

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»
Факультет інформатики
Кафедра інформатики

РОЗРОБКА ЗАСТОСУНКУ ДЛЯ ОРГАНІЗАЦІЇ ТЕСТУВАННЯ У
ШКОЛАХ
Текстова частина до курсової роботи
за спеціальністю 121 «Інженерія програмного забезпечення»

Керівник курсової роботи
к-т фіз.-мат. наук, доцент
Жежерун О. П.
(прізвище та ініціали)

(підпис)
“ ” 2020 р.

Виконала студент
Данилова І. А.
(прізвище та ініціали)

(підпис)
“ ” 2020 р.

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»
Кафедра мультимедійних систем факультету інформатики

ЗАТВЕРДЖУЮ

Завідуючий кафедри мультимедійних систем
к-т фіз.-мат. наук, доцент _____ Жежерун О.П.
(підпис)

« ____ » _____ 2019р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ

на курсову роботу

студента Данилової Ірини Анатоліївни факультету інформатики 3 р.н.

бакалаврської програми

ТЕМА: Розробка застосунку для організації тестування у школах

Зміст текстової частини до курсової роботи:

Вступ

Аналіз предметної області

Проектування програмного рішення

Розробка програмного рішення

Висновки

Список літератури

Додатки (за необхідністю)

Дата видачі « ____ » _____ 2019 р.

Керівник _____
(підпис)

Завдання отримав _____
(підпис)

Тема: Розробка веб-застосунку для організації тестування у школах

Календарний план виконання роботи:

№	Назва етапу	Термін виконання	Примітка
1	Отримання теми курсової роботи	23.10.2019	
2	Пошук та ознайомлення з літературою	10.12.2019	
3	Вивчення аналогів	20.01.2020	
4	Планування веб-сайту	06.02.2020	
5	Створення бази даних	20.02.2020	
6	Написання першого розділу	15.03.2020	
7	Написання другого розділу	01.04.2020	
8	Написання практичної частини	23.04.2020	
9	Написання третього розділу	27.04.2020	
10	Написання висновків	01.05.2020	
11	Створення презентації	05.05.2020	

Студент Данилова І. А.

Керівник Жежерун О. П.

“ ”

Зміст

ВСТУП.....	5
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ.....	6
1.1 Аналіз існуючих методів вирішення поставленої задачі.....	6
1.2 Обґрунтування вибору рішення.....	7
РОЗДІЛ 2. ПРОЕКТУВАННЯ ПРОГРАМНОГО РІШЕННЯ.....	11
2.1 Вибір принципу дії системи.....	11
2.2 Розробка структурної і функціональної схеми.....	17
РОЗДІЛ 3. РОЗРОБКА ПРОГРАМНОГО РІШЕННЯ.....	21
3.1 Експериментальні дослідження.....	21
3.2 Алгоритмічне та програмне забезпечення.....	26
ВИСНОВКИ.....	29
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ.....	30

ВСТУП

Актуальність дослідження. В сучасних умовах особливо актуально організувати процес навчання так, щоб його освітній результат виявлявся в розвитку власної внутрішньої мотивації навчання, мислення, уяви, творчих здібностей, стійкого пізнавального інтересу учнів, у формуванні системи життєво важливих, практично затребуваних знань і умінь, екологічної культури, що дозволяє учням адаптуватися до життя і ставитися до неї активно, творчо.

Мета навчання в сучасній школі передбачає не тільки засвоєння знань, а й загальний розвиток учнів. Для цього розробляються все нові і нові технології: програмованого навчання, поетапного формування розумових дій (П. Я. Гальперін), розвиваючого навчання (В. В. Давидов, Л. В. Занков, Д. Б. Ельконін), особистісно орієнтованого навчання (В. І. Загвязинский, Е. Ф. Зеер, В. П. Зінченко, І. С. Якиманська)[1].

Актуальність розробки тестування учнів обумовлена наступними обставинами: в рамках практико-орієнтованого підходу значно підвищується ефективність навчання завдяки підвищенню особистісного статусу учня і практико-орієнтованого змісту досліджуваного матеріалу; в процесі взаємодії в системі «вчитель-учень» постійно діють канали зворотного зв'язку; система розвиває інтерес учнів до творчості, дозволяє їм пізнати радість творчої діяльності.

Постановка завдання:

- вивчити практико-орієнтовану технологію;
- розробити практико-орієнтовані завдання для формування і оцінки знань та умінь школярів.

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Аналіз існуючих методів вирішення поставленої задачі

Тестування широко застосовується для оцінки рівня знань в навчальних закладах, при прийомі на роботу, для оцінки кваліфікації персоналу установ, тобто практично у всіх сферах діяльності людини. Випробуваному пропонується ряд питань (тест), на які він повинен відповісти.

Зазвичай до кожного питання дається кілька варіантів відповіді, з яких треба обрати правильний. Кожному варіанту відповіді відповідає деяка оцінка. Підсумовуванням оцінок за відповіді виходить загальний бал, на основі якого робиться висновок про рівень підготовленості випробуваного.

Основна відмінність тесту полягає в тому, що він завжди передбачає вибір. Оцінка, що виставляється за підсумками тестування, відрізняється більшою об'єктивністю і незалежністю від можливого суб'єктивізму того, хто проводить співбесіду або тестування усно. Головна відмінна риса тесту - об'єктивність, що гарантується вимірюванням, функція якого полягає в тому, щоб давати кількісну (бальну) оцінку.

Максимально підвищити надійність вимірювання і валідність використання тесту можна, якщо слідувати трьом основним етапам його створення:

- дати чітке і недвозначне теоретичне визначення умінь, які треба перевірити;
- точно встановити умову і операції, які повинні дотримуватися при проведенні тесту;
- кількісно визначити результати спостережень, щоб переконатися, що мірні шкали володіють всіма необхідними якостями.

Під комп'ютерним тестом будемо розуміти реалізовану за допомогою персонального комп'ютера систему взаємопов'язаних предметним змістом завдань стандартної форми, що мають певну шкалу значень, виконання яких виявляє наявність знань, умінь і навичок і рівня підготовки фахівця.

Формування умінь і навичок доцільно перевіряти, враховуючи такі взаємодоповнюючі критерії:

1. Правильність: результат будь-якої діяльності відповідає меті виконання.
2. Осмисленість: готовність пояснити кожную з застосовуваних операцій.
3. Усвідомленість: орієнтація на суттєві зв'язки і відносини, задані в умові при здійсненні діяльності.
4. Раціональність: вибір такого способу виконання завдань, який швидше (оптимально) призведе до мети.
5. Узагальнення: здатність школяра виконувати необхідні дії в варіативних умовах.
6. Абстрактність: можливість опису оперативного складу виконаної дії і послідовності виконання операцій, що входять в неї, відтворення необхідного правила.

1.2 Обґрунтування вибору рішення

В даний час існує безліч систем, призначених для автоматизації тестування знань школярів, як платних, так і безкоштовних.

Розглянемо програми, орієнтовані на специфіку навчання у вітчизняних школах. У таблиці 1.1 наведено порівняльну характеристику автоматизованих систем тестування.[9] [10]

Таблиця 1.1 – Порівняння автоматизованих систем тестування

Ноп / п	Назва	Робота через інтернет	платформа	Ліцензія
1	Система «Синтез»	+	PHP, MySQL	платна
2	«Конструктор тестів»	-	C ++	Безкоштовна
3	«OpenTest 2.0»	+	HTML, PHP,	платна

			JavaScript	
--	--	--	------------	--

В ході аналізу програм аналогів були виявлені недоліки у перерахованих вище систем тестування, а саме функціональна перевантаженість, платна ліцензія, а також не всі системи є масштабованими і не працюють через інтернет. Таким чином можемо зробити висновок, що розробка власних коштів автоматизованого тестування знань позбавленого виділених у програм аналогів недоліків є доцільною.

Необхідно розробити програмний комплекс, який, по своїй суті, є тестовою системою. Тестова система включає блок питань, що дозволяють визначити наявність навичок аналізувати приблизні програми навчальних курсів. У блок повинні бути включені питання, які містять фактичні та чисельні показники програм, для перевірки обізнаності та рівня знань структури і змісту конкретних програм.

Дана програма розробляється як прототип електронного засобу для застосування в навчальному процесі. Наповнення бази питань є тестовим і переслідує основну мету - продемонструвати працездатність програмного засобу. Програма повинна проектуватися таким чином, щоб блок питань міг бути розширений або повністю замінений без змін (перекомпіляції) самої програми. Тобто питання повинні бути представлені в зовнішніх даних, що підключаються до програми.

Основним завданням розробки є модуль тестування, але з огляду на дослідження, проведені в першому розділі - архітектура програмного комплексу повинна будуватися гнучкою.

Крім безпосередньо модуля тестування, комплекс може включати і теоретичний, і практичний блоки. Комплекс може припускати наявність деяких інтерактивних завдань або «живих схем». Інтерактивність, закладена в реалізації, повинна сприяти кращому розумінню матеріалу, практичні приклади забезпечують розуміння застосування теорії, а система тестування дозволяє перевірити рівень засвоєння знань.

Основні вимоги:

Зовнішній вигляд програми не повинен відволікати школярів від засвоєння викладеного в програмі матеріалу.

Діалог з користувачем здійснюється за допомогою меню, представленого на кожній формі програми.

Програма повинна являти собою електронний підручник, орієнтований як на стаціонарне, так і на дистанційне вивчення або для використання в процесі самоосвіти. Програма також може використовуватися для забезпечення практичних робіт в очному курсі вивчення матеріалів предмета.

З огляду на основну мету розробки забезпечення освітнього процесу, програма повинна передбачати подачу матеріалу в різних формах:

- теоретична складова;
- практична складова;
- контролюючий модуль.

Програма повинна мати зрозумілий приємний інтерфейс. У програмі повинні бути передбачені всі вище перераховані функціональні можливості.

Автоматизована система тестування повинна виконувати наступні функції:

- прийом призначених для користувача даних;
- перевірка введених даних користувачем в полях авторизації;
- підрахунок правильних і неправильних відповідей;
- звіт про результат тестування;
- зберігання результатів тестування.

Категорії користувачів

1) Гість - неавторизований користувач, має права:

- тестова частина - тільки ознайомлення з темою тесту, без переходу до тестування;
- бали - немає доступу;
- сторінка авторизації - перегляд, введення даних.

2) Школяр - авторизований користувач, має права:

- тестова частина - перехід до тестування;
 - бали - є доступ по завершенню тесту;
 - сторінка авторизації - перегляд, введення даних.
- 3) Викладач - авторизований користувач, має права:
- тестова частина - перехід до тестування;
 - бали - є доступ до результатів всіх авторизованих користувачів;
 - розділ «Нотатки» - перегляд, створення нових заміток;
 - сторінка авторизації - перегляд, введення даних.
- 4) Адміністратор - авторизований користувач, має права:
- тестова частина - перехід до тестування, створення нового тесту;
 - бали - є доступ до результатів всіх авторизованих користувачів;
 - паролі - є доступ до списку зберігання всіх паролів користувачів;
 - сторінка авторизації - перегляд, введення даних.

РОЗДІЛ 2. ПРОЕКТУВАННЯ ПРОГРАМНОГО РІШЕННЯ

2.1 Вибір принципу дії системи

Виділимо бізнес-процеси, що протікають в системі і занесемо їх в таблицю 2.1.

Таблиця 2.1 – Бізнес-процеси предметної області

№ п / п	Бізнес-процес	Виконавець	Вхідні дані		Вихідні дані	
			Постачальник	Те, що передає	Споживач	Те, що отримує
1	Додати тест	Клієнтський додаток	Викладач	Новий тест	База даних	Новий тест
2	Вилучити тест	Клієнтський додаток	Викладач	Тест, який необхідно вилучити	База даних	Оновлений список тестів
3	Редагувати тест	Клієнтський додаток	Викладач	відредагований тест	База даних	відредагований тест
4	Додати користувача	Клієнтський додаток	Викладач / школяр	Дані про новому користувача	База даних	Дані про нового користувача
5	Отримати перелік тестів	Клієнтський додаток	База даних	Предмет, за яким необхідно отримати тести	Викладач / школяр	Список тестів
6	Отримати перелік питань	Клієнтський додаток	База даних	Тест, з якого необхідно отримати питання	Викладач / школяр	Перелік запитань
7	Додати інформацію про тестування	Клієнтський додаток	клієнтське додаток	Інформація про проведений тестуванні	База даних	Інформація про проведений тестуванні

8	Подивитися результати тестування	Клієнтський додаток	база даних	результати тестування	викладач	Результати тестування
---	----------------------------------	---------------------	------------	-----------------------	----------	-----------------------

На підставі виділених сутностей, їх атрибутів і зв'язків побудуємо концептуальну модель предметної області в нотації draw.io (рис. 2.1).

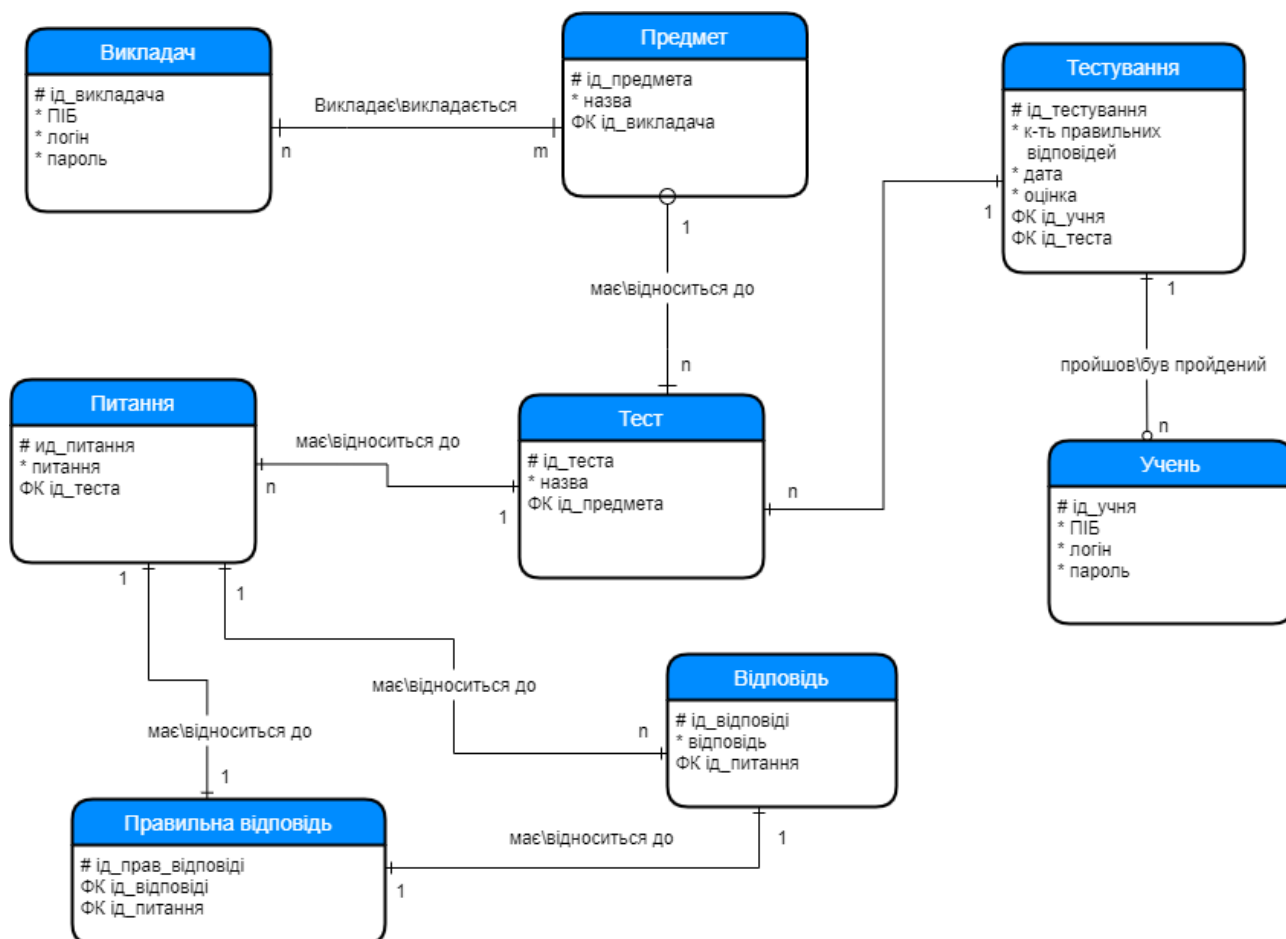


Рисунок 2.1 - Концептуальна модель системи

В ході подальшої роботи буде здійснено перехід від концептуальної моделі до логічної.

Аналіз функціональних вимог.

Вимогу можна визначити, як «докладний опис того, що повинно бути реалізовано». Користувачів клієнтського додатку виділяється 2 типи:

- викладач;
- школяр.

Класи і характеристики користувачів представлені в таблиці 2.7.

Відповідно до поставленим завданням була побудована модель прецедентів предметної області. Діаграма варіантів використання представлена на рисунку 2.2.

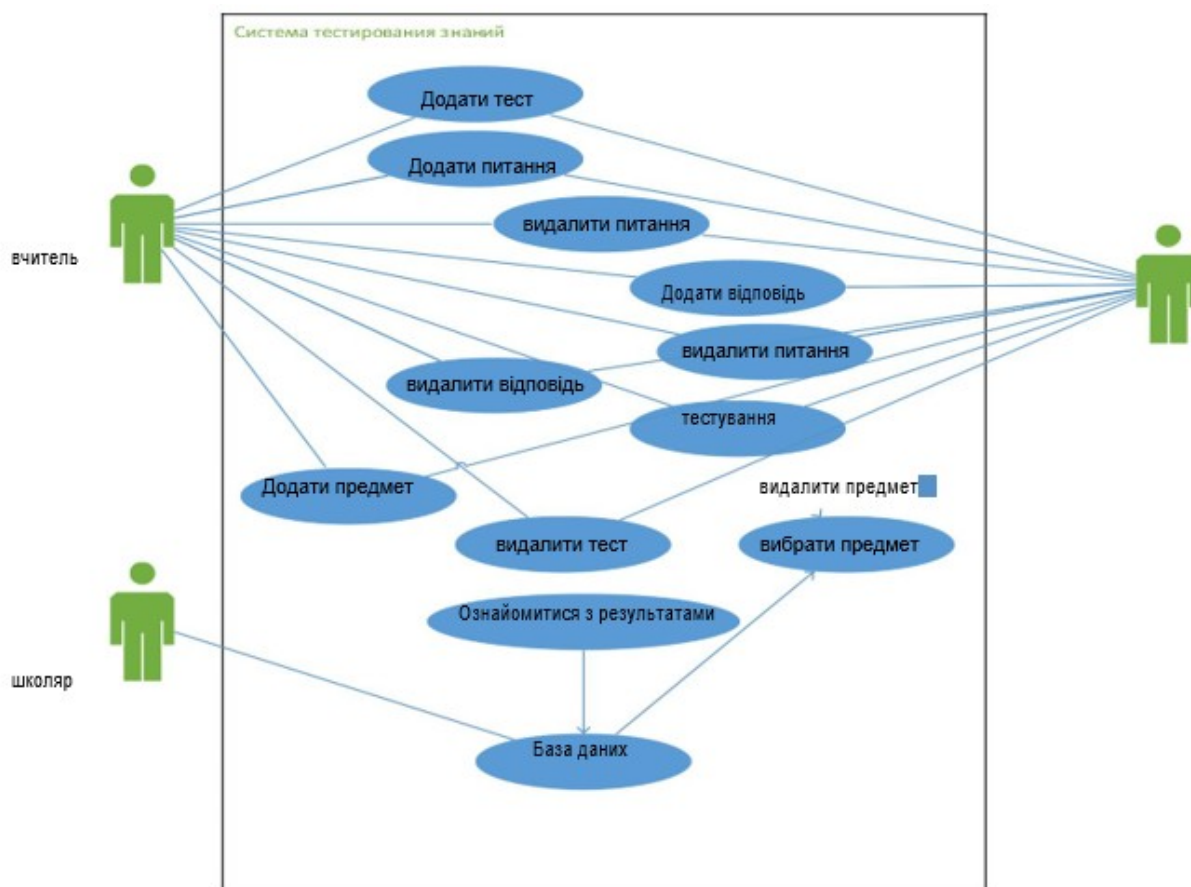


Рисунок 2.2 - Діаграма варіантів використання

Після створення діаграми прецедентів і виявлення акторів приступають до точного визначення кожного прецеденту. Цей процес називається деталізацією прецедентів. Як підсумок цієї діяльності буде отримано більш деталізований прецедент, що включає, як правило, ім'я прецеденту і його специфікацію. Специфікації прецедентів представлені для: додати тест, додати питання, пройти тест, вибрати предмет, для альтернативного прецеденту виконання тесту перервано.

Таблиця 2.2 – специфікація прецеденту «Додати тест».

Прецедент: додати тест.
ID: 1.
Короткий опис: додавання нового тесту.
Користувач: вчитель.

Предумови: прецедент починається, коли вчитель обрав опцію “додати тест”.
Основний потік: <ol style="list-style-type: none"> 1. Система запитує у викладача назву тесту. 2. Якщо тест з таким ім’ям вже існує. <ol style="list-style-type: none"> 2.1 Видати інформацію про помилку. 3. Інше. <ol style="list-style-type: none"> 3.1 Система додає новий тест. 4. Оновлений список тестів зберігається у базі даних.
Постумови: <ol style="list-style-type: none"> 1. Система оновлює список тестів.
Альтернативні потоки: немає.

Таблиця 2.3 – специфікація прецеденту «Додати питання».

Прецедент: додати питання.
ID: 2.
Короткий опис: додавання нового питання.
Користувач: вчитель.
Предумова: прецедент починається, коли вчитель обрав опцію “додати питання”.
Основний потік: <ol style="list-style-type: none"> 1. Система запитує у викладача ввід даних. 2. Викладач вводить питання, можливі варіанти відповіді та правильну відповідь на запитання. 3. Викладач підтверджує додавання нового питання. 4. Система перевіряє коректність введених даних, а саме наявність питання, відповідей та правильної відповіді. 5. Якщо перевірка даних не була завершена успішно. <ol style="list-style-type: none"> 5.1 Видати повідомлення про введення некоректних даних. 6. Інше. <ol style="list-style-type: none"> 6.1 Видати повідомлення про введення некоректних даних. 7. Нове питання зберігається в базі даних.
Постумова: <ol style="list-style-type: none"> 1. Система оновлює список питань на екрані.
Альтернативні потоки: немає.

Таблиця 2.4 – специфікація прецеденту «Пройти тест».

Прецедент: пройти тест.
ID: 3.
Короткий опис: проходження тесту школяром.
Користувач: школяр.
Предумова: немає.
Основний потік: <ol style="list-style-type: none"> 1. Include (Обрати предмет). 2. Система відображає список тестів з обраного предмету. 3. Школяр обирає тест та опцію “виконати”. 4. Система ініціює початок тестування і видає школяру питання, поки всі питання не задані. 5. Школяр повинен відповісти на поточне питання, щоб перейти до наступного. 6. Після проходження тесту система видає на екран результати тестування.
Постумова: <ol style="list-style-type: none"> 1. Система фіксує результат виконання тестування в базу даних.
Альтернативні потоки: виконання тесту перервано.

Таблиця 2.5 – специфікація альтернативного прецеденту «Виконання тесту перервано».

Прецедент: виконання тесту перервано.
ID: 3.1.
Короткий опис: школяр скасовує проходження тесту.
Користувач: школяр.
Предумова: немає.
Основний потік: <ol style="list-style-type: none"> 1. Система запитує у школяра підтвердження скасування тестування. 2. Якщо школяр підтвердив вибір. <ol style="list-style-type: none"> 2.1 Система скидає результати проходження даного тестування. 3. Інше. <ol style="list-style-type: none"> 3.1 Система повертає школяра до виконання тесту.
Постумова: немає.
Альтернативні потоки: немає.

Таблиця 2.6 – специфікація прецеденту «Обрати предмет».

Прецедент: обрати предмет.
ID: 4.
Короткий опис: обирання предмету для редагування/проходження тестів.
Користувач: викладач, школяр.
Предумова: виконаний вхід до системи.
Основний потік: <ol style="list-style-type: none"> 1. Система запитує у викладача/школяра предмет для редагування/проходження тесту. 2. Обраний предмет зберігається в системі для подальшої роботи.
Постумова: немає.
Альтернативні потоки: немає.

2.2 Розробка структурної і функціональної схеми

Побудуємо логічну модель предметної області в нотації IDEF1X за допомогою case-засоби ERwin Data Modeler. Логічна модель представлена на рисунку 2.2.

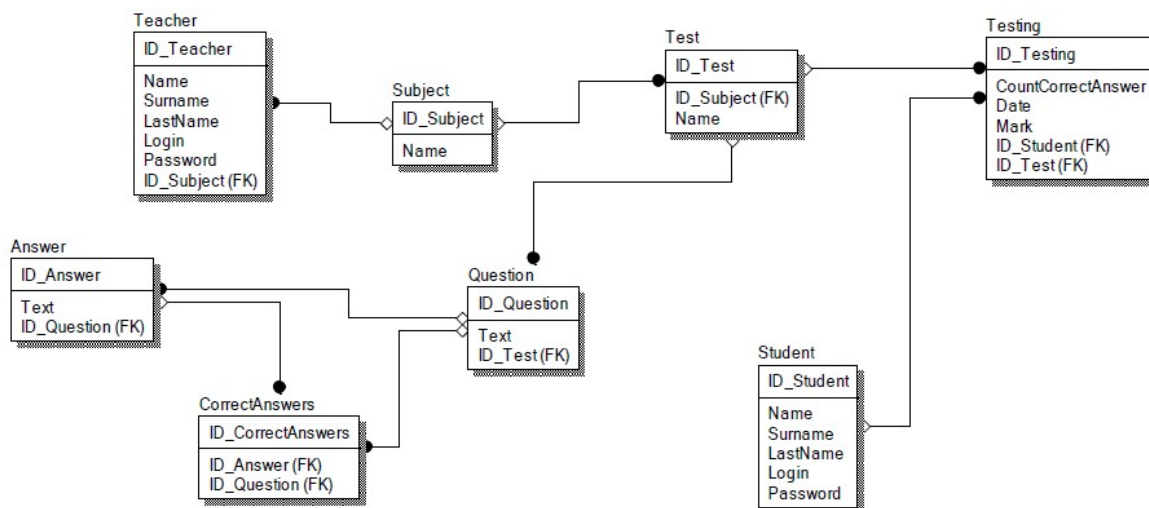


Рисунок 2.2 - Логічна модель системи

Розроблена логічна модель бази даних не містить транзитивних залежностей у відносинах. Таким чином, модель відповідає третій нормальній формі.

Виділимо типи сутностей і занесемо результати в таблицю 2.7.

Таблиця 2.7 – Типи сутностей предметної області

№	Назва	Опис	Категорія	К-ть екземплярів	Режим оновлення
1	Відповідь	Містить можливі відповіді на питання	Базова	1000	90% в рік
2	Правильна відповідь	Містить правильну відповідь на конкретне питання	Довідкова	1000	90% в рік
3	Питання	Містить питання для тестів	Довідкова	1000	90% в рік
4	Школяр	Містить інформацію про школяра	Базова	1000	90% в рік
5	Предмет	Містить можливі предмети для тестування	Довідкова	100	5% в рік
6	Вчитель	Містить інформацію про вчителя	Базова	1000	70% в рік
7	Тест	Містить перелік тестів по конкретному предмету	Довідкова	1000	90% в рік
8	Тестування	Містить результати проходження тестування	Базова	1000	90% в рік

Виділимо відносини між типами сутностей і занесемо результати в таблицю 2.8.

Таблиця 2.8 – Типи зв'язків між типами сутностей

№	Тип сутності 1	Тип сутності 2	Опис	потужність	Обов'язковість	Обмеження
1	питання	відповідь	питання <i>має</i> відповідь	1 : M	обов'язкова (Питання не може не мати відповіді)	без обмежень
2	питання	правильна відповідь	питання <i>має</i> правильну відповідь	1 : 1	обов'язкова	без обмежень

3	тест	питання	тест <i>має</i> питання	1 : М обов'язков а (Тест не може не мати жодного питання)	обов'язкова	без обмежень
4	предмет	тест	предмет <i>має</i> тест	1 : М	необов'язко ва (У предмета може не бути тестів)	без обмежень
5	школяр	тест	Тестуванн я опис	М : М потужність	необов'язко ва (Школяр може не пройти жодного тесту)	без обмежень
6	тестуванн я	тест	Тестуванн я	М : 1	обов'язкова (Тестування не може існувати без тесту)	без обмежень
7	вчитель	предмет	Вчитель викладає предмети	М : М	обов'язкова (Вчитель не може не викладати жодного предмета)	без обмежень

РОЗДІЛ 3. РОЗРОБКА ПРОГРАМНОГО РІШЕННЯ

3.1 Експериментальні дослідження

Інтерфейс починається зі входу в систему тестування (рис.3.1).

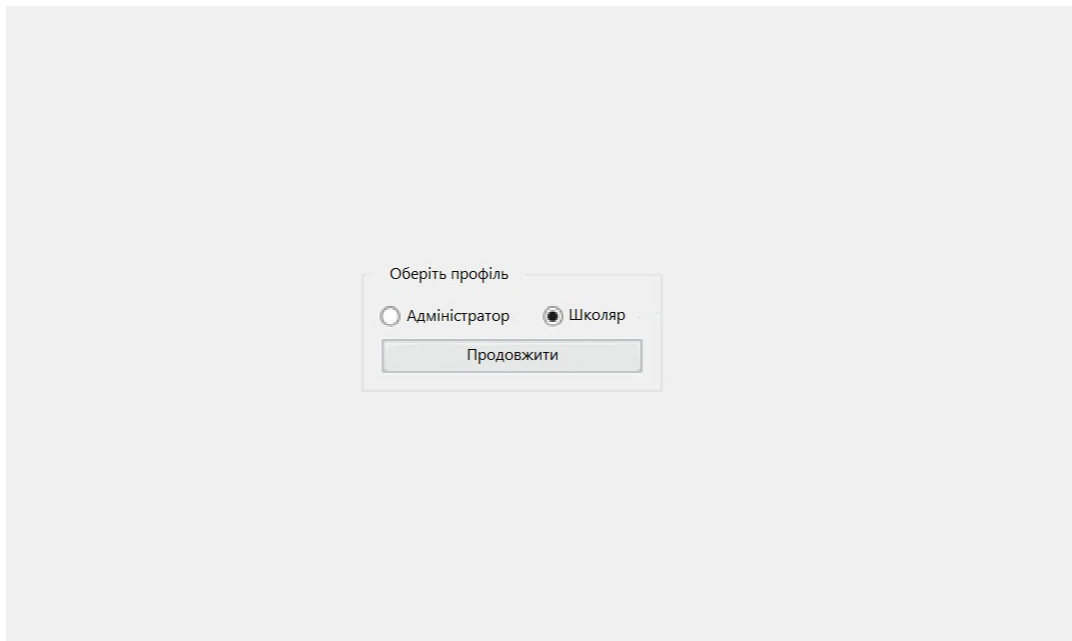


Рисунок 3.1. Вхід у систему. Користувач – Школяр

Далі вводимо логін / пароль.

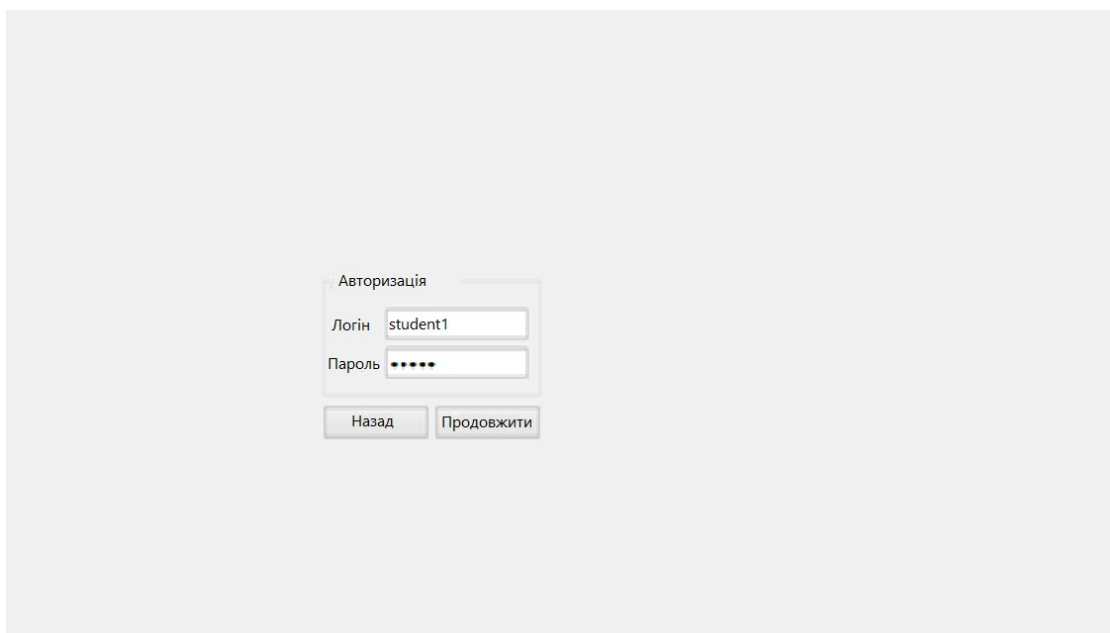


Рисунок 3.2. Авторизація

Натискаємо кнопку Продовжити і переходимо в робоче вікно.

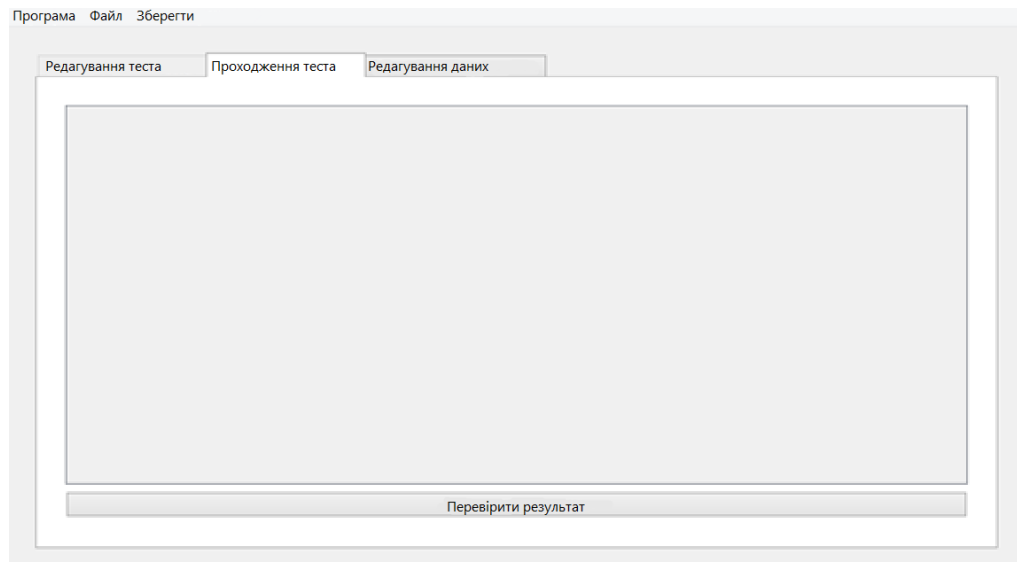


Рисунок 3.3. Головне вікно системи тестування

Тут будуть дані тести, так само є можливість відкриття (підвантаженими) вже готових раніше тестів.

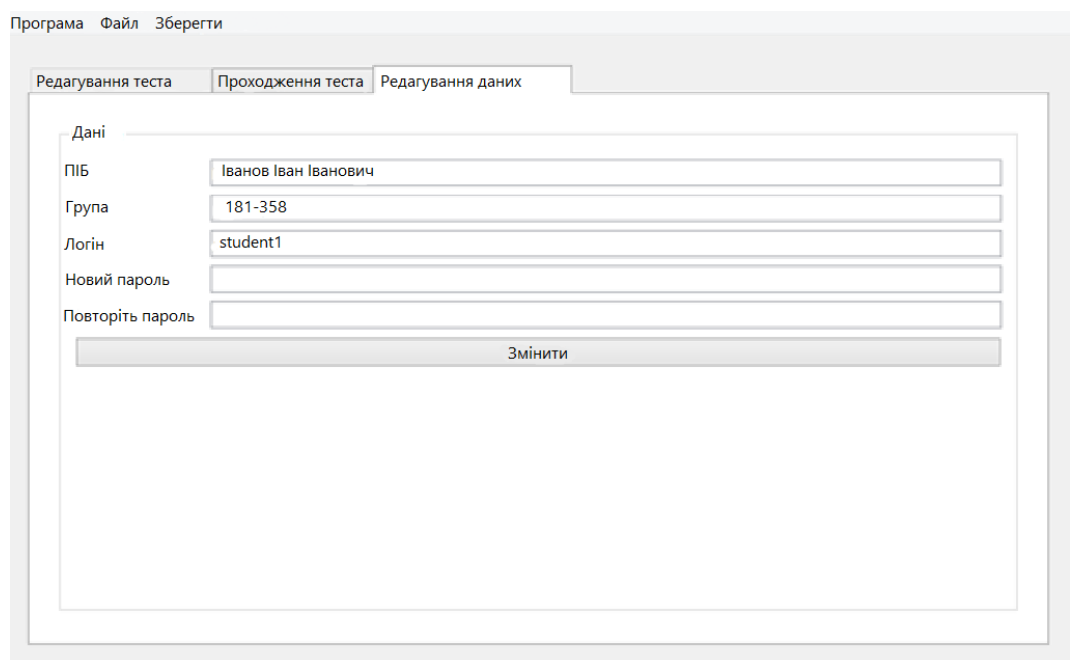


Рисунок 3.4. Вкладка редагування даних

Вхід від імені адміністратора. Створення тесту.

Програма Файл Зберегти

Редагування тесту Проходження теста Редагування даних

Назва теста

* правильний варіант відповіді помічати зірочкою(*) вкінці

Тест

Питання	1 варіант	2 варіант	3 варіант	4 варіант
1				

Додати строку Видалити строку

Рисунок 3.5. Створення тесту

Редагування даних.

Програма Файл Зберегти

Редагування тесту Проходження теста Редагування даних

Змінити пароль адміністратора

Новий пароль

Повторіть пароль

Змінити

Дані

ПІБ

Група

Логін

Змінити

Додати

ПІБ

Група

Логін

Пароль

Додати

Рисунок 3.6. Редагування даних

В даному розділі детально описана архітектура системи тестування. Проаналізовано концептуальна і даталогічна модель системи. Визначено основні алгоритми роботи системи. А також, наведено інтерфейс користувача системи.

Введення в експлуатацію і інструкція користувача АС.

The screenshot shows a web application window with a menu bar at the top containing 'Програма', 'Тест', 'Зберегти', and 'Мова'. Below the menu bar is a tabbed interface with three tabs: 'Робота з тестом(створення, збереження)', 'Проходження теста', and 'Панель управління'. The 'Панель управління' tab is active. It contains a section titled 'Дані' with a list of input fields: 'ПІБ', 'Група', 'Логін', 'Новий пароль', and 'Повторіть пароль'. Below these fields is a 'Змінити' button. Underneath is a section titled 'Список доступних тестів для проходження' with a large empty rectangular box. At the bottom, there is a label 'Останній результат:'.

Рисунок 3.7. Панель управління користувача

The screenshot shows the same web application window as in Figure 3.7, but with the 'Проходження теста' tab active. The 'Робота з тестом(створення, редагування)' tab is also visible. The main content area displays two test questions. Question 1 is 'Оберіть правильну відповідь: Яку столицю має Фрація?' with four radio button options: 'Париж' (selected), 'Барселона', 'Київ', and 'Мілан'. Question 2 is 'Оберіть правильну відповідь: Яку висоту має гора Поп-Іван?' with four radio button options: '2061 м' (selected), '2022 м', '1936 м', and '1883 м'. The interface includes a vertical scrollbar on the right side of the question list.

Рисунок 3.8. Проходження тесту

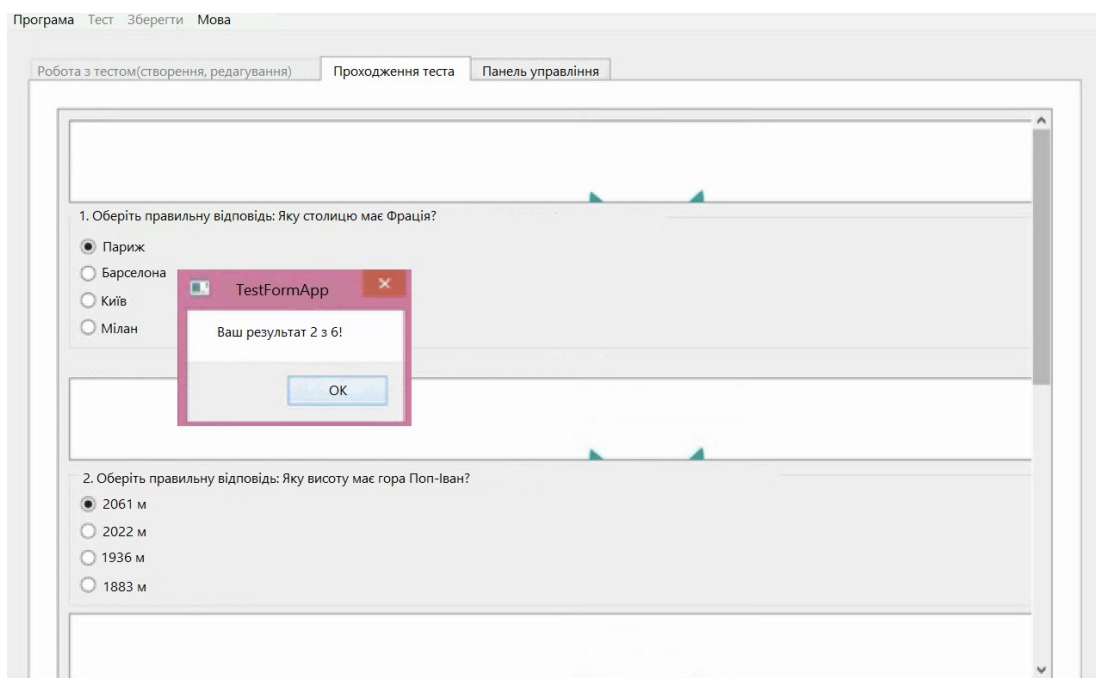


Рисунок 3.9. Результат тесту

Стандартні бланкові тестові методики з множинним вибором відповідей і їх комп'ютерні аналоги мають відносно просту лінійну структуру і як правило припускають, що тестові оцінки прямо визначають ту чи іншу шкалу, відповідну вимірюваної концептуальної змінної.

Разом з тим у багатьох випадках психологічної оцінки зв'язку між емпіричними виборами або оцінками і концептуальними змінними можуть і повинні бути більш складними.

Наприклад, це можуть бути ситуаційні вимірювання з ієрархічно заданими відносинами або так звані не реактивні вимірювання, що оцінюють відносини (такі як переваги, упередження, несвідому поведінку і ін.).

Розробка і проведення таких тестів істотно полегшуються з використанням сучасних комп'ютерних технологій.

Разом з тим пропозицію комп'ютерних програм, призначених для створення і проведення цих тестів, невелика (треба провести аналіз таких пропозицій в літературі та інтернеті).

3.2 Алгоритмічне та програмне забезпечення

Для наочного уявлення поведінки прецедентів зручно застосовувати діаграму діяльності. Діаграма діяльності для прецеденту «Додати питання» (рисунок 3.10), діаграма діяльності для прецеденту «Пройти тест» (рисунок 3.11).



Рисунок 3.10. Діаграма діяльності прецеденту «Додати питання»



Рисунок 3.11. Діаграма діяльності прецеденту «Пройти тест»

Для відображення взаємодії між виділеними класами на етапі побудови моделі аналізу розглянемо діаграму послідовностей для прецеденту «Додати питання» (рис 3.12).

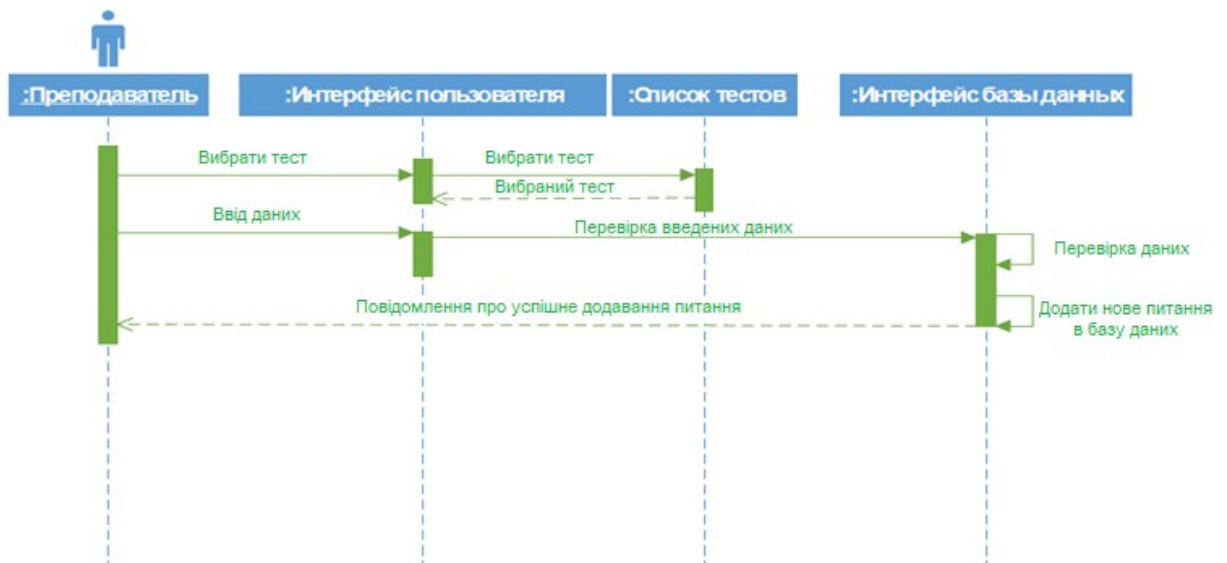


Рисунок 3.12. Діаграма послідовності прецеденту «Додати питання»

Діаграма послідовності допомагає встановити тимчасовий взаємозв'язок між класами, іншими словами вона лише показує реалізацію поведінки прецеденту, але не є точним поданням кожного його кроку. На діаграмі бере участь призначений для користувача інтерфейс, яким не займаються всерйоз аж до етапу проектування, тому на діаграмі послідовностей він міг бути опущений. В аналізі нас цікавить тільки поведінка класів аналізу.

В даному випадку розглядається успішний варіант розвитку сценарію, при якому викладач обрав тест, ввів зміст питання, можливі відповіді і вказав правильний з них. Система перевірила дані і за результатами перевірки додала нове питання в базу даних.

ВИСНОВКИ

Вважається, що екзаменаційний тест є не тільки більш об'єктивним методом оцінювання, а й більш щадним порівняно з усним іспитом.

Використання тестових завдань при оцінці знань і умінь має ряд переваг і забезпечує:

- об'єктивність, достовірність і однозначність - правильна відповідь на кожне тестове завдання заздалегідь визначена розробником і не залежить від суб'єктивного ставлення особи, яка здійснює перевірку;
- рівень вимоги до визначення рівня знань і умінь - використання завдань однакової складності, обсягу та змісту, єдиного критерію і норм оцінки;
- оперативність, скорочення витрат часу на проведення контрольних заходів - перевірка великого обсягу знань одночасно за відносно невеликий проміжок часу;
- повноту охоплення матеріалу, рівень освоєння якого планується визначити;
- можливість багаторазового повторення умов перевірки для з'ясування змін в рівні підготовки та засвоєнні навчального матеріалу.

Однак, складання тестових завдань вимагає високого рівня кваліфікації та досвіду розробника, часу і зусиль.

Тестування проводиться з метою поточного оцінювання засвоєння знань з окремої теми, розділу, предмета; самоконтролю; визначення рівня професійної підготовки. Структурною одиницею тесту є тестове завдання.

В результаті виконання курсової роботи розроблена керуюча програма тестування. Але не всі поставлені завдання були виконані. Кількість категорій користувачів була зменшена до двох: школяр, адміністратор (викладач) та не була реалізована така категорія користувачів як гість.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. <https://nevacrossfit.ru/uk/resheniya-po-cvetam/innovacionnye-tehnologii-v-pedagogike-i-obrazovanii-metodicheskii-doklad-na/>
2. Аванесов В.С. Наукові основи тестового контролю знань, 2014.
3. Аванесов В. С. Композиція тестових завдань, 2012.
4. Баранов Р. Д., Кошелева Т. М. Здійснення контролю знань студентів методом тестування з використанням технічних засобів.- <http://www.ineu.megalog.ru/konf/>.
5. Галеев, Д.Л. Храмов, А.П. Светлаков, О.В. Колосов. Адаптивного навчання и тестування. // Матеріали Всеросійської науково-методичної конференції «Розвиток методів и ЗАСОБІВ комп'ютерного адаптивного тестування», 2013.
6. Нейман Ю. М., Хлебніков В. А. Педагогічне тестування як вимір, 2017.
7. Кабанова Т. А., Новиков В. А. Тестування в сучасній освіті. Уч. Посібник. - М.: Вища школа, 2016.
8. Казиев В. М. Введення в практичне тестування. - М.: Інтуїт.ру, Біном. Лабораторія Знань, 2018.
9. <http://www.sintest.ru>
10. <http://opentest.com.ua>