготовки студента до початку навчання або в процесі навчання. Адаптивна система навчання дозволить відстежувати ступінь успішності, контролювати студентів і корегувати процес навчання залежно від знання або не знання студентів.

Створення адаптивної системи навчання – це один з перспективних способів підвищення ефективності процесу навчання. Адаптивний контроль навчання являє собою сучасний комп'ютерний варіант реалізації відомого педагогічного

принципу індивідуалізації навчання – одного з найважливіших для підготовки кваліфікованих фахівців. Цей принцип не може бути реалізований ефективним чином в умовах традиційних класно-урочних (лекційних) форм занять.

1. ДОСЛІДЖЕННЯ СИСТЕМ АДАПТИВНОГО НАВЧАННЯ ТА МЕТОДІВ ОЦІНКИ ЗНАНЬ

1.1 Адаптивне навчання в учбовому процесі.

Адаптивне навчання виникло на основі аналізу сучасних тенденцій удосконалення навчального процесу за допомогою формування індивідуальної програми студента. Класичні методи навчання подають матеріал всім студентам однаково. Тому різний рівень засвоюваності матеріалу не дозволяє нормально оцінити повноту отриманих знань. Так як один студент засвоює один матеріал краще, а інший цей же матеріал гірше. І це позначається на підсумкових результатах студента.

Навчання, як один з видів діяльності людини за допомогою адаптивного навчання, стає активною самостійною діяльністю, створює умови для розумного включення в навчальний процес. Адаптивне навчання має новий підхід до змісту навчання і процесів, які забезпечують гармонійний розвиток студентів. Класичні методи навчання подають матеріал всім студентам однаково. Тому адаптивне навчання є один з кращих способів підвищення ефективності освіти та створення зворотного зв'язку між викладачем та студентом.

Зворотний зв'язок — це процес визначення результатів освітньої діяльності учнів педагогом для виявлення, аналізу, оцінювання й корекції навчання. Зворотній зв'язок може бути двох видів: внутрішній і зовнішній. Внутрішній зворотний зв'язок — це інформація, яка надходить від навчальної програми до студента у відповідь на його дії при виконанні завдань. Вона призначена для самокорекції суб'єктом навчання своєї навчальної діяльності і дає можливість тому, кого навчають зробити усвідомлений висновок про успішність чи помилковості навчальної діяльності. Вона є стимулом до подальших дій, допомагає оцінити і скорегувати результати навчальної діяльності. Розрізняють консультативний і результативний внутрішній зворотний зв'язок. Консультація може бути різною: допомога, роз'яснення, підказка. Результативна зворотній зв'язок також може бути різною: від "вірно-невірно" до демонстрації правильного результату або

способу дії. Зовнішній зворотний зв'язок – це інформація, яка надходить від навчальної програми до викладача. З її допомогою він коригує навчальну діяльність студента, а також може скорегувати сценарій навчання [1].

Адаптивний зворотний зв'язок є важливою складовою в подачі досліджуваного матеріалу. Якщо ніхто не зміг відповісти правильно на якесь одне завдання це може бути пов'язано з складною формулюванням завдання, але так само можливо при поганому викладі матеріалу або поганому засвоєнні цього матеріалу учням. Проблемою може бути погане розуміння певної теми або кількох тем конкретної дисципліни. Або навіть того матеріалу який вивчався раніше в іншій дисципліні, але був засвоєний на недостатньому рівні.

Вирішенням цієї проблеми є повторне вивчення учням погано засвоєного матеріалу. Можливість підказувати учню, залежно від його результатів, матеріал який допоможе йому позбавиться від прогалин у знаннях, є великим кроком для адаптивного навчання. Це дозволить не тільки знайти індивідуальний підхід до кожного учня і дасть можливість йому самому вивчити матеріал, але і стане відмінним помічником для викладача.

Використання систем адаптивного навчання (САН) в установах освіти, державних структурах, підприємствах і компаніях дозволяє автоматизувати безліч процесів, пов'язаних з організацією електронного навчання, управлінням навчальним процесом, оцінкою та розвитком персоналу. САН дозволяє підвищити якість навчання, об'єктивність оцінки рівня знань співробітників і перейти від примусового навчання до самоосвіти. Використання САН ефективно як для підприємств з невеликою чисельністю співробітників, так і для великих організацій з більшим числом персоналу та філіями в різних регіонах. Основним завданням САН є – управління навчальним процесом: планування навчання, контроль подаються знань протягом усього строку навчання, аналіз навчального процесу виходячи з отриманих результатів [2].

На відміну від більшості систем, що оцінюють рівень знань у яких відсутня єдина методика оцінки якості тестових матеріалів та відсутній аналіз якості засвоєння різного матеріалу (наприклад засвоєння різних розділів дисциплін в університеті) використання САН дозволить посилити увагу на проблемні місця

у знаннях студента і таким чином підтягти його загальні знання на певний рівень. САН дозволить зробити підготовку студентів більш якісною.

1.2 Застосування інформаційних технологій як один із засобів оцінювання знань студентів.

Основна мета контролю знань і умінь полягає у виявленні досягнень, успіхів студентів, у вказівці шляхів вдосконалення, поглиблення знань, умінь. Ця мета в першу чергу пов'язана з визначенням якості засвоєння студентами навчального матеріалу – рівня оволодіння знаннями, вміннями та навичками передбачених програмою. По-друге, конкретизація основної мети контролю пов'язана з навчанням прийомам взаємоконтролю і самоконтролю, формуванням потреби в самоконтролі і взаємоконтролю. Контроль виконує наступні функції [3].

Контролююча функція. Полягає у виявленні стану знань і умінь студентів. За допомогою контролю визначається вихідний рівень для подальшого оволодіння знаннями, вміннями і навичками, вивчається глибина і обсяг їх засвоєння. Порівнюється плановане з дійсними результатами, встановлюється ефективність викладання викладачем матеріалу.

Навчальна функція. Полягає у вдосконаленні знань і умінь, їх систематизації. У процесі перевірки студенти повторюють і закріплюють вивчений матеріал. Вони не тільки відтворюють раніше вивчене, а й застосовують знання і вміння в нову ситуацію. Перевірка допомагає виділити головне, основне в досліджуваному матеріалі, зробити знання та вміння, що перевіряються, більш ясними і точними. Контроль сприяє також узагальненню та систематизації знань [4].

Діагностуюча функція. Сутність діагностичної функції контролю – в отриманні інформації про помилки, недоліки і прогалини в знаннях і вміннях студентів і породжують причини труднощів студентів в оволодінні навчальним матеріалом, про кількість, характер помилок. Результати діагностичних перевірок допомагають вибрати найбільш інтенсивну методику навчання, а

також уточнити напрямок подальшого вдосконалення змісту методів і засобів навчання.

Прогностична функція. Прогностична функція перевірки служить для отримання випереджаючої інформації про навчально-виховний процес. В результаті перевірки отримують підстави для прогнозу про хід певного відрізка навчального процесу: чи достатньо сформовані конкретні знання, вміння та навички для засвоєння наступної порції навчального матеріалу (розділу, теми) [5]. Результати прогнозу використовують для створення моделі подальшої поведінки студента, допускає сьогодні помилки даного типу або має певні прогалини в системі прийомів пізнавальної діяльності. Прогноз допомагає отримати вірні висновки для подальшого планування і здійснення навчального процесу.

Орієнтуюча функція. Сутність орієнтуючої функції контролю – в отриманні інформації про ступінь досягнення мети навчання окремим студентом і групою в цілому – наскільки засвоєний і як глибоко вивчений навчальний матеріал. Контроль орієнтує студентів на подолання труднощів та досягнення. Розкриваючи прогалини, помилки і недоліки студентів, він вказує їм напрями докладання зусиль щодо вдосконалення знань і умінь. Контроль допомагає студенту краще пізнати самого себе, оцінити свої знання і можливості [6].

Існують такі методи контролю.

1. Усна перевірка. Вона організується по-різному, в залежності від її мети та від змісту перевіряється матеріалу. Серед цільових установок перевірки можна виділити наступні: перевірити виконання домашнього завдання, виявити підготовленість студентів до вивчення нового матеріалу, перевірити ступінь розуміння і засвоєння нових знань. Залежно від змісту вона проводиться за матеріалом попереднього заняття або по окремих розділах і темах курсу. Методика усної перевірки містить у собі дві основні частини: складання перевірочних питань і їх завдання; відповідь студентів на поставлені питання;

Складання перевірочних питань і завдань – важливий елемент усної перевірки. Якість питань визначається їхнім змістом, характером виконуваних студентами при відповіді на питання розумових дій, а також словесним

формулюванням. При складанні питань завжди виходять з того, що перевіряти слід ті знання, які є основними в даному курсі або відносно важко засвоюються студентами, або які необхідні для успішного засвоєння подальших розділів і тем курсу. На добір питань впливає вид перевірки: для уточнення змісту питань для поточної перевірки необхідний аналіз зв'язків досліджуваного матеріалу з раніше пройденим, а для тематичної і підсумкової перевірки – виділення провідних знань і способів оперування ними. Причому усну перевірку вважають ефективною, якщо вона спрямована на виявлення свідомості сприйняття знань і усвідомленості їхнього використання, якщо вона стимулює самостійність і творчу активність студентів. Якість питання визначається характером розумових дій, які виконують студенти при відповіді на питання. Тому серед перевірочних завдань виділяють питання, що активізують пам'ять (на відтворення вивченого), мислення (на порівняння, доказ, узагальнення), мову. Велике значення мають проблемні питання, що змушують застосовувати отримані знання в практичній діяльності. Другою складовою частиною усної перевірки є відповідь студента на питання. При оцінці відповіді студента звертають увагу на правильність і повноту відповіді, послідовність викладу, якість мови.

2. Модульна перевірка в письмовій формі. Цей метод має свої якісні особливості: велика об'єктивність та економія часу у порівнянні з усною перевіркою, охоплення потрібного числа тих що перевіряються. Застосування письмових робіт використовується для:

- перевірки знання теоретичного матеріалу;
- уміння застосовувати його до рішення задач.

У методиці письмово-графічних робіт виділяють чотири основних етапи, яким треба приділяти увагу. Це підготовка, організація, проведення, аналіз результатів.

При підготовці потрібно: виділити мету перевірки, відібрати зміст об'єктів перевірки, скласти перевірочні завдання. Велику допомогу при цьому надають навчально-методичні посібники, зразки перевірочних робіт в журналах. При організації перевірочної роботи студентом повідомляється, які завдання їм призначені, як озаглавити роботу, як оформити рішення, час виконання роботи.

При цьому потрібно стежити за самостійністю виконання роботи кожним студентом. Аналіз відповідей студентів ефективний тоді, коли він проводиться за певними схемами (схемами поелементного аналізу). Ретельно проведений аналіз дозволяє глибоко вивчити прогалини і досягнення окремих студентів, виділити типові помилки й основні труднощі студентів, вивчити причини їх появи і намітити шляхи їх усунення.

3. Перевірка практичних і лабораторних робіт. За допомогою цього методу одержують дані про уміння студентів застосовувати отримані знання при вирішенні практичних завдань. Викладач одержує звіт студента, у якому наводиться тільки результат або схематично описані план практичної роботи та її результати.

Сучасні інформаційні технології з кожним роком все більше інтегруються в навчальний процес викладання фахових дисциплін в вищих навчальних закладах. Застосування цих трьох методів мають суттєві недоліки. При усній перевірці у викладача не достатньо часу опитати всіх студентів. Хоча письмова перевірка економить час у порівнянні з усною перевіркою, але викладач витрачає багато часу на перевірку завдання кожного студента. Також студенти мають можливість списувати один у одного. При лабораторних неможливо дізнатися теоретичні знання студенту [7].

У сфері освіти йде пошук нового змісту та нових форм навчання, створюються нові освітні технології, розширюється використання особистісно-орієнтованих методів навчання. Тому при контролі знань потрібні нові способи оцінювання знань. А також швидкий та зручний зворотній зв'язок між студентом та викладачем. Адаптивне навчання – це спосіб організації навчального процесу з урахуванням індивідуального рівня підготовки студента до початку навчання або в процесі навчання [8]. Адаптивна система контролю знань дозволить відстежувати ступінь успішності, контролювати студентів і корегувати процес навчання залежно від знання або не знання студентів. Така система має ряд переваг:

- Дозволяє зменшити непродуктивні витрати живої праці викладача, який в цьому випадку перетворюється на технолога сучасного навчального процесу, в

якому провідна роль відводиться не так і не тільки навчальній діяльності педагога, скільки навчанню самих студентів;

- Диференційоє підхід до студентів, заснований на визнанні того факту, що у різних студентів попередній досвід і рівень знань в одній області різні, кожен учень приходить до процесу оволодіння новими знаннями зі своїм власним інтелектуальним багажем, який і визначає ступінь розуміння ним нового матеріалу і його інтерпретацію, тобто здійснюється поворот від оволодіння всіма студентами одного і того ж матеріалу до оволодіння різними студентами різного матеріалу;

- Підвищує оперативність і об'єктивність контролю та оцінки результатів навчання;

- Гарантує безперервний зв'язок у відносинах "викладач-студент";

- Сприяє індивідуалізації навчальної діяльності (диференціація темпу навчання, труднощі навчальних завдань тощо).

В результаті аналізу перспективних інноваційних технологій визначено, що використання адаптивних систем має ряд переваг у порівнянні з існуючими методами контролю, це дає можливість більш детально розібрати прогалини в навчанні та способи та усунення. Адаптивна система потребує певних відомостей про стан знань студентів, а для цього то потрібна правильна система оцінювання.

1.3 Методи оцінки знань випробуваних та тестових завдань.

Сучасне тестування представляє собою комплекс стандартизованих методів вимірювання параметрів, через які визначають рівень підготовки людини і відповідність освітнім стандартам у конкретній галузі знань. При цьому широко використовуються математичні методи планування й опрацювання результатів тестування, а також сучасні технології опрацювання даних. Об'єктивний контроль знань, вмінь і навичок вдається здійснити при критеріально-орієнтованій інтерпретації тестування. Критеріально-орієнтоване тестування призначене не тільки для оцінювання рівня знань, а й для визначення рівня індивідуальних досягнень відносно певного критерію на підставі аналізу змісту

завдань. Тому, враховуючи індивідуалізацію навчання, конструювання критеріально-орієнтованих тестів є одним із провідних та найактуальніших напрямків розвитку теорії тестів.

В сучасних навчальних системах потрібно знати складність кожного завдання тесту. Така складність апробується і визначається емпірично. Так виникає одне із головних питань теорії тестів – питання побудови ефективного тесту.

Історично виділяють два основні підходи до створення тестів. Перший з них набув широкого розвитку в рамках класичної теорії тестів. Згідно з ними, рівень знань учасників тестування оцінюється за допомогою індивідуальних балів. Бал обчислюють як алгебраїчну суму оцінок виконання кожного завдання тесту. Класична теорія тестів ґрунтується на статистичних методах аналізу результатів тестування.

Складність завдань можна визначати двома способами:

- на основі оцінки передбачуваного числа і характеру розумових операцій, необхідних для вдалого виконання завдань;
- на основі емпіричної перевірки завдань, з підрахунком частки неправильних відповідей.

У класичній теорії тестів багато років розглядалися тільки емпіричні показники складності. У сучасних теоріях навчальних тестів, які використовуються в дистанційному навчанні, більше уваги приділяється характеру розумової діяльності у процесі виконання тестових завдань різних форм.

Наступною вимогою до тестових завдань є варіація балів. Якщо на деяке завдання правильно відповідають всі студенти, то таке завдання стає не тестовим. Учасники тестування відповідають на нього однаково: між ними немає варіації. Нетестовим вважається також завдання, на яке немає жодної правильної відповіді. Варіація при цьому теж дорівнює нулеві. Нульова варіація означає практичну необхідність вилучення завдання із тесту.

Штучність деяких припущень класичної теорії тестів і деякі її практичні недоліки помітно вплинули на розвиток критичних тенденцій. Цьому в першу

чергу сприяли сумніви в об'єктивності емпіричних оцінок складності завдань тесту. А саме виникло питання про правомірність традиційного оцінювання складності завдань за допомогою частки правильних чи неправильних відповідей.

При традиційному підході до зміни рівня складності завдань на різних вибірках відповідей студентів з різними рівнями підготовки залишається відкритим питання про об'єктивність значень параметра складності завдань тесту. Спроба введення вагових коефіцієнтів, що відображають вагу завдання в індивідуальному балі студента, суттєво не виправляє такі недоліки. Значення цих коефіцієнтів можна, в свою чергу, поставити під сумнів. Деякі з них визначаються суб'єктивно, на основі думки педагога про складність завдання. Оцінки решти з них базуються на емпіричних даних тестування і, відповідно, залежать від рівня знань опитаних студентів.

Другий підхід до створення тестів та опрацювання результатів тестування використовуються в так званій сучасній теорії тестування, що набула широкого розвитку в 1960-1980 роках в багатьох країнах [9].

Item Response Theory (IRT) – математична теорія параметричного оцінювання тестових завдань і результатів тестування. Відповідно до цієї теорії встановлено, що між результатом виконання, що спостерігається, і рівнем знань учасників тестування є деяка залежність, яку можна виразити за допомогою деякої функції.

До найбільш вагомих переваг IRT відносять:

- стійкі об'єктивні оцінки параметра складності завдань, що не залежать від властивостей вибірки відповідей студентів, які виконують тест;
- вимірювання значень параметрів знань студентів і завдань тесту в одній і тій самій шкалі, що дозволяє поставити у відповідність рівень знань кожного учасника тестування із рівнем складності кожного завдання тесту;
- можливість оцінити ефективність різних за рівнем складності завдань для вимірювання рівня знань студента.

IRT переводили такими словами, як "теорія латентних рис", "теорія характеристичних кривих завдань", "теорія моделювання та параметризації педагогічних тестів", "сучасна теорія тестів" і т.д.

На думку деяких авторів IRT – це сучасна теорія педагогічних вимірювань, долає недоліки інших теорій і що вона здатна вирішити багато, якщо не всі проблеми підвищення якості педагогічних вимірювань.

Система тестування на основі моделі Раша володіє важливими достоїнствами, серед яких, насамперед, необхідно відзначити наступні.

Модель Раша перетворює вимірювання, зроблені в дихотомічних і порядкових шкалах в лінійні вимірювання. В результаті якісні дані аналізуються за допомогою кількісних методів. Це дозволяє використовувати широкий спектр статистичних процедур.

Оцінка труднощі тестових завдань не залежить від вибірки досліджуваних, на яких була отримана і, аналогічно, оцінка рівня знань піддослідних не залежить від використовуваного набору тестових завдань.

Пропуск даних для деяких комбінацій тестових завдань не є критичним.

Сама система тестування досить проста, порівняно з іншими аналогічними системами вона характеризується найменшим числом параметрів – тільки один параметр рівня знань для кожного випробуваного і тільки один параметр труднощі для кожного завдання [10].

Модель Раша спирається на чіткі та конструктивні поняття "труднощі завдання" і "рівень знань". Так, одне завдання вважається більш важким, ніж інше, якщо ймовірність правильної відповіді на перше завдання менше, ніж на друге, незалежно від того, хто їх виконує. Аналогічно, більш підготовлений студент має велику ймовірність правильно відповісти на всі завдання, ніж менш підготовлений.

Завдяки простій структурі моделі існують зручні обчислювальні процедури для багатоаспектної перевірки адекватності моделі: для всього набору тестових результатів, для кожного випробуваного, для кожного завдання і для кожного конкретного відповіді.

"Залишки", одержувані при апроксимації результатів тестування моделлю можна використовувати для виділення різних типів піддослідних.

Для практики одним з найбільш важливих критеріїв є точність оцінювання. Чим більше точність, тим краще працює модель. У разі відсутності помилок вимірювання будь-яка модель в сенсі точності вимірювання працює ідеально. Але на практиці помилки завжди є і тому важливо знати, наскільки точні оцінки дозволяє отримувати та чи інша модель.

Досліджується точність оцінювання рівнів знань і труднощів завдань, а також число ітерацій, необхідних для обчислення цих оцінок (методом найбільшої правдоподібності) в багатofакторної ситуації в залежності від:

- діапазону рівнів знань піддослідних;
- діапазону труднощів завдань;
- ступеня відповідності діапазонів рівнів знань піддослідних і труднощів завдань;
- числа піддослідних;
- числа завдань;
- ступеня відповідності даних моделі;
- частки пропущених даних.

1.4 Статистичні методи аналізу результатів тестування.

Згідно з основним підходом створення тестів, рівень знань учасників тестування оцінюється за допомогою їх індивідуальних балів. Бал обчислюють як алгебраїчну суму оцінок виконання кожного завдання тесту.

Класична теорія тестів ґрунтується на статистичних методах аналізу результатів тестування. Розглянемо найпростіші і необхідні процедури статистичної обробки результатів тестування знань і методи оцінки якості тесту.

В усіх відомих теоріях тестування розглядається як процес протистояння учасника із запропонованими йому завданнями. Позначимо через x_{ij} числову оцінку успішності виконання j -ого завдання i -им студентом. Результати тестування звичайно представляються у вигляді матриці x_{ij} , що задана n

рядками та m стовпцями ($i = \overline{1, n}$, $j = \overline{1, m}$). Матриця тестових результатів показує результат виконання всіх завдань учасниками тестування. На практиці прийнято, як правило, використовувати дихотомічну шкалу оцінок результатів. У результаті правильного виконання завдання студент отримує один бал ($x_{ij} = 1$), в протилежному випадку – нуль балів ($x_{ij} = 0$).

Потім потрібно знайти значення індивідуального бале i -го випробуваного X_i :

$$X_i = \sum_{j=1}^M a_{ij}.$$

Потім знаходимо кількість вірних відповідей R_j , на j -е завдання.

$$R_j = \sum_{i=1}^N a_{ij}.$$

Знаючи R_j ми множимо знайти кількість невірних відповідей W_j на j -е завдання

$$W_j = N - R_j$$

Після цього знаходимо частку вірних відповідей p_j і частка невірних відповідей q_j

$$p_j = \frac{R_j}{N};$$
$$q_j = 1 - p_j.$$

Важливим параметром тестового завдання є варіація (дисперсія) тестових балів p_j , q_j . Чим більше варіація, тим краще завдання диференціює випробовуваних.

При $p_j = 0$ та $p_j = 1$ дисперсія завдання дорівнює нулю. Іншими словами, якщо на завдання не відповів жоден випробуваний або успішно відповіли всі, то завдання не може їх диференціювати за рівнем підготовленості.

Потім нам необхідно обчислити частки вірних p_i і невірних $q_i = 1 - p_i$ відповідей піддослідних.

$$p_i = \frac{X_i}{M},$$

де X_i – індивідуальний бал випробуваного, M – кількість завдань у тесті.

Далі обчислюємо початкові значення рівня підготовленості випробуваних за формулою

$$\theta_i^0 = \ln \frac{p_i}{q_i}.$$

Далі обчислюємо початкове значення складності завдань β_j

$$\beta_j^0 = \ln \frac{p_j}{q_j}.$$

Тепер ми можемо обчислити середні значення рівня підготовленості випробуваних і складності завдань:

$$\bar{\theta} = \frac{\sum_{i=1}^N \theta_i^0}{N};$$

$$\bar{\beta} = \frac{\sum_{j=1}^M \beta_j^0}{M}.$$

Маючи значення параметрів на різних інтервальних шкалах треба звести їх в єдину шкалу стандартних оцінок. Для цього необхідно обчислити дисперсії S_θ та S_β :

$$S_\theta = \frac{\sum_{i=1}^N (\theta_i^0)^2 - N(\bar{\theta})^2}{N-1};$$

$$S_\beta = \frac{\sum_{j=1}^M (\beta_j^0)^2 - M(\bar{\beta})^2}{M-1}.$$

Далі обчислюємо кутові коефіцієнти

$$a_\theta = \sqrt{\frac{1 + S_\beta / 2,89}{1 - S_\theta S_\beta / 8,35}};$$

$$a_{\beta} = \sqrt{\frac{1 + S_{\theta}/2,89}{1 - S_{\theta}S_{\beta}/8,35}}.$$

Нарешті, ми можемо записати оцінки параметрів θ и β за єдиною інтервальною шкалою:

$$\theta_i = a_{\theta}\theta_i^0 + \bar{\beta};$$

$$\beta_j = a_{\beta}\beta_j^0 + \bar{\theta}.$$

Залишилося обчислити стандартні помилки вимірювання $S_E(\theta_i)$ та $S_E(\beta_j)$ для θ_i и β_j :

$$S_E(\theta_i) = \frac{a_{\theta}}{\sqrt{p_i(M - X_i)}} = \frac{a_{\theta}}{\sqrt{Mp_i(1 - p_i)}} = \frac{a_{\theta}}{\sqrt{Mp_iq_i}};$$

$$S_E(\beta_j) = \frac{a_{\beta}}{\sqrt{p_j(N - R_j)}} = \frac{a_{\beta}}{\sqrt{Np_j(1 - p_j)}} = \frac{a_{\beta}}{\sqrt{Np_jq_j}}.$$

На основі отриманих даних можемо зробити наступні висновки:

Якщо завдань з позитивними β_j більше ніж з негативними, та сума всіх значень $\beta_j > 0$, то даний тест має більше важких завдань, тобто він більш складний. Навпаки, якщо більше негативних β_j , та сума всіх значень $\beta_j < 0$, то такий тест більш легкий. В ідеалі, тест можна вважати збалансованим за умови коли сума всіх значень $\beta_j = 0$, але на практиці значення має наближатися до 0.

Для атестації за болонською системою студент має мати 60% правильних відповідей. Тобто рівень знань студента θ_i повинен знаходитись в таких межах $1 \leq \theta_i \leq 5$. Студенти які мають рівень знань θ_i менший за 1, мають шанс бути не атестованими. Чим менший рівень знань тим більший шанс бути не атестованим.

Математична теорія параметричного оцінювання тестових завдань і результатів тестування має багато переваг. Але на мою думку її можливості можна вдосконалити за допомогою деталізації яка зможе створити зворотній зв'язок. Модель дає узагальнену оцінку в цілому по всьому тесту. Але якщо розбити тест на сегменти (тематичні групи питань). То ми отримаємо можливість проаналізувати результати більш детально. В результаті ми

отримаємо зворотній зв'язок. Викладач зможе побачити який сегмент запитань більш складніший а який легкий. Та на основі отриманих результатів буде мати змогу скорегувати навчальний план для окремої групи.

Розбиття тесту на сегменти дасть можливість розраховувати складність сегменту та рівень знання студентом сегменту теми. Отримані результати дадуть можливість більш точно відобразити рівень знань студентів та їх підготовленість[13].

1.5 Математичні моделі IRT у тестуванні навчальних досягнень.

В якості математичної моделі, що пов'язує успіх випробуваного з рівнем його підготовленості і труднощами завдання вибирається логістична функція. Для моделі Раша вона має вигляд:

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

$$P_i(\beta) = \frac{e^{1,7(\theta_i - \beta)}}{1 + e^{1,7(\theta_i - \beta)}}. \quad (1.2)$$

Масштабний множник 1,7 використовується для сумісності моделі G. Rasch з моделлю A. Fergusson, де ймовірність правильної відповіді на завдання виражена інтегралом нормального розподілу, що дозволяє використовувати замість логістичних кривих добре вивчену інтегральну функцію нормованого нормального розподілу

$$P_j(\theta) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\theta - \beta_j} e^{-\frac{1}{2}x^2} dx.$$

Модель Раша носить назву "1 Parametric Logistic Latent Trait Model" (1PL), а модель A. Fergusson - "1 Parametric Normal Ogive Model" (1PN). Оскільки модель Раша описує вірогідність успіху випробуваного як функцію одного параметра $(\theta_i - \beta_j)$, то іноді її називають однопараметричною моделлю IRT [13].

Взаємодія двох множин θ_i і β_j утворює дані, що володіють властивістю "спільної адитивності" (conjoint additivity). Правильне використання моделі Раша дозволяє відокремити оцінки випробуваних від оцінок складності завдань і

навпаки. Ця властивість Rasch Measurement носить назву *separability parameter estimates* – "незалежність оцінок завдань від випробовуваних і оцінок випробовуваних від параметрів завдань".

На рис. 1.1 показані три характеристичні криві згідно рівняння (1.1) з труднощами завдань -2 , 0 і $+2$ логіт (перше найлегше, друге – середнє, третє найважче).

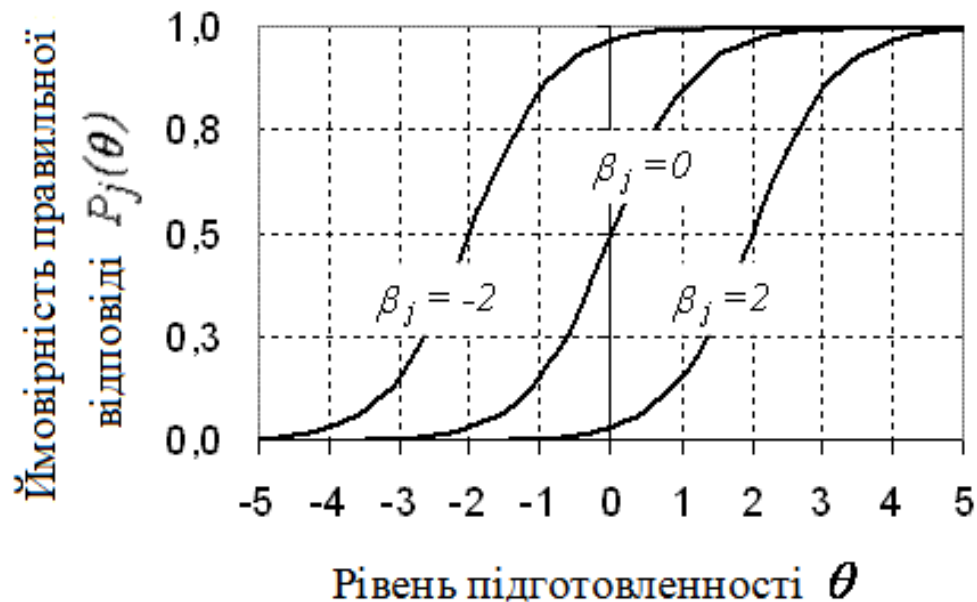


Рисунок 1.1 – Характеристичні криві завдань (ICC) в моделі (1PL).

З наведених залежностей видно, що чим вищий рівень підготовленості θ випробуваного, тим вище ймовірність успіху в тому чи іншому завданні. Наприклад, для випробуваного з $\theta = 0$ ймовірність правильно відповісти на перше завдання близька до одиниці, на другому дорівнює $1/2$ і на третьому майже дорівнює нулю. Відзначимо, що в точках, де $\theta = \beta$ ймовірність правильної відповіді дорівнює $0,5$. Тобто, якщо труднощі завдання дорівнює рівню підготовленості (ability) випробуваного, то він з однаковою ймовірністю може впоратися або не впоратися з цим завданням.

Характеристичні (логістичні) криві для завдань тесту в англomовній літературі називаються Item Characteristic Curve (ICC).

На рисунку 1.2 показані три характеристичні криві випробовуваних згідно рівняння (1.2) – "Person Characteristic Curve" (PCC). Показані графіки для трьох

випробовуваних з рівнем підготовленості -2 логіт (найслабший), 0 логіт (середній) і $+2$ логіт (сильний випробуваний).

З наведених залежностей видно, що чим вище рівень підготовленості, тим вище ймовірність правильної відповіді на завдання. Наприклад, завдання з труднощами $\beta = 0$ перший випробуваний ($q = -2$) практично не зможе виконати, другий ($q = 0$) має ймовірність виконання завдання рівну $0,5$, третій ($q = +2$) легко впорається із завданням, так як для нього ймовірність успіху майже дорівнює одиниці.

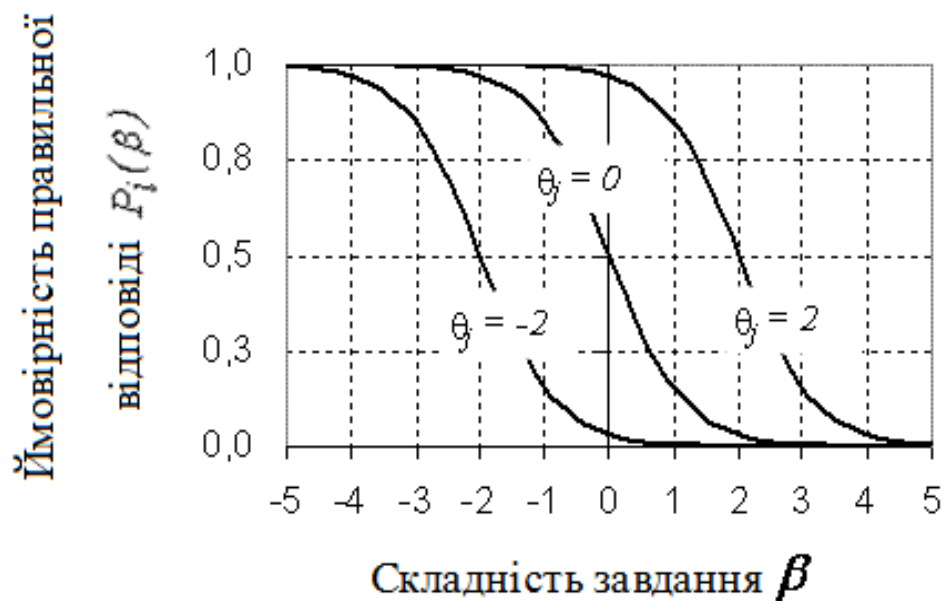


Рисунок 1.2 – Характеристичні криві досліджуваних (PCC) в моделі 1PL.

Як видно з наведених залежностей, крутизна характеристичних кривих в області $P_j = 0,5$ однакова, тобто диференціююча здатність є константою. Для дихотомічної моделі ця константа дорівнює $0,25$.

Якщо тест містить завдання з різною диференціюючою здатністю, то однопараметрична модель 1PL не може описати такі емпіричні дані. Для подолання цієї труднощі А. Бірнбаум (A. Birnbaum) ввів ще один параметр – item discrimination parameter.

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

$$P_i(\beta) = \frac{e^{1,7a_j(\theta_i - \beta)}}{1 + e^{1,7a_j(\theta_i - \beta)}}. \quad (1.4)$$

Параметр a_j визначає нахил (крутизну) характеристичної кривої j -го завдання. Приклади характеристичних кривих показані на рисунку 1.3. Видно, що чим більше a_j тим крутіше йде крива, тим вище диференціююча здатність завдання.

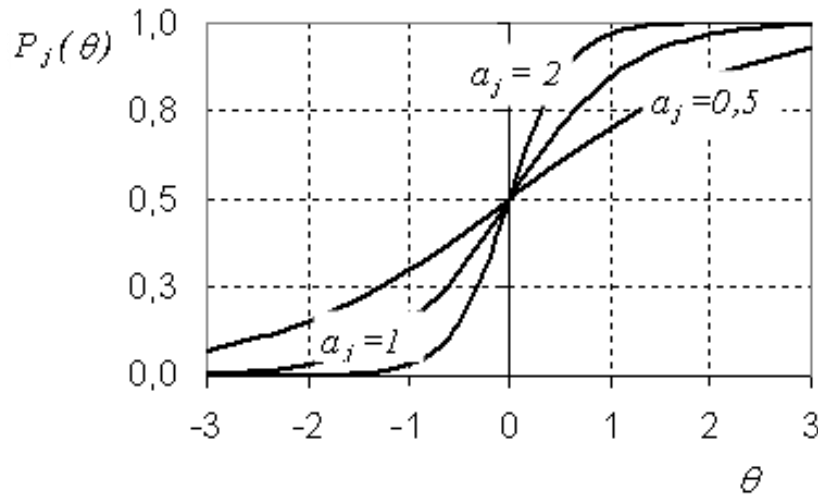


Рисунок 1.3 – ІСС в двухпараметричній моделі 2PL

Для ще кращої відповідності емпіричним даним А. Бірнбаум ввів третій параметр c – параметр вгадування.

$$P_j(\theta) = c_j + (1 - c_j) \frac{e^{1,7a_j(\theta - \beta_j)}}{1 + e^{1,7a_j(\theta - \beta_j)}}; \quad (1.5)$$

$$P_i(\beta) = c_j + (1 - c_j) \frac{e^{1,7a_j(\theta_i - \beta)}}{1 + e^{1,7a_j(\theta_i - \beta)}}. \quad (1.6)$$

З рівнянь (1.5) і (1.6) видно, що при $c_j = 0$ і $a_j = 1$ ці рівняння переходять в однопараметричну модель. З цієї причини іноді говорять, що модель Раша є окремим випадком дво- і трипараметричних моделей Бірнбаума. Формально це так, але по суті це невірно.

На рисунку 1.4 наведені приклади характеристичних кривих для трьох завдань з труднощами $\beta = 1$, дискримінаційним параметром $a_j = 1$ і різними параметрами вгадування $c_j = 0$, $c_j = 0,25$, $c_j = 0,5$.

З наведених графіків видно, що наявність параметра вгадування призводить до пропорційного зміщення ICC вгору на величину c_j .

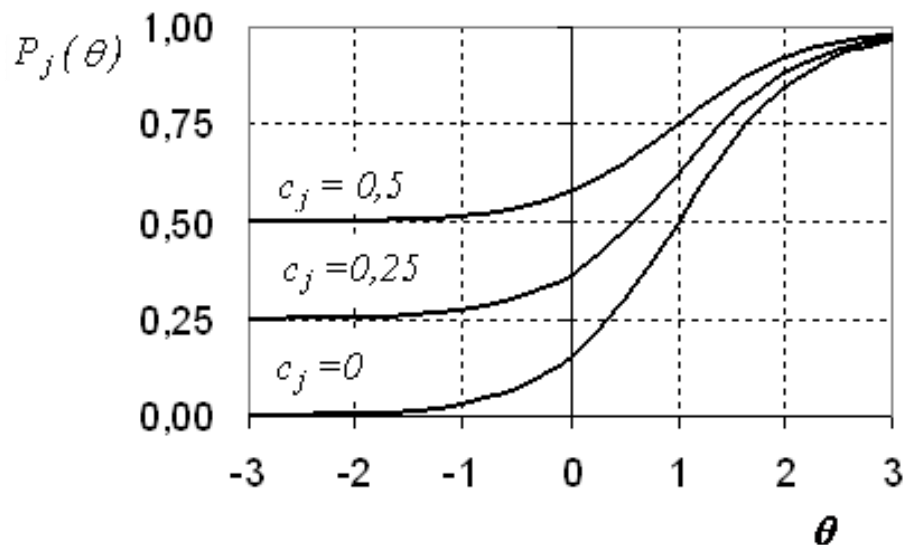


Рисунок 1.4 – ICC в трипараметричній моделі 3PL, $a_j = 1, \beta_j = 1$.

В якості теоретичної оцінки c_j можна використовувати зворотну величину від кількості відповідей в завданнях з вибором. Наприклад, в тесті використовуються завдання з чотирма відповідями, тоді $c_j = \frac{1}{4} = 0,25$. Це значення має уточнюватися при аналізі емпіричних даних.

Характерною особливістю моделі Раша є те, що характеристичні криві (ICC) не перетинаються (рисунок 1.1). Це означає, що якщо деяке завдання "А" легше завдання "Б", то це співвідношення зберігається у всьому інтервалі зміни θ .

Зовсім інша картина спостерігається для двох- і трипараметричних моделей. На рисунку 1.3 це добре видно. Завдання з $a_j = 0,5$ в області позитивних значень θ є найважчим з представлених трьох завдань, то є ймовірність правильної відповіді на це завдання найнижча. В області ж негативних значень θ це ж завдання тепер уже найлегше – ймовірність правильної відповіді на нього найбільша. Виходить, що для слабких учнів це найлегше завдання, а для сильних учнів – найважче.

Аналогічна картина спостерігається і для трипараметричної моделі. На рисунку 1.4 показаний рідкісний випадок непересічних характеристичних кривих, так як для них обрані однакові параметри $\beta_j = 1$ і $a_j = 1$, тобто всі три завдання мають однакову трудність і однаковий параметр диференціюючої здібності. На рисунку 1.5 наведено інший приклад.

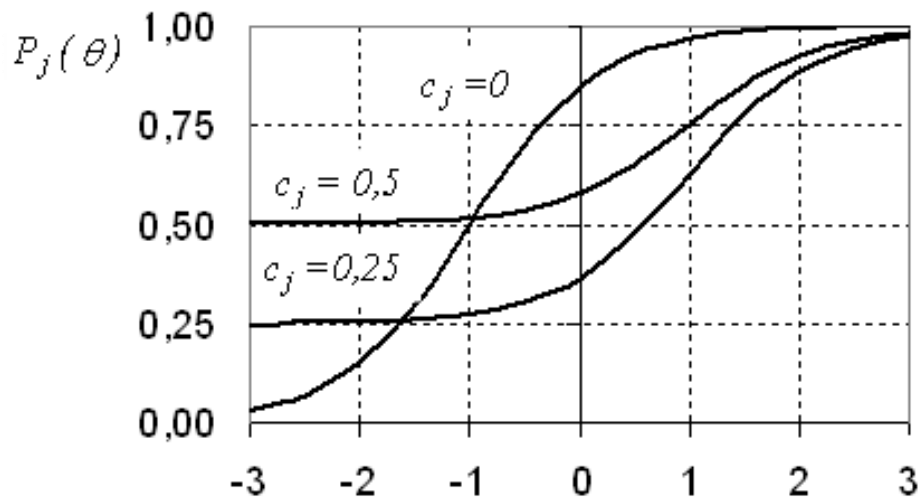


Рисунок 1.5 – Пересічні ІСС в трипараметричній моделі.

Тут у завдання з параметром $c_j = 0$ змінена складність $\beta_j = -1$, що негайно викликало перетин характеристичних кривих. Завдання з $c_j = 0$ в області $\theta < -2$ є найважчим. В області $-1,5 < \theta < -1$ завдання легше завдання з $c_j = 0,25$ і важче завдання з $c_j = 0,5$. В області $\theta > -1$ завдання з $c_j = 0$ є найлегшим.

Подібне перетин ІСС практично завжди відбувається для двох- і трипараметричних моделей.

Таким чином, тільки однопараметрична модель Раша відповідає вимогам, що пред'являються до якісної вимірювального інструментарію. Саме модель Rasch Measurement найбільше придатна для побудови тесту, як вимірювального інструмента [14].

1.6 Аналіз існуючих систем.

У результаті аналізу можливостей великої кількості програм виявилось, що більшість їх призначене тільки для складання тестів і проведення тестування. Можливості таких систем коливаються від найпростіших, що дозволяють тільки проводити тестування і зберігати його результати, а потім надавати їх викладачеві, до досить складних, забезпечених різними засобами аналізу результатів навчання (статистична обробка результатів), оформлення звітів за різними критеріями, установки параметрів питань (наприклад, коефіцієнт складності), параметрів тестів (наприклад, обмеження часу тестування), розмежування прав доступу і т.д. Деякі системи можуть використовуватися при роботі в мережі. Розрізняються системи за типами питань: найчастіше мається можливість створювати тільки одно варіантні запитання (дається кілька варіантів відповідей, з яких тільки один є правильним). Більш складні системи дозволяють використовувати декілька типів питань: багатоваріантні (кілька варіантів відповідей, з яких правильних може бути декілька або всі), відкриті (учень повинен ввести відповідь сам), на знаходження відповідності (дається два списки, потрібна для кожного елемента першого списку встановити відповідний до нього елемент другого списку) [15].

Прикладами таких програм є система КИОС, МастерТест, TestMaker, мережева Система електронного тестування, SunRay TestOfficePro, Test Commander, Асистент, Екзаменатор, AnyTest, Teaching Templates (система для веб-тестування) та багато інших.

Наступним класом програм є засоби розробки власне навчальних систем, які дозволяють окрім тестування підключати теоретичний матеріал. Деякі з них дозволяють здійснювати зв'язок тестових питань з теоретичними темами: наприклад, при неправильній відповіді на питання навчають, може отримати пояснення, в чому полягає його помилка або повернутися до вивчення пов'язаного теоретичного матеріалу.

Прикладами таких програм є Адонис, КАДИС, Наставник.

Окремо стоять системи, призначені для створення мультимедійних довідників, енциклопедій, словників. Основне призначення таких систем – зручність подання інформації, швидкість пошуку, можливість швидко

створювати складні інформаційні системи, швидко об'єднувати безліч файлів в єдину систему. Про контроль і тренування тут мова практично не йде. Прикладом такої програми є пакет HyperMethod.

І, нарешті, нечисленною групою виявилися засоби, що дозволяють створювати навчальні системи з вирішенням завдань (тобто з тренеруючою частиною). Всі ці програми дозволяли вирішувати завдання по одній, певній темі або за кількома близьким темам. Треба також зазначити, що всі розглянуті програми мали досить великий обсяг.

Прикладами таких програм є ComputerMentor, ComputerMaster, Універсальний Математичний Решатель.

Можна також розглянути засоби розробки, що мають дуже широкі можливості, вбудовані мови скриптів, що дозволяють програмувати окремі фрагменти створюваних систем, що підтримують велику кількість графічних, аудіо форматів, відео форматів, що дозволяють створювати платформо-незалежні навчальні системи і використовувати їх окремо від засобів розробки. Однак, вони дуже дорогі та недоступні широкому колу користувачів.

Прикладами таких засобів розробки є Toolbook, AuthorWare, Director, Media Objects, LearningSpace та інші.

Розглянемо деякі з перерахованих систем. А саме SunRay TestOfficePro, Master Test 1.0, VeralTest Professional, ComputerMentor Pro, Moodle, EJudge. Критерії по яким будемо розглядати є такі:

1. Візуалізація – використання в тестах крім звичайних текстових методів, використання зображень, таблиць та аудіо файлів.

2. Багатокористувацька – можливість користування системою різними користувачами та різними правами доступу (адміністратор, викладач, студент). Реєстрація нових користувачів в системі.

3. Аналіз результатів індивіду – звіт з результатами контролю в якому показані правильні та не правильні результати одного студента. Ця функція створена для визначення прогалин у знаннях вивченого студентом матеріалу.

4. Адаптивність – формування індивідуальної програми для студента на основі його результатів тестування.

5. Аналіз результатів тестових завдань – розрахунок складності/легкості завдань.

6. Пояснення не правильних відповідей – у разі наявності неправильних відповідей. Роз'яснення матеріалу у якому була допущена помилка. У вигляді теоретичного матеріалу.

Таблиця 1.1 Порівняння існуючих систем

Система	Візуалізація	Багато-користувацька	Аналіз результатів індивіду	Адаптивність	Аналіз результатів тестових завдань	Пояснення неправильних відповідей
SunRav TestOfficePro	+	+	+	–	–	–
Master Test 1.0	–	–	+	–	–	–
VeralTest Professional	–	+	+	–	–	–
ComputerMentor Pro	+	–	+	–	–	+
Moodle	–	+	+	–	–	–
EJudge	–	+	+	–	+	–

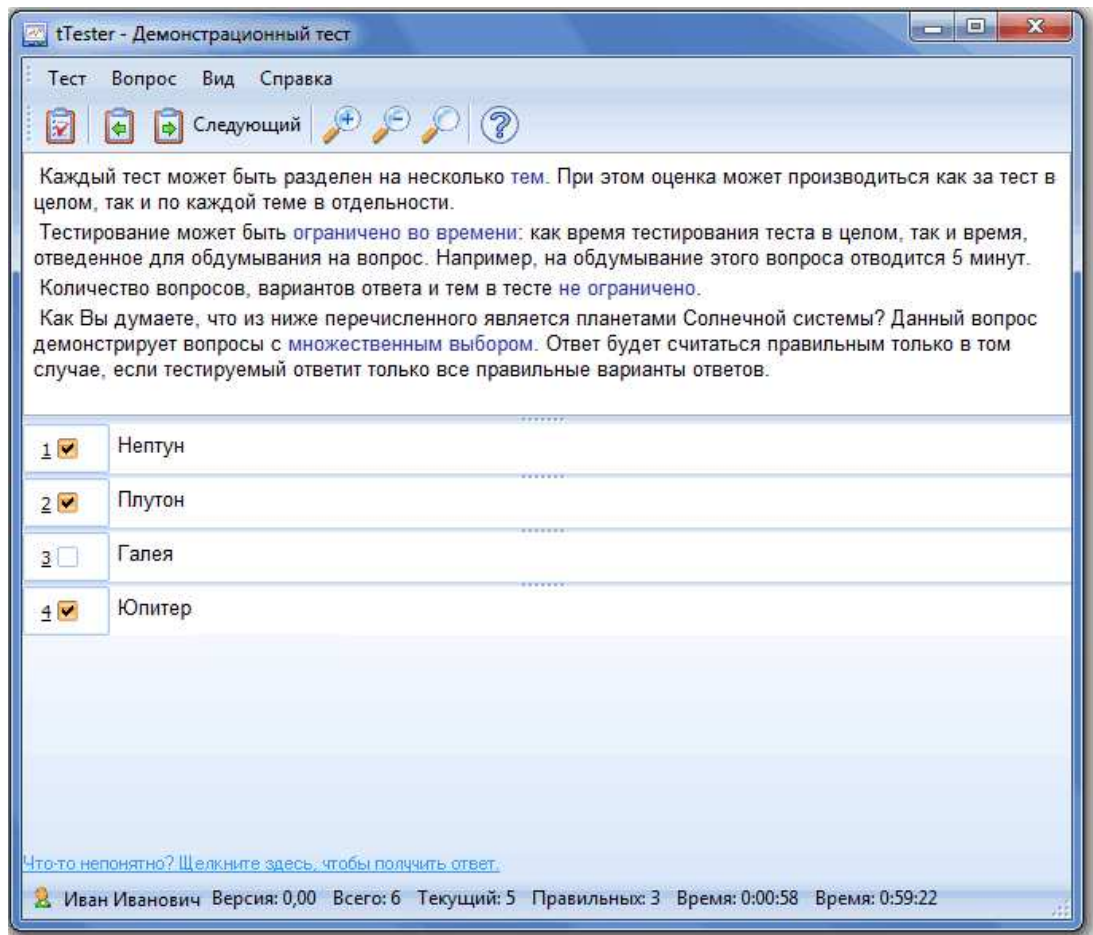


Рисунок 1.6 – Пакет программ для создания и прохождения тестов SunRay TestOfficePro

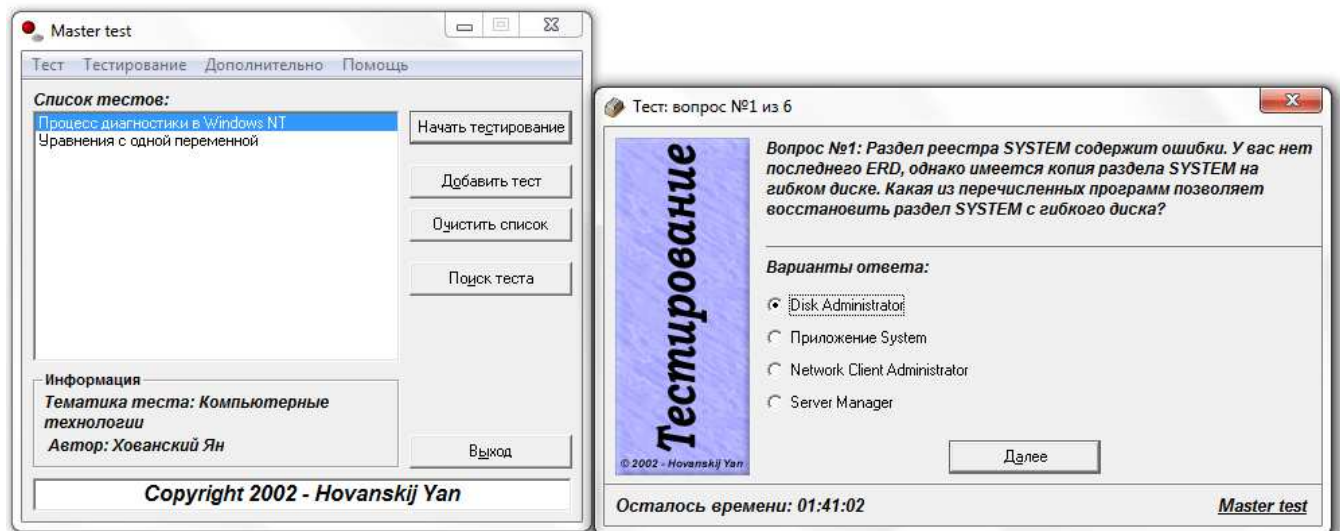


Рисунок 1.7 – Программа для создания тестов Master Test 1.0

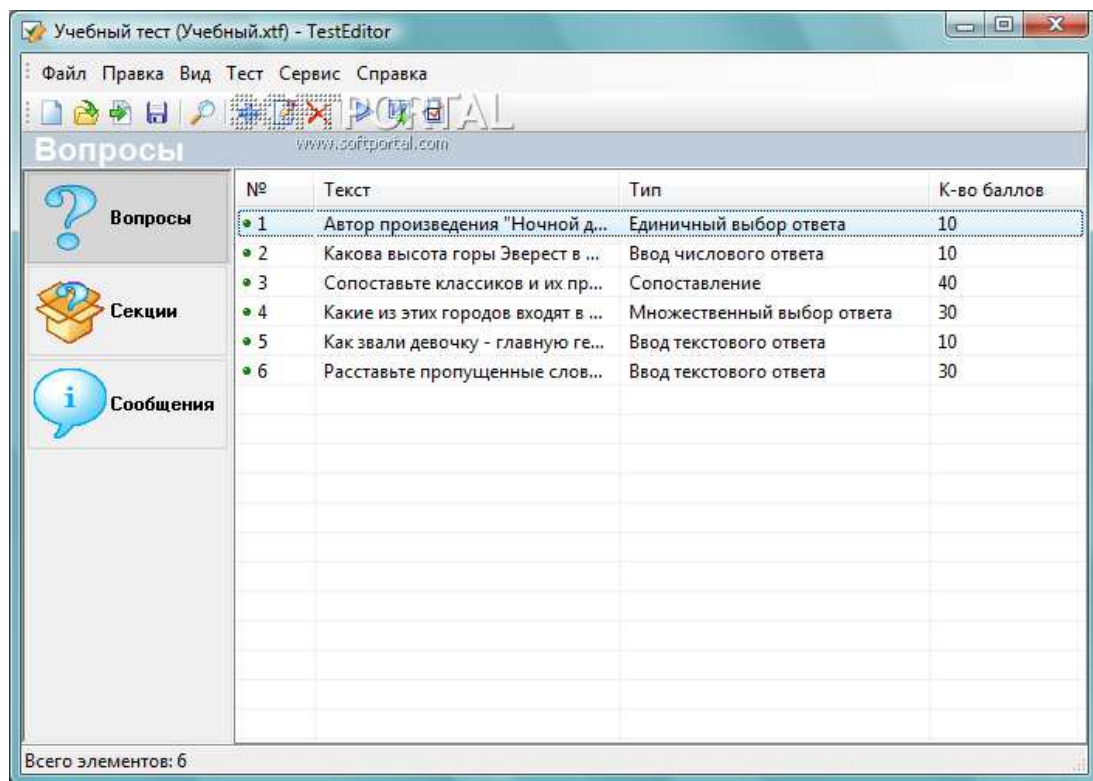


Рисунок 1.8 – Пакет программ для створення та проходження тестів **VerbalTest Professional**

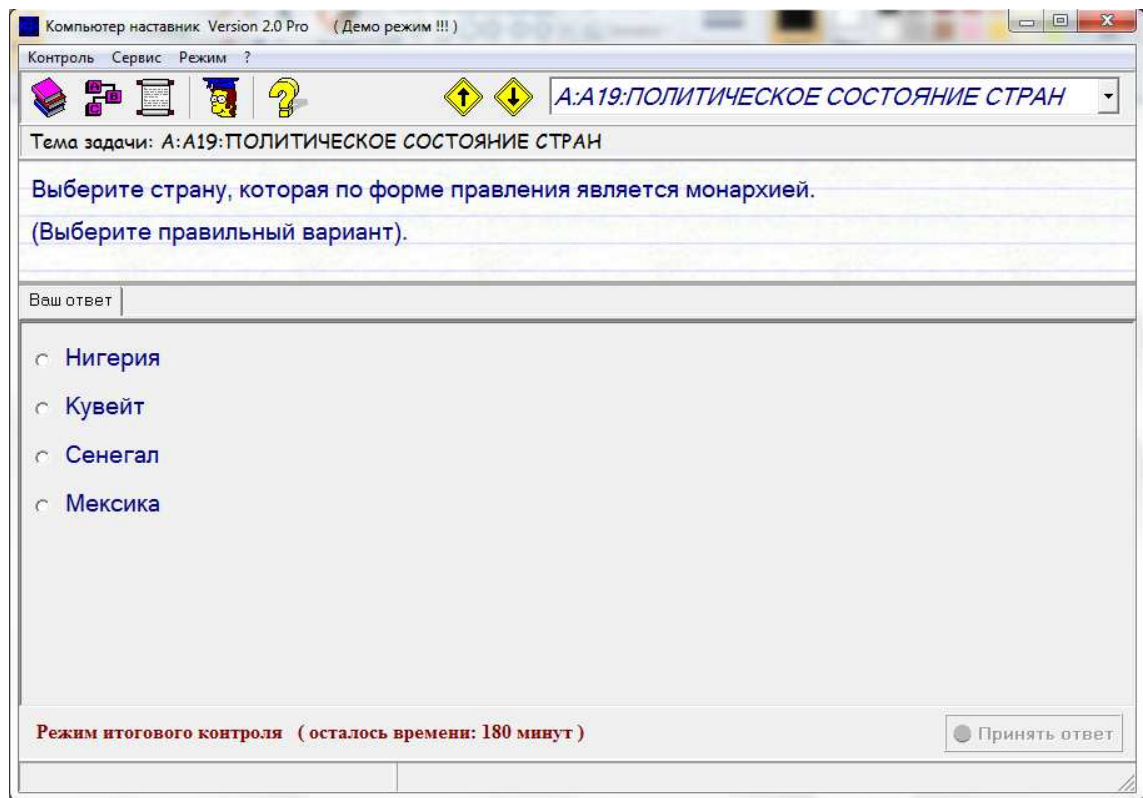


Рисунок 1.9 – Програма для проходження тестів **ComputerMentor Pro**

Вступление
Результаты
Просмотр
Редактировать

Тест
Вопросы
Категории
Импорт
Экспорт

Вопросы этого теста

Пока не добавлено ни одного вопроса

Банк вопросов

Название категории

Default for Гуманная педагогика (3)

☒ Отображать вопросы находящиеся в подкатегориях

☐ Также показывать старые вопросы

☐ Отображать содержание вопроса в списке

The default category for questions shared in context 'Гуманная педагогика'.

Создать новый вопрос

Выбрать...

Сортировать по типу, названию

Действие
Название вопроса
Тип

<< < > >> x
Высказывания
:::

<< < > >> x
Педагоги-гуманисты
:::

<< < > >> x
Макаренко
:

Выбрать все / Убрать выделение

С выбранными:

< Добавить в тест
Удалить
Переместить в >>

Default for Гуманная педагогика (3)

Добавить 1 случайных вопросов
Добавить

Рисунок 1.10 – Система управління курсами (електронне навчання) - Moodle

sidorov [Тест 10 класс. Простейшие вычисления]: Сдать решение

Настройки
Выйти из системы [sidorov]

Инфо
Итог
Посылки
Положение участников
Отправить вопрос
Сообщения

19:56:11 / ТУРНИР ИДЕТ / Остается: 830:13:19

A
B
C
D

Задача В-Средний балл второй группы 10Г по физике.

Задача В Найти средний балл за контрольную

Ограничение времени: 1 с
Ограничение памяти: 64 М

Вторая группа 10Г класса гимназии состоит из 8 человек. На контрольной по физике Владимир Александрович выставил некоторые оценки от 2 до 5 баллов.

Найти средний балл по группе за эту контрольную.

Каждое число последовательности не меньше 2 и не больше 5.

На стандартный поток вывода напечатайте ЦЕЛОЕ число. Ответ 4.0078000E+000 или 4.00 не будет считаться правильным, поскольку такой ответ не форматирован.

Например, если было 3 оценки 3 3 5, пример ввода 3-х оценок read(a,b,c). Именно read !!!

Примеры

Входные данные	Результат работы
3 3 5 4 5 5 3 4 4	4.4
5 5 5 5 5 5 5 5 5	5
4 4 5 4 5 5 3 4 4	4.4

Сдать решение

Язык: fpc - Free Pascal 2.2.0

Файл
Обзор...

Отправить!

[Предыдущая задача](#)
[Следующая задача](#)

Рисунок 1.11 – Система для проведения олімпіад EJudge

За даними порівняльної таблиці можна зробити висновок, що існуючі системи мають різний функціонал, але в них немає адаптивності та аналізу результатів тестових завдань. EJudge має аналіз результатів тестових завдань, але функціонально вона створена для олімпіад. Наявність цих функцій в адаптивній системі може створити зворотний зв'язок між засобами контролю (тести) і засобами подання матеріалу (викладач і план викладу матеріалу по якому відбувається навчальний процес). Основною проблемою є те що у більшості існуючих систем зворотній зв'язок та аналіз результатів тестових завдань відсутній.

1.7 Розробка механізмів вдосконалення зв'язку "викладач-студент".

Створення адаптивної системи контролю знань – це один з перспективних способів підвищення ефективності процесу навчання. Адаптивний контроль навчання являє собою сучасний комп'ютерний варіант реалізації відомого педагогічного принципу індивідуалізації навчання – одного з найважливіших для підготовки кваліфікованих фахівців. Цей принцип не міг бути реалізований ефективним чином в умовах традиційних класно-урочних форм занять (лекційних).

Весь навчальний процес пронизаний можливостями адаптації до індивідуальних особливостей студентів, в умовах колективного навчання. Перехід до розвиваючого навчання без адаптації до індивідуальних особливостей студентів практично неможливий. Саме під час індивідуального контакту вчителя з студентом важливо мати інструмент для контрольного тестування рівнів навченості. Контроль проводиться викладачем і не впливає на оцінку, він дозволяє побачити стан навченості кожного студента та всієї групи і внести відповідні корекції в навчальний процес. Важливо знати чи засвоєний навчальний базовий мінімум. Решта матеріалу опрацьовується з орієнтацією на запам'ятовування, що розширює можливості кожного студента, зайнятого активною творчою діяльністю. Такий процес вимагає комп'ютерної технології створення, зберігання тестів, проведення тестування і обробки результатів тестування. По суті, ця технологія дає початок новій організації навчального

процесу в цілому, на більш високому науковому рівні. В умовах масової освіти адаптивне навчання дає можливість ефективної практичної реалізації принципу індивідуалізації навчання. Застосування адаптованої системи оцінки знань дозволить підвищити об'єктивність в оцінці індивідуального рівня навченості студента так і оцінці колективного рівня навченості. Система дасть можливість моніторити зв'язок "викладач-студент". Моніторинг – це безперервні контролюючі дії в зв'язки "викладач-студент", що дозволяють спостерігати (і коригувати в міру необхідності) просування навчання від незнання до знання [16]. Це регулярне відстеження якості засвоєння знань і умінь у навчальному процесі. Він відрізняється від звичайної оцінки знань тим, що забезпечує викладача оперативної зворотним зв'язком про рівень засвоєння студентами обов'язкового навчального матеріалу. Система моніторингу включає в себе створення безпосередньо інструментів контролю знань і умінь і коригувальну методику, орієнтовану безпосередньо на особистість студента з урахуванням його індивідуальних досягнень у навчальному процесі. Об'єктом управління при навчанні є той хто навчається (студент). Суб'єктами управління виступають викладач і система (тому що вона не замінює, а доповнює викладача). Моніторинг модернізує існуючий зв'язок "викладач-студент" рисунок 1.12 [16].

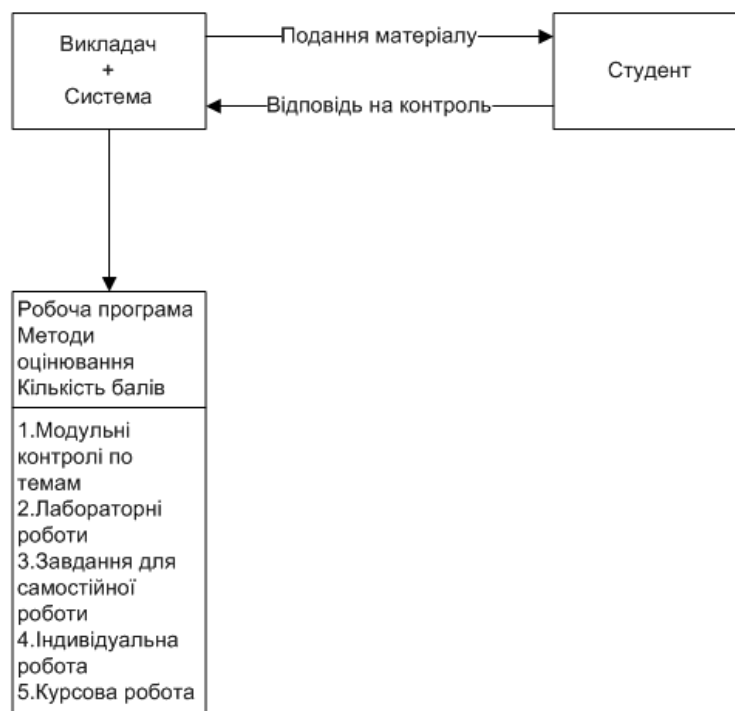


Рисунок 1.12 – Зв'язок між об'єктом та суб'єктом навчання

Спільною проблемою більшості перерахованих систем є відсутність аналізу результатів тестування по сегментам та відсутність адаптивності. Цей аналіз може створити зворотній зв'язок[17]. Тому рішенням цієї проблеми є створення адаптивної системи оцінки знань яка має такі функції:

1. Багатокористувальницька.
2. Аналіз результатів одного студенту.
3. Адаптивне пристосування матеріалу системою до кожного студента в залежності від його знань.
4. Аналіз загальних результатів тестування.
5. Аналіз результатів тестування по сегментам.

Новизною цієї системи є аналіз результатів тесту по сегментам та адаптивне пристосування до кожного студента. Така система допоможе створити зворотній зв'язок між засобами контролю і засобами подання матеріалу. Це, в свою чергу, допоможе якісно оцінити знання як одного студента так і групи студентів. Це надасть можливість знайти індивідуальний підхід до кожного студента, вдосконалити існуючі учбові плани що, в решті-решт, покращить учбовий процес.

2. СТАТИСТИЧНА ОБРОБКА ЗНАНЬ СТУДЕНТІВ ТА СКЛАДНОСТІ ЗАВДАННЯ.

2.1 Статистична обробка результатів тестування.

Виконання статистичної обробки результатів тестування починається з формування так званої матриці тестових результатів

Матриця результатів тестування a_{ij} – це матриця розмірності $N \times M$, що містить числові позначення градації індикатора, пов'язаного з досліджуваної латентної змінної, де M – число індикаторів, N – число піддослідних.

Ця матриця являє собою таблицю, рядки якої відповідають випробуваним, а стовпці – індикаторним змінним. У разі тестування навчальних досягнень індикаторними змінними є тестові завдання. На перетині рядків і стовпців знаходиться число, відповідне відповіді даного студента на дане завдання.

Профіль студентів – це послідовність значень індикатора у впорядкованій матриці тестових результатів.

У дихотомічному разі відповіді випробуваного характеризується двома символами (цифрами) – 0 і 1. Нулю відповідає невірну відповідь, одиниці – вірну відповідь.

На практиці найчастіше використовується дихотомічний випадок, тому надалі ми будемо розглядати саме його.

Бінарною матрицею називається матриця результатів тестування для дихотомічного випадку. Приклад дихотомічної матриці наведено в таблиці 2.1

Таблиця 2.1 Бінарна матриця (11×9).

	1	2	3	4	5	6	7	8	9
1. Иванов	0	0	1	0	1	0	1	1	1
2. Сидоров	0	0	0	0	1	0	0	1	1
3. Петров	0	0	0	0	0	1	0	1	0
4. Алексеев	0	0	0	0	1	0	1	1	1
5. Михайлов	1	0	0	0	1	0	0	1	0
6. Федоров	1	1	1	0	0	0	1	1	1
7. Антонов	1	1	1	1	1	1	1	1	1
8. Кузнецов	1	0	1	1	0	1	1	1	1
9. Болдырев	0	0	0	0	0	0	1	1	0
10. Яковлев	1	0	1	0	1	1	1	1	1
11. Громов	1	1	1	0	1	1	1	1	1

Для подальшого аналізу, нам будуть потрібні значення X_i – індивідуального бали i -го випробуваного, кількість вірних відповідей R_j , на j -е завдання, кількість невірних відповідей W_j на j -е завдання, частка вірних відповідей p_j і частка невірних відповідей q_j .

$$X_i = \sum_{j=1}^M a_{ij}.$$

У нашому випадку ($M = 9$) індивідуальний тестовий бал, наприклад, для другого випробуваного ($i = 2$) дорівнює:

$$\begin{aligned} X_2 &= \sum_{j=1}^9 a_{2j} = a_{21} + a_{22} + a_{23} + \dots + a_{29} = \\ &= 0 + 0 + 0 + 0 + 1 + 0 + 0 + 1 + 1 = 3. \end{aligned}$$

$$R_j = \sum_{i=1}^N a_{ij}.$$

$$W_i = N - R_j.$$

У нашому випадку ($N = 11$) для третього завдання ($j = 3$) отримуємо:

$$\begin{aligned} R_3 &= \sum_{i=1}^{11} a_{i3} = a_{13} + a_{23} + a_{33} + \dots + a_{113} = \\ &= 1 + 0 + 0 + 0 + 0 + 1 + 1 + 1 + 0 + 1 + 1 = 6. \\ W_3 &= 11 - R_3 = 11 - 6 = 5. \end{aligned}$$

Частка вірних відповідей p_j на j -е завдання дорівнює:

$$p_j = \frac{R_j}{N}.$$

Параметр p_j прийнято називати мірою складності завдання, хоча логічно було б називати його мірою легкості тестового завдання.

Частка невірних відповідей дорівнює:

$$q_j = 1 - p_j.$$

У нашому випадку для третього завдання отримаємо:

$$p_3 = \frac{R_3}{N} = \frac{6}{11} = 0,545.$$

$$q_3 = 1 - p_3 = 1 - 0,545 = 0,455.$$

У таблиці 2.2 наведені обчислені значення X_i і R_j . Для зручності візуального аналізу комірки з нульовими значеннями заштриховані. На відміну від попередньої таблиці, тут прізвища випробовуваних замінені їх номерами. Крайній правий стовпець містить значення індивідуальних тестових балів випробовуваних X_i . Сама нижня рядок містить кількість правильних відповідей на завдання R_j .

Таблиця 2.2 Бінарна матриця (11×9) з індивідуальними тестовими балами піддослідних.

	1	2	3	4	5	6	7	8	9	X_i
1	0	0	1	0	1	0	1	1	1	5
2	0	0	0	0	1	0	0	1	1	3
3	0	0	0	0	0	1	0	1	0	2
4	0	0	0	0	1	0	1	1	1	4
5	1	0	0	0	1	0	0	1	0	3
6	1	1	1	0	0	0	1	1	1	6
7	1	1	1	1	1	1	1	1	1	9
8	1	0	1	1	0	1	1	1	1	7
9	0	0	0	0	0	0	1	1	0	2
10	1	0	1	0	1	1	1	1	1	7
11	1	1	1	0	1	1	1	1	1	8
R_j	6	3	6	2	7	5	8	11	8	56

Далі, необхідно впорядкувати бінарну матрицю. Спочатку виконаємо впорядкування за величиною X_i . Для цього рядок з найбільшим значеннями X_i перемістимо вгору, з меншими значеннями – вниз. В результаті, рядок № 7 пересунулася на самий верх, а рядок № 9 – вниз. Аналогічно виконується впорядкування за значенням R_j . У цьому випадку переміщуються стовпці. Стовпець № 8 стає крайнім лівим, а № 4 – крайнім правим (таблиця 2.3).

Зазначимо, що після упорядкування, нумерація рядків і стовпців порушилася. Стовпець № 8 є першим, № 7 – другим, № 9 – третій і т.д. Нумери стовпців збережені саме такими, якими були номери завдань у тесті. Це важливо для встановлення взаємно однозначної відповідності між завданнями в

впорядкованої і вихідної матрицях. Це ж стосується й номерам рядків (випробовуваних).

Таблиця 2.3 Бінарна матриця (11×9), впорядкована по X_i і по R_j .

	8	7	9	5	1	3	6	2	4	X_i
7	1	1	1	1	1	1	1	1	1	9
11	1	1	1	1	1	1	1	1	0	8
8	1	1	1	0	1	1	1	0	1	7
10	1	1	1	1	1	1	1	0	0	7
6	1	1	1	0	1	1	0	1	0	6
1	1	1	1	1	0	1	0	0	0	5
4	1	1	1	1	0	0	0	0	0	4
2	1	0	1	1	0	0	0	0	0	3
5	1	0	0	1	1	0	0	0	0	3
3	1	0	0	0	0	0	1	0	0	2
9	1	1	0	0	0	0	0	0	0	2
R_j	11	8	8	7	6	6	5	3	2	56

З отриманої таблиці видно, що завдання № 8 успішно виконали всі 11 досліджуваних. Це завдання не дозволяє диференціювати піддослідних, тому його слід видалити з тесту. Відзначимо, що ця вимога стосується тільки нормативно-орієнтованих тестів. Відповідно, якби на якесь завдання не відповів жоден випробуваний ($R_j = 0$), то це завдання теж повинно бути видалено з тісту. Крім того, випробуваний № 7 успішно виконав всі завдання тесту. Тест не дає інформації про досліджуваного, за винятком того, що для нього всі завдання занадто легкі. Рядок № 7 також слід видалити з матриці.

Після видалення стовпця № 8 і рядки № 7 ми отримаємо редуційовану бінарну матрицю (таблиця 2.4).

У цій таблиці для зручності додана нова нумерація рядків і стовпців. У нижній частині таблиці наведені значення R_j , W_j , p_j , q_j .

Важливим параметром тестового завдання є варіація (дисперсія) тестових балів p_j , q_j . Чим більше варіація, тим краще завдання диференціює

випробовуваних.

При $p_j = 0$ та $p_j = 1$ дисперсія завдання дорівнює нулю. Іншими словами, якщо на завдання не відповів жоден випробуваний або успішно відповіли всі, то завдання не може їх диференціювати за рівнем підготовленості.

Таблиця 2.4 Відредагована бінарна матриця 10×8 .

№	нові	1	2	3	4	5	6	7	8		
нові	старі	7	9	5	1	3	6	2	4	X_i	X_i^2
1	11	1	1	1	1	1	1	1	0	7	49
2	8	1	1	0	1	1	1	0	1	6	36
3	10	1	1	1	1	1	1	0	0	6	36
4	6	1	1	0	1	1	0	1	0	5	25
5	1	1	1	1	0	1	0	0	0	4	16
6	4	1	1	1	0	0	0	0	0	3	9
7	2	0	1	1	0	0	0	0	0	2	4
8	5	0	0	1	1	0	0	0	0	2	4
9	3	0	0	0	0	0	1	0	0	1	1
10	9	1	0	0	0	0	0	0	0	1	1
R_j		7	7	6	5	5	4	2	1	37	181
W_j		3	3	4	5	5	6	8	9		
p_j		0,7	0,7	0,6	0,5	0,5	0,4	0,2	0,1		
q_j		0,3	0,3	0,4	0,5	0,5	0,6	0,8	0,9		
$p_j * q_j$		0,21	0,21	0,24	0,25	0,25	0,24	0,16	0,09		

Бінарна матриця (таблиця 2.4) має характерну особливість – майже всі нулі і одиниці розподілилися щодо діагоналі, яка йде з лівого нижнього кута в правий верхній.

Згідно Гуттманн це розмежування має бути ідеальним. Якщо випробуваний, вірно відповів на важке завдання, то він тим більше, повинен впоратися з більш легкими завданнями. Це повинно приводити до суворого розмежування одиниць і нулів діагоналлю матриці.

Насправді ж це не зовсім так. Наприклад, у нашій бінарної матриці профіль випробуваного № 2 (№8 за старою нумерацією) сильно відхиляється від правила Гуттмана. Цей випробуваний впорався з найважчим завданням № 8 (№ 4 за старою нумерацією), але не впорався з більш легкими завданнями № 7 та № 3. Профіль випробуваного спотворений. Якби одиниці і нулі помінялися місцями,

тобто випробовуваний, вірно відповів на важкі питання, але не впорався з легкими, то кажуть, у нього інвертований профіль.

Інвертований профіль свідчить або про невірну структуру знань випробуваного, або про порушення процедури тестування (списування, вгадування і т.д.), або про недоліки тестових завдань (за формою і (або) за змістом).

2.2 Розрахунок рівня підготовленості студентів.

Розглянемо процедуру обчислення рівня підготовленості θ_i . В якості вихідних даних візьмемо бінарну матрицю з таблиці 2.4.

Спочатку необхідно обчислити частки вірних p_i і невірних q_i відповідей студентів.

$$p_i = \frac{X_i}{M}.$$

$$q_i = 1 - p_i.$$

де X_i – індивідуальний бал випробуваного, M – кількість завдань у тесті.

Наприклад, для 2-го випробуваного маємо:

$$p_2 = \frac{6}{8} = 0,75.$$

$$q_2 = 1 - p_2 = 1 - 0,75 = 0,25.$$

Далі обчислюємо початкові значення рівня підготовленості випробуваних за формулою

$$\theta_i^0 = \ln \frac{p_i}{q_i}.$$

Для 2-го випробуваного маємо

$$\theta_2^0 = \ln \frac{0,75}{0,25} = 1,099.$$

Аналогічні розрахунки виконуються для всіх десяти піддослідних (таблиця 2.4) і заносяться в таблицю 2.5.

Таблиця 2.5 Початкові значення рівня підготовленості випробуваних

i	X_i	p_i	q_i	q_i^0	$(q_i^0)^2$
1	7	0,875	0,125	1,946	3,786
2	6	0,750	0,250	1,099	1,207
3	6	0,750	0,250	1,099	1,207
4	6	0,750	0,250	1,099	1,207
5	4	0,500	0,500	0,000	0
6	3	0,375	0,625	-0,511	0,261
7	2	0,250	0,750	-1,099	1,207
8	2	0,250	0,750	-1,099	1,207
9	1	0,125	0,875	-1,946	3,786
10	1	0,125	0,875	-1,946	3,786
				$\sum (q_i^0)^2 =$	17,655

Тепер ми можемо обчислити середні значення рівня підготовленості випробуваних.

$$\bar{\theta} = \frac{\sum_{i=1}^N \theta_i^0}{N} = \frac{1,946 + 1,099 + 1,099 + 1,099 + 0 - 0,511 - 1,099 - 1,099 - 1,946 - 1,946}{10} = -0,136.$$

Отримані результати нам треба звести в єдину шкалу стандартних оцінок. Для цього необхідно обчислити дисперсію S_θ , використовуючи дані з таблиці 2.5.

$$S_\theta = \frac{\sum_{i=1}^N (\theta_i^0)^2 - N(\bar{\theta})^2}{N-1} = \frac{17,655 - 10 \cdot (-0,136)^2}{10-1} = 1,941.$$

Далі обчислюємо кутовий коефіцієнт

$$a_\theta = \sqrt{\frac{1 + S_\beta / 2,89}{1 - S_\theta S_\beta / 8,35}} = \sqrt{\frac{1 + 1,153 / 2,89}{1 - 1,941 \cdot 1,153 / 8,35}} = 1,911.$$

Нарешті, ми можемо записати оцінки параметра θ на єдиній інтервальній шкалі

$$\theta_i = a_\theta \theta_i^0 + \bar{\beta} = 1,911 \theta_i^0 + 0,236.$$

Нам залишилося обчислити стандартну помилку вимірювання $S_E(\theta_i)$ для θ_i

$$S_E(\theta_i) = \frac{a_\theta}{\sqrt{p_i(M - X_i)}} = \frac{a_\theta}{\sqrt{M p_i(1 - p_i)}} = \frac{a_\theta}{\sqrt{M p_i q_i}}.$$

Наприклад, для першого випробуваного отримаємо

$$S_E(\theta_i) = \frac{1,911}{\sqrt{8 \cdot 0,845 \cdot 0,125}} = 2,043.$$

Обчислені значення стандартних помилок наведені в таблицях 2.6 (третій стовпчик).

Таблиця 2.6 Розрахункові параметри для рівня підготовленості випробуваних

i	θ_i	$S_E(\theta_i)$	θ_i'
1	3,955	2,043	2,436
2	2,335	1,560	1,365
3	2,335	1,560	1,365
4	2,335	1,560	0,523
5	0,236	1,351	-0,157
6	-0,740	1,396	-0,781
7	-1,863	1,560	-1,431
8	-1,863	1,560	-1,431
9	-3,483	2,043	-2,217
10	-3,483	2,043	-2,217

Якщо студент має рівень знань $\theta_i < 1$, то цей студент має суттєві прогалини в знаннях від яких потрібно позбавитися, повторно пройшовши матеріал. Чим менше значення θ_i тим гірша підготовленість у студента. Для отримання атестації студент повинен мати рівень знань $\theta_i \geq 1$. Чим більше значення θ_i тим краще підготовлений студент. Такому студенту потрібно давати більш складний матеріал. Сума значень θ_i показує загальний рівень підготовленості групи студентів.

2.3 Розрахунок складності завдання.

Для розрахунку рівня складності завдань для початку потрібно, обчислюємо початкове значення складності завдань β_j .

$$\beta_j^0 = \ln \frac{p_j}{q_j}$$

Тут j приймає значення від 1 до M , де M – кількість випробовуваних. Як приклад розрахуємо початкове значення труднощі 2-го завдання. Величини p_j та q_j розраховані нами раніше і наведені в таблиці 2.4.

$$\beta_2^0 = \ln \frac{p_2}{q_2} = \ln \frac{0,3}{0,7} = -0,847.$$

Далі розраховуємо для всіх завдань їх початковий рівень складності.

Таблиця 2.7 Початкові значення складності завдань

j	R_j	p_i	q_i	b_j^0	$(b_j^0)^2$
1	7	0,700	0,300	-0,847	0,718
2	7	0,700	0,300	-0,847	0,718
3	6	0,600	0,400	-0,405	0,164
4	5	0,500	0,500	0,000	0
5	5	0,500	0,500	0,000	0
6	4	0,400	0,600	0,405	0,164
7	2	0,200	0,800	1,386	1,922
8	1	0,100	0,900	2,197	4,828
				$\sum (b_j^0)^2 = 8,514$	

Тепер ми можемо обчислити середнє значення складності завдань.

$$\bar{\beta} = \frac{\sum_{j=1}^M \beta_j^0}{M} = \frac{-0,847 - 0,847 - 0,405 + 0 + 0 + 0,405 + 1,386 + 2,197}{8} = 0,236.$$

Отримані результати нам треба звести в єдину шкалу стандартних оцінок. Для цього необхідно обчислити дисперсію S_β , використовуючи дані з таблиці 2.7.

$$S_\beta = \frac{\sum_{j=1}^M (\beta_j^0)^2 - M(\bar{\beta})^2}{M-1} = \frac{8,514 - 8 \cdot 0,236^2}{8-1} = 1,153.$$

Далі обчислюємо кутовий коефіцієнт

$$a_\beta = \sqrt{\frac{1 + S_\theta/2,89}{1 - S_\theta S_\beta/8,35}} = \sqrt{\frac{1 + 1,941/2,89}{1 - 1,941 \cdot 1,153/8,35}} = 2,284.$$

Нарешті, ми можемо записати оцінку параметра β на єдиній інтервальній шкалі

$$\beta_j = a_\beta \beta_j^0 + \bar{\theta} = 2,284 \beta_j^0 - 0,136.$$

Відповідні результати зведені в таблиці 2.8 (другий стовпець).

Таблиця 2.8 Розрахункові параметри для труднощі завдань тесту

j	β_j	$S_E(\beta_j)$	β_j'
1	-2,071	1,576	-1,545
2	-2,071	1,576	-1,669
3	-1,062	1,474	-0,603
4	-0,136	1,445	-0,502
5	-0,136	1,445	-0,256
6	0,790	1,474	0,102
7	3,030	1,806	1,854
8	4,882	2,408	2,620

З таблиці 2.8 випливає, що

$$\sum_{\beta} = \sum_{j=1}^N \beta_j = 3,226.$$

Тобто, завдань з позитивними β_j більше, ніж з негативними. Даний тест не збалансований, він містить більше важких завдань, ніж легких. Рекомендується прагнути до того, щоб $\sum_{\beta}$ було близько до нуля.

Нам залишилося обчислити стандартну помилку вимірювання $S_E(\beta_j)$ для β_j .

$$S_E(\beta_j) = \frac{a_{\beta}}{\sqrt{p_j(N - R_j)}} = \frac{a_{\beta}}{\sqrt{Np_j(1 - p_j)}} = \frac{a_{\beta}}{\sqrt{Np_jq_j}}.$$

Для першого завдання стандартна помилка дорівнює

$$S_E(\beta_1) = \frac{2,284}{\sqrt{10 \cdot 0,7 \cdot 0,3}} = 1,576.$$

Якщо значення β_j дуже мале то запитання є легким. Тому це запитання потрібно прибрати. Та якщо β_j дуже велике то запитання є складним. Тому це запитання потрібно теж прибрати. Сума значень β_j показує нам загальну складність або легкість тесту. На основі результатів досліджень ми прийшли до висновків що:

Адаптивне навчання – це спосіб організації навчального процесу з урахуванням індивідуального рівня підготовки студента до початку навчання або в процесі навчання. Його використання дає можливість вдосконалити навчальний процес.

IRT – це сучасна теорія педагогічних вимірювань, долає недоліки інших теорій і що вона здатна вирішити багато, якщо не всі проблеми підвищення якості педагогічних вимірювань.

Математична теорія параметричного оцінювання тестових завдань і результатів тестування має багато переваг. Але на мою думку її можливості можна вдосконалити за допомогою деталізації яка зможе створити зворотній зв'язок. Викладач зможе побачити який сегмент запитань більш складніший а який легкий. Та на основі отриманих результатів буде мати змогу скорегувати навчальний план для окремої групи.

Більшість існуючих систем створені для простого проходження тестів та отримання студентів балу. В них відсутні аналізу результатів тестування по сегментам та адаптивності. Тому рішенням цієї проблеми є створення адаптивної системи оцінки знань яка має такі функції:

1. Багатокористувальницька.
2. Аналіз результатів одного студенту.
3. Адаптивне пристосування матеріалу системою до кожного студента в залежності від його знань.
4. Аналіз результатів загального результату тестування.
5. Аналіз результатів тестування по сегментам.

На підставі проведених досліджень переходимо до розробки проекту та створення адаптивної системи.

3. РОЗРОБКА ПРОЕКТУ СИСТЕМИ АДАПТИВНОГО НАВЧАННЯ.

3.1 Інформаційне забезпечення.

3.1.1 Діаграми потоків даних.

Маємо наступні зовнішні сутності: "Студент", "Адміністратор" та "Викладач" які взаємодіють з "Адаптивною системою навчання".

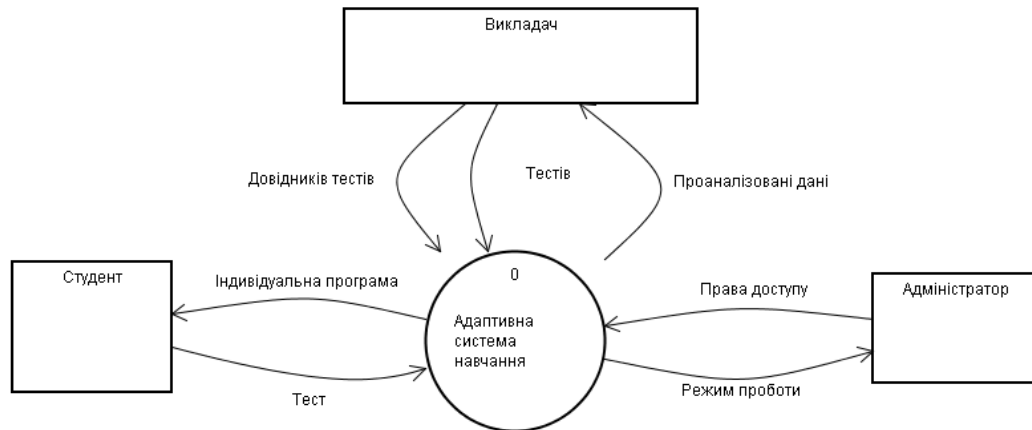


Рисунок 3.1 – Концептуальна діаграма

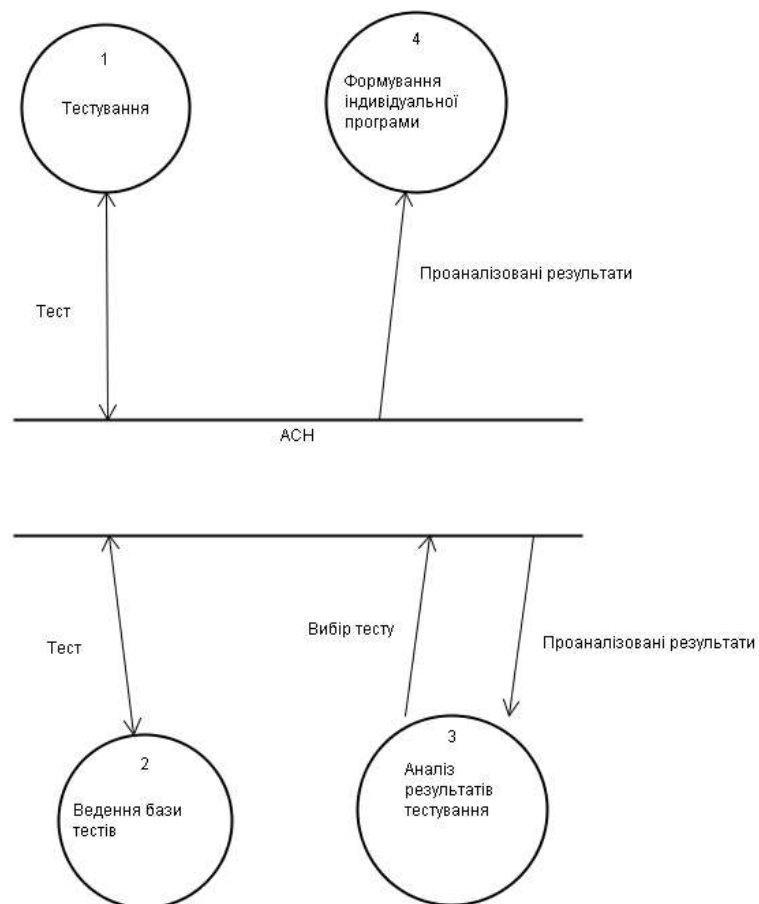


Рисунок 3.2 – Діаграма потоків даних 0-го рівня

3.1.2 Розробка інформаційної моделі програмного забезпечення (ER-діаграма).

Інформаційна модель бази даних визначає систему основних понять і правил їхнього комбінування, які не залежать від засобів. На основі раніше розробленої моделі потоків даних (діаграма DFD), розробимо інформаційну модель даних ERD (використовуючи нотацію "сутність-зв'язок") для бази даних реєстрації фізичних осіб. Дана модель зображена на рисунку 3.3.

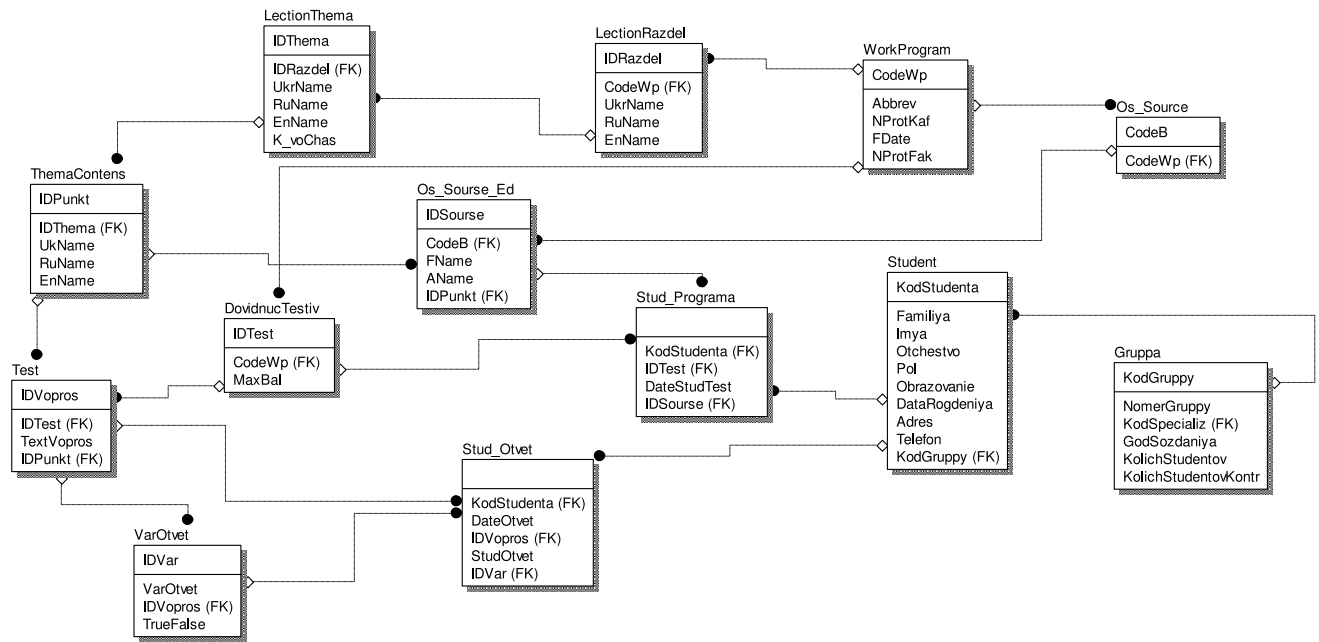


Рисунок 3.3 – Концептуальна модель даних ERD

Модель даних має такі сутності: Група, Студент, Відповідь студента, Програма студента, Довідник тестів, Тест, Пункт теми, Тема лекції, Розділ лекції, Робоча програма.

Сутність "Відповідь студента" має такі атрибути: Код студента, Дата відповіді, Номер запитання, Відповідь студента.

Сутність "Програма студента" має такі атрибути: Код студента, Номер тесту, Дата проходження тесту, Код розділу.

Сутність "Довідник тестів" має такі атрибути: Номер тесту, Код робочої програми, Максимальний бал, Кількість варіантів.

Сутність "Література" має такі атрибути: Код робочої програми, Код книги.

Сутність "Розділ книги" має такі атрибути: Код розділу, Назва розділу,

Електронна адреса, Пункт теми, Код книги.

Сутність "Тест" має такі атрибути: Номер запитання, Текст запитання, Номер тесту, Код пункту теми, Відповідь 1, Відповідь 2, Відповідь 3, Правильна відповідь.

Сутність "Пункт теми" має атрибути: Код пункту, Українська назва, Російська назва, Англійська назва, Номер теми.

Сутність "Тема лекції" має атрибути: Номер теми, Українська назва, Російська назва, Англійська назва, Кількість годин, Номер модулю.

Сутність "Розділ лекції" має атрибути: Номер модулю, Код робочої програми, Українська назва, Російська назва, Англійська назва.

Сутність "Робоча програма" має атрибути: Код робочої програми, Коротка назва дисципліни, Профілюючий факультет, Профілююча кафедра, Фактична дата.

Сутність "Група" має атрибути: Код групи, Номер групи, Код спеціалізації, Кількість студентів, Кількість студентів контрактників, Рік створення.

Сутність "Студент" має атрибути: Код студента, Код групи, Прізвище, Ім'я, По батькові, Освіта, Стать, Дата народження, Адреса, Телефон,

3.1.3 Розробка логічної моделі бази даних.

Базою для побудови логічної моделі даних є концептуальна модель. Логічна модель є більш детальним представленням даних. В ній сутності трансформуються в таблиці даних, одна сутність може представлятися декількома таблицями, атрибути сутностей стають полями таблиць, уточнюються відношення між таблицями, виділяються первинні ключі.

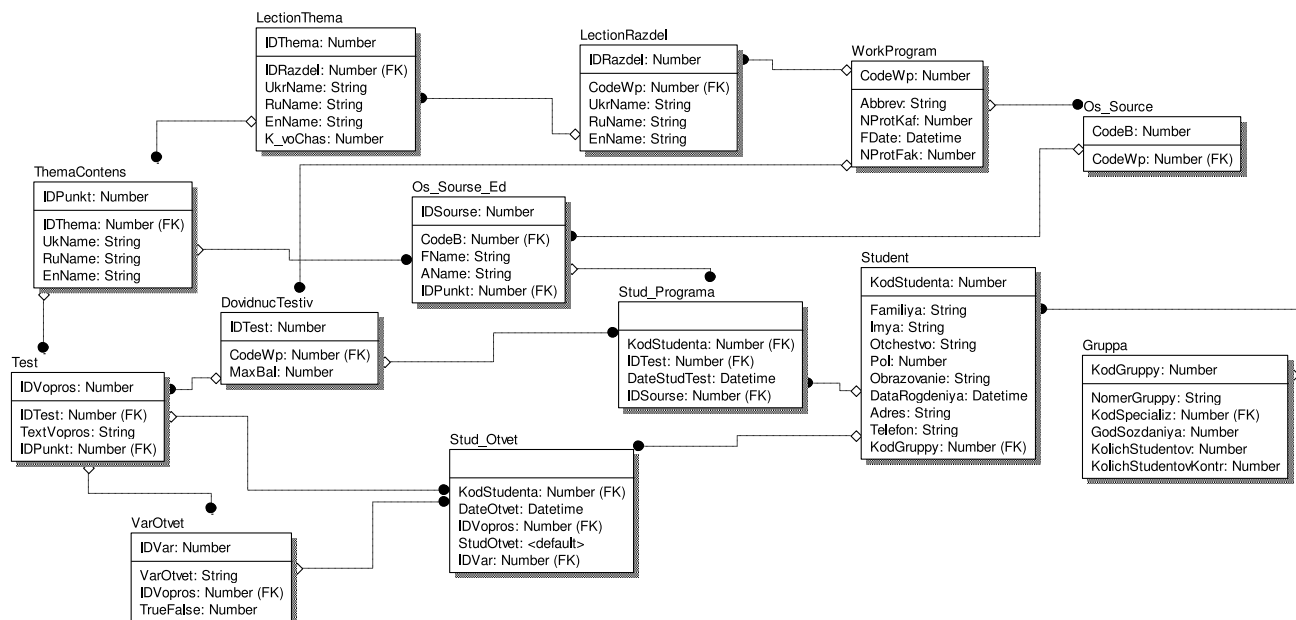


Рисунок 3.4 – Логічна модель даних

3.1.4 Розробка фізичної моделі бази даних.

Фізична модель даних ґрунтується на логічній моделі. В ній визначається тип даних та умовні імена для кожного поля в таблицях даних логічного рівня, а також формат кожного поля. Фізична модель орієнтується на вибрану СУБД – MS Access. Блок схема зроблена в All Function ERWin Data Modeler, Database: Access.

При переході від логічної моделі до фізичної необхідно забезпечити цілісність даних. Всі назви таблиць і полів таблиць, а також зв'язки між таблицями у фізичній моделі і логічній моделі повинні бути повністю відповідними.

Дані, за якими необхідно представити фізичну модель даних в СУБД – MS Access на основі раніше розробленої логічної моделі даних БД.

Дані фізичної моделі даних можна також представити у вигляді таблиці (таблиці 3.1-3.13).

Таблиця 3.3 – Фізична модель таблиці "Тема лекції"

Поле	Тип	Ім'я	Ключ	Null /Not Null
Номер теми	Числовий	IDTema	PK	Not Null
Номер модулю	Числовий	IDModul	FK	Null
Українська назва	Текстовий	UrkName		Null
Російська назва	Текстовий	RuName		Null
Англійська назва	Текстовий	EnName		Null
Кількість годин	Числовий	K_voChas		Null

Таблиця 3.4 – Фізична модель таблиці "Пункт теми"

Поле	Тип	Ім'я	Ключ	Null /Not Null
Номер пункту	Числовий	IDPunkt	PK	Not Null
Українська назва	Текстовий	UkrName		Null
Російська назва	Текстовий	RuName		Null
Англійська назва	Текстовий	EnName		Null
Номер теми	Числовий	IDTema	FK	Null

Таблиця 3.5 – Фізична модель таблиці "Література"

Поле	Тип	Ім'я	Ключ	Null /Not Null
Код книги	Числовий	CodeB	PK	Not Null
Код робочої програми	Числовий	CodeWP		Null

Таблиця 3.6 – Фізична модель таблиці "Розділ книги"

Поле	Тип	Ім'я	Ключ	Null /Not Null
Код розділу	Числовий	IDSourse	PK	Not Null
Назва розділу	Текстовий	FName	80	Null
Електрона адреса	Текстовий	AName	80	Null
Пункт теми	Числовий	IDPunkt	FK	Null
Код книги	Числовий	CodeB	FK	Null

Таблиця 3.7 – Фізична модель таблиці "Тест"

Поле	Тип	Ім'я	Ключ	Null /Not Null
Номер запитання	Числовий	IDVopros	PK	Not Null
Текст Запитання	Текстовий	TextVopros		Null
Номер тесту	Числовий	IDTest	FK	Null
Номер пункту	Числовий	IDPunkt	FK	Null

Таблиця 3.8 – Фізична модель таблиці "Довідник тестів"

Поле	Тип	Ім'я	Ключ	Null /Not Null
Номер тесту	Числовий	IDTest	PK	Not Null
Код робочої програми	Числовий	CodeWP	FK	Null
Максимальний бал	Числовий	MaxBal		Null

Таблиця 3.9 – Фізична модель таблиці "Програма студента"

Поле	Тип	Ім'я	Ключ	Null /Not Null
Код студента	Числовий	KodStudenta	FK	Not Null
Номер тесту	Числовий	IDTest	FK	Null
Дата проходження тесту	Дата/час	DateStudTest		Null
Код розділу	Числовий	CodeB	FK	Null

Таблиця 3.10 – Фізична модель таблиці "Відповідь студента"

Поле	Тип	Ім'я	Ключ	Null /Not Null
Код студента	Числовий	KodStudenta	FK	Not Null
Дата відповіді	Дата/час	DateOtvot		Null
Номер запитання	Числовий	IDVopros	FK	Null
Відповідь студента	Текстовий	StudOtvot		Null

Таблиця 3.11 – Фізична модель таблиці "Студент"

Поле	Тип	Ім'я	Ключ	Null /Not Null
Код студента	Числовий	KodStudenta	PK	Not Null
Прізвище	Текстовий	Familiya		Null
Ім'я	Текстовий	Imya		Null
По батькові	Текстовий	Otchestvo		Null
Освіта	Текстовий	Obrazovanie		Null
Стать	Текстовий	Pol		Null
День народження	Дата/час	DataRogdenya		Null
Адреса	Текстовий	Adres		Null
Телефон	Числовий	Telefon		Null
Код групи	Числовий	KodGruppy	FK	Null

Таблиця 3.12 – Фізична модель таблиці «Група»

Поле	Тип	Ім'я	Ключ	Null /Not Null
Код групи	Числовий	KodGruppy	PK	Not Null
Номер групи	Числовий	NomerGruppy		Null
Код спеціалізації	Числовий	KodSpecializ		Null
Рік створення	Числовий	GodSozdaniya		Null
Кількість студентів	Числовий	KolichStudentov		Null
Кількість студентів контрактників	Числовий	KolichStudentovKontr		Null

Таблиця 3.13 – Фізична модель таблиці "Варіант відповіді"

Поле	Тип	Ім'я	Ключ	Null /Not Null
Номер варіанту	Числовий	KodGruppy	PK	Not Null
Варіант відповіді	Текстовий	VarOtvvet		Null
Номер запитання	Числовий	IDVorpos	FK	Null
ІстинаАбоНі	Числовий	TrueFalse		Null

3.2 Технічне забезпечення.

Комп'ютерним класом має бути кабінет оснащений не менше ніж 6-ю комп'ютерами і використовуваний в навчальному процесі з дисциплін навчального плану, освітній стандарт для яких передбачає використання комп'ютерів в рамках навчального плану.

Комп'ютери навчальних класів повинні бути підключені до локальної мережі університету і мати вихід в мережу Інтернет. Налаштування мережевих параметрів комп'ютерів здійснюється відповідно до єдиної політикою управління локальною мережею.

У навчальних комп'ютерних класах встановлюється тільки легально придбане програмне забезпечення. Реєстр програмного забезпечення, рекомендованого для використання в комп'ютерних класах, розробляється і актуалізується. Системні програмні засоби, що використовуються програмою, повинні бути представлені ліцензійної локалізованої версією операційної системи сімейства Windows починаючи з Windows XP SP1. Допускається

використання пакетів оновлень.

Програма поставляється на дистрибутивному (зовнішньому оптичному) носії (компакт-диску). Комп'ютери повинні мати такі характеристики апаратного забезпечення:

Процесор: Intel Celeron G460 1800 МГц.

Відеокарта: Intel HD Graphics.

Оперативна пам'ять: DDR3 2048 Мб 1333 МГц.

Жорсткий диск: SATA 500 Gb.

Оптичний привід: DVD + /-RW.

Монітор: 19 дюймів.

Клавіатура, миша.

3.3 Програмне забезпечення

3.3.1 Мова програмування Object Pascal.

Object Pascal (англ. Об'єктний Паскаль) – мова програмування, розроблена у фірмі Apple Computer в 1986 році групою Ларрі Теслера, який консультувався з Ніклаусом Віртом.

Мова програмування Object Pascal є останньою версією сімейства мов Паскаль, що реалізує принципи об'єктно-орієнтованого програмування. Ця мова є основою системи візуального програмування Delphi. Найбільш істотною відмінністю від традиційної мови Паскаль є наявність досить складних структур даних (класи) і можливість засобами Pascal звертатися до функцій вікна API для створення повноцінних Windows-додатків. Object Pascal дозволяє використовувати безліч самих різноманітних типів і структур даних. Всі типи даних можна розбити на дві групи: прості (базові) та структуровані (користувальницькі) типи, які створюються на основі базових і об'єднують кілька змінних різних типів в одній структурі даних.

3.3.2 Середовище програмування Delphi.

Embarcadero Delphi, раніше Borland Delphi та Code Gear Delphi – це інтегроване середовище швидкої розробки програмного забезпечення.

Середовище Embarcadero Delphi працює під Microsoft Windows і підтримує розробку програмного забезпечення для операційних систем Microsoft Windows x86 та x64, Mac OS x86, Apple iOS та Android на мові програмування Delphi (Object Pascal). Embarcadero Delphi входить до групи засобів розробки Embarcadero RAD Studio і, в залежності від набору функціоналу, постачається в редакціях *Starter*, *Professional*, *Enterprise* та *Architect*.

Delphi – це комбінація кількох найважливіших технологій:

- Високопродуктивний компілятор в машинний код.
- Об'єктно-орієнтована модель компонент.
- Візуальне (а, отже, і швидкісне) побудова додатків з програмних прототипів.
- Масштабовані засоби для побудови баз даних.

Завдяки такій архітектурі, додатки, виготовлені за допомогою Delphi, працюють надійно і стійко. Delphi підтримує використання вже існуючих об'єктів, включаючи DLL, написані на C і C ++, OLE сервера, VBX, об'єкти, створені за допомогою Delphi. З готових компонент працюючі додатки збираються дуже швидко. Крім того, оскільки Delphi має повністю об'єктну орієнтацію, розробники можуть створювати свої повторно використовувані об'єкти для того, щоб зменшити витрати на розробку.

Delphi пропонує розробникам як у складі команди, так і індивідуальним відкриту архітектуру, що дозволяє додавати компоненти, де б вони не були виготовлені, і оперувати цими нововведеними компонентами у візуальному конструкторі. Розробники можуть додавати CASE-інструменти, кодові генератори, а також авторські help'и, доступні через меню Delphi. Two-way tools – однозначна відповідність між візуальним проектуванням і класичним написанням тексту програми Це означає, що розробник завжди може бачити код, відповідний тому, що він побудував за допомогою візуальних інструментів і навпаки.

3.3.3. СУБД Microsoft Access.

"Microsoft Access" (повна назва Microsoft Office Access) – система управління базами даних від компанії Майкрософт, програма, що входить до складу пакету офісних програм Microsoft Office. Має широкий спектр функцій, включаючи зв'язані запити, сортування по різних полях, зв'язок із зовнішніми таблицями і базами даних.

Access, при роботі з базою даних, інакше взаємодіє з жорстким (або гнучким) диском, ніж інші програми. В інших програмах, файл-документ, при відкритті, повністю завантажується в оперативну пам'ять, і нова редакція цього файлу (змінений файл) цілком записується на диск тільки при натисканні кнопки "зберегти".

В Access нова редакція вмісту зміненої комірки таблиці записується на диск (зберігається) відразу, як тільки курсор клавіатури буде поміщений в іншу комірку (або нова редакція зміненого запису записується на диск відразу, як тільки курсор клавіатури буде поставлений в іншу запис (рядок)). Таким чином, якщо раптово відключать електрику, то пропаде тільки зміна того запису, якого не встигли покинути.

Цілісність даних в Access забезпечується також за рахунок механізму транзакцій.

Кнопка "Зберегти" в Access теж є, але в Access у режимі перегляду даних вона потрібна, в першу чергу, для збереження зміненого режиму показу таблиці або іншого об'єкта, тобто, для збереження таких змін, як:

- Зміна ширини стовпців і висоти рядків.
- Перестановка стовпців в режимі перегляду даних, "закріплення" стовпців і звільнення закріплених стовпців.
- Зміна сортування.
- Застосування нового фільтра.
- Зміна шрифту; кольору тексту, сітки і тла.

Крім того, в Access ця кнопка потрібна в режимі "Конструктор" для збереження змін структури об'єкта бази даних, зроблених в цьому режимі.

3.4 Результати розробки.

Була розроблена адаптивна система навчання яка має такі функції:

- Формування тесту.
- Проходження тесту.
- Отримання індивідуальної програми студента.
- Обробка результатів тестування.

Створення тестових завдань

Дисципліна: СКБД Розділ: Реляційні БД Тема: Реляційна модель даних

Робоча програма | Розділи РП | Теми розділу | Підрозділи теми | Довідник тестів | Тестові завдання

Формулює тестові завдання

№ тесту: 1 Код підрозділу: 1223 Код завдання: 12238

Зміст завдання

. Спеціальний комп'ютер, виділений для спільного використання учасниками мережі, називається:

Варіант 1: Сервером

Варіант 2: Адміністратором

Варіант 3: Директором

Варіант 4: Процесора

Варіант вірний: 1

Додати завдання

Визначити № запитання

Вибір підрозділу теми

Підрозділ	Назва підрозділу
1111	Основні функції СУБД
1122	Базові поняття РБД
1131	Таблиці. Записи
1132	Ключові поля
1221	Сортування записів
1222	Визначення атрибутів суті
1223	Домен, кортеж, ключі
1311	Ключі прості та складні

Завдання тесту

IDTest	IDpunk	IDVopros	TextVopros	Var1	Var2	Var3	var4	Vime
1	1111	11111	(МЕМО)	(МЕМО)	(МЕМО)	(МЕМО)	(МЕМО)	
1	1122	11221	(МЕМО)	(МЕМО)	(МЕМО)	(МЕМО)	(МЕМО)	
1	1122	11222	(МЕМО)	(МЕМО)	(МЕМО)	(МЕМО)	(МЕМО)	
1	1131	11311	(МЕМО)	(МЕМО)	(МЕМО)	(МЕМО)	(МЕМО)	
1	1131	11312	(МЕМО)	(МЕМО)	(МЕМО)	(МЕМО)	(МЕМО)	
1	1131	11313	(МЕМО)	(МЕМО)	(МЕМО)	(МЕМО)	(МЕМО)	
1	1131	11314	(МЕМО)	(МЕМО)	(МЕМО)	(МЕМО)	(МЕМО)	

Рисунок 3.6 – Формування тесту

Тестування

Регістрація

N тесту: 1 Дата: []

Студент: 115

Вибір своє прізвище зі списку

KodStudenta	Familiya
115	Іванов
116	Сидоров
117	Мала
118	Сулема
119	Гаврилов

Почати тестування

Тестові завдання

Що таке база даних?

Будь-який текстовий файл

Організована структура для зберігання інформації

Будь-яка інформація, представлена в табличній формі

Будь-яка електронна таблиця

Вибір варіант

Варіант 1

Варіант 2

Варіант 3

Варіант 4

Прийняти відповідь

Кількість вірних відповідей

Вихід

Із 9 завдань ви дали вірну відповідь на 8

Рисунок 3.7 – Проходження тесту студентом

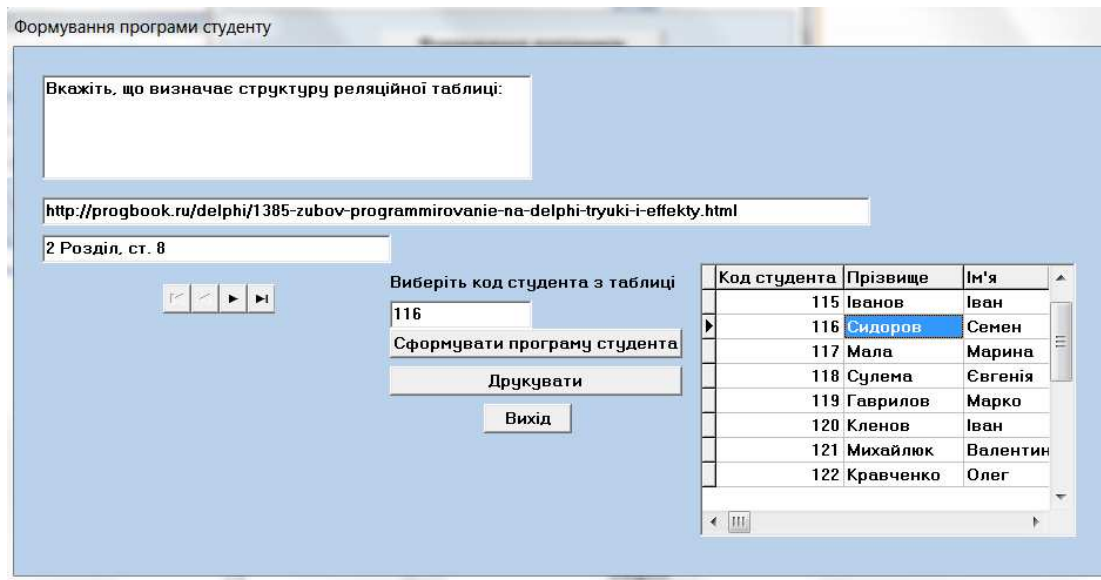


Рисунок 3.8 – Формування програми студента

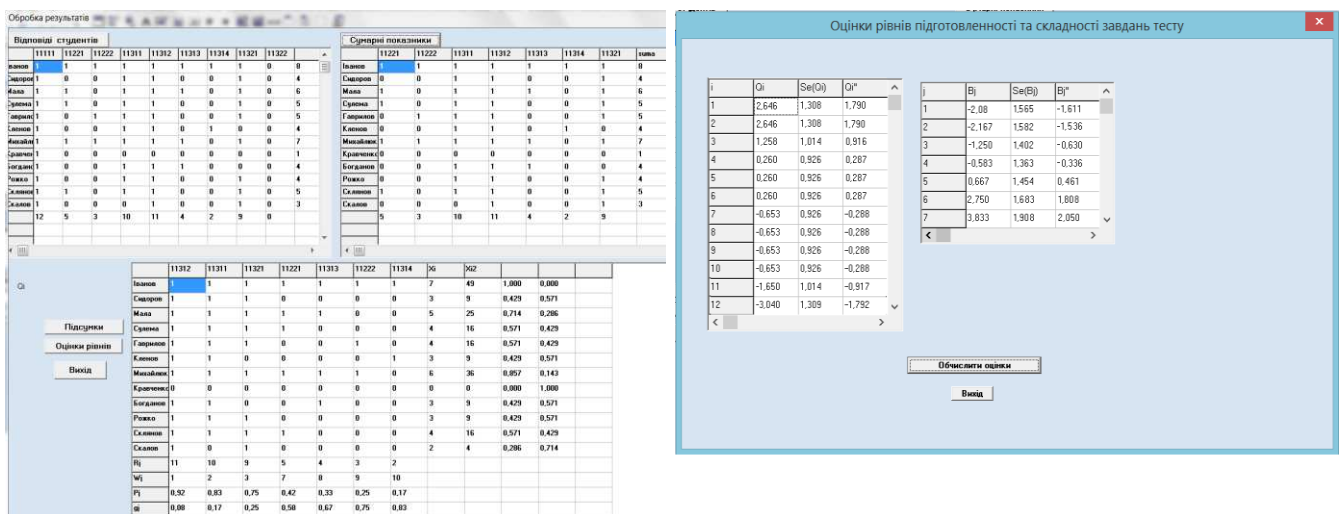


Рисунок 3.9 – Обробка результатів тестування

4. РОЗРАХУНОК ЕКОНОМІЧНОЇ ЕФЕКТИВНОСТІ ВПРОВАДЖЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.

4.1 Вступ.

Сучасні комп'ютери, які приймають, запам'ятовують і передають інформацію, здійснюють контроль, являють собою важливий технічний засіб навчання. Сучасні засоби програмування дозволяють вирішувати задачі, що стояли в нещодавно перед цілими колективами наукових співробітників.

Розвиток нових технологій і ріст застосування обчислювальної техніки привели до широкого застосування ЕОМ для вирішення проблем, пов'язаних з обробкою інформації, її контролем і аналізом. Сьогодні на основі засобів обчислювальної техніки розробляються і впроваджуються різні автоматизовані інформаційні системи (проектування, технологічної підготовки виробництва, адаптивного навчання).

Використання даного програмного продукту приведе до значної економії, що виражається в наступних показниках:

- підвищення рівня знань;
- аналіз отриманої інформації;
- економія часу на обробку інформації.

Найбільш важливим моментом для розробника, з економічної точки зору, є процес формування вартості програмного продукту. Очевидно, що вона являє собою дуже специфічний товар з безліччю особливостей.

Створення системи адаптивного навчання вимагає одноразових витрат на її розробку, придбання необхідних технічних засобів і поточних витрат на функціонування системи. Відношення цієї економії до витрат на створення системи характеризує економічну ефективність капітальних вкладень. Економічні показники визначаються по діючим на момент розрахунку оптовим цінам, тарифам і ставкам заробітної плати.

4.2 Розрахунок витрат на створення та експлуатацію інформаційної системи.

Витрати на розробку системи складаються з витрат на зарплату розробника, на амортизацію ЕОМ, на якій виконується розробка, на експлуатацію цієї ЕОМ, на засоби розробки та витрат на матеріали і комплектуючі.

Припустимо розробка програмного забезпечення виконується програмістом, місячний оклад якого складає 15000 грн. Додаткова заробітна плата складає 20 % від основної. Виходячи з цього, основна і додаткова заробітна плата розроблювача системи 18000 грн./міс., а вартість сучасної ПК складає 24000 грн. (приведена вартість машини на базі Intel Core i5: Intel Core i5-7400 (3.0-3.5 ГГц)/RAM 16 ГБ/HDD 1 ТБ/nVidia GeForce GTX1050 Ti, 4 ГБ/без ОД/LAN/без ОС. При вартості кіловат-години електроенергії рівної 4,32 грн. (для населення, яке споживає понад 100 кВт·год до 600 кВт·год електроенергії на місяць), розраховується вартість розробки програми.

Витрати на допоміжні матеріали приведені в табл. 4.1.

Таблиця 4.1 – Витрати на допоміжні матеріали

Пункти витрат	Сума грн.
Папір	100,00
Заправлення картриджа до принтера	120,00
Канцтовари	180,00
Література	2000,00
Непередбачені витрати	500,00
Разом:	2800,00

Вартість програми розраховуємо по формулі:

$$C_{np} = (Z_{zn} + Z_{cz} + Z_{ze} + Z_e) \cdot T + Z_m, \quad (4.1)$$

де T – тривалість розробки, міс.

Z_{zn} – основна і додаткова заробітна плата обслуговуючого персоналу, грн.

Z_{cz} – відрахування на соціальні заходи (38 % від основної і додаткової заробітної плати), грн.

Z_{ze} – загальногосподарські витрати (10 % від основної заробітної плати), грн.

Z_e – витрати на електроенергію.

Z_m – витрати на основні і допоміжні матеріали.

Z_e при споживанні потужності 0.5 кВт, тривалості роботи на місяць, рівної $22 \cdot 6 = 132$ години, вартості кіловат-години електроенергії 4,32 грн обчислимо:

$$Z_e = 132 \cdot 4,32 \cdot 0,5 = 285,12 \text{ грн.}$$

Таблиця 4.2. – Витрати на розробку системи.

Найменування витрат	Одиниця виміру	Кількість
Тривалість розробки	Місяць	1
Основна і додаткова заробітна плата	Гривня	18000
Відрахування на соціальні заходи	Гривня	6840
Загальногосподарські витрати	Гривня	1500
Витрати на допоміжні матеріали	Гривня	2800
Витрати на електроенергію	Гривня	110,88

Відповідно до формули 4.1 вартість програми:

$$C_{np} = (18000 + 6840 + 1500 + 285,12) \cdot 1 + 2800 = 29250,88 \text{ грн.}$$

Амортизаційні відрахування на устаткування складають 60 % балансової вартості в рік:

$$A_{об} = 24000 \cdot 0,6 = 14400 \text{ грн.}$$

У масштабах підприємства річні витрати на основні і допоміжні матеріали (носії інформації, папір) визначаються в розмірі 5 % вартості основного устаткування:

$$B_m = 24000 \cdot 0,06 = 1200 \text{ грн.}$$

Річний обсяг робіт ПК у годинах визначається в такий спосіб:

$$\Phi_m = 264,5 \cdot T_z, \quad (4.2)$$

де T_z – середнє місячне завантаження устаткування (близько 4 годин);

264,5 – середня кількість робочих днів у році.

Отже, річний обсяг роботи ПЕОМ складе:

$$\Phi_m = 264,5 \cdot 4 = 1058 \text{ годин.}$$

Витрати на електроенергію Z_e при 1058 годинах роботи устаткування в рік складуть

$$Z_e = 1058 \cdot 1,68 \cdot 0,5 = 888,72 \text{ грн.}$$

Експлуатаційні витрати для ПК за рік складуть:

$$Z_{zp} = A_{об} + B_{м} + \Phi_{м} = 14400 + 1200 + 888,72 = 16488,72 \text{ грн.}$$

Отже, у перший рік витрати на створення й експлуатацію програми складуть:

$$Z_{ce} = C_{np} + Z_{zp} = 29250,88 + 16488,72 = 45739,60 \text{ грн.}$$

4.3 Економічна ефективність розробки і впровадження системи.

Основним показником економічної ефективності функціонування системи є підвищення ефективності керування інформацією у вигляді зниження витрат на керування при одночасному збільшенні швидкості і якості одержання потрібного результату.

Крім багатьох інших негативних ефектів, ручна обробка інформації спричиняє наступні негативні економічні ефекти:

- високі витрати на складування паперових документів (сейфи, шафи, папки й ін.);
- підвищені витрати на канцтовари;
- витрати, пов'язані з роботою виявлення раніше допущених помилок (людський фактор).

До числа основних факторів, що визначають приріст прибутку в зв'язку з упровадженням програми, відносяться:

- підвищення продуктивності праці;
- вивільнення робочого часу.

Крім того, не піддається прямій грошовій оцінці підвищення оперативності керування, якість одержуваних результатів, поліпшення організації праці і т.д.

Обов'язковою умовою визначення економічної ефективності програми є порівнянність усіх показників у часі, за цінами й іншими нормами, використовуваними для визначення показників, за змістом і колом елементів витрат.

Визначимо пряму економічну ефективність, ґрунтуючись на тому, що

впровадження програмного продукту вивільняє 0,7 працівника (за експертною оцінкою фахівців підприємства).

Зарплата 0,7 працівника в рік складає:

$$15000 \cdot 12 \cdot 0,7 = 126000 \text{ грн.}$$

Річний економічний ефект розраховується по формулі:

$$\mathcal{E}_{\text{год}} = \Delta C_n - E_n \cdot k,$$

де ΔC_n – вивільнені кошти після впровадження програмного продукту мінус експлуатаційні витрати: $\Delta C_n = 126000 - 16488,72 = 109511,3$ грн.;

E_n – коефіцієнт ефективності (дорівнює коефіцієнту амортизації – 0,6);

k – одноразові витрати на впровадження продукту (45739,60).

$$\mathcal{E}_{\text{год}} = 109511,3 - 0,6 \cdot 45739,60 = 82067,52 \text{ грн.}$$

Строк окупності системи розраховується по формулі:

$$T = \frac{k}{\Delta C} = \frac{45739,60}{109511,3} \approx 0,42 \text{ років.}$$

Отже строк окупності системи складає приблизно 5 місяців.

5. ОХОРОНА ПРАЦІ.

5.1 Вступ.

Охорона праці – це система правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних і лікувально-профілактичних заходів та засобів, спрямованих на збереження життя, здоров'я і працездатності людини у процесі трудової діяльності.

Ця інструкція призначена для запобігання несприятливої дії на людину шкідливих факторів, які супроводжують роботи із засобами обчислювальної техніки та периферійним обладнанням.

Ця інструкція підлягає обов'язковому і безумовному виконанню. За порушення інструкції винні несуть відповідальність в адміністративному та судовому порядку залежно від характеру наслідків порушення. Дотримання правил безпечної роботи є необхідною умовою попередження виробничого травматизму. Ця інструкція поширюється на персонал, експлуатаційні засоби обчислювальної техніки і периферійне устаткування. Інструкція містить загальні вказівки щодо безпечного застосування електроустаткування в установі. Вимоги цієї інструкції є обов'язковими, відступу від неї не допускаються.

5.2 Аналіз шкідливих і небезпечних факторів в лабораторіях обчислювальної техніки.

5.2.1 Електробезпека при роботі.

Заходи щодо усунення небезпеки ураження електричним струмом зводяться до правильного розміщення устаткування та електричних кабелів. Інші заходи щодо забезпечення електробезпеки, збігаються з загальними заходами пожежо- та електробезпеки.

В якості профілактичних заходів для забезпечення пожежної безпеки слід використовувати скриту електромережу, надійні розетки з пожежобезпечних матеріалів, силові мережі живлення устаткування виконувати кабелями, розрахованими на підключення в 3-5 разів більшого навантаження, включати й виключати живлення обладнання за допомогою штатних вимикачів. Треба

регулярно робити очистку внутрішніх частин комп'ютерів, іншого устаткування від пилу, розташовувати комп'ютери на окремих неспалюваних столах. Для запобігання іскріння необхідно рідше встромляти і виймати штепсельні вилки з розеток.

5.2.2 Освітлення.

Система освітлення повинна відповідати таким вимогам:

- освітленість на робочому місці повинна відповідати характеру зорової роботи, який визначається трьома параметрами: об'єктом розрізнення – найменшим розміром об'єкта, що розглядається на моніторі ПК; фоном, який характеризується коефіцієнтом відбиття; контрастом об'єкта і фону;
- необхідно забезпечити достатньо рівномірне розподілення яскравості на робочій поверхні монітора, а також в межах навколишнього простору;
- на робочій поверхні повинні бути відсутні різкі тіні;
- в полі зору не повинно бути відблисків (підвищеної яскравості поверхонь, які світяться та викликають осліплення);
- величина освітленості повинна бути постійною під час роботи;
- слід обирати оптимальну спрямованість світлового потоку і необхідний склад світла.

5.2.3 Вимоги до монітору.

Основним обладнанням робочого місця користувача комп'ютера є монітор, системний блок та клавіатура.

Робочі місця мають бути розташовані на відстані не менше 1,5 м від стіни з вікнами, від інших стін на відстані 1 м, між собою на відстані не менше 1,5 м. Відносно вікон робоче місце доцільно розташовувати таким чином, щоб природне світло падало на нього збоку, переважно зліва.

Робочі місця слід розташовувати так, щоб уникнути попадання в очі прямого світла. Джерела освітлення рекомендується розташовувати з обох боків екрану паралельно напрямку погляду. Для уникнення світлових відблисків екрану, клавіатури в напрямку очей користувача, від світильників загального

освітлення або сонячних променів, необхідно використовувати антиполюскові сітки, спеціальні фільтри для екранів, захисні козирки, на вікнах – жалюзі.

Екран дисплея повинен бути розташованим перпендикулярно до напрямку погляду. Якщо він розташований під кутом, то стає причиною сутулості. Відстань від дисплея до очей повинна трохи перевищувати звичну відстань між книгою та очима. Перед екраном монітора, особливо старих типів, повинен бути спеціальний захисний екран. При його відсутності треба сидіти на відстані витягнутої руки від монітора.

Фільтри з металевої або нейлонової сітки використовувати не рекомендується, тому що сітка спотворює зображення через інтерференцію світла. Найкращу якість зображення забезпечують скляні поляризаційні фільтри. Вони усувають практично всі відблиски, роблять зображення чітким і контрастним.

Ще одним моментом, який стосується зору, є необхідність створення неоднорідного поля зору. Для цього можна розвісити на поверхнях (стінах) плакати та картини, виконані у спокійних тонах. Наприклад, пейзажі.

При роботі з текстовою інформацією (в режимі введення даних та редагування тексту, читання з екрану) найбільш фізіологічним правильним є зображення чорних знаків на світлому (чорному) фоні.

Монітор повинен бути розташований на робочому місці так, щоб поверхня екрана знаходилася в центрі поля зору на відстані 400-700 мм від очей користувача. Рекомендується розміщувати елементи робочого місця так, щоб витримувалася однакова відстань очей від екрана, клавіатури, тексту.

5.2.4 Робоча поза.

Зручна робоча поза при роботі з комп'ютером забезпечується регулюванням висоти робочого столу, крісла та підставки для ніг. Рациональною робочою позою може вважатися таке положення, при якому ступні працівника розташовані горизонтально на підлозі або підставці для ніг, стегна зорієнтовані у горизонтальній площині, верхні частини рук - вертикальні. Кут ліктьового суглоба коливається в межах 70-90°, зап'ястя зігнуті під кутом не більше ніж

20°, нахил голови 15-20°.

Важливою є форма спинки крісла, яка повинна повторювати форму спини. Висота крісла повинна бути такою, щоб користувач не почував тиску на куприк або стегна. Крісло бажано обладнати бильцями. Його потрібно встановити так, щоб не треба було тягтися до клавіатури. Періодично користувачу необхідно рухатися, вчасно змінювати положення тіла і робити перерви у роботі.

При напруженій роботі за комп'ютером щогодини необхідно робити перерву на 15 хвилин через кожну годину і треба займатися іншою справою. Декілька разів на годину бажано виконувати серію легких вправ для розслаблення.

Для нейтралізації зарядів статичної електрики в приміщенні, де виконується робота на комп'ютерах, в тому числі на лазерних та світлодіодних принтерах, рекомендується збільшувати вологість повітря за допомогою кімнатних зволожувачів. Не рекомендується носити одяг з синтетичних матеріалів.

5.3 Розрахунок захисного заземлення приміщення лабораторії.

Визначити фактичний опір струму. Вхідними даними розрахунку є допустимий опір заземленого пристрою, тип ґрунту, довжина труби заземлення і її діаметр.

Задано:

Допустимий опір: $R_d = 4 \text{ Ом}$;

Тип ґрунту: садовий ґрунт;

Довжина труби заземлювача: $l_d = 4 \text{ м}$;

Діаметр труби заземлювача $d = 0,045 \text{ м}$;

Глибина закладення труби від поверхні: $t_0 = 0,5 \text{ м}$.

Розрахунок:

1. Питомий опору ґрунту за даними таблиці $\rho = 50 \text{ Ом/м}$.

2. Для обраного типу заземлювача і його розмірів визначаємо опір розтікання одного заземлювача. Для трубного заземлювача, розташованого в

грунті, опір розтікання:

$$R_1 = \frac{\rho}{2\pi \cdot l} \left(\ln \frac{2l}{d} + \frac{1}{2} \ln \frac{4t+l}{4t-l} \right)$$

$$\text{де } t = \frac{l}{2} + t_0 = \frac{4}{2} + 0,5 = 2,5.$$

Тоді:

$$R_1 = \frac{50}{2 \cdot 3,14 \cdot 4} \left(\ln \frac{2 \cdot 4}{0,045} + \frac{1}{2} \ln \frac{4 \cdot 2,5 + 4}{4 \cdot 2,5 - 4} \right) = 11,149 \text{ Ом.}$$

3. Якщо опір одного заземлювача не перевищує допустимий опір пристрою $R_1 \leq R_d$, то приймаємо 1 штучний заземлювач. Відповідно, якщо $R_1 > R_d$, то необхідно приймати кілька штучних заземлювачів, з'єднаних паралельно. В нашому випадку $R_1 > R_d$.

4. Визначаємо необхідне число паралельно з'єднаних заземлювачів:

$$n = \frac{R_1}{\tau_0 \cdot R_d},$$

де τ_0 – коефіцієнт використання заземлювача ($\tau_0 = 1$).

$$n = \frac{11,149}{1 \cdot 4} = 2,787 \approx 3.$$

Таким чином, необхідно 3 заземлювача.

5. Для зв'язку вершин заземлювачів приймають сполучну межу. Опір розтікання сполучної риси визначають за формулою:

$$l_c = 1,05 \cdot a \cdot n \cdot l = 1,05 \cdot 1 \cdot 3 \cdot 4 = 12,6 \text{ м.}$$

$$R_{lc} = \frac{\rho}{2\pi \cdot l_c} \ln \frac{l^2}{d \cdot t} = \frac{50}{2 \cdot 3,14 \cdot 12,6} \ln \frac{4^2}{0,045 \cdot 2,5} = 4,29 \text{ Ом.}$$

$$R_{\text{цєрми}} = \frac{R_{lc}}{\tau_c} = 4,06 \text{ Ом.}$$

6. Визначаємо фактичний опір заземлювального пристрою:

$$R_3 = \frac{R_\phi \cdot R_{\text{цєрми}}}{R_\phi + R_{\text{цєрми}}} = \frac{4,64 \cdot 4,06}{4,64 + 4,06} = 2,16 \text{ Ом.}$$

$$R_3 \leq R_\delta.$$

5.4 Техніка безпеки на робочому місці.

Загальні положення при роботі з ПК.

При виконанні робіт на комп'ютерах необхідно дотримуватись вимог загальної та даної інструкції з охорони праці.

До самостійної роботи на комп'ютерах допускаються особи, які пройшли медичний огляд, навчання по професії, вступний інструктаж з охорони праці та первинний інструктаж з охорони праці на робочому місці. В подальшому вони проходять повторні інструктажі з охорони праці на робочому місці один раз на півріччя, періодичні медичні огляди один раз на два роки.

Згідно статті 18 Закону України "Про охорону праці" працівник зобов'язаний:

а) знати і виконувати вимоги нормативних актів про охорону праці, правила поведінки з устаткуванням та іншими засобами виробництва, користуватися засобами колективного та індивідуального захисту;

б) дотримуватись зобов'язань щодо охорони праці, передбачених колективним договором та правилами внутрішнього трудового розпорядку підприємства;

в) співробітничати з власником у справі організації безпечних і нешкідливих умов праці, особисто вживати посильних заходів щодо усунення будь-якої виробничої ситуації, яка створює загрозу його життю чи здоров'ю, або людей, які його оточують, повідомляти про небезпеку свого безпосереднього керівника або іншу посадову особу.

Вимоги безпеки перед початком роботи:

- увімкнути систему кондиціонування в приміщенні;
- перевірити надійність встановлення апаратури на робочому столі. Повернути монітор так, щоб було зручно дивитися на екран – під прямим кутом (а не збоку) і трохи зверху вниз, при цьому екран має бути трохи нахиленим, нижній його край ближче до оператора;
- перевірити загальний стан апаратури, перевірити справність електропроводки, з'єднувальних шнурів, штепсельних вилок, розеток, заземлення захисного екрана;

- відрегулювати освітленість робочого місця;
- відрегулювати та зафіксувати висоту крісла, зручний для користувача нахил його спинки;
- приєднати до системного блоку необхідну апаратуру. Усі кабелі, що з'єднують системний блок з іншими пристроями, слід вставляти та виймати при вимкненому комп'ютері;
- ввімкнути апаратуру комп'ютера вимикачами на корпусах в послідовності: монітор, системний блок, принтер (якщо передбачається друкування);
- відрегулювати яскравість свічення монітора, мінімальний розмір світної точки, фокусування, контрастність. Не слід робити зображення надто яскравим, щоб не втомлювати очей.

Рекомендується:

- яскравість свічення екрана - не менше 100 kg/m^2 ;
- відношення яскравості монітора до яскравості оточуючих його поверхонь в робочій зоні – не більше 3:1;
- мінімальний розмір точки свічення не більше 0, мм для монохромного монітора і не менше 0,6 мм для кольорового, контрастність зображення знаку – не менше 0,8.

При виявленні будь-яких несправностей роботу не розпочинати, повідомити про це керівника.

Вимоги безпеки під час виконання роботи:

- необхідно стійко розташовувати клавіатуру на робочому столі, не опускати її хитання. Під час роботи на клавіатурі сидіти прямо, не напружуватися;
 - для забезпечення несприятливого впливу на користувача пристроїв типу "миша" належить забезпечувати вільну велику поверхню столу для переміщення "миші" і зручного упору ліктьового суглоба;
 - не дозволяються сторонні розмови, подразнюючі шуми;
- періодично при вимкненому комп'ютері прибирати ледь змоченою мильним

розчином бавовняною ганчіркою порох з поверхонь апаратури. Екран ВДТ та захисний екран протирають ганчіркою, змоченою у спирті. Не дозволяється використовувати рідинні або аерозольні засоби чищення поверхонь комп'ютера.

6. ОХОРОНА НАВКОЛИШНЬОГО СЕРЕДОВИЩА.

6.1 Вступ.

За останні десятиліття кількість шкідливих, відходів підприємств настільки зросло, що природа поступово втрачає здатність до самоочищення. Стало очевидним, що будь-яка нова техніка і технологія, навіть високопродуктивна й економічно вигідна, але не задовольняючим сучасним вимогам охорони праці і навколишнього середовища, не є прогресивною і не відповідає основному напрямку технічного прогресу. Тому охорона навколишнього середовища в даний час здобуває першорядне значення у всіх областях діяльності, людини.

Сучасна політика держав в області охорони навколишнього середовища від забруднення ґрунтується на принципі "забруднив – плати". Уперше на міжнародному рівні даний принцип був висунутий Організацією економічного співробітництва і розвитку в 1972 році. З цього часу цей принцип став активно використовуватися в законодавчій практиці європейських і інших країн світу.

Для постійного контролю за станом навколишнього середовища застосовують різні методи, одним з них є екологічний моніторинг навколишнього середовища.

6.2 Моніторинг як система спостереження і контролю навколишнього середовища.

Моніторингом навколишнього середовища називаються регулярні, виконувані по заданій програмі спостереження природних середовищ, природних ресурсів, рослинного і тварини світу, що дозволяють виділити їхній стан і процеси що відбуваються в них під впливом антропогенної діяльності.

Під екологічним моніторингом варто розуміти організований моніторинг навколишнього середовища, при якому, по-перше, забезпечується постійна оцінка екологічних умов середовища життя людини і біологічних об'єктів (рослин, тварин, мікроорганізмів і т.д.), а також оцінка стану і функціональної цінності екосистеми, по-друге, створюються умови для визначення корегувальних дій у тих випадках, коли цільові показники екологічних умов не

досягаються.

У систему моніторингу повинні входити наступні основні процедури:

- виділення (визначення) об'єкта спостереження;
- обстеження виділеного об'єкта спостереження;
- складання інформаційної моделі для об'єкта спостереження;
- планування спостережень;
- оцінка стану об'єкта спостереження й ідентифікація його інформаційної моделі;
- прогнозування зміни стану об'єкта спостереження;
- представлення інформації в зручній для використання формі і доведення її до споживача.

- Основні задачі екологічного моніторингу:
- спостереження за джерелом антропогенного впливу;
- спостереження за фактором антропогенного впливу;
- спостереження за станом природного середовища під впливом факторів антропогенного впливу й оцінка прогнозованого стану природного середовища.

При розробці проекту екологічного моніторингу необхідна наступна інформація:

- джерело надходження забруднюючих речовин у навколишнє середовище – викиди забруднюючих речовин в атмосферу промисловими, енергетичними, транспортними й іншими об'єктами; скидання стічних вод у водяні об'єкти; поверхневі змиви забруднюючих і біогенних у поверхневі води суші і моря; внесення на земну поверхню і (чи) у ґрунтовий шар забруднюючих і біогенних речовин разом з добривами і ядохімікатами при сільськогосподарській діяльності; місця поховання і складування промислових і комунальних відходів; техногенні аварії, що приводять до викиду в атмосферу небезпечних речовин чи розливу рідких забруднюючих і небезпечних речовин, і т.д.;
- переноси забруднюючих речовин – процеси атмосферного переносу; процеси переносу і міграції у водяному середовищі;
- процеси ландшафтно-геохімічного перерозподілу забруднюючих

речовин – міграція забруднюючих речовин по ґрунтовому профілю до рівня ґрунтових вод; міграція забруднюючих речовин по ландшафтно-геохімічному сполученню з урахуванням геохімічних бар'єрів і біохімічних круговороті. Спостереження за цими процесами доцільно проводити періодично на спеціально виділеній системі пунктів: контрольні водозбори – катени – площадки – створи;

- дані про стан антропогенних джерел емісії – потужність джерела емісії і місце розташування його, гідродинамічні умови надходження емісії в навколишнє середовище.

6.3 Методи контролю забруднюючих речовин в об'єктах навколишнього середовища.

Сучасні методи контролю хімічних речовин, що забруднюють навколишнє середовище – це по суті фізико-хімічні методи. Іноді їх поєднують терміном "інструментальні методи аналізу". Дана тема величезна, тому ми розглянемо лише найбільш важливі з фізико-хімічних методів, що оптимально сполучають у собі цілий ряд якостей: високу точність і відтворюваність результатів аналізу, високу чутливість і, незважаючи на ці тверді вимоги, приступність апаратури і можливість швидкого її освоєння.

Газова хроматографія. В основу методу газової хроматографії покладений наступний принцип: аналіз суміші речовин у результаті розподілу компонентів між фазами, що не змиваються, одна з яких рухлива – інертний газ (азот, гелій і ін.), інша – нерухома (висококип'яча рідина чи тверда фаза).

Цей метод має два варіанти: газоабсорбційна і газорідинна хроматографія.

Поділ компонентів суміші відбувається в хроматографічному стовпчику. Хроматографічні стовпчики: набивні (довжина – 1-3 м, діаметр – 4 мм, матеріал – скло, сталь і ін.) і капілярні (довжина – до 50 м, матеріал – скло, кварц).

Вибір нерухомої фази (Нф). Ефективність стовпчика (здатність розділяти складні суміші на окремі компоненти) залежить від розміру часток, на які нанесена рідка фаза. Вона зростає при використанні однорідних часток малого розміру. Для стандартних набивних стовпчиків оптимальний розмір часток 0,12-

0,17 мм. Необхідно враховувати їхню близькість до аналізованих з'єднань. Для аналізу полярних компонентів застосовують полярні фази, для аналізу неполярних компонентів – менш полярні чи цілком неполярні.

Неполярні фази для газоадсорбційної хроматографії силікагель, оксид алюмінію, цеоліти, полімерні сорбенти (наприклад, полисорб, поропак і ін.).

Рухливі фази: азот, гелій, аргон, пари води.

Детектори. Історія розвитку газової хроматографії – це історія появи і розвитку детекторів для хроматографії. Застосовуються кілька типів детекторів.

1. Детектор теплопровідності (ДТП) чи катетометр. Принцип його дії заснований на розходженні теплопровідностей аналізованої речовини і газу-носія.

2. У детекторі іонізаційно плазмовому (ПІД чи ДПІ) використовується залежність електропровідності простору між електродами від числа іонізованих часток, що знаходяться в ньому, що утворюються у водневому полум'ї під дією термічних і окисних процесів при влученні в нього молекул аналізованої речовини.

3. Електронно-загарбний детектор (ЕЗД), чи детектор по захопленню електронів, як і ДІП, заснований на залежності електропровідності проміжку між електродами і числімо іонів, що знаходяться в цьому проміжку, що зв'язано з числом молекул, що надходять у детектор. Однак механізм і спосіб утворення іонів принципово відрізняються від такого у випадку ДІП – іони утворюються в результаті взаємодії молекул аналізованої речовини і потоку електронів у камері детектора в результаті бета-розпаду радіоактивної речовини.

4. Детектор термоіонний (ДТІ) за принципом дії аналогічний ДІП. Однак додатково у водневе полум'я безупинно надходить потік іонів лужних металів (калій, натрій, цезій). У їхній присутності різко зростає ефективність іонізації з'єднань, що містять азот, фосфор, хлор і ін. ДТІ застосовують для визначення ФОС і азотовмісних з'єднань.

5. Полум'яно-фотометричний детектор (ПФД) селективний і має підвищену чутливість стосовно з'єднань, що містять сірку.

Високоєфективна рідинна хроматографія (ВЕРХ) – хроматографічний

метод, що дозволяє розділити висококип'ячі рідини і (чи) тверді речовини, що важко або недоцільно визначати методом газорідинної хроматографії, наприклад полі циклічні ароматичні вуглеводні, амінокислоти, ПАВ, пестициди, лікарські препарати, вуглеводи й ін.

Тонкошарова хроматографія (ТШХ). Поділ відбувається на спеціальних пластинках для тонкошарової хроматографії. Нерухома фаза в ТШХ: силікагель, оксид алюмінію, іонообмінні смоли з добавками крохмалю і гіпсу.

Аналізовану суміш наносять на стартову лінію мікрошприцом чи мікропіпеткою. Пластинку чи папір з нанесеною пробєю поміщають у закриту камеру, що містить розчинник, який переміщається по шару сорбенту (чи по паперові) під дією капілярних сил. Компоненти суміші переміщуються разом з розчинником з різними швидкостями. По закінченні поділу пластинку чи папір виймають з камери, випаровують розчинник, обробляючи струменем теплого повітря. Обумовлені речовини з'являються на хроматограмі у виді плям у результаті обробки спеціальним реактивом (наприклад, нігдрином при аналізі амінокислот) чи методом флюоресценції. Зміст аналізованого компонента пропорційно площі плям. Кількісну оцінку чи проводять безпосередньо на пластинці за допомогою планіметра, чи шляхом зняття пофарбованої плями з хроматограми екстракції речовини розчинником і визначення його зміст фотометричним чи методом за допомогою денситометра.

При аналізі катіонів колонку для поділу заповнюють сульфованими катіонами низкою ємності, а гнітючу колонку-аніонітом високої ємності. У якості елюентів використовують розчини HCl HNO_3 , гідро хлориду піридину й ін.

В останні роки розвивається іонна хроматографія без придушення фонового сигналу елемента і з різними способами детектування: фотометричний, атомно абсорбційний, іонометричний.

Достоїнства методу: низька межа визначення – 1-10 мг/мол, селективність, можливість одночасного визначення неорганічних і органічних іонів експресність, широкий діапазон обумовлених концентрацій.

Застосовують вітчизняний хроматограф "Колір-3006", кондуктометричний

детектор, мікропроцесор. Межа виявлення по хлориду натрію – 3,10 мг/мол.

Хроматомасс-спектрометрія (ХМС) – це в сутності газова хроматографія з мас-спектрометром як детектор (наприклад, МІ-1201). Даний метод дозволяє розшифровувати склад складних сумішей, що містять сотні невідомих компонентів, і визначати їх по одній пробі.

Полярографія. Це один з електрохімічних методів аналізу. Полярограма – залежність сили струму від величини прикладеної напруги на електроди. При цьому методі не відбувається фізичного поділу суміші на окремі компоненти. Як катод найчастіше застосовують ртутний електрод, що капає, (РКЕ), поверхня якого безупинна обновляється, що дозволяє одержувати полярограми і проводити аналіз з високою відтворюваністю результатів.

Пряме визначення можливе лише при наявності речовин, здатних відновлюватися на РКЕ: іони металів, органічні сполуки, що містять галоїд-, нітро-, карбонільні з'єднання, пероксиди, дисульфідів, і т.д. Це трохи обмежує можливості методу, однак при визначенні поліграфічних активних з'єднань дозволяє досягти високої селективності визначення без попереднього поділу складних сумішей на окремі компоненти.

Основні типи полярографії – постійна-токова (класична) і перемінно-токова. Остання має різні назви (підрозділи): у залежності від форми амплітуди перемінного струму – квадратно-хвильова.; у залежності від полярності електрода, що використовують як індикаторний, – катодна (відновлення) чи анодна (окислювання). Останню іноді називають вольтамперометрією. В анодній полярографії на відміну від катодної використовують тільки твердий електрод (наприклад, графітовий). Застосовують фоновий чи індиферентний електроліт (називаний просто – тло), тобто розчин кислоти, солі, буферний розчин більш складного складу, у якому розчиняють аналізовану пробу.

ВИСНОВКИ

1. Досліджено використання адаптивного навчання в навчальному процесі для вдосконалення рівня підготовленості.
2. Проаналізовані методи оцінки знань випробуваних та тестових завдань.
3. Запропоновано вдосконалити існуючу теорію параметричного оцінювання тестових завдань і результатів тестування за допомогою сегментації запитань яка дасть змогу створити зворотній зв'язок між студентом та викладачем.
4. Проаналізовані існуючі системи для оцінки знань.
5. На основі наукових досліджень була створена та розроблена адаптивна система навчання яка дозволяє аналізувати отримані результати тестування.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Онопрієнко О. В. Дидактико-методичні засади контролю й оцінювання навчальних досягнень молодших школярів : монографія / Онопрієнко Оксана Володимирівна. — Київ: Педагогічна думка, 2020. — 400 с.
2. Адаптивне навчання студентів професії вчителя: теорія і практика: монографія / В. І. Бондар, І. М. Шапошнікова, Т. Л. Опалюк, Т. Й. Франчук ; за заг. ред. В. І. Бондаря. — Київ : Вид-во НПУ імені М. П. Драгоманова, 2018. — 308 с.
3. Комп'ютерні технології в освіті : навч. посібн. / Ю. С. Жарких, С. В. Лисоченко, Б. Б. Сусь, О. В. Третяк. — К.: Видавничополіграфічний центр "Київський університет", 2012. — 239 с.
4. Дистанційне навчання в системі професійно-технічної освіти : монографія / авт. кол. В. В. Ягупов, Л. М. Петренко, С. Г. Кравець та ін. / За наук. ред. В. В. Ягупова. Житомир: «Полісся», 2019. — 234с.
5. Моніторинг навчальної діяльності: навчальний посібник. / Д.М. Бодненко, О.Б. Жильцов, О.Л. Лещинський, Н.П. Мазур — К : Київський університет імені Бориса Грінченка, 2014. — 276 с. ISBN 978-966-7548-98-8
6. Растригин, Л.А. Адаптивное обучение с моделью обучаемого. [Текст] / Л.А Растригин, М.Х. Эренштейн — Рига: Зинатне, 1988. — 160 с.
7. Адамова І., Багрій К. Тестування як форма контролю та діагностики знань студентів. Витоки педагогічної майстерності. Серія: Педагогічні науки, 2012. С. 43-92.
8. Алексейчук І. С. Про технологію створення системи тестування. Нові технології навчання: Науково-методичний збірник. Київ : НМЦВД, 2000. С. 43-92.
9. Бабкіна О. Проблеми підвищення якості вищої освіти в Україні у контексті Болонських реформ. Освіта і управління. Тернопіль, 2006. № 1. С. 91–94.
10. Безрученков Ю. В. Педагогічний контроль в системі дистанційного навчання. Вісник Національної академії Державної прикордонної служби

України. 2015. Вип. 2.

11. Бербец В. В. Поєднання форм і методів контролю навчальних досягнень учнів в процесі проектно-технологічної діяльності. Київ : Міленіум, 2004. С. 27-34.

12. Берещук М. Я., Бархаєв Ю. П., Стадник Г. В. Тестовий контроль і рейтинг в освіті: Навчальний посібник. Харків : ХНАМГ, 2006. 106 с.

13. Берещук М. Я., Дмитрієв І. Б. Тестовий контроль та рейтингова оцінка знань студентів: Методичні рекомендації до застосування. Харків : ХДАМГ, 2001. 43 с.

14. Біла книга національної освіти / за ред. В. Г. Кременя. Київ, 2009. 376 с.
9. Болюбаш Я. Я., Булах І. Є., Мруга М. Р. Педагогічне оцінювання і тестування. Правила, стандарти, відповідальність. Київ : Майстер-клас, 2007. 272 с.

15. Булах І. Є. Основи педагогічного оцінювання: Навчальний посібник. Частина 1. Київ, 2006. С. 36-91. 11. Булах І. Є. Створюємо якісний тест: навч. посібн. Київ: Майстер-клас, 2006. 160 с.

16. Булах І. Є., Філончук І. В. Проблеми стандартизації і засоби педагогічної діагностики. Післядипломна освіта в Україні. 2004. № 2. С. 90–93.

17. Булгакова Н. Б. Вища освіта і Болонський процес. Педагогіка вищої школи: навч.-метод. посібник. Національний авіаційний ун-т, Спільний європейський проект «Навчання українських інженерів комп'ютерному промисловому проектуванню». Київ : НАУ-друк, 2009. 84 с.

18. Вдовиченко Р. Між професійною підготовкою та компетентністю. Управління освітою (Шкільний світ). 2003. № 15-16. С. 2-3. 15. Використання електронних відкритих систем для інформаційно-аналітичної підтримки педагогічних досліджень: короткий термінологічний словник. Київ : ІТЗН НАПН України, 2017. 67 с.

Технічне завдання на розробку адаптивної системи навчання системи

1. Загальні відомості.

1.1 Назва системи.

Найменування системи: "Адаптивна система навчання".

1.2 Підстави для розробки.

Документ: завдання на магістерську роботу.

Організація: НУК імені адмірала Макарова, кафедра Інформаційних управляючих систем та технологій.

1.3 Порядок оформлення та пред'явлення Замовнику результатів робіт.

Робота по створенню АСН здається поетапно у відповідності з календарним планом Проекту.

2 Призначення та цілі створення системи.

2.1 Призначення системи.

Призначенням підсистеми є формування індивідуальної програми студента на основі його результатів тестування та аналіз цих результатів для викладача.

Метою створення підсистеми є підвищення рівня знань студентів за допомогою системи та аналіз отриманих результатів для корегування викладачем навчальної програми.

2.2 Цілі створення системи.

Призначенням підсистеми є формування індивідуальної програми студента на основі його результатів тестування та аналіз цих результатів для викладача.

Метою створення підсистеми є підвищення рівня знань студентів за допомогою системи та аналіз отриманих результатів для корегування викладачем навчальної програми.

3 Вимоги до інформаційної системи.

3.1 Вимоги до структури системи.

Система складається з декількох підсистем, представлених програмними модулями та інструментами, що відповідають за різні групи функцій:

- 1) Модуль роботи з базою даних
- 2) Модуль створення тестів
- 3) Модуль проходження тестів
- 4) Модуль індивідуальної програми студента
- 5) Модуль аналізу результатів тестування

3.2 Вимоги до функціональних характеристик.

В системі мають виконуватися наступні функції:

- a) Робота с БД.
- b) Формування тестів.
- c) Проходження тестів.
- d) Формування індивідуальної програми студента.
- e) Аналіз результатів тестування.
- f) Авторизований доступ до даних на основі прав доступу.

Результатом роботи АСН має бути конфігурація налаштувань графічних панелей інструментів в тому вигляді, який задає користувач.

3.3 Вимоги до надійності.

Повинна бути забезпечена надійність збереження інформації при аварійних ситуаціях, таких як відключення електроживлення та вихід з ладу компонентів системи.

Надійність системи має бути забезпечена за рахунок:

- 1) застосування технічних засобів, системного та базового програмного забезпечення, які відповідають класу задач, що вирішуються в системі;
- 2) дотримання правил експлуатації та технічного обслуговування програмно-апаратних засобів.

3.4 Вимоги до надійності технічних засобів.

- 1) апаратно-системний комплекс АСН повинен мати засоби для відновлення при збої;
- 2) в ІС має бути організований алгоритм роботи, за якого створюються резервні копії даних.
- 3) система має пройти процес тестування та налагодження, що забезпечить стійкість системи.

3.5 Додаткові вимоги.

ІС має розроблятися і експлуатуватися на наявному у замовника апаратно-технічному комплексі.

4. Стадії та етапи розробки.

Стадії та етапи розробки відповідають календарному плану, затвердженому кафедрою ІУСТ.

5. Порядок контролю та приймання.

Приймання створеної системи здійснюється представниками виконавця і замовника відповідно до технічного завдання. Складається двосторонній акт, що стає підставою для розрахунків.

Код програми

Головний файл проекту

```
program Prest25;
uses
  Forms,
  Unrob in 'Unrob.pas' {Form1},
  UnDov in 'UnDov.pas' {Fdovidniki},
  UnDM in 'UnDM.pas' {DM1: TDataModule},
  Untest in 'Untest.pas' {Ftest},
  Unrezultat in 'Unrezultat.pas' {Frezultat},
  Unsart in 'Unsart.pas' {Fstart},
  Unoc in 'Unoc.pas' {Focin},
  UStud_prog in 'UStud_prog.pas' {FStud_prog};

{$R *.res}

begin
  Application.Initialize;
  Application.CreateForm(TFstart, Fstart);
  Application.CreateForm(TForm1, Form1);
  Application.CreateForm(TFdovidniki, Fdovidniki);
  Application.CreateForm(TDM1, DM1);
  Application.CreateForm(TFtest, Ftest);
  Application.CreateForm(TFrezultat, Frezultat);
  Application.CreateForm(TFocin, Focin);
  Application.CreateForm(TFStud_prog, FStud_prog);
  Application.Run;
end.
```

Програмний код модуля стартової форми

```
unit Unsart;

interface

uses Windows, SysUtils, Classes, Graphics, Forms, Controls,
  StdCtrls,
  Buttons, ExtCtrls, Dialogs, Mask;

type
  TFstart = class(TForm)
    OKButton: TButton;
    Label1: TLabel;
    evx: TMaskEdit;
    Btstart: TButton;
    Label2: TLabel;
    Button1: TButton;
    procedure OKButtonClick(Sender: TObject);
    procedure BtstartClick(Sender: TObject);
    procedure Button1Click(Sender: TObject);
  private
```

```

        { Private declarations }
    public
        { Public declarations }
    end;

var
    Fstart: TFstart;

implementation

uses Unrob, Untest;

{$R *.dfm}

procedure TFstart.OKButtonClick(Sender: TObject);
begin
    close
end;

procedure TFstart.BtstartClick(Sender: TObject);
begin
    if evx.text='111' then
    begin
        Form1.showModal;

    end
    else
        ShowMessage('Пароль не вірний');
    end;

procedure TFstart.Button1Click(Sender: TObject);
begin
    Ftest.Show;
end;

end.

```

Модуль вибору режиму роботи

```

unit Unrob;

interface

uses

    Windows, Messages, SysUtils, Variants, Classes, Graphics,
    Controls, Forms,
    Dialogs, StdCtrls;

type
    TForm1 = class(TForm)
        Button1: TButton;
        Button2: TButton;
        Button3: TButton;

```

```

    BtRezult: TButton;
    procedure Button1Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure Button3Click(Sender: TObject);
    procedure BtRezultClick(Sender: TObject);
private
    { Private declarations }
public
    { Public declarations }
end;

var
    Form1: TForm1;

implementation

uses UnDm, UnDov, Untest, Unrezultat, Unsart, UStud_prog;

{$R *.dfm}

procedure TForm1.Button1Click(Sender: TObject);
begin
    Fdovidniki.show
end;

procedure TForm1.FormCreate(Sender: TObject);
begin
    close
end;

procedure TForm1.Button2Click(Sender: TObject);
begin

close;
end;

procedure TForm1.Button3Click(Sender: TObject);
begin
    FStud_prog.Show;
end;

procedure TForm1.BtRezultClick(Sender: TObject);
begin
    Frezultat.show;
end;

end.

```

Модуль формування довідників

```

unit UnDov;

interface

```

```
uses Windows, SysUtils, Classes, Graphics, Forms, Controls,
StdCtrls,
  Buttons, ExtCtrls, Grids, Dialogs, DBGrids, ComCtrls, Mask,
DBCtrls;
```

```
type
```

```
TFdovidniki = class(TForm)
  OKButton: TButton;
  PageControl1: TPageControl;
  TSwp: TTabSheet;
  TabSheet1: TTabSheet;
  TabSheet2: TTabSheet;
  TabSheet3: TTabSheet;
  TabSheet4: TTabSheet;
  TabSheet5: TTabSheet;
  DBGrid1: TDBGrid;
  EkWP: TEdit;
  EAbr: TEdit;
  DTk: TDateTimePicker;
  dTf: TDateTimePicker;
  CBkf: TComboBox;
  CBfk: TComboBox;
  Label1: TLabel;
  Label2: TLabel;
  Label5: TLabel;
  Label6: TLabel;
  Label7: TLabel;
  Label8: TLabel;
  BWpzm: TButton;
  BPzmV: TButton;
  BPadd: TButton;
  BpDel: TButton;
  DBGrid2: TDBGrid;
  DBGrid3: TDBGrid;
  DBGrid4: TDBGrid;
  DBGrid6: TDBGrid;
  e1: TDBEdit;
  Label17: TLabel;
  Label18: TLabel;
  e2: TDBEdit;
  e3: TDBEdit;
  Label19: TLabel;
  DBGrid5: TDBGrid;
  Label9: TLabel;
  Ewp: TEdit;
  Eab: TEdit;
  Edata: TEdit;
  Ekf: TEdit;
  Efdt: TEdit;
  Efk: TEdit;
  Shapel: TShape;
  DBEdit1: TDBEdit;
  DBEdit2: TDBEdit;
  Label3: TLabel;
```



```

Label4: TLabel;
Label10: TLabel;
DBEdit3: TDBEdit;
Label11: TLabel;
DBEdit4: TDBEdit;
GroupBox1: TGroupBox;
M1: TMemo;
M2: TMemo;
M3: TMemo;
M4: TMemo;
M5: TMemo;
DBGrid7: TDBGrid;
LbEdvP: TLabel;
Label12: TLabel;
Label13: TLabel;
Label14: TLabel;
Button1: TButton;
Label15: TLabel;
Label16: TLabel;
Label20: TLabel;
Label21: TLabel;
Label22: TLabel;
Label23: TLabel;
LbEPnk: TLabel;
Evir: TEdit;
UpDown1: TUpDown;
EdNT: TEdit;
BTNVpr: TButton;
Label24: TLabel;
DBNavigator1: TDBNavigator;
DBNavigator2: TDBNavigator;
DBNavigator3: TDBNavigator;
DBNavigator4: TDBNavigator;
procedure OKButtonClick(Sender: TObject);
procedure BWpzmClick(Sender: TObject);
procedure BPzmVClick(Sender: TObject);
procedure BPaddClick(Sender: TObject);
procedure TSWpEnter(Sender: TObject);
procedure TabSheet1Enter(Sender: TObject);
procedure DBGrid7CellClick(Column: TColumn);
procedure DBGrid5CellClick(Column: TColumn);
procedure Button1Click(Sender: TObject);
procedure BTNVprClick(Sender: TObject);

private
{ Private declarations }
public
{ Public declarations }
end;

var
Fdovidniki: TFdovidniki;

implementation

```

```

{$R *.dfm}
uses Unrob, UnDM;
procedure TFdovidniki.OKButtonClick(Sender: TObject);
begin
close
end;

procedure TFdovidniki.BWpzmClick(Sender: TObject);
//Скидання запису для коригування
begin
With Dm1 do
begin
ekwp.Text:= twp.Fields.Fields[0].asString;
eabr.Text:= twp.Fields.Fields[1].Value;
dtk.Date:=twp.Fields.Fields[2].Value;
dtf.Date:=twp.Fields.Fields[4].Value;

end;
end;

procedure TFdovidniki.BPzmVClick(Sender: TObject);
//Запис змін
begin
dm1.TWP.Edit;
dm1.TWP.SetFields([strtoint(ekwp.Text),eabr.Text,int(dtk.Date),
strtoint(cbkf.Items.Strings[cbkf.ItemIndex]),int(dtf.Date),
strtoint(cbkf.Items.Strings[cbkf.ItemIndex]))];
dm1.TWP.Post;
end;
procedure TFdovidniki.BPaddClick(Sender: TObject);
begin
if (ekwp.Text>'') and (eabr.Text>'') Then
begin
if dm1.TWP.Locate('KodeWp',StrToInt(ekwp.text),[]) =true
then
begin
ShowMessage('Програма з таки кодом існує');
ekwp.SetFocus;
end
else

dm1.TWP.InsertRecord([strtoint(ekwp.Text),eabr.Text,int(dtk.Date),
strtoint(cbkf.Items.Strings[cbkf.ItemIndex]),int(dtf.Date),
strtoint(cbkf.Items.Strings[cbkf.ItemIndex]))];
end;
end;

procedure TFdovidniki.TSWpEnter(Sender: TObject);
begin
e2.Visible:=false;
e3.Visible:=false;
end;

```

```

procedure TFdovidniki.TabSheet1Enter(Sender: TObject);
begin
e1.Visible:=true;
e2.Visible:=true;
end;

procedure TFdovidniki.DBGrid7CellClick(Column: TColumn);
begin
ednt.Text:=dm1.TDovTes.Fields.Fields[0].AsString;
lbepnk.Text:=dm1.TTC.Fields.Fields[0].AsString;
end;

procedure TFdovidniki.DBGrid5CellClick(Column: TColumn);
begin
with dm1 do
begin

//lbedvp.Text:=TTes.Fields.Fields[2].AsString;

end;
end;

procedure TFdovidniki.Button1Click(Sender: TObject);
VAR i,n:integer;
begin
with dm1 do
begin

ttes.InsertRecord([strtoint(ednt.text),strtoint(lbepnk.text),strtoi
nt(lbedvp.text),
m1.text,m2.text,m3.text,m4.text,m5.Text,(evir.text)]);
end;

end;
procedure TFdovidniki.BTNVprClick(Sender: TObject);
begin
with dm1 do
begin
if QNvp.Active then QNvp.Close;
qnvP.Parameters.ParamByName('p').Value:=StrToInt(lbepnk.Text);
QNvp.Open;
QNvp.Last;
lbedvp.Text:=inttostr(dm1.qnvP.Fields.Fields[1].Value +1);
end;
end;

end.

```

Модуль тестування студента

```

unit Untest;

interface

```

```
uses Windows, SysUtils, Classes, Graphics, Forms, Controls,
StdCtrls,
  Buttons, ExtCtrls, DBCtrls, Mask, Grids, DBGrids,
Dialogs, ComCtrls, DB, ADODB;
```

```
type
```

```
  TFtest = class(TForm)
    OKButton: TButton;
    Panel1: TPanel;
    Panel2: TPanel;
    Panel3: TPanel;
    DBMemo1: TDBMemo;
    DBMemo2: TDBMemo;
    DBMemo3: TDBMemo;
    DBMemo4: TDBMemo;
    DBMemo5: TDBMemo;
    rg: TRadioGroup;
    v2: TRadioButton;
    v3: TRadioButton;
    v4: TRadioButton;
    v1: TRadioButton;
    Btvidpovid: TButton;
    DBGrid1: TDBGrid;
    DBGrid2: TDBGrid;
    dtpk: TDateTimePicker;
    Label1: TLabel;
    Et: TEdit;
    Label2: TLabel;
    Label3: TLabel;
    DBGrid3: TDBGrid;
    tst: TADOTable;
    DBGrid4: TDBGrid;
    Button2: TButton;
    Button3: TButton;
    EkSt: TEdit;
    Epriz: TEdit;
    Button4: TButton;
    DBGrid5: TDBGrid;
    Label4: TLabel;
    Label5: TLabel;
    Label6: TLabel;
    Button1: TButton;
    lr: TLabel;
    procedure BtvidpovidClick(Sender: TObject);
    procedure OKButtonClick(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure DBGrid2CellClick(Column: TColumn);
    procedure Button4Click(Sender: TObject);
    procedure Button1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
```

```

var
  Ftest: TFtest;

implementation

{$R *.dfm}
uses UnDm, UnDov, Unrob;
procedure TFtest.BtvidpovidClick(Sender: TObject);
var vv:integer; pp:string;
begin
  with dm1 do

begin
  if ttes.Eof then
    showmessage (' The end')
  else
    Begin
    if v1.Checked then vv:=1
    else
      if v2.Checked then vv:=2
    else
      if v3.Checked then vv:=3
    else
      if v4.Checked then vv:=4
    else vv:=0;

totvet.InsertRecord([1, strtoint(ekst.Text), int(dtpk.date), strtoint(
et.Text),
  ttes.Fields.Fields[1].value, ttes.Fields.Fields[2].value, vv]);
  if vv=ttes.Fields.Fields[8].Value then vv:=1
  else
    vv:=0;

trez.InsertRecord([strtoint(et.Text), tstd.Fields.Fields[0].value,
ttes.Fields.Fields[2].value, vv, ttes.Fields.Fields[1].value]);
  if vv=0 then

tstprg.InsertRecord([strtoint(ekst.Text), int(dtpk.date), ttes.Fields
.Fields[2].value]);
  ttes.Next;
  end;
End;
end;

procedure TFtest.OKButtonClick(Sender: TObject);
var n,m,i,j,k:integer;
begin
{n:=dm1.tstd.recordcount;
m:=dm1.TTes.RecordCount;
  for i:=0 to dm1.tstd.recordcount-1 do
    begin
      t1.cells[0,i]:=dm1.tstd.Fields.Fields[1].value;

```

```

        for j:=0 to dm1.tRez.RecordCount-1 do
        begin
            k:=1;
            if dm1.tstd.Fields.Fields[0].value=
dm1.tRez.Fields.Fields[0].Value
            then
                begin
                    t1.Cells[k,i]:= dm1.tRez.Fields.Fields[3].AsString;
                    k:=k+1;
                    if not(dm1.tRez.Eof) then dm1.trez.Next;
                end;
            end;
            dm1.TStd.Next
        end;
        for i:=1 to m do
            t1.cells[i,0]:=inttostr(i+1);        }

end;
procedure TFtest.Button2Click(Sender: TObject);
begin
close
end;

procedure TFtest.DBGrid2CellClick(Column: TColumn);
begin
with dm1 do
begin
ekst.Text:=tstd.Fields.Fields[0].AsString;
epriz.Text:=tstd.Fields.Fields[1].Value+'
'+tstd.Fields.Fields[2].Value;
end;
end;

procedure TFtest.Button4Click(Sender: TObject);
begin
dm1.TTes.First;
v1.Checked:=false;
v2.Checked:=false;
v3.Checked:=false;
v4.Checked:=false;
end;

procedure TFtest.Button1Click(Sender: TObject);
var i,z, n:integer;
begin

z:=DM1.Ttes.RecordCount;
n:=0;
With DM1 do
begin
trez.First;
for i:=0 to trez.recordcount-1 do
    begin
        if (trez.fields.fields[1].value=strtoint(ekst.Text))

```

```

        and (trez.fields.fields[3].value=1) then n:=n+1;
        trez.next;
    end;
    end ;
    lr.caption:='Із '+inttostr(z)+' завдань ви дали вірну відповідь
на '+inttostr(n);
        end;
end.

```

Модуль обробки результатів тестування

```

unit Unrezultat;

interface

uses Windows, SysUtils, Classes, Graphics, Forms, Controls,
StdCtrls,
    Buttons, ExtCtrls, dialogs, Grids;

type
    TFrezultat = class(TForm)
        OKButton: TButton;
        tr1: TStringGrid;
        Button1: TButton;
        Button2: TButton;
        tr2: TStringGrid;
        Button3: TButton;
        tr3: TStringGrid;
        Button4: TButton;
        Label1: TLabel;
        Label2: TLabel;
        procedure OKButtonClick(Sender: TObject);
        procedure Button1Click(Sender: TObject);
        procedure Button2Click(Sender: TObject);
        procedure Button3Click(Sender: TObject);
        procedure Button4Click(Sender: TObject);
    private
        { Private declarations }
    public
        { Public declarations }
    end;

var
    Frezultat: TFrezultat;

implementation

{$R *.dfm}
    uses UnDm, UnDov, unrob, Unoc;
    procedure TFrezultat.OKButtonClick(Sender: TObject);
    begin
    close;
    end;

```

```

procedure TFrezultat.Button1Click(Sender: TObject);

var n,m,i,j,k:integer;
begin
n:=dm1.tstd.recordcount;
m:=dm1.TTes.RecordCount;
tr1.RowCount:=n+5;
tr1.ColCount:=m+2;
i:=1;
while not dm1.TTes.Eof do
begin
    tr1.cells[i,0]:=dm1.ttes.Fields.Fields[2].asString;
    tr2.cells[i,0]:=dm1.ttes.Fields.Fields[2].asString;
    dm1.TTes.Next;
    i:=i+1;
end;
dm1.TTes.First;

    dm1.TStd.First;
    for i:=1 to n do
        begin
            // prizv stud
            tr1.cells[0,i]:=dm1.tstd.Fields.Fields[1].value;
            //rezult studenta
            if dm1.Qrez.Active then dm1.Qrez.Close;

dm1.qrez.Parameters.ParamByName('p').Value:=dm1.tstd.Fields.Fields[
0].value;
            dm1.Qrez.Open;
            for j:=0 to dm1.Qrez.RecordCount-1 do
                begin

tr1.Cells[j+1,i]:=dm1.Qrez.Fields.Fields[2].AsString;
                    if not(dm1.qrez.Eof) then
                        dm1.qrez.Next;
                    end;
                if not(dm1.tstd.eof) Then
                    dm1.tstd.next;
                end;

end;

end;

procedure TFrezultat.Button2Click(Sender: TObject);
var
i,j,n,m,mx,mn,nmx,nmn,c,r,k:integer;
begin
    mx:=0;
    mn:=TR1.ROWCOUNT;

    with tr1 do
        begin
            n:=colcount-2;
            m:=rowcount-5;
            //SUM PO ROW

```



```

        for J:=1 to m do
        begin
            cells[n+1,j]:='0';
            for i:=1 to n do
                cells[
n+1,j]:=inttostr(strtoint(cells[n+1,j])+strtoint(cells[i,j]));
            end;

            //suma po stovb
            for i:=1 to n do
            begin
                cells[i,m+1]:='0';
                for j:=1 to m do

cells[i,m+1]:=inttostr(strtoint(cells[i,m+1])+strtoint(cells[i,j]))
;
                    if strtoint(cells[i,m+1])< mn then
                    begin
                        mn:=strtoint(cells[i,m+1]); nm:=i;
                    end;
                    if strtoint(cells[i,m+1])> mx then
                    begin
                        mx:=strtoint(cells[i,m+1]); nm:=i;
                    end;
                end ;

                tr2.colcount:=(tr1.colcount);
                tr2.rowcount:=(tr1.rowcount);
                k:=0;
                for i:=0 to tr1.ColCount do
                    if (i<>nm) and (i<>nm) then
                    begin
                        for j:=0 to tr1.RowCount+1 do
                            tr2.Cells[k,j]:=tr1.Cells[i,j];
                        k:=k+1;
                    end;

                end ;
                tr2.Cells[k-2,0]:='suma';
            end;
        procedure TFrezultat.Button3Click(Sender: TObject);
        var mx,i,k,r,c,p,m,j:integer;
        max:integer;
        Qi,Bi:array[1..20,1..4] of real;
        begin
            r:=dm1.TStd.recordcount;
            c:=dm1.TTes.RecordCount-2;
            tr3.RowCount:=r+3;
            tr3.colcount:=c+3;
            max:=1;
            k:=1;

            with tr2 do
            begin
                for i:=1 to r do
                //prizv

```

```

        tr3.cells[0,i]:=cells[0,i];

//rydi
// for p:=1 to r+1 do
//tr3.Cells[c+1,p] :=cells[c+1,p];
For k:=1 to c do
begin
    max:=1;  mx:=1;
    for i:=1 to  c do
        if strtoint(cells[i,r+1])>max then
        begin
            max:=strtoint(cells[i,r+1]);  mx:=i;
            //      showmessage(inttostr(max)+' 1 '+inttostr(i));
        end;
    for p:=1 to r+1 do
    begin
        tr3.Cells[k,p]:=cells[mx,p];
        tr3.Cells[k,0]:=cells[mx,0];
    end;

    Showmessage(  tr3.Cells[k,r+1]+' '+ tr3.Cells[k,0]);

dm1.Tend.InsertRecord([strtoint(tr3.Cells[k,0]),strtoint(tr3.Cells[
k,r+1])]);
    tr2.cells[mx,r+1]:='0';
end;
end;

WITH tr3 do
BEGIN
//suma po rydu
mx:=0;
for i:=1 to (DM1.TStd.RecordCount) do
begin
    mx:=0; //i-ryd
    for j:=1 to  (DM1.TTes.RecordCount-2) do
        mx:=mx+strtoint(cells[j,i]);

    cells[j,i]:=inttostr(mx);
    cells[j+1,i]:=inttostr(sqr(mx));
    mx:=0;
// showmessage( cells[j,i]+ ' '+      cells[j+1,i]);
end;
    cells[j,0]:='Xi'; cells[j+1,0]:='Xi2';
END;

{  for i:=1 to (Dm1.TTes.RecordCount-2) do
    mx:=0;
    begin
        for j:=1 to dm1.TStd.RecordCount do
        begin
            mx:=mx+ strtoint(cells[i,j]);

```

```

        cells[i,j+1]:=inttostr(mx);
    end;
    //      showmessage(  tr3.Cells[i,0]+' v ' + inttostr(mx));
    //  dm1.tend.insertrecord([strtoint(tr3.Cells[i,0]),mx]);

```

```

END;

```

```

end; }

```

```

With tr3 do
begin
    rowcount:=r+5; colcount:=c+6;
    cells[0,r+1]:='Rj'; Cells[0,r+2]:='Wj'; Cells[0,r+3]:='Pj';
    Cells[0,r+4]:='gj'; Cells[0,r+5]:='Pj*qj';
    for i:=1 to c do
    begin
        cells[i,r+2]:=inttostr(R-strtoint(cells[i,r+1]));
        cells[i,r+3]:=floattostrF(strtoint(cells[i,r+1])/R,ffFixed, 5,
2);
        cells[i,r+4]:=floattostrF((1-strtofloat(cells[i,r+3])),ffFixed,
5, 2);

        cells[i,r+5]:=floattostrF((strtofloat(cells[i,r+3])*strtofloat(cells[i,r+4])),ffFixed, 5, 2);
    end;
    FOR P:=1 TO R do
    begin
        cells[c+2,p]:=inttostr(sqr(strtoint(cells[c+1,p])));
        cells[c+3,p]:=
floattostrf((strtoint(cells[c+1,p])/7),fffixed,5,3);
        cells[c+4,p]:= floattostrf((1-
strtoint(cells[c+1,p])/7),fffixed,5,3);
    end;
end;

end;

end;

```

```

procedure TFrezultat.Button4Click(Sender: TObject);
begin
    if dm1.qstpн.active then
        dm1.qstpн.Close;
        dm1.qstpн.Open;

        if dm1.qtrн.active then
            dm1.qtrн.Close;
            dm1.qtrн.Open;
Focin.show

```

end;

end.

Модуль оцінки завдань та рівня підготовленості

unit Unoc;

interface

uses Windows, SysUtils, Classes, Graphics, Forms, Controls,
StdCtrls,
Buttons, ExtCtrls, Grids, dialogs, DBGrids;

type

TFocin = class(TForm)
 OKButton: TButton;
 tst: TStringGrid;
 tvp: TStringGrid;
 DBGrid1: TDBGrid;
 Button1: TButton;
 procedure OKButtonClick(Sender: TObject);
private
 { Private declarations }
public
 { Public declarations }
end;

var

Focin: TFocin;

implementation

{ \$R *.dfm }
uses unDM ;
procedure TFocin.OKButtonClick(Sender: TObject);
var i,j,m,n:integer;
Qsr,qsr2,Bsr,bsr2,Mo,Nb,Xi,Sb,Sq,aqk,bqk:real;
begin
With DM1 do
Begin
Qstpн.first;
 Qsr:=0;
 while not Qstpн.Eof do
 begin
qsr:=qsr+qstpн.Fields.fields[3].Value;
qsr2:=qsr2+qstpн.Fields.fields[4].Value;
qstpн.Next;
end;
qsr:=qsr/12;
 SQ:=(qsr2-12*sqr(qsr))/11;
 //showmessage(floattostr(sq));
 qtrн.First;

```

Bsr:=0;
Bsr2:=0;
qtrn.first;
while not qtrn.Eof do
    begin
        bsr:=bsr+qtrn.Fields.Fields[4].Value;
        bsr2:=bsr2+qtrn.Fields.Fields[5].Value;
        qtrn.Next;
    end;
    bsr:=bsr/7;
    SB:= (bsr2-7*sqr(bsr))/6;
    // showmessage(floattostr(sb));
    aqk:= sqrt( (1+sb/2.89)/(1-sq*sb/8.35));
    bqk:= sqrt( (1+sq/2.89)/(1-sq*sb/8.35));

Qstpн.first;
Qsr:=0; i:=1;
tst.Cells[0,0]:='i';
tst.Cells[2,0]:='Se(Qi)';
tst.Cells[1,0]:='Qi';
tst.Cells[3,0]:='Qi"';
//for j:=0 to Qstpн.RecordCount-1 do
while not Qstpн.Eof do
    begin

tst.Cells[1,i]:=floattostrf((qstpн.Fields.fields[3].Value),ffixed,
5,3);

tst.Cells[3,i]:=floattostrf((qstpн.Fields.fields[3].Value*aqk+bsr),
ffixed,5,3);

tst.Cells[2,i]:=floattostrf((aqk/sqrt(12*qstpн.Fields.fields[1].Val
ue*qstpн.Fields.fields[2].Value)),ffixed,5,3);
tst.Cells[0,i]:=inttostr(i);
i:=i+1;
qstpн.Next;
    end;

Qtrн.First;
j:=1;
tvp.Cells[0,0]:='j';
tvp.Cells[2,0]:='Se(Bj)';
tvp.Cells[1,0]:='Bj';
tvp.Cells[3,0]:='Bj"';
while not Qtrн.Eof do
    begin
        // showmessage(inttostr(Qtrн.RecNo));

tvp.Cells[1,j]:=floattostrf((Qtrн.Fields.fields[3].Value),ffixed,5
,3);

tvp.Cells[3,j]:=floattostrf((Qtrн.Fields.fields[3].Value*bqk+bsr),f
ffixed,5,3);

```

```

tvp.Cells[2,j]:=floattostrf((bqk/sqrt(7*Qtrn.Fields.fields[1].Value
*Qtrn.Fields.fields[2].Value)),ffixed,5,3);
    tvp.Cells[0,j]:=inttostr(j);
    j:=j+1;
    qtrn.Next;
end
End;

end;

end.
Формування програми студента
unit UStud_prog;

interface

uses Windows, SysUtils, Classes, Graphics, Forms, Controls,
StdCtrls,
    Buttons, ExtCtrls, DBCtrls, Mask, Grids, DBGrids;

type
    TFStud_prog = class(TForm)
        OKButton: TButton;
        Button1: TButton;
        Button2: TButton;
        DBEdit2: TDBEdit;
        DBEdit3: TDBEdit;
        DBMemo1: TDBMemo;
        DBNavigator1: TDBNavigator;
        DBGrid1: TDBGrid;
        kod: TEdit;
        Button3: TButton;
        Label1: TLabel;
        Button4: TButton;
        Button5: TButton;
        procedure Button2Click(Sender: TObject);
        procedure FormCreate(Sender: TObject);
        procedure Button3Click(Sender: TObject);
        procedure DBGrid1CellClick(Column: TColumn);
        procedure Button5Click(Sender: TObject);
    private
        { Private declarations }
    public
        { Public declarations }
    end;

var
    FStud_prog: TFStud_prog;

implementation

{$R *.dfm}
uses unDM;

```

```

procedure TFStud_prog.Button2Click(Sender: TObject);
begin
close;
end;

procedure TFStud_prog.FormCreate(Sender: TObject);
begin
if dml.qstpn.active then
    dml.qstpn.Close;
    dml.qstpn.Open;
end;

procedure TFStud_prog.Button3Click(Sender: TObject);
begin
if dml.qstPrg.active then
    dml.qstPrg.Close;

dml.qstPrg.Parameters.ParamByName('p').Value:=strtoint(kod.Text);
    dml.qstPrg.Open;
end;

procedure TFStud_prog.DBGrid1CellClick(Column: TColumn);
begin
kod.Text:=dml.TStd.Fields.Fields[0].AsString;
end;

procedure TFStud_prog.Button5Click(Sender: TObject);
begin
close
end;

end.

```

Тексти SQL запитів

Визначення номеру останнього питання тесту

```

SELECT ThemaContens.IDPunkt, Max(Test.IDVopros) AS [Max-IDVopros]
FROM ThemaContens INNER JOIN Test ON ThemaContens.IDPunkt =
Test.IDpunkt
GROUP BY ThemaContens.IDPunkt
HAVING (((ThemaContens.IDPunkt)=:p));

```

Визначення результату студента

```

SELECT TRez.IdStud, TRez.Nvop, TRez.bal
FROM TRez
GROUP BY TRez.IdStud, TRez.Nvop, TRez.bal
HAVING (((TRez.IdStud)=:p));

```

Сумарний бал студентів

```

SELECT TRez.IdStud, Sum(TRez.bal) AS [Sum-bal]
FROM TRez
GROUP BY TRez.IdStud;

```

Сумарний результати по питанням тесту

```

SELECT TRez.Nvop, Sum(TRez.bal) AS [Sum-bal]

```

```
FROM TRez
GROUP BY TRez.Nvop
ORDER BY TRez.Nvop;
```

Розрахунок показників якості відповідей

```
SELECT stendl.Bal AS Xi, [stendl]![Bal]/7+0.000001 AS pi, 1-
[pi]+0.0002 AS qi, Log([pi]/[qi]) AS Qli, [qli]*[qli] AS
Qli2,stendl.familiya
FROM stendl
ORDER BY stendl.Bal DESC;
```

Розрахунок якості питань

```
SELECT TEnd.Idvopros, TEnd.tbal, [tbal]/12+0.0000001 AS Pbi, 1-
[Pbi]+0.00001 AS Qbi, Log([qbi]/[pbi]) AS bi, [bi]*[bi] AS bi2
FROM TEnd;
```

Вибір даних для програми студентів

```
SELECT Stud_Programa.IDVopros, ThemaContens.UkrName,
Os_Source.NameBook, Os_Sourse_ED.Fame, Stud_Programa.KodStudenta,
Os_Sourse_ED.Aname, Test.TextVopros
FROM Stud_Programa AS Stud_Programa_1, ((Stud_Programa INNER JOIN
Test ON Stud_Programa.IDVopros = Test.IDVopros) INNER JOIN
ThemaContens ON Test.IDpunkt = ThemaContens.IDPunkt) INNER JOIN
(Os_Source INNER JOIN Os_Sourse_ED ON Os_Source.CodeB =
Os_Sourse_ED.CodeB) ON ThemaContens.IDPunkt = Os_Sourse_ED.IDPunkt
WHERE (((Stud_Programa.KodStudenta)=:p));
```


Інструкція користувача.

Під час запуску програми користувач потрапляє на головну форма ПЗ, яка представлена на рисунку 3.1. В залежності від його прав користувач отримає можливість "Пройти тестування" якщо він студент та "Робота з базою" якщо він викладач рис В.1.

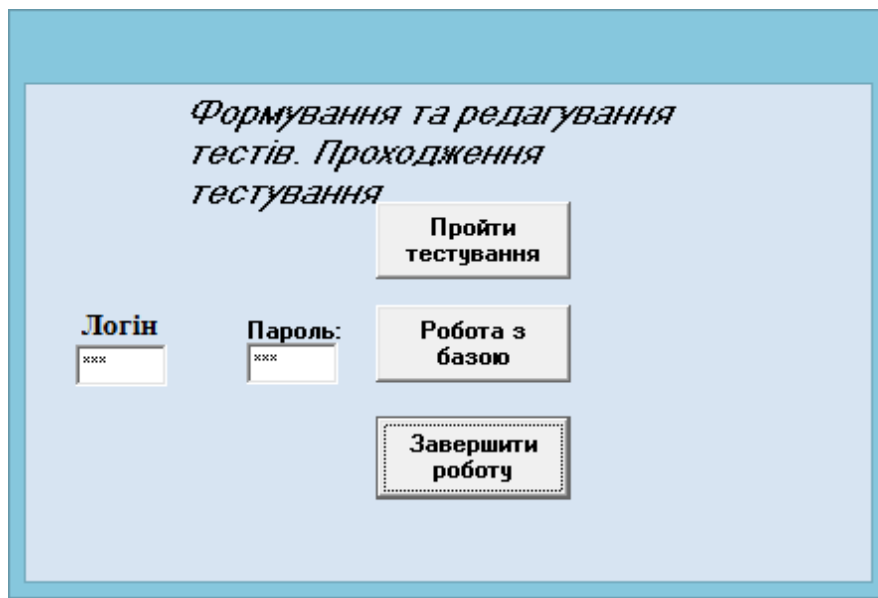


Рисунок В.1 – Головна форма

Після обрання та проходження тесту , рис В.2, студент має змогу побачити свій індивідуальний план натиснувши на "Формування програми студента" рис. В.3.

Тестування

Реєстрація

N тесту Дата

Студент

Виберіть своє прізвище зі списку

KodStudenta	Familiya
115	Іванов
116	Сидоров
117	Мала
118	Сулена
119	Гаврилов

Тестові завдання

Що таке база даних?

Будь-який текстовий файл

Організована структура для зберігання інформації

Будь-яка інформація, представлена в табличній формі

Будь-яка електронна таблиця

Із 9 завдань ви дали вірну відповідь на 8

Виберіть варіант

☐ Варіант1

☐ Варіант2

☐ Варіант3

☐ Варіант4

Рисунок В.2 – Форма проходження тесту

Формування програми студенту

Вкажіть, що визначає структуру реляційної таблиці:

Виберіть код студента з таблиці

Код студента	Прізвище	Ім'я
115	Іванов	Іван
116	Сидоров	Семен
117	Мала	Марина
118	Сулена	Євгенія
119	Гаврилов	Марко
120	Кленов	Іван
121	Михайлюк	Валентин
122	Кравченко	Олег

Рисунок В.3 – Форма індивідуальної програми студента

Викладач натиснувши на "Формування довідників" рис В.4 має можливість редагувати довідники рис В.5 для тестів та створювати/редагувати тести рис В.6.

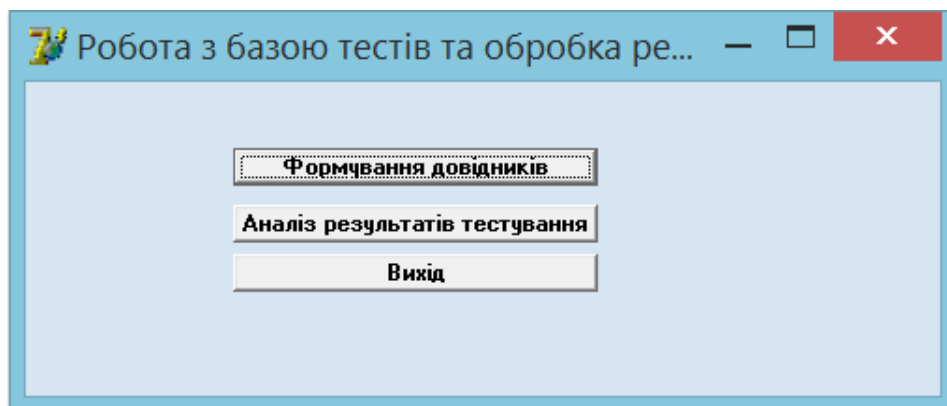


Рисунок В.4 – Форма роботи з базою тестів та обробки результатів

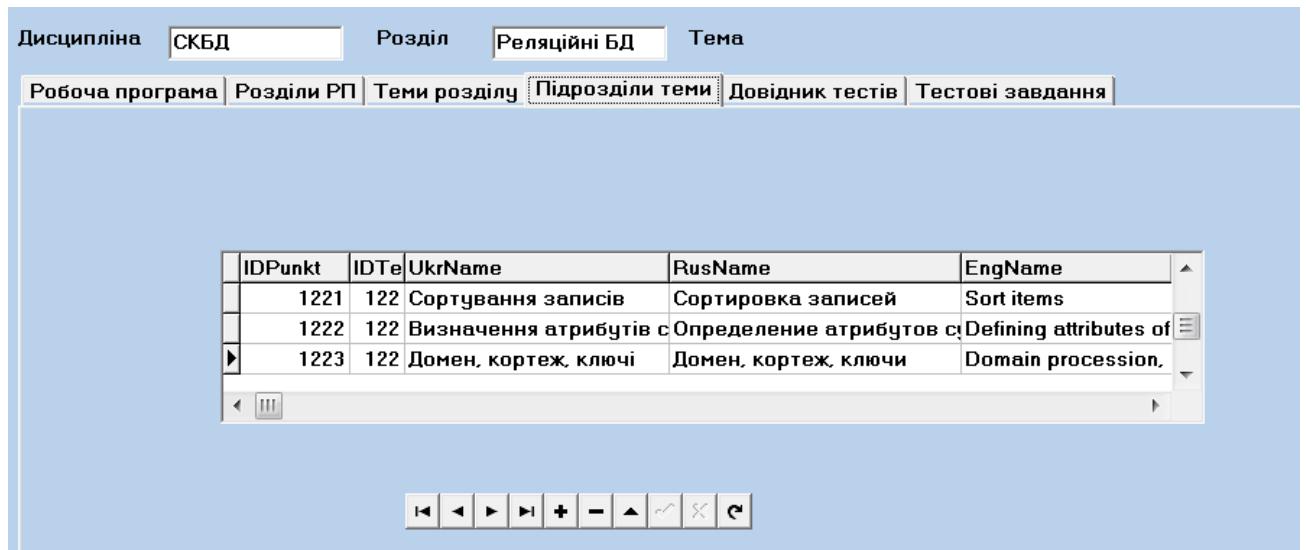


Рисунок В.5 – Форма створення тестових завдань "Підрозділи теми"

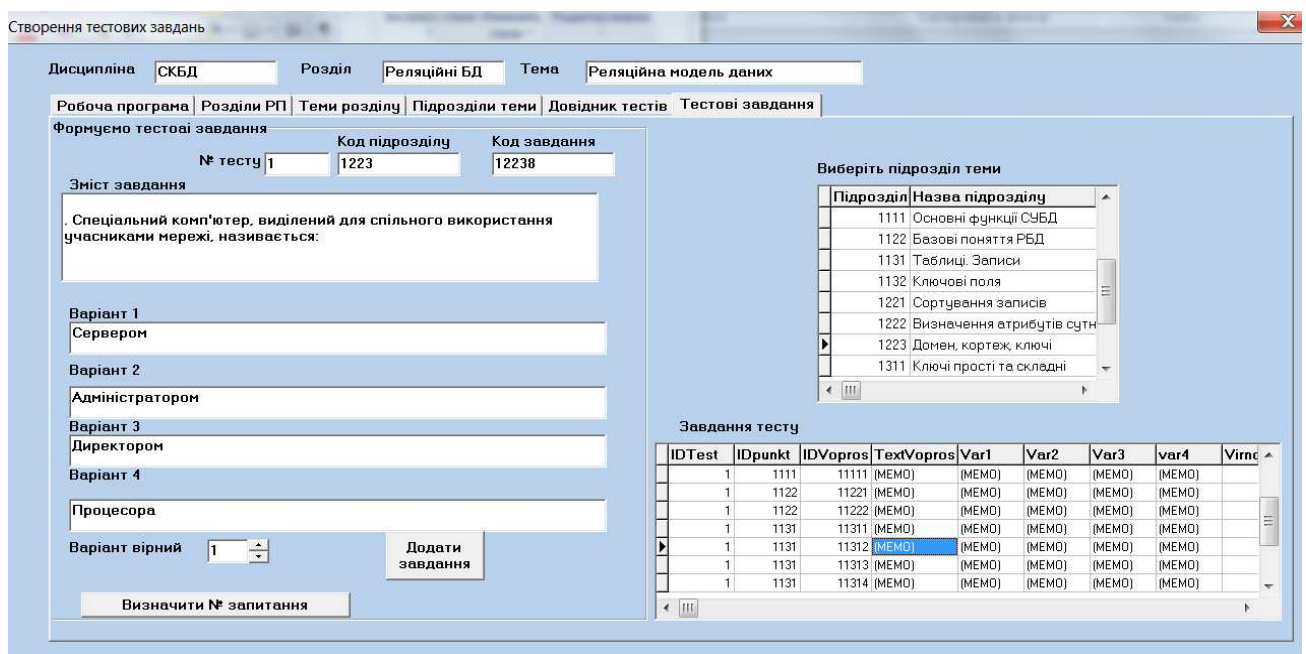


Рисунок В.6 – Форма створення тестових завдань "Тестові завдання"

Натиснувши на "Аналіз результатів тестування" відкривається форма "Обробка результатів" на якій викладач має можливість отримати результати

тестування та їх аналіз.

Обробка результатів

Відповіді студентів									
	11111	11221	11222	11311	11312	11313	11314	11321	11322
Іванов	1	1	1	1	1	1	1	1	0
Сидоров	1	0	0	1	1	0	0	1	0
Мала	1	1	0	1	1	1	0	1	0
Сулема	1	1	0	1	1	0	0	1	0
Гаврилов	1	0	1	1	1	0	0	1	0
Кленов	1	0	0	1	1	0	1	0	0
Михайлюк	1	1	1	1	1	1	0	1	0
Кравченко	1	0	0	0	0	0	0	0	0
Богданов	1	0	0	1	1	1	0	0	0
Рожко	1	0	0	1	1	0	0	1	0
Склянов	1	1	0	1	1	0	0	1	0
Скалов	1	0	0	0	1	0	0	1	0

Сумарні показники									
	11221	11222	11311	11312	11313	11314	11321	11322	сума
Іванов	1	1	1	1	1	1	1	1	8
Сидоров	0	0	1	1	0	0	0	1	4
Мала	1	0	1	1	1	0	1	1	6
Сулема	1	0	1	1	0	0	1	1	5
Гаврилов	0	1	1	1	0	0	1	1	5
Кленов	0	0	1	1	0	1	0	1	4
Михайлюк	1	1	1	1	1	0	1	1	7
Кравченко	0	0	0	0	0	0	0	0	1
Богданов	0	0	1	1	1	0	0	0	4
Рожко	0	0	1	1	0	0	1	1	4
Склянов	1	0	1	1	0	0	1	1	5
Скалов	0	0	0	1	0	0	1	1	3
	5	3	10	11	4	2	9		

	11312	11311	11321	11221	11313	11222	11314	X_i	X_{i2}		
Іванов	1	1	1	1	1	1	1	7	49	1,000	0,000
Сидоров	1	1	1	0	0	0	0	3	9	0,429	0,571
Мала	1	1	1	1	1	0	0	5	25	0,714	0,286
Сулема	1	1	1	1	0	0	0	4	16	0,571	0,429
Гаврилов	1	1	1	0	0	1	0	4	16	0,571	0,429
Кленов	1	1	0	0	0	0	1	3	9	0,429	0,571
Михайлюк	1	1	1	1	1	1	0	6	36	0,857	0,143
Кравченко	0	0	0	0	0	0	0	0	0	0,000	1,000
Богданов	1	1	0	0	1	0	0	3	9	0,429	0,571
Рожко	1	1	1	0	0	0	0	3	9	0,429	0,571
Склянов	1	1	1	1	0	0	0	4	16	0,571	0,429
Скалов	1	0	1	0	0	0	0	2	4	0,286	0,714
R_j	11	10	9	5	4	3	2				
W_i	1	2	3	7	8	9	10				
P_i	0,92	0,83	0,75	0,42	0,33	0,25	0,17				
q_i	0,08	0,17	0,25	0,58	0,67	0,75	0,83				

Підсумки
Оцінки рівнів
Вихід

Рисунок В.7 – Форма обробка результатів

Оцінки рівнів підготовленості та складності завдань тесту

i	Q_i	$Se(Q_i)$	Q_i''
1	2,646	1,308	1,790
2	2,646	1,308	1,790
3	1,258	1,014	0,916
4	0,260	0,926	0,287
5	0,260	0,926	0,287
6	0,260	0,926	0,287
7	-0,653	0,926	-0,288
8	-0,653	0,926	-0,288
9	-0,653	0,926	-0,288
10	-0,653	0,926	-0,288
11	-1,650	1,014	-0,917
12	-3,040	1,309	-1,792

j	B_j	$Se(B_j)$	B_j''
1	-2,08	1,565	-1,611
2	-2,167	1,582	-1,536
3	-1,250	1,402	-0,630
4	-0,583	1,363	-0,336
5	0,667	1,454	0,461
6	2,750	1,683	1,808
7	3,833	1,908	2,050

i	S_1	S_2	S_3
1	3,646	2,308	-2,790
2	4,646	3,308	-1,790
3	3,258	1,014	-2,050
4	3,650	2,926	-2,287
5	2,260	-1,987	-1,653
6	2,260	0,901	-2,916
7	1,653	0,926	-2,288
8	1,688	1,288	-3,226
9	1,653	1,926	-4,288
10	1,626	-1,901	-3,288
11	2,050	-3,014	-2,917
12	3,040	-4,309	-4,792

Обчислити оцінки
Вихід

Рисунок В.8 – Форма оцінки рівнів підготовленості та складності завдань тесту