

МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ
КРИВОРІЗЬКИЙ ФАХОВИЙ КОЛЕДЖ
ДЕРЖАВНОГО НЕКОМЕРЦІЙНОГО ПІДПРИЄМСТВА
«ДЕРЖАВНИЙ УНІВЕРСИТЕТ «КИЇВСЬКИЙ АВІАЦІЙНИЙ ІНСТИТУТ»
Циклова комісія комп'ютерних систем та мереж
(повна назва циклової комісії)

Допустити до захисту

Голова випускової циклової комісії
комп'ютерних систем та мереж

(повна назва циклової комісії)

Ірина КРАВЧУК

(ім'я, ПРІЗВИЩЕ)

« 10 » 06 2025 р.

КВАЛІФІКАЦІЙНА РОБОТА
(ПОЯСНЮВАЛЬНА ЗАПИСКА)
ВИПУСКНИКА ОСВІТНЬО-ПРОФЕСІЙНОГО СТУПЕНЯ
ФАХОВИЙ МОЛОДШИЙ БАКАЛАВР

Тема: Програмування додатку для статичного аналізу якості освіти
засобами C# та WPF

Група: 3-011 Спеціальність: 123 «Комп'ютерна інженерія»

Здобувач освіти

АМ

(підпис)

Артур НАТИКАНИЙ

(ім'я, ПРІЗВИЩЕ)

Керівник роботи

ММ

(підпис)

Роман МІНЕНКО

(ім'я, ПРІЗВИЩЕ)

Консультант з оформлення
пояснювальної записки

ОС

(підпис)

Оксана ОСАДЧА

(ім'я, ПРІЗВИЩЕ)

Кривий Ріг 2025 р.

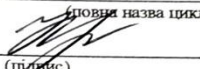
КРИВОРІЗЬКИЙ ФАХОВИЙ КОЛЕДЖ
ДЕРЖАВНОГО НЕКОМЕРЦІЙНОГО ПІДПРИЄМСТВА
«ДЕРЖАВНИЙ УНІВЕРСИТЕТ «КИЇВСЬКИЙ АВІАЦІЙНИЙ ІНСТИТУТ»

Відділення комп'ютерної та програмної інженерії
Циклова комісія комп'ютерних систем та мереж
Освітньо-професійний ступінь фаховий молодший бакалавр
Спеціальність 123 «Комп'ютерна інженерія»

ЗАТВЕРДЖУЮ

Голова випускової циклової комісії
комп'ютерних систем та мереж

(повна назва циклової комісії)


(підпис) Ірина КРАВЧУК
(ім'я, ПРІЗВИЩЕ)

« 01 » 03 2025 р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧУ ОСВІТИ

НАТИКАНОГО Артура Анатолійовича

(прізвище, ім'я, по батькові)

1. Тема роботи Програмування додатку для статичного аналізу якості освіти засобами C# та WPF

Керівник роботи Міненко Роман Вадимович, к.ф.-м.н, викладач

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

затверджені наказом по коледжу від « 04 » 04 2025 року № 50-ст

2. Строк подання здобувачем освіти роботи з _____ по _____

3. Вихідні дані до роботи Розробка практичного інструменту на C# та WPF для аналізу освітніх даних

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

Теоретичні основи аналізу якості освіти та технологій розробки ПЗ

Статистичний аналіз та можливості його автоматизації засобами ЕОМ

Розробка додатку для статичного аналізу якості освіти

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

Презентація Microsoft PowerPoint

6. Консультанти розділів роботи (проекту)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	<i>Узгодження технічного завдання з керівником кваліфікаційної роботи</i>	<i>04.04.2025-07.04.2025</i>	<i>виконано</i>
2	<i>Підбір та вивчення науково-технічної літератури за темою кваліфікаційної роботи</i>	<i>08.04.2025-14.04.2025</i>	<i>виконано</i>
3	<i>Теоретичні основи аналізу якості освіти та технологій розробки ПЗ</i>	<i>15.04.2025-21.04.2025</i>	<i>виконано</i>
4	<i>Статистичний аналіз та можливості його автоматизації засобами ЕОМ</i>	<i>22.04.2025-28.04.2025</i>	<i>виконано</i>
5	<i>Розробка додатку для статичного аналізу якості освіти</i>	<i>29.04.2025-23.05.2025</i>	<i>виконано</i>
6	<i>Написання та оформлення пояснювальної записки</i>	<i>26.05.2025-30.05.2025</i>	<i>виконано</i>
7	<i>Попередній захист кваліфікаційної роботи</i>	<i>09.06.2025-12.06.2025</i>	<i>виконано</i>
8	<i>Захист кваліфікаційної роботи</i>		

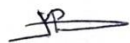
Здобувач освіти


(підпис)

Артур НАТИКАНИЙ

(ім'я, ПРІЗВИЩЕ)

Керівник роботи


(підпис)

Роман МІНЕНКО

(ім'я, ПРІЗВИЩЕ)



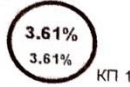
Звіт подібності

метадані

Назва організації
Ukrainian national aviation university
 Заголовок
Натиканий_Артур_Анатолійович_3-011_2025_123
 Автор Науковий керівник / Експерт
НатиканийМіненко Р
 підрозділ
Криворізький Фаховий коледж

Обсяг знайдених подібностей

Коефіцієнт подібності визначає, який відсоток тексту по відношенню до загального обсягу тексту було знайдено в різних джерелах. Зверніть увагу, що високі значення коефіцієнта не автоматично означають плагіат. Звіт має аналізувати компетентна / уповноважена особа.



25

Довжина фрази для коефіцієнта подібності 2

8254

Кількість слів

69899

Кількість символів

Тривога

У цьому розділі ви знайдете інформацію щодо текстових спотворень. Ці спотворення в тексті можуть говорити про МОЖЛИВІ маніпуляції в тексті. Спотворення в тексті можуть мати навмисний характер, але частіше характер технічних помилок при конвертації документа та його збереженні, тому ми рекомендуємо вам підходити до аналізу цього модуля відповідально. У разі виникнення запитань, просимо звертатися до нашої служби підтримки.

Заміна букв		1
Інтервали		0
Мікропробіли		0
Білі знаки		0
Парафрази (SmartMarks)		20

Подібності за списком джерел

Нижче наведений список джерел. В цьому списку є джерела із різних баз даних. Копір тексту означає в якому джерелі він був знайдений. Ці джерела і значення Коефіцієнту Подібності не відображають прямого плагіату. Необхідно відкрити кожне джерело і проаналізувати зміст і правильність оформлення джерела.

10 найдовших фраз

ПОРЯДКОВИЙ НОМЕР	НАЗВА ТА АДРЕСА ДЖЕРЕЛА URL (НАЗВА БАЗИ)	Копір тексту КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ)
1	https://financial.lnu.edu.ua/wp-content/uploads/2020/09/syllabus_Instrumenty-vymiruvannia-i-otsiniuvannia-SEP_2024_na-sayt.pdf	28 0.34 %
2	http://kk.nau.edu.ua/article/1183	27 0.33 %
3	http://kk.nau.edu.ua/article/1183	23 0.28 %
4	http://kk.nau.edu.ua/article/1183	16 0.19 %

РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи «Програмування додатку для статичного аналізу якості освіти засобами C# та WPF» містить 48 сторінок, 10 рисунків, 2 таблиці, 2 додатки, 15 використаних джерел.

СТАТИСТИЧНИЙ АНАЛІЗ, ЯКІСТЬ ОСВІТИ, АВТОМАТИЗАЦІЯ, C#, WPF, УСПІШНІСТЬ СТУДЕНТІВ, ОСВІТНЯ АНАЛІТИКА

Кваліфікаційна робота присвячена розробці програмного застосунку для статичного аналізу якості освіти з використанням мови програмування C# та технології *Windows Presentation Foundation (WPF)*.

Об'єктом дослідження є процес аналізу успішності студентів на основі табличних даних, а предметом — методика автоматизації статистичної обробки цих даних для освітніх потреб.

Метою роботи є створення ефективного та зручного програмного засобу, який дозволяє проводити обчислення середніх балів, формувати рейтинг студентів, виявляти проблемні дисципліни та студентів, а також виводити результати у наочному форматі.

У процесі дослідження були розглянуті основи статистичного аналізу в освітній сфері, обґрунтовано доцільність використання параметричних і непараметричних методів залежно від типу даних, а також досліджено можливості автоматизації таких процедур за допомогою ЕОМ. У третьому розділі детально описано процес створення програми, її структуру, інтерфейс користувача, а також результати тестування на базі тестових відомостей.

Розроблений застосунок продемонстрував високу ефективність у вирішенні прикладних задач аналізу навчальних даних. Його використання дозволяє спростити аналітичну роботу викладачів та адміністраторів, забезпечити візуальне представлення результатів і сформулювати підґрунтя для прийняття обґрунтованих рішень у сфері забезпечення якості освіти.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ.....	6
ВСТУП.....	7

РОЗДІЛ 1 ТЕОРЕТИЧНІ ОСНОВИ АНАЛІЗУ ЯКОСТІ ОСВІТИ ТА ТЕХНОЛОГІЙ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	9
1.1 Поняття та методи аналізу якості освіти	9
1.2 Системи збору та обробки статистичних освітніх даних	
11 1.3 Огляд та класифікація програмних засобів аналізу якості освіти	14
1.4 Висновок до першого розділу	18
РОЗДІЛ 2 СТАТИСТИЧНИЙ АНАЛІЗ ТА МОЖЛИВОСТІ ЙОГО АВТОМАТИЗАЦІЇ ЗАСОБАМИ ЕОМ	20
2.1 Вступ.....	20 2.2
Основні статистичні методи в дослідженнях	20 2.3
Умови вибору статистичних методів	21 2.4
Параметричні та непараметричні тести.....	22 2.5
Автоматизація статистичного аналізу.....	23 2.6
Типові помилки при виборі статистичного методу.....	23 2.7
Висновок до другого розділу.....	23
РОЗДІЛ 3 РОЗРОБКА ДОДАТКУ ДЛЯ СТАТИЧНОГО АНАЛІЗУ ЯКОСТІ ОСВІТИ	25
3.1 Вступ.....	25 3.2
Розробка структура та інтерфейсу додатку.....	26 3.3
Розробка функціональних модулів на мові C#	29 3.4
Розробка інтерфейсу користувача мовою XAML	32 3.5
Тестування проекту.....	35 3.6
Висновки до третього розділу	41
ВИСНОВКИ.....	43
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ	46
ДОДАТОК А.....	48
ДОДАТОК Б	55

ПЕРЕЛІК СКОРОЧЕНЬ

API - application programming interface

NLP - Natural Language Processing

SaaS - Software as a Service

SWOT - Strengths, Weaknesses, Opportunities, Threats

UIS - Institute for Statistics

WPF - Windows Presentation Foundation

XAML - eXtensible Application Markup Language

АСУ - автоматизована система управління

БД - база даних

ВНЗ - вищий навчальний заклад

ЄВІ - єдиний вступний іспит

ЄДЕБО - Єдина державна електронна база з питань освіти

ЄФВВ - єдине фахове вступне випробування

ЗНО - зовнішнє незалежне оцінювання

УЦОЯО - Український центр оцінювання якості освіти

7

ВСТУП

Прийняття рішень на основі даних є фундаментальною складовою інженерної діяльності. У цьому контексті статистика відіграє ключову роль, забезпечуючи кількісну основу для аналізу, оцінки та вдосконалення освітніх програм. Вона допомагає не лише в оцінюванні успішності студентів, а й у визначенні ефективності навчальних стратегій, якості програм, а також у процесах акредитації.

Значна частина процесів у вищій освіті, зокрема оцінювання курсових робіт та академічної успішності, базується на статистичному аналізі. Завдяки статистиці можна виявити прогалини в навчальних підходах, оцінити вплив змін у навчальній програмі та запропонувати обґрунтовані покращення. Наприклад, описова статистика дозволяє узагальнювати дані за допомогою таких показників, як середнє значення, медіана, мода, дисперсія та стандартне відхилення.

Статистика також необхідна для відповідності вимогам акредитаційних організацій, які часто вимагають аналізу показників, таких як співвідношення студентів і викладачів, рівень задоволеності навчанням, та показники працевлаштування випускників.

Незважаючи на її важливість, багато студентів інженерних спеціальностей сприймають курси зі статистики як складні, нецікаві та відірвані від практики. Часто матеріал подається у вигляді прикладів з підкиданням монет або киданням гральних кубиків, які, на думку студентів, не мають нічого спільного з реальними інженерними проблемами. Такий підхід створює розрив між теорією та практикою, ускладнюючи засвоєння матеріалу.

Проте ситуація змінюється. Сучасні викладачі все частіше використовують реальні дані, моделювання та приклади з інженерної практики, що дозволяє зробити навчання більш прикладним і релевантним. Це відповідає потребам галузі, де зростає попит на фахівців, здатних приймати обґрунтовані рішення на основі даних.

8

Одним з потужних методів статистичного аналізу, що широко застосовується у вищій освіті, є дисперсійний аналіз (*ANOVA*). Цей метод дозволяє порівнювати середні значення трьох і більше груп та визначати, чи є спостережувані відмінності між ними статистично значущими. *ANOVA* особливо корисна при оцінці успішності студентів різних спеціальностей, програм або курсів.

У межах цього дослідження застосовано *ANOVA* для аналізу екзаменаційних оцінок здобувачів освіти Криворізького фахового коледжу Державного некомерційного підприємства «Державний університет «Київський авіаційний інститут» (КРФК КАІ). Метою є виявлення статистично значущих відмінностей між середніми оцінками, що дозволить оцінити ефективність викладання та визначити напрями для покращення освітнього процесу.

9

РОЗДІЛ 1

ТЕОРЕТИЧНІ ОСНОВИ АНАЛІЗУ ЯКОСТІ ОСВІТИ ТА ТЕХНОЛОГІЙ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

1.1 Поняття та методи аналізу якості освіти

Якість освіти — це багатовимірне поняття, яке охоплює сукупність характеристик освітнього процесу, результатів навчання, інституційного середовища, ресурсного забезпечення та відповідності очікуванням стейкхолдерів (учнів, викладачів, роботодавців, держави). У широкому розумінні якість освіти відображає ступінь відповідності освітньої діяльності та її результатів встановленим стандартам, нормативам і соціальним очікуванням.

В умовах інформатизації суспільства та переходу до цифрових технологій аналіз якості освіти стає критично важливим. Він дозволяє не лише ідентифікувати проблеми в освітній системі, а й виявляти шляхи її вдосконалення. Водночас традиційні методи оцінювання, зосереджені на кількісних показниках, все частіше доповнюються інтелектуальними технологіями обробки даних, що відкриває нові можливості для об'єктивного та глибокого аналізу [1].

У науковій та практичній літературі існує кілька підходів до трактування якості освіти:

- нормативний підхід — орієнтується на відповідність результатів освітньої діяльності державним стандартам, програмам і критеріям акредитації;
- споживацький підхід — акцентує увагу на задоволенні потреб і очікувань споживачів освітніх послуг, зокрема студентів, батьків, роботодавців;
- педагогічний підхід — фокусується на відповідності процесу навчання педагогічним принципам, методам, умовам розвитку особистості;
- системний підхід — розглядає якість освіти як властивість усієї освітньої системи, яка формується взаємодією різних її елементів (кадрового, методичного, матеріально-технічного, управлінського забезпечення).

10

Зазвичай в освітньому менеджменті використовуються елементи всіх підходів — для створення цілісного уявлення про якість освітніх послуг. Аналіз якості освіти здійснюється за допомогою як кількісних, так і якісних методів. Основні серед них [2]:

- анкетування і опитування — дозволяють зібрати думки студентів, викладачів та роботодавців щодо якості навчального процесу, змісту дисциплін, організації освітнього середовища;

- контент-аналіз — використовується для оцінки змісту навчальних програм, методичних матеріалів, дипломних робіт тощо;

- статистичний аналіз — застосовується для обробки даних про успішність студентів, результати ЗНО/ЄВІ працевлаштування випускників, рівень відвідуваності, тощо;

- *SWOT*-аналіз — дозволяє виявити сильні та слабкі сторони навчального закладу, а також зовнішні можливості та загрози для підвищення якості;

- *Benchmarking* (еталонування) — метод порівняння показників свого закладу з найкращими практиками або результатами інших установ освіти. - методи експертної оцінки — залучення кваліфікованих експертів для оцінки програм, результатів, компетентностей тощо.

Крім того, останнім часом все більшого поширення набувають інтелектуальні методи аналізу — на основі машинного навчання, аналізу великих даних (*Big Data*), обробки природної мови (*NLP, Natural Language Processing*) та кластеризації. Вони дозволяють виявляти приховані залежності, прогнозувати ризики неуспішності студентів, аналізувати великі обсяги текстових і числових даних у реальному часі [2].

Для кількісного оцінювання якості освіти використовують такі індикатори:

- рівень академічної успішності студентів;
- процент працевлаштованих випускників за спеціальністю; - частка викладачів із науковими ступеннями;

11

- оцінка якості викладання за результатами анкетувань;
- кількість наукових публікацій та участь у міжнародних проектах; - матеріально-технічне забезпечення (наявність лабораторій, доступу до сучасного ПЗ, тощо);
- відповідність програм сучасним вимогам ринку праці.

Ці показники використовуються для внутрішнього контролю якості (*self assessment*) і зовнішнього оцінювання (акредитація, незалежний аудит). Попри

значну кількість методик, проблема об'єктивного аналізу якості освіти залишається актуальною. Серед основних викликів:

- фрагментарність даних — освітні установи часто не мають єдиної системи збору та обробки освітньої аналітики.
- суб'єктивність оцінок — у багатьох випадках результати опитувань або експертних оцінок не мають достатньої валідності.
- низька інтеграція ІТ-рішень — застосування програмних засобів для аналізу є обмеженим і не стандартизованим.

Розв'язання цих проблем можливе шляхом розробки спеціалізованих додатків, що дозволяють автоматизувати процес збору та аналізу даних про якість освіти. Саме таке завдання ставить перед собою розробка програмного засобу статичного аналізу, опис якого наведено у наступних розділах.

1.2 Системи збору та обробки статистичних освітніх даних

Системи збору та обробки статистичних даних є фундаментальним компонентом процесу аналізу якості освіти. Вони забезпечують збирання, впорядкування, зберігання, аналіз і візуалізацію інформації, що характеризує різні аспекти функціонування освітньої системи: академічну успішність студентів, роботу викладачів, результати акредитацій, рівень працевлаштування випускників, ресурсне забезпечення закладів освіти тощо.

12

У контексті цифровізації освіти ці системи перетворюються на інтегровані інформаційні середовища, які підтримують прийняття обґрунтованих управлінських рішень на основі достовірних та актуальних даних.

Збір статистичної інформації може здійснюватися як на рівні окремих закладів освіти, так і на національному чи міжнародному рівні. Основні джерела таких даних включають:

- електронні журнали успішності студентів;
- автоматизовані системи управління закладом освіти (АСУ) —

наприклад, «Деканат» [3], «Є-кампус», *Moodle* [4], *Google Workspace for Education* [5];

- державні статистичні звіти — форми звітності для МОН, Державної служби якості освіти тощо;
- платформи зовнішнього оцінювання знань — ЗНО, ЄВІ, ЄФВВ; - анкетування студентів, викладачів, випускників і роботодавців; - національні освітні реєстри — ЄДЕБО (Єдина державна електронна база з питань освіти), УЦОЯО, база академічної доброчесності. У межах цих джерел накопичується значний обсяг як структурованих (табличних), так і неструктурованих (текстових, графічних) даних, які можуть бути використані для різних видів аналітики.

Типова система збору та аналізу освітніх статистичних даних складається з кількох функціональних компонентів:

- підсистема введення/збору даних. Дані можуть вводитися вручну або імпортуватися з інших джерел за допомогою *API*, форм обміну (*CSV*, *XML*, *JSON*) або через вебінтерфейси;
- база даних (БД). Всі зібрані дані зберігаються у централізованій або розподіленій БД. Найчастіше використовуються реляційні СУБД (*MySQL*, *PostgreSQL*, *MS SQL Server*), хоча для великих обсягів інформації можуть застосовуватись *NoSQL*-рішення (*MongoDB*, *Cassandra*);

13

- модулі попередньої обробки. Вони забезпечують очищення, нормалізацію, валідацію даних (видалення дублікатів, перевірку повноти та коректності записів);
- аналітичний модуль. Тут реалізуються методи статистичного аналізу, побудова зведених таблиць, трендовий аналіз, кореляції, кластеризація, а також простіші алгоритми машинного навчання;
- інтерфейс користувача. Зазвичай включає інформаційні панелі (дашборди), графіки, діаграми, звітні форми. Інтерфейси можуть бути реалізовані як десктопні, так і веб-застосунки, доступні через браузер.

У світовій практиці реалізовано багато інформаційно-аналітичних

систем, які підтримують освітню статистику:

- *UNESCO UIS (Institute for Statistics)* — збирає глобальні освітні дані для міжнародних порівнянь;

- *OECD Education at a Glance* — платформа для міжнародного порівняння показників якості освіти;

- національна платформа відкритих даних про вищу освіту в Україні — містить статистику за університетами, ліцензованими спеціальностями, рейтингами;

- аналітичні модулі ЄДЕБО — забезпечують державний контроль за динамікою студентських потоків, розподілом бюджетних місць, ступенем завантаження викладачів.

На рівні окремих університетів часто створюються локальні аналітичні системи, які адаптовані до конкретної структури даних та потреб адміністрації. З метою обробки освітньої статистики все ширше застосовуються наступні програмні рішення:

- *MS Excel, LibreOffice Calc* — базові засоби для ручного аналізу та побудови діаграм;

- *Power BI, Tableau, Google Data Studio* — професійні *BI*-інструменти для інтерактивної візуалізації;

14

- *R, Python (pandas, matplotlib, seaborn)* — мови програмування для гнучкого аналізу та створення кастомних звітів;

- *SPSS, SAS, STATA* — статистичні пакети для наукових досліджень; - *C# + WPF* або *.NET MAUI* — засоби для створення десктопних програм з підтримкою візуалізації та баз даних.

У межах цієї кваліфікаційної роботи планується розробка власного програмного засобу на *C#* з використанням технології *WPF*, що реалізує функції статичного аналізу освітньої якості на основі введених даних.

Незважаючи на технічний прогрес, у сфері освітньої статистики спостерігаються такі проблеми:

- фрагментарність систем — багато університетів та коледжів

використовують несумісні між собою рішення;

- низька якість введених даних — помилки при заповненні, застаріла інформація;

- відсутність автоматизації — значна частина аналізу здійснюється вручну або в *Excel*;

- недостатній рівень захисту персональних даних — порушення принципів *GDPR* або українського законодавства.

Для подолання цих викликів необхідна стандартизація форматів даних, підвищення цифрової грамотності працівників, та розробка інструментів з елементами інтелектуального аналізу, які будуть доступні не лише аналітикам, а й широкому колу користувачів.

1.3 Огляд та класифікація програмних засобів аналізу якості освіти

У сучасних умовах цифрової трансформації освіти зростає роль програмних засобів у забезпеченні якості освітніх процесів. Зокрема, вони дозволяють здійснювати моніторинг, виявляти проблемні зони, формувати аналітичні звіти для прийняття рішень на різних рівнях управління — від викладача до Міністерства освіти. Програмні засоби для аналізу якості освіти

15

охоплюють широкий спектр функцій — від збору та візуалізації даних до побудови прогнозних моделей на основі штучного інтелекту.

Для систематизації різноманітних програмних засобів доцільно використовувати наступні класифікаційні критерії:

- за рівнем застосування: локальні — використовуються на рівні окремих навчальних курсів, кафедр або викладачів; інституційні — охоплюють весь навчальний заклад; національні/міжнародні — розраховані на використання освітніми установами всієї країни або у глобальному масштабі;

- за функціональним призначенням: збір і обробка статистичних даних; моніторинг і аналітика освітніх процесів; управління якістю (включаючи підтримку процесів акредитації); прогнозування та виявлення ризиків;

оцінювання та зворотний зв'язок;

- за способом реалізації: настільні (десктопні) програми; веборієнтовані рішення (*SaaS*); мобільні додатки; інтегровані освітні платформи; - за технологічною основою (табл. 3.1): програмні продукти загального призначення (*Excel, R, Python*); спеціалізовані системи аналізу освітніх даних (наприклад, *Power BI*-аналітика в *Moodle*); індивідуально розроблені рішення під конкретну установу.

Огляд найбільш поширених програмних рішень:

1. *MS Excel / Google Sheets*. Найпоширеніші серед непрофесійних користувачів інструменти. Дозволяють збирати та агрегувати дані, будувати діаграми, застосовувати формули й фільтри. Мають обмежені можливості масштабування, проте залишаються популярними для базового аналізу успішності студентів, ведення обліку відвідуваності, моніторингу динаміки академічних показників.

2. *Power BI (Microsoft)* [6]. Сучасний *BI*-інструмент для інтерактивної візуалізації даних. Може інтегруватися з базами даних, *Excel*, сервісами *SharePoint*, хмарними платформами. У контексті освіти дозволяє створювати дашборди успішності, аналізувати навантаження викладачів, співвідносити дані

16

різних джерел. Підтримує авторизацію, рівні доступу, автоматичне оновлення даних.

3. *Tableau*. Аналог *Power BI* із потужними засобами візуального аналізу. Популярний серед університетів США, Канади, країн ЄС. Має широкий набір графіків, можливість інтерактивного дослідження тенденцій і шаблони для освітньої аналітики.

4. *SPSS, SAS, R, Python*. Це інструменти для складного статистичного аналізу. Наприклад, у *R* та *Python* можна реалізувати обробку масивів освітніх даних, аналіз кореляцій, регресій, кластеризацію, обчислення індексу ефективності викладання. Вони використовуються в академічних дослідженнях та дипломних роботах, хоча вимагають глибших знань статистики та програмування.

5. *Moodle* з аналітичними плагінами (*Learning Analytics*). Платформа *Moodle* дозволяє інтегрувати модулі освітньої аналітики, які збирають дані про активність студентів: відвідування курсів, проходження тестів, спілкування на форумах. Ці дані аналізуються з метою виявлення груп ризику, прогнозування відсіву, вдосконалення педагогічного дизайну.

6. *OpenEdu Analytics* (Україна) [7]. Проєкт на базі платформи відкритих освітніх даних, підтримується МОН. Надає інформацію про заклади освіти, кількість студентів, кадрове забезпечення. Дозволяє порівнювати ВНЗ за рядом показників, підтримує відкрите *API* для інтеграції з іншими системами.

7. Індивідуальні програмні рішення. Деякі навчальні заклади розробляють власні інструменти аналізу якості освіти, що враховують специфіку внутрішньої структури та потреб. Найчастіше це *C#/.NET*-застосунки, вебплатформи на *PHP* або *Python*, які інтегруються з внутрішніми базами (наприклад, АСУЗ) і дають змогу будувати звіти на рівні деканатів, факультетів, кафедр.

У даній кваліфікаційній роботі розробка буде реалізована за допомогою мови програмування *C#* та платформи *WPF*. Це дозволить створити зручний графічний інтерфейс, з інтерактивним введенням даних, збереженням у локальну БД або файл, побудовою графіків і можливістю формування звітів.

17

Таблиця 1.1 – Порівняльна таблиця популярних програмних засобів

Засіб	Складність освоєння	Можливість візуалізації	Підтримка БД	Підтримка AI	Застосування
<i>Excel</i>	Низька	Базова	Ні	Ні	Оцінка успішності
<i>Power BI</i>	Середня	Висока	Так	Частково	Аналітика на рівні ВНЗ

<i>R/Python</i>	Висока	Гнучка	Так	Так	Наукові дослідження
<i>Moodle</i>	Середня	Середня	Так	Так (плагіни)	Аналіз е-курсів
<i>C#/WPF</i>	Середня	Висока	Так	Залежно від реалізації	Кастомні рішення

Такий підхід забезпечить автономність системи та простоту використання без потреби у доступі до хмари чи зовнішніх сервісів.

На ринку освітньої аналітики спостерігається поступовий перехід від ручної обробки даних до інтелектуальних систем підтримки прийняття рішень (*DSS*), які включають модулі машинного навчання, адаптивного навчання, когнітивного аналізу.

Такі системи можуть:

- автоматично виявляти аномалії в навчальному процесі;
- прогнозувати успішність окремих студентів;
- рекомендувати педагогічні стратегії для викладачів;
- виявляти неефективні курси або методики.

Окремим трендом є відкритість та прозорість освітньої аналітики, з можливістю публічного доступу до агрегованих даних (у рамках дотримання вимог щодо конфіденційності).

18

1.4 Висновок до першого розділу

У першому розділі кваліфікаційної роботи було проведено теоретичний огляд проблеми оцінювання якості освіти в контексті цифровізації та розвитку інформаційних технологій. Аналіз засвідчив, що якість освіти є багатограним поняттям, яке охоплює не лише академічні досягнення здобувачів освіти, але й ефективність організаційних процесів, рівень педагогічної взаємодії,

відповідність результатів навчання вимогам ринку праці та соціальним запитам. З метою системного вимірювання та аналізу якості освіти застосовується ціла низка методів: статистичні, соціологічні, психометричні, порівняльні, а також експертні.

Зібрані дані є основою для об'єктивного оцінювання процесів, що відбуваються в освітньому середовищі. У п.п 1.2 розглянуто джерела освітньої статистики, включно з національними системами збору даних, такими як ЄДЕБО в Україні, а також глобальними платформами. Збір даних відбувається на різних рівнях — інституційному, регіональному та міжнародному — із застосуванням як традиційних методів (анкети, звітність), так і автоматизованих цифрових систем (електронні журнали, освітні платформи, *LMS*, тощо). Особливу увагу було приділено важливості забезпечення цілісності та достовірності освітніх даних, а також дотриманню принципів конфіденційності.

У п.п. 1.3 було проведено детальний огляд програмних засобів, що використовуються для аналізу якості освіти. Системи поділяються за масштабом застосування, функціональністю, способом реалізації та технологічною основою. Серед найбільш поширених інструментів було розглянуто *Excel*, *Power BI*, *R*, *Python*, *Moodle* з аналітичними модулями, а також індивідуальні кастомні розробки на *C#* або інших мовах. Було встановлено, що потужні аналітичні функції, інтеграція з БД, гнучка візуалізація, а також можливості прогнозування роблять програмні рішення ключовим інструментом у підвищенні якості управління освітнім процесом.

19

У підсумку, аналіз існуючих методів і програмних засобів підтвердив актуальність створення спеціалізованого додатку для підтримки аналітики якості освіти. Особливої ваги набувають інструменти, які дозволяють не лише відображати поточний стан справ, але й виявляти тенденції, прогнозувати результати та формувати стратегічні рекомендації для удосконалення освітнього середовища.

20

РОЗДІЛ 2

СТАТИСТИЧНИЙ АНАЛІЗ ТА МОЖЛИВОСТІ ЙОГО АВТОМАТИЗАЦІЇ ЗАСОБАМИ ЕОМ

2.1 Вступ

У сучасних умовах інформатизації суспільства статистичні методи є одним з основних інструментів аналізу результатів наукових досліджень, у тому числі в освітній сфері. Статистичний аналіз включає планування дослідження, збирання даних, їх обробку, інтерпретацію результатів та побудову обґрунтованих висновків. Автоматизація цього процесу з використанням ЕОМ значно підвищує точність обчислень, ефективність аналізу та швидкість прийняття управлінських рішень.

Цей розділ базується на аналізі ключових понять зі сфери прикладної статистики, розглядає відмінності між описовими та інференційними методами, а також обґрунтовує переваги автоматизації статистичних операцій у програмному забезпеченні. Подано адаптований зміст із сучасних публікацій, що дозволяє розкрити питання застосування параметричних і непараметричних методів у дослідженнях.

2.2 Основні статистичні методи в дослідженнях

Статистичні методи можна умовно поділити на дві великі категорії: описова статистика та інференційна статистика. Описова статистика дозволяє узагальнити характеристики вибірки, використовуючи такі показники, як середнє значення, медіана, мода, дисперсія та стандартне відхилення. Ці методи не дають змоги робити висновки про генеральну сукупність, але забезпечують базовий огляд отриманих даних.

Інференційна статистика навпаки, дозволяє робити узагальнення на основі вибірки щодо всієї сукупності. Вона включає перевірку гіпотез, оцінку

параметрів, порівняння груп. Застосовуються такі методи як *t*-тест (парний, непарний), *F*-тест (*ANOVA*), регресійний аналіз, кореляція тощо. Вибір методу залежить від типу змінної, розподілу даних та дослідницької мети.

Параметричні методи використовуються тоді, коли дані мають нормальний розподіл і вимірюються на інтервальній або відношеній шкалі. Вони дозволяють аналізувати середні значення. При порушенні умов нормальності доцільніше використовувати непараметричні методи, що базуються на рангах або медіанах, наприклад, критерій Манна-Уїтні, Вілкоксона, Крускала-Уолліса. Непараметричні методи також застосовуються для категоріальних або порядкових змінних.

2.3 Умови вибору статистичних методів

Вибір конкретного статистичного методу залежить від кількох ключових чинників:

- Мета дослідження — для порівняння, оцінки зв'язків або прогнозування застосовуються різні підходи. Наприклад, для оцінки ефективності навчальної програми доцільно застосовувати парний *t*-тест до даних «до» та «після»;

- тип змінних — числові змінні поділяються на дискретні (кількість студентів) та неперервні (середній бал), тоді як категоріальні можуть бути номінальними (стать) або порядковими (рівень задоволеності);

- кількість груп — для однієї групи з двома замірами застосовується парний аналіз, а при порівнянні трьох і більше груп — багатofакторні методи, наприклад *ANOVA* чи її непараметричні аналоги;

- тип вибірки: парна чи непарна — якщо порівнюються однакові респонденти в різний час, застосовуються парні методи; якщо різні — непарні; - розподіл даних — при нормальному розподілі перевага надається параметричним тестам; у протилежному випадку — непараметричним.

2.4 Параметричні та непараметричні тести

Параметричні методи засновані на припущенні про наявність нормального розподілу в генеральній сукупності. До них належать: - *t*-тести (одно- та двозразкові);

- *F*-тест (однофакторний або багатофакторний *ANOVA*);
- пірсонівська кореляція;
- лінійна регресія.

Альтернативні непараметричні тести:

- знак-тест або Вілкоксона (аналог 1-зразкового *t*-тесту);
- тест Манна-Уїтні (аналог 2-зразкового *t*-тесту);
- тест Крускала-Уолліса (аналог *ANOVA*);
- спірманівська кореляція;
- нелінійна регресія або логістична регресія.

У табл. 2.1 наведено відповідності між параметричними та непараметричними методами:

Таблиця 2.1 – Відповідності між параметричними та непараметричними методами

Завдання	Параметричний тест	Альтернатива (непараметричний)
Порівняння 2 вибірок (незалежні)	<i>t</i> -тест	Манн-Уїтні
Порівняння 2 вибірок (парні)	парний <i>t</i> -тест	тест Вілкоксона
Порівняння > 2 вибірок	<i>ANOVA</i>	Крускала-Уолліса / Фрідмана
Кореляція	Пірсон	Спірман
Регресія	Лінійна регресія	Нелінійна / логістична регресія

Сучасні обчислювальні засоби значно спрощують роботу зі статистичними даними. Програмне забезпечення дозволяє:

- швидко імпортувати таблиці даних з *Excel*, *CSV* або баз даних; -
- виконувати обчислення без ручного введення формул;
- генерувати графіки, діаграми та звіти;
- мінімізувати вплив людського фактору;
- забезпечити інтерактивний інтерфейс для користувача без статистичної підготовки.

Популярні системи — *SPSS*, *SAS*, *R*, *Python* (*pandas*, *seaborn*, *scikit-learn*), а також прикладні програми на *C#* та *WPF*, як розроблено в межах даної кваліфікаційної роботи.

2.6 Типові помилки при виборі статистичного методу

Неправильне застосування статистичних методів призводить до помилкових висновків. Наприклад, використання парного *t*-тесту замість незалежного призводить до заниження *p*-значення та помилкової інтерпретації результатів як статистично значущих. Для уникнення таких помилок дослідник повинен володіти хоча б базовими знаннями в галузі статистики або залучати консультантів на етапах планування дослідження та аналізу даних.

2.7 Висновок до другого розділу

У результаті аналізу теоретичних положень статистики та практичних підходів до обробки даних було з'ясовано, що правильний вибір методу аналізу залежить від багатьох чинників: структури даних, їх розподілу, цілей дослідження, типу змінних та формату спостережень. Автоматизація аналізу за допомогою ЕОМ — ефективний інструмент у руках дослідника, що дозволяє

уникнути помилок, прискорити обробку, візуалізувати результати та зробити дослідження об'єктивним. У наступному розділі буде представлено реалізацію

РОЗДІЛ 3

РОЗРОБКА ДОДАТКУ ДЛЯ СТАТИЧНОГО АНАЛІЗУ ЯКОСТІ

ОСВІТИ 3.1 Вступ

У сучасному програмуванні вибір мови та середовища розробки відіграє ключову роль у забезпеченні ефективності, масштабованості та зручності використання програмного забезпечення. Для реалізації програмного продукту, що виконує аналіз якості освіти на основі табличних даних, у даній роботі було обрано мову програмування C# у поєднанні з технологією WPF (*Windows Presentation Foundation*). Такий вибір зумовлений низкою переваг, які надає ця платформа для створення зручного, функціонального та надійного користувацького інтерфейсу.

Мова програмування C# [8] — це сучасна, об'єктно-орієнтована мова, що розроблена компанією *Microsoft* і є основною для створення застосунків під .NET-платформу. Вона поєднує в собі високу продуктивність, строгість синтаксису та багатий набір бібліотек. C# забезпечує чудову інтеграцію з *Windows API*, зручну роботу з файлами, таблицями, базами даних і має широкий спектр засобів для обробки подій, роботи з формами, колекціями та мережевими протоколами. Завдяки цьому C# є ідеальним інструментом для розробки прикладних десктопних додатків.

WPF, у свою чергу, є потужною платформою для побудови графічного інтерфейсу користувача. Вона підтримує декларативний підхід до опису інтерфейсу через мову *eXtensible Application Markup Language (XAML)* [9], що дозволяє відокремити логіку програми від її візуального представлення. WPF надає широкі можливості для стилізації та візуалізації елементів, а також підтримує механізм прив'язки даних (*data binding*), шаблонів, анімацій та роботи з подіями. Це значно спрощує створення адаптивного та динамічного інтерфейсу користувача.

Серед інших переваг *C#* і *WPF* можна виділити:

- швидку розробку за рахунок інструментів *Visual Studio*;
- активну спільноту розробників і наявність великої кількості прикладів та документації;
- підтримку сторонніх бібліотек для роботи з *Excel*, графікою, побудовою діаграм тощо.

У контексті даної кваліфікаційної роботи вибір *C#* та *WPF* дозволив реалізувати інтуїтивно зрозумілий інтерфейс для завантаження табличних даних, їх обробки, виводу звітів та візуалізації результатів аналізу, що в сукупності забезпечує зручність користування та розширюваність системи.

3.2 Розробка структура та інтерфейсу додатку

У межах кваліфікаційної роботи буде реалізовано програмне забезпечення для аналізу якості освіти, що використовує вхідні дані у вигляді *Excel*-файлу з оцінками студентів. Основна мета програми — обчислення статистичних показників (середні бали студентів і дисциплін), формування рейтингу студентів, виявлення проблемних учнів та дисциплін, а також візуалізація результатів аналізу у зручній формі. Програма буде реалізована мовою *C#* з використанням технології *WPF*, яка дозволяє створити гнучкий і адаптивний графічний інтерфейс користувача.

Блок схема роботи програми обробки статистичних даних приведена на рисунку 3.1.

Програму умовно можна поділити на кілька основних функціональних модулів:

- модуль завантаження та обробки даних. Він відповідає за імпорт даних з *Excel*-файлу у внутрішню структуру програми. Дані автоматично зчитуються з першого листа файлу. Перша колонка містить прізвища студентів, інші колонки — оцінки за відповідними дисциплінами. Для зчитування

використовується бібліотека *ClosedXML*, яка забезпечує простий і безпечний

доступ до структурованих табличних даних.

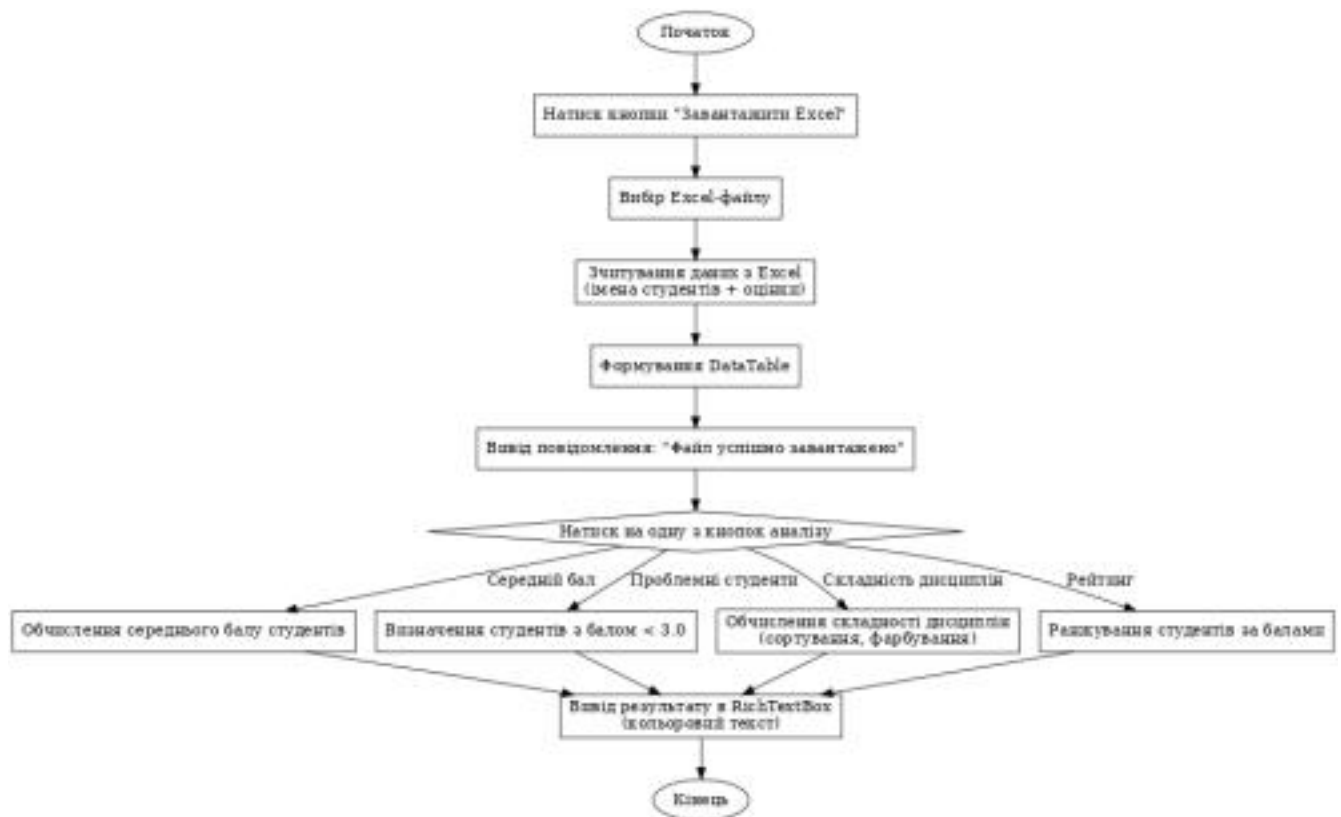


Рисунок 3.1 – Блок схема модуля обробки статистичних даних

- модуль розрахунків: Після завантаження даних програма виконує обчислення: середнього балу для кожного студента; середнього балу для кожної дисципліни; формує рейтинг студентів за успішністю; виділяє проблемних студентів (середній бал < 3.0); сортує дисципліни за складністю (середній бал по дисципліні зростає).

- модуль візуалізації. Обчислені результати виводяться у вигляді таблиць та кольорових текстових блоків в інтерфейсі користувача. Використовується елемент *RichTextBox*, який дозволяє формувати текст із використанням різних кольорів, що особливо зручно для позначення ступеня складності дисциплін: червоний колір — дисципліни з найменшим середнім балом (нижня третина); жовтий — середні за складністю; зелений — дисципліни з найвищими середніми балами (верхня третина).

- інтерфейс користувача. Інтерфейс програми має бути спроектовано таким чином, щоб забезпечити простоту у використанні навіть для користувачів без спеціальної технічної підготовки. Основне вікно програми має містити такі

елементи: кнопка «Завантажити *Excel*» (відкриває вікно вибору файлу, після чого розпочинається процес обробки); блок виводу результатів (середній бал кожного студента; список студентів із середнім балом менше 3.0 (проблемні студенти));

середній бал для кожної дисципліни; дисципліни, розфарбовані за рівнем складності);

- додаткові повідомлення: мають виводитись підказки про успішність завантаження даних, кількість студентів, кількість дисциплін, тощо. Інтерфейс має бути побудований за допомогою *XAML*-розмітки, що дозволить розділити візуальну частину від логіки обробки даних. Усі основні події, як-от натискання кнопок або обробка винятків, мають бути реалізовані у кодї *C#*.

- алгоритмічна логіка. У процесі обчислень мають використовуватися стандартні колекції типу *List<double>* та словники *Dictionary<string, double>* для зберігання імен студентів, назв дисциплін та відповідних середніх балів. Після заповнення колекцій має виконуватися сортування проблемних дисциплін та студентів за допомогою *LINQ*-запитів. Колірна схема візуалізації дисциплін має реалізовуватися динамічно — залежно від позиції в упорядкованому списку, дисципліни мають розподілятися на три рівні складності.

Заплановані результати запропонованого підходу:

- масштабованість: за потреби легко можна додати нові функції (експорт звітів, інтеграція з базами даних тощо);

- гнучка візуалізація: підтримка кольорів, шрифтів, абзаців дозволяє адаптувати подання інформації під потреби користувача;

- надійність: програма має обробку винятків, перевірку на помилки у вхідному файлі;

- легка підтримка: чітка структурованість коду спрощує модифікації та обслуговування.

У результаті, реалізована програма має поєднувати в собі зручність, функціональність і візуальну інформативність. Вона може бути використана як аналітичний інструмент у закладах освіти для моніторингу успішності

студентів, а також для прийняття рішень щодо вдосконалення навчального процесу.

3.3 Розробка функціональних модулів на мові C#

Програмне забезпечення, наведене в даному кодї (див. нижче «Код програми на мові C#»), реалізує інструмент для аналізу якості освіти на основі табличних даних, отриманих з *Excel*-файлу.

Код програми на мові C# приведено у додатку А.

Воно розроблене за допомогою мови програмування C# з використанням фреймворку *WPF*. Для кращого розуміння логіки функціонування програми, розглянемо її основні функціональні компоненти.

3.3.1 Ініціалізація та налаштування середовища

У конструкторі класу *MainWindow* викликається метод *InitializeComponent()*, що ініціалізує компоненти інтерфейсу. Також встановлюється контекст ліцензування для бібліотеки *EPPlus*, яка використовується для обробки *Excel*-файлів:

ExcelPackage.LicenseContext = LicenseContext.NonCommercial; Цей фрагмент забезпечує коректне завантаження файлів з *Excel* у некомерційних цілях.

3.3.2 Завантаження *Excel*-файлу

Метод *LoadExcel_Click* спрацьовує при натисканні на відповідну кнопку. Відкривається діалог вибору файлу, після чого дані зчитуються з першого аркуша *Excel*-файлу:

- перший рядок сприймається як заголовки колонок (імена дисциплін);
- інші рядки — як записи студентів (ім'я + оцінки).

Дані зчитуються у *DataTable*, яка прив'язується до елемента *DataGridStudents* для візуалізації таблиці в інтерфейсі. Після завершення

імпорту виводиться повідомлення про успішне завантаження даних у вигляді параграфа у *RichTextBox*.

3.3.3 Обчислення середнього балу студентів

Метод *AvgStudentScores_Click* призначений для обчислення середніх балів кожного студента. Алгоритм працює таким чином:

- для кожного рядка таблиці (студента) сумуються всі числові значення в рядку;
- обчислюється середнє арифметичне та виводиться результат у вигляді форматovanого тексту.
- ці значення відображаються в *RichTextBox* з заголовком, виділеним жирним шрифтом.

3.3.4 Визначення проблемних студентів

Метод *FindProblemStudents_Click* здійснює аналіз на предмет виявлення студентів, чий середній бал нижчий за встановлений поріг (у даному випадку — 3.0). Ці студенти виводяться червоним кольором у звіті. Механізм розрахунку ідентичний попередньому методу, але перед виводом результатів здійснюється перевірка на умову $avg < 3.0$.

3.3.5 Аналіз складності дисциплін

Метод *AnalyzeSubjectDifficulty_Click* аналізує середні оцінки по кожній дисципліні:

- для кожної колонки (дисципліни) обчислюється середнє арифметичне значення оцінок усіх студентів;
- отриманий список сортується за зростанням;
- після сортування дисципліни розподіляються на три групи: нижня третина (найнижчі бали) — виділяється червоним кольором, середня — жовтим, верхня третина — зеленим. Це дозволяє швидко виявити як найбільш складні дисципліни, так і ті, які засвоюються студентами найкраще.

3.3.6 Формування рейтингу студентів

Метод *RankStudents_Click* формує рейтинг студентів за зменшенням середнього балу. Для цього:

- обчислюється середній бал для кожного студента;
 - формується список кортежів типу (ПІБ, середній бал);
 - список сортується за спаданням;
 - результат виводиться у вигляді пронумерованого списку у *RichTextBox*.
- Цей підхід дає змогу швидко визначити найуспішніших студентів групи.

3.3.7 Вивід результатів у *RichTextBox*

Всі результати аналізу виводяться у *RichTextBox*, що дозволяє форматувати текст (жирність, колір). Це реалізується у двох допоміжних методах: *UpdateReport(Paragraph paragraph)*

Очищає попередній вміст *RichTextBox* та додає новий параграф з результатами.

ShowMessage(string message, SolidColorBrush color)

Використовується для виводу коротких повідомлень (наприклад, про успішне завантаження файлу) з можливістю вказати колір тексту.

3.3.8 Взаємодія з інтерфейсом користувача

Клас *MainWindow* є вікном головного інтерфейсу користувача. Усі дії користувача активуються через події (натискання кнопок), які викликають відповідні методи. Всі обчислення виконуються в коді-*behind* (C#), а інтерфейс описаний у файлі *XAML* у наступному підрозділі.

Представлена програма є простим, але потужним інструментом для статистичного аналізу успішності студентів. Вона об'єднує у собі сучасні

технології читання табличних даних (*EPPlus*), обробки колекцій (*LINQ*), візуалізації результатів (*WPF, RichTextBox*) та взаємодії з користувачем. Чітке розділення логіки та візуалізації робить програму масштабованою та зручною для подальшого розвитку, наприклад, для додавання збереження результатів або

побудови діаграм. Завдяки використанню зрозумілих алгоритмів та стандартних структур даних, додаток легко підтримується та адаптується до нових умов.

3.4 Розробка інтерфейсу користувача мовою XAML

Підпункт 3.4 кваліфікаційної роботи присвячений опису розробки інтерфейсу користувача з використанням мови розмітки XAML (*eXtensible Application Markup Language*), яка є складовою технології WPF.

Код, який утворює інтерфейс користувача на мові XAML, представлено у додатку Б:

XAML дозволяє описувати графічний інтерфейс у декларативному стилі, відокремлюючи візуальне представлення від логіки програми, що реалізується на мові C#. Такий підхід сприяє гнучкості, зручності розробки та підтримці додатку.

3.4.1 Загальний опис інтерфейсу

Інтерфейс додатку побудовано у віконному середовищі (*Window*) з фіксованими розмірами (600 пікселів заввишки та 900 пікселів завширшки). Головне вікно має заголовок «Аналіз якості освіти». Вміст вікна розміщений у контейнері *Grid*, який дозволяє створити гнучке розташування елементів за допомогою рядків та стовпців:

```
1 <Grid.RowDefinitions>
2 <RowDefinition Height = "Auto" / >
3 <RowDefinition Height = "*" / >
4 <RowDefinition Height = "2*" / >
5 < / Grid.RowDefinitions>
```

Рисунок 3.2 – Вміст вікна розміщений у контейнері *Grid*

33

Цей фрагмент визначає три рядки сітки:

- перший рядок має висоту, що автоматично підлаштовується під вміст;
- другий рядок займає одну частину залишкового простору;
- третій — дві частини.

Такий поділ дозволяє логічно розмежувати: верхню панель кнопок керування, центральний блок для табличних даних і нижній блок для текстових звітів.

3.4.2 Панель керування (*StackPanel*)

У верхньому рядку *Grid* розміщено *StackPanel* з горизонтальною орієнтацією. Він містить п'ять кнопок (*Button*), які виконують основні функції програми. Кожна кнопка має атрибут *Click*, що вказує на відповідний обробник події у кодї на C#:

```
1 <Button Name = "LoadExcel" Content = "Завантажити Excel" Click = "LoadExcel_Click" / >
2 <Button Name = "AvgStudentScores" Content = "Середній бал студентів" Click = "AvgStudentScores_Click" / >
3 <Button Name = "FindProblemStudents" Content = "Проблемні студенти" Click = "FindProblemStudents_Click" / >
4 <Button Name = "AnalyzeSubjectDifficulty" Content = "Складність дисциплін" Click = "AnalyzeSubjectDifficulty_Click" / >
5 <Button Name = "RankStudents" Content = "Рейтинг студентів" Click = "RankStudents_Click" / >
```

Рисунок 3.3 – Програмування атрибуту *Click*

Кожна з кнопок виконує певну логіку, яка стосується аналізу даних, завантажених з *Excel*. Завдяки зрозумілим написам на кнопках користувач легко орієнтується в функціоналі.

3.4.3 Таблиця з даними (*DataGrid*)

У другому рядку *Grid* розміщено компонент *DataGrid*, що відповідає за відображення зчитаних даних у табличному вигляді. Він має властивість *AutoGenerateColumns="True"*, що дозволяє автоматично будувати колонки на основі структури *DataTable*, прив'язаної у C# кодї:

```
<DataGrid Name="DataGridStudents" AutoGenerateColumns="True"
IsReadOnly="True" />
```

Також встановлено атрибут *IsReadOnly="True"*, що запобігає зміні даних користувачем, що є доцільним, оскільки програма виконує лише аналіз, а не редагування.

3.4.4 Вивід звітів (*RichTextBox*)

У третьому рядку розташовано компонент *RichTextBox*, який використовується для форматowanego текстового виводу результатів обробки даних. Завдяки цьому елементу результати можуть бути виведені з використанням кольорів, шрифтів та інших візуальних ефектів:

```
<RichTextBox Name="TextBoxReport" IsReadOnly="True"
```

```
VerticalScrollBarVisibility="Auto" FontSize="14" Padding="5" />
```

Атрибут *IsReadOnly="True"* забезпечує лише перегляд тексту, без можливості його редагування користувачем. *VerticalScrollBarVisibility="Auto"* означає, що вертикальна прокрутка з'явиться лише у випадку переповнення вмісту. Атрибути *FontSize* і *Padding* відповідають за візуальну привабливість відображення результатів.

3.4.5 Стиль і зручність інтерфейсу

Завдяки використанню контейнерів *Grid* та *StackPanel*, інтерфейс виглядає впорядковано та інтуїтивно зрозуміло. Всі елементи мають однакову висоту, чіткі відступи (*Margin*), що забезпечує зручність користування. Уся функціональність зосереджена у верхній панелі, що дозволяє виконувати всі дії послідовно та швидко.

Кожна кнопка логічно відповідає окремій функції аналізу, що відповідає принципам модульного дизайну. Крім того, реалізовано принцип відокремлення інтерфейсу від логіки: *XAML* відповідає за візуальне представлення, а *C#* — за функціональність. Це робить код легким для підтримки та масштабування.

35

3.4.6. Переваги використання *XAML* у даному проєкті

Розділення обов'язків: *XAML* зосереджений на інтерфейсі, а *C#* — на бізнес-логіці.

Гнучкість верстки: За допомогою сіток та панелей можна адаптувати розмітку під різні розміри вікна або доповнити її новими елементами.

Інтуїтивність: Розробник може швидко зрозуміти структуру інтерфейсу без аналізу коду-*behind*.

Можливість розширення: У майбутньому можна додати стилі, шаблони,

тригери чи анімації без значних змін у логіці.

Інтерфейс користувача, реалізований за допомогою *XAML*, є прикладом ефективного підходу до створення сучасних десктопних застосунків. Його структура логічно впорядкована, компоненти — функціонально розмежовані, а візуальне оформлення — зручне для сприйняття. Такий підхід дозволяє не лише покращити взаємодію користувача з додатком, але й спрощує процес подальшої підтримки та розширення функціональності. Розроблений *UI* повністю відповідає поставленим задачам кваліфікаційної роботи щодо наочності та зручності статистичного аналізу успішності студентів.

3.5 Тестування проекту

Після завершення етапів проектування інтерфейсу та реалізації функціональної логіки наступним критичним етапом є тестування ПЗ. Мета тестування — перевірити коректність роботи основних функцій програми, забезпечити точність обчислень, коректну обробку вхідних даних та правильність виведення результатів у візуальному інтерфейсі. Для цього було використано тестову таблицю *Test_data.xlsx* (рис. 3.4), яка містить умовні дані щодо успішності студентів за низкою дисциплін.

Усі тести проводилися на робочій версії програми в середовищі *Windows* із встановленими бібліотеками *EPPlus* та *.NET Framework*. Дані з тестової таблиці були попередньо перевірені на коректність структури: перший рядок —

36

заголовки (імена дисциплін), перший стовпець — ПІБ студентів, інші клітинки — оцінки за п'ятибальною шкалою (від 2 до 5).

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	ПІВ здобувача освіти	Архітектура освітньої	Алгоритми та структури даних	Математичний аналіз	Фізика	Програмування	Фізична культура	Операційні системи	Базис даних	Комп'ютерні мережі	Екологія підприємства	Українська мова	Цифрова освітотехніка	
2	ВЕСТ Кіаті Рексович	3	3	3	3	3	3	5	5	4	3	4	3	
3	ВІНФРІ Опра Вернелліана	4	3	4	3	3	3	4	5	4	4	3		
4	ГАГА Леді Діозефіана	5	4	5	4	3	3	5	5	4	4	5	4	
5	ГЕЙТС Білл Вільямович	5	4	3	3	3	5	4	5	4	5	3	4	
6	ГЕНІС Том Томасович	5	3	5	5	3	4	5	5	4	4	3	5	
7	ДІ КАПРІО Леонардо Джорджович	5	3	3	4	5	4	5	5	4	4	4	3	
8	ЙОГАНССОН Оскарлетт Карсонівна	3	4	5	3	3	4	5	3	5	3	5	5	
9	ЛОРІ-БЛУ Адель Марковна	5	5	5	4	4	4	4	4	4	4	4	5	
10	МАСК Ілон Ерролович	4	4	3	3	5	5	4	3	3	4	4	3	
11	НОУЛЗ Бейонсе Метьюсіана	5	4	4	4	4	4	4	4	4	5	4	5	
12	ПЕРРІ Неті Морісіана	4	3	5	3	4	4	4	3	5	5	4	3	
13	ПРЕСЛІ Еліас Верноніан	5	5	3	4	4	5	5	3	4	5	3	3	
14	РІВЗ Кіану Семюлович	3	2	2	2	2	3	2	3	2	3	2	3	
15	РІЗУЛЛ Шадра Вільямівна	4	3	5	5	5	4	5	5	5	3	3	4	
16	СВІРТ Тейлор Скоттіана	3	2	3	2	3	2	2	2	3	3	2	2	
17	ТУНБЕРГ Грета Свантіанівна	2	3	3	2	2	2	3	2	2	2	2	3	
18	ФЕНТІ Рінна Рональдівна	3	4	3	5	4	5	5	3	5	4	4	3	
19	ФРИМЕН Морган Вікентович	3	3	2	2	3	3	2	2	3	3	2	2	
20	ЦУКЕРБЕРГ Марк Діаїлібович	5	5	4	3	4	5	3	3	4	3	5	5	
21	ЧІКОНЕ Мадонна Сільвіана	4	3	3	5	5	3	4	5	3	5	4	3	
22	ШВАРЦЕНЕГЕР Арнольд Густавович	5	4	3	5	5	3	4	5	5	5	3	3	

Рисунок 3.4 – Тестова таблиця у форматі *.xlsx*

3.5.1 Завантаження *Excel*-файлу

Перше, що перевіряється, — можливість імпортувати *Excel*-файл з даними. Тестування включає:

- відкриття діалогового вікна після натискання кнопки «Завантажити *Excel*»;
- вибір файлу *Test_data.xlsx*;
- обробку файлу: зчитування заголовків, перетворення таблиці у внутрішній об'єкт *DataTable*;
- виведення даних у візуальну таблицю *DataGridStudents*; - повідомлення про успішне завантаження.

Очікуваний результат: таблиця завантажується без помилок, заголовки та дані виводяться коректно, у нижньому полі *RichTextBox* з'являється повідомлення «Файл успішно завантажено» (рис. 3.5).

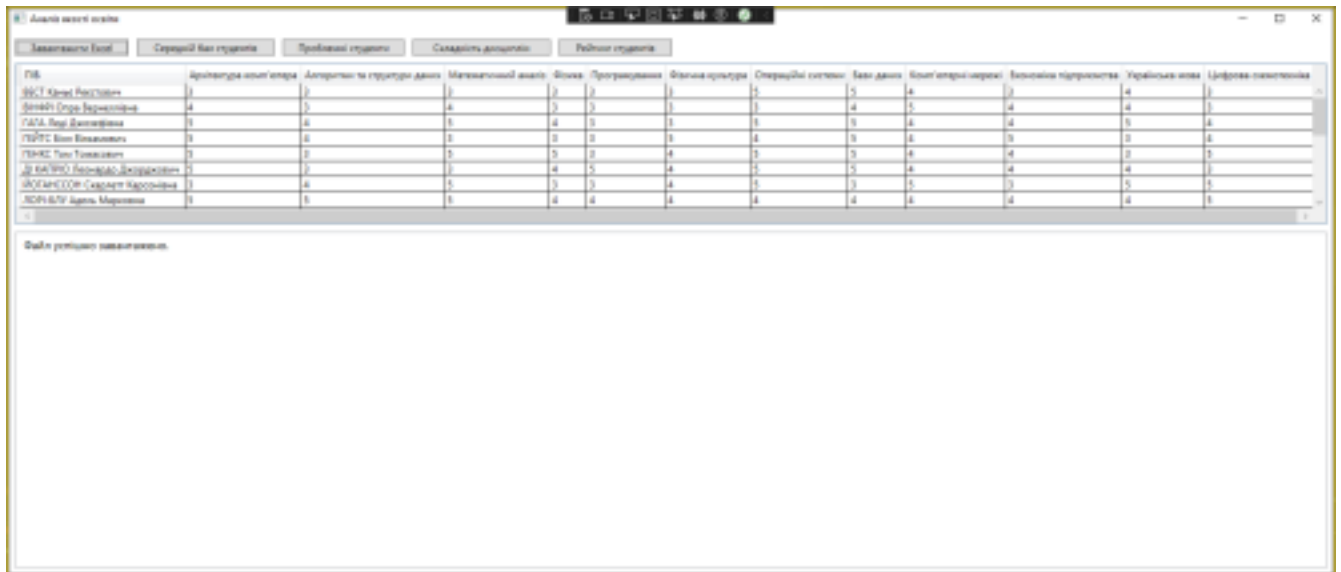


Рисунок 3.5 – Інтерфейс додатка після завантаження таблиці

3.5.2 Обчислення середнього балу студентів

Після завантаження таблиці перевіряється функція «Середній бал студентів» (рисунок 3.6).

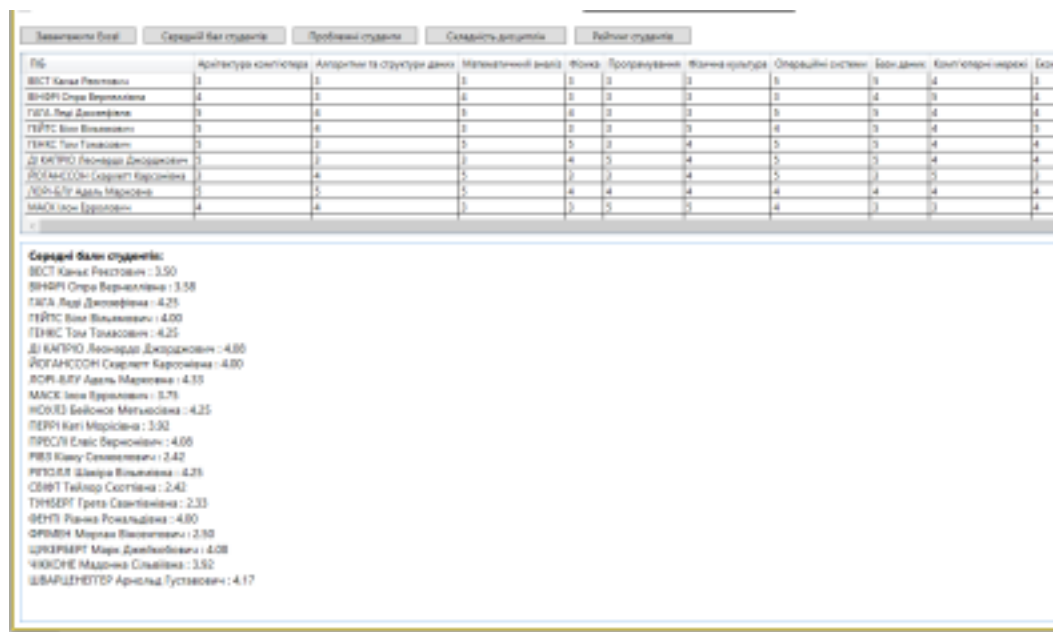


Рисунок 3.6 – Скріншот результату з розрахунками середнього балу

Тестування включає натискання відповідної кнопки, що активує обчислення:

- для кожного студента підраховується середнє арифметичне всіх його оцінок;

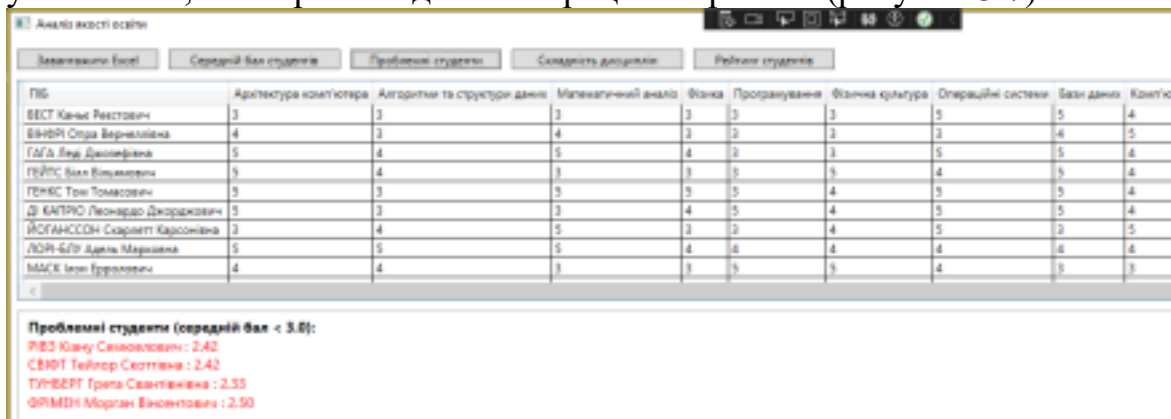
- результати виводяться у *RichTextBox* у вигляді списку: Ім'я: середній бал.

Очікуваний результат: для кожного студента відображається коректне середнє значення з точністю до двох знаків після коми. Усі результати мають відповідати ручним розрахункам.

3.5.3 Виявлення проблемних студентів

Ця функція дозволяє визначити студентів, чий середній бал нижчий за 3.0. Після натискання кнопки «Проблемні студенти»:

- розраховуються середні бали;
- результати фільтруються за умовою $avg < 3.0$;
- прізвища таких студентів виділяються червоним кольором. Очікуваний результат: список містить лише студентів, які дійсно мають низький рівень успішності; кольорове виділення працює коректно (рисунок 3.7).



The screenshot shows a web application interface with a table of student grades and a list of problematic students below it. The table has columns for student names and various subjects. The list below the table shows the names of problematic students in red text, along with their average scores.

ПІБ	Архітектура комп'ютера	Алгоритми та структури даних	Математичний аналіз	Фізика	Програмування	Фізична культура	Операційні системи	Бази даних	Квантис
ВЕСТ Кіане Раєстоєн	3	3	3	3	3	3	5	5	4
ВАНБІ Опра Вернеліана	4	3	4	3	3	3	3	4	5
ГАГА Лана Діксоніана	5	4	5	4	3	3	5	5	4
ГЕЛТІ Сіан Вільямівна	5	4	3	3	3	5	4	5	4
ГЕНКС Тей Томасович	5	3	5	5	5	4	5	5	4
ДІ КАПРО Леонардо Джорджович	5	3	3	4	5	4	5	5	4
ЙОГАНССОН Скарлетт Карсонівна	3	4	5	3	3	4	5	3	5
ЛОРИ-БЛЮ Дана Маршівна	5	5	5	4	4	4	4	4	4
МАКЕ Іван Еуралович	4	4	3	3	3	5	4	3	3

Проблемні студенти (середній бал < 3.0):
ВЕСТ Кіане Савковичівна : 2.42
ВАНБІ Тейлер Скоттівна : 2.42
ТЛІМБЕРГ Трейла Савантівна : 2.33
ФРІМБІН Морган Ванковичівна : 2.50

Рисунок 3.7 – Список проблемних студентів з червоним текстом у *RichTextBox*

3.5.4. Аналіз складності дисциплін

Функція «Складність дисциплін» аналізує середні оцінки за кожною дисципліною, визначаючи:

- середній бал по кожному предмету;
- сортування дисциплін за зростанням середнього балу;
- кольорове виділення: червоний — найскладніші дисципліни (нижня третина), жовтий — середня складність, зелений — легкі дисципліни (верхня третина).

Очікуваний результат: результати виводяться у вигляді списку з коректними середніми балами; сортування відповідає очікуванням; кольори відповідають позиції у списку (рисунок 3.8).

ІВБ	Архітектура комп'ютера	Алгоритми та структури даних	Математичний аналіз	Фізика	Програмування	Фізична культура	Операційні системи	Бази даних
ВЕСТ Бачи Ростислав	5	5	5	5	5	5	5	5
ВНОЯ Опра Вероніка	4	5	4	5	5	5	5	4
ГАГА Леви Діана	5	4	5	4	5	5	5	5
ГЕЯС Віта Вікторівна	5	4	5	5	5	5	4	5
ГЕНІС Том Тимович	5	5	5	5	5	4	5	5
Д. КАПРИО Ірина	5	5	5	4	5	4	5	5
ЙОГАНССОН Скарлет Кароліна	5	4	5	5	5	4	5	5
ЛОЯ-БІУ Адріан Маркович	5	5	5	4	4	4	4	4
МАК Іван Іванович	4	4	5	5	5	4	5	5

Середні бали по дисциплінах:
 Алгоритми та структури даних: 3.52
 Фізика: 3.52
 Українська мова: 3.52
 Цифрова симетрія: 3.52
 Математичний аналіз: 3.62
 Програмування: 3.67
 Фізична культура: 3.71
 Бази даних: 3.76
 Економіка підприємства: 3.86
 Комп'ютерні мережі: 3.90
 Операційні системи: 3.95
 Архітектура комп'ютера: 4.05

Рисунок 3.8 - Кольоровий вивід складності дисциплін

3.5.5 Ранжування студентів за успішністю

Після натискання кнопки «Рейтинг студентів» відбувається:

- повторне обчислення середнього балу для кожного студента; - сортування за спаданням;
- формування пронумерованого списку в *RichTextBox*.

Очікуваний результат: студенти розміщені у порядку зменшення успішності, значення балів відповідають очікуваням.

Результат повністю збігається з вручну сформованим рейтингом на основі вхідної таблиці (рисунок 3.9).

ГІВ кафедра освіти	Архітектура освітнього процесу	Діджиталізація освітнього процесу	Математичний аналіз	Фізика	Програмування	Фізична культура	Операційні системи	Бази даних	Інформаційні системи	Векторний аналіз	Українська мова	Цифрова освітні технології
ІНСТ Київський Національний								5	4	3	4	3
ІНСТ Київський Національний								4	5	4	4	3
ІНСТ Київський Національний								5	4	4	5	4
ІНСТ Київський Національний								5	4	3	5	4
ІНСТ Київський Національний								5	4	4	4	5
ІНСТ Київський Національний								5	4	4	4	5
ІНСТ Київський Національний								5	4	3	5	4
ІНСТ Київський Національний								4	4	4	4	5
ІНСТ Київський Національний								5	4	4	4	5
ІНСТ Київський Національний								5	4	4	4	5

Файл успішно завантажено.

Рисунок 3.10 – Приклад роботи з неповним або зміненим файлом

Проведене тестування підтвердило стабільність роботи додатку, логічну послідовність операцій та відсутність критичних помилок при роботі з реальними та змодельованими даними. Програма готова до практичного використання як інструмент внутрішнього аналізу успішності в навчальному закладі.

3.6 Висновки до третього розділу

У третьому розділі кваліфікаційної роботи було докладно розглянуто практичну реалізацію програмного засобу для аналізу якості освіти студентів на основі табличних даних. Основна увага була зосереджена на технічній побудові програми, логіці її функціонування, розробці графічного інтерфейсу та тестуванні.

Під час проєктування функціоналу були реалізовані основні модулі, що відповідають за:

- завантаження даних з *Excel*-файлу;
- обчислення середніх балів студентів;
- формування рейтингу;

- виявлення студентів із низькою успішністю;
- визначення дисциплін, які викликають найбільші труднощі; - кольорову візуалізацію результатів у зручному для користувача форматі.

Інтерфейс користувача було створено з використанням *XAML* у межах технології *WPF*, що дозволило досягти відокремлення логіки програми від її візуального представлення. Усі елементи інтерфейсу розміщено логічно, відповідно до принципів зручності та доступності. Кнопки розташовані у верхній частині вікна, що забезпечує швидкий доступ до ключових функцій, а результати відображаються у нижній частині з використанням форматowanego *RichTextBox*.

Проведене тестування програми підтвердило її стабільність, правильність обчислень та відповідність очікуваним результатам. Особливу увагу було приділено роботі з неповними або некоректно заповненими файлами, і програма продемонструвала здатність адекватно обробляти подібні ситуації.

Таким чином, реалізований програмний продукт повністю виконує поставлені завдання та може використовуватися для внутрішнього аналізу якості навчання в освітніх установах. Водночас його структура та модульність дозволяють у подальшому розширити функціонал: додати збереження звітів, графічну аналітику, експорт у *PDF* або підключення до бази даних. Програма демонструє приклад практичного застосування засобів *C#* та *WPF* для вирішення актуальних задач в освітньому середовищі.

43

ВИСНОВКИ

У процесі виконання кваліфікаційної роботи на тему "Програмування додатку для статичного аналізу якості освіти засобами *C#* та *WPF*" було розроблено та реалізовано програмний продукт, який дозволяє ефективно аналізувати показники навчальної успішності студентів на основі даних у форматі *Excel*. В результаті дослідження були досягнуті основні поставлені цілі, а саме: опрацьовано теоретичні основи статистичного аналізу в освітньому середовищі, досліджено можливості його автоматизації засобами ЕОМ,

створено програму для аналізу освітніх даних, протестовано її функціональність та інтерфейс.

У першому розділі було розглянуто загальні підходи до аналізу якості освіти, визначено ключові статистичні показники, які використовуються в системах оцінювання. Було проаналізовано сучасні підходи до збору та обробки освітніх даних, зокрема у вигляді електронних таблиць та звітів. Окрему увагу приділено питанням наочності, доступності і точності інтерпретації даних, які є критично важливими для управлінців, викладачів та дослідників у галузі освіти.

У другому розділі було здійснено аналіз методології статистичного аналізу, зокрема описано базові поняття описової та інференційної статистики, умови застосування параметричних та непараметричних методів, наведено приклади їхнього використання для освітніх досліджень. Особливу увагу приділено критеріям вибору методу в залежності від типу даних, їх розподілу, обсягу вибірки та мети дослідження. Також було розглянуто переваги автоматизації статистичного аналізу із застосуванням сучасних програмних засобів, таких як *SPSS*, *R*, *Python*, а також власних програмних розробок. Визначено ключові фактори, які впливають на якість аналізу: правильний вибір методу, коректність вихідних даних, доступність інтерфейсу та можливість візуалізації результатів.

У третьому розділі докладно описано процес створення програмного забезпечення для аналізу освітніх даних. Програма була реалізована за

44

допомогою мови *C#* з використанням технології *WPF*. Інтерфейс користувача розроблений засобами *XAML*, що забезпечило чітке розділення логіки та візуального представлення. У додатку реалізовано наступний функціонал: - завантаження даних з *Excel*-файлу;

- розрахунок середнього балу кожного студента;
- формування рейтингу студентів за успішністю;
- виявлення студентів з низькими середніми балами (проблемних); -
- аналіз складності дисциплін на основі середнього балу;
- кольорове маркування результатів для полегшення сприйняття. У ході

тестування було використано спеціально підготовлений файл із тестовими даними, що дозволило перевірити всі функції програми в умовах, наближених до реальних. Було встановлено, що всі основні модулі функціонують стабільно, результати обчислень відповідають очікуваням. Програма коректно обробляє як повні, так і частково заповнені файли *Excel*, не допускає аварійного завершення роботи при нестандартних вхідних даних.

Особливо слід відзначити переваги розробленої програми над класичними електронними таблицями:

- швидкість — обробка навіть великої кількості студентів відбувається миттєво;
- гнучкість — програма не потребує адаптації під конкретну кількість предметів або студентів;
- зручність — простий інтерфейс, чітко структуровані кнопки функцій;
- візуальність — кольорове маркування дозволяє швидко виявити проблемні ділянки;
- локальність — вся обробка виконується на комп'ютері користувача, що забезпечує конфіденційність даних.

Таким чином, поставлені у кваліфікаційній роботі завдання були повністю виконані. В результаті розроблено ефективний інструмент для аналізу освітньої інформації, який може бути використаний адміністрацією навчальних закладів,

45

викладачами, кураторами академічних груп, а також дослідниками у галузі педагогіки та освітнього менеджменту.

Розроблена система має широкі перспективи розвитку. У майбутньому можливо:

- реалізувати збереження результатів у вигляді *PDF* або *Excel*-звітів;
- додати модуль графічної аналітики (графіки, діаграми);
- забезпечити інтеграцію з базами даних чи електронними журналами навчальних закладів;
- реалізувати веб-версію для віддаленого використання.

Враховуючи зростаючі вимоги до якості освітніх послуг, цифрову трансформацію закладів освіти та актуальність моніторингу результатів навчання, створення та впровадження таких систем має стратегічне значення. Кваліфікаційна робота підтвердила доцільність використання мов програмування високого рівня (C#) та технологій WPF у сфері освітньої аналітики, продемонструвала можливість створення ефективного застосунку у стислі терміни та з обмеженими ресурсами.

Отже, практичні результати кваліфікаційної роботи можуть бути реалізовані як у закритих локальних системах, так і в масштабованих проєктах з відкритим кодом, що сприятиме підвищенню ефективності освітнього процесу та забезпеченню якості освіти на новому рівні.

46

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. О. В. Раєвська, І. В. Аксьонова, О. І. Бровко ; за заг. ред. д-ра екон. наук, професора О. В. Раєвської. Статистика : навчальний посібник/ – Харків : ХНЕУ ім. С. Кузнеця, 2019. – 389 с.

2. І. Мосякова. Освітній менеджмент: Тенденції ефективного використання технологій управління багатoproфільним закладом позашкільної освіти. Нова педагогічна думка. 2022. № 4 (112)., DOI: 10.37026/2520-6427-2022-112-4-38-42.

3. Веб Деканат. Інтернет-посилання. [Метод доступу]: <https://vuz.osvita.net/> (дата звернення 12.05.2025)

4. Moodle community. Інтернет-посилання. [Метод доступу]: <https://moodle.org> (дата звернення 12.05.2025)

5. Google for Education. Інтернет-посилання. [Метод доступу]: <https://edu.google.com/workspace-for-education> (дата звернення 12.05.2025) 6.

Power BI. Інтернет-посилання. [Метод доступу]: <https://power.bi.com.ua/> (дата звернення 12.05.2025)

7. Open Education Analytics. Інтернет-посилання. [Метод доступу]: <https://openeducationanalytics.org/> (дата звернення 12.05.2025) 8. C Sharp.

Інтернет-посилання. [Метод доступу]: https://uk.wikipedia.org/wiki/C_Sharp (дата звернення 12.05.2025) 9. Платформа XAML. Інтернет-посилання. [Метод доступу]: <https://learn.microsoft.com/windows/uwp/xaml-platform/> (дата звернення 12.05.2025)

10. Трофименко О. Г. С++. Алгоритмізація та програмування : підручник / О. Г. Трофименко, Ю. В. Прокоп, Н. І. Логінова, О. В. Задерейко. 2-ге вид. перероб. і доповн. – Одеса : Фенікс, 2019. – 477 с.

11. Лобода Ю. Г., Трофименко О. Г. Офісні інформаційні технології : навч.-метод. посіб. [Електронне видання] / Ю. Г. Лобода, О. Г. Трофименко ; Нац.

47

ун-т «Одес. юрид. академія». – Одеса : Фенікс, 2025. – 158 с. Режим доступу: <https://hdl.handle.net/11300/29638>

12. Чанишев Р. І. Інформаційні технології та *Data mining* в соціальних науках: навчально-методичний посібник для підготовки здобувачів вищої освіти галузі знань 05 «Соціальні та поведінкові науки» [Електронне видання] / Р. І. Чанишев, М. А. Яценко ; Нац. ун-т «Одеська юрид. академія». – Одеса: Фенікс, 2024. – 706 с. – Режим доступу: <https://doi.org/10.32837/11300.27676>

13. Т. М. Паянок, Т. М. Задорожня. Статистичний аналіз даних : навчальний посібник . – Ірпінь : Університет державної фіскальної служби України, 2020. – 312 с. – (Серія «На допомогу студенту УДФСУ»; т. 60).

14. Негрей М., Гнот Т. Аналітика з R: навчальний посібник / Негрей М., Гнот Т. Київ: ФОП Ямчинський О. В., 2020. 236 с.

15. Ю.Г. Лисенко; Н.С. Педченко; В.М. Кравченко; В.В. Мандра; М.О. Мизнікова; В.М. Берлін; В.М. Лев та ін. Модернізація фінансових систем: методологія та інструменти управління. За ред. чл.кор. НАН України, д-ра екон. наук, проф. Лисенко Ю.Г.; д-ра екон. наук, доц. Жерліцина Д.М. – Полтава, 2017. – 348 с.

48

ДОДАТОК А

Код програми на мові С#

```

using Microsoft.Win32;
using OfficeOpenXml;
using System;
using System.Collections.Generic;
using System.Data;
using System.IO;
using System.Linq;
using System.Windows;
using System.Windows.Documents;
using System.Windows.Media;
namespace EducationStatsApp
{
    public partial class MainWindow : Window
    {
        private DataTable _dataTable;
        public MainWindow()
        {
            InitializeComponent();
            ExcelPackage.LicenseContext = LicenseContext.NonCommercial; }
        private void LoadExcel_Click(object sender, RoutedEventArgs e) {
            var dialog = new OpenFileDialog
            {
                Filter = "Excel Files (*.xlsx)|*.xlsx"
            };
            if (dialog.ShowDialog() == true)
            {
                string path = dialog.FileName;
                using (var package = new ExcelPackage(new FileInfo(path))) {
                    ExcelWorksheet worksheet = package.Workbook.Worksheets[0]; var dt =
                    new DataTable();

```

```

for (int col = worksheet.Dimension.Start.Column; col <=
worksheet.Dimension.End.Column; col++)
dt.Columns.Add(worksheet.Cells[1, col].Text);
for (int row = 2; row <= worksheet.Dimension.End.Row; row++) {
var dataRow = dt.NewRow();
for (int col = worksheet.Dimension.Start.Column; col <=
worksheet.Dimension.End.Column; col++)
dataRow[col - 1] = worksheet.Cells[row, col].Text;
dt.Rows.Add(dataRow);
}

_dataTable = dt;
DataGridStudents.ItemsSource = dt.DefaultView;
ShowMessage("Файл успішно завантажено.", Brushes.Black); }
}
}

private void AvgStudentScores_Click(object sender, RoutedEventArgs e) {
if (_dataTable == null) return;
var paragraph = new Paragraph();
paragraph.Inlines.Add(new Run("Середні бали студентів:\n") { FontWeight =
FontWeights.Bold });

foreach (DataRow row in _dataTable.Rows)
{
string name = row[0].ToString();
double sum = 0;
int count = 0;
for (int i = 1; i < _dataTable.Columns.Count; i++)
{
if (double.TryParse(row[i].ToString(), out double score)) {
sum += score;
}
}
}
}

```

```

count++;
}
}
double avg = count > 0 ? sum / count : 0;
paragraph.Inlines.Add(new Run($"{name}: {avg:F2}\n")); }
UpdateReport(paragraph);
}
private void FindProblemStudents_Click(object sender, RoutedEventArgs e) {
if (_dataTable == null) return;
var paragraph = new Paragraph();
paragraph.Inlines.Add(new Run("Проблемні студенти (середній бал < 3.0):\n") {
FontWeight = FontWeights.Bold });
foreach (DataRow row in _dataTable.Rows)
{
string name = row[0].ToString();
double sum = 0;
int count = 0;
for (int i = 1; i < _dataTable.Columns.Count; i++)

{
if (double.TryParse(row[i].ToString(), out double score)) {
sum += score;
count++;
}
}
double avg = count > 0 ? sum / count : 0;
if (avg < 3.0)
paragraph.Inlines.Add(new Run($"{name}: {avg:F2}\n") { Foreground =
Brushes.Red });
}
UpdateReport(paragraph);
}
}

```

}

```
private void AnalyzeSubjectDifficulty_Click(object sender, RoutedEventArgs e) {  
    if (_dataTable == null) return;  
    var averages = new List<(string Subject, double Average)>();  
    for (int i = 1; i < _dataTable.Columns.Count; i++)  
    {  
        string subject = _dataTable.Columns[i].ColumnName;  
        double sum = 0;  
        int count = 0;  
        foreach (DataRow row in _dataTable.Rows)  
        {  
            if (double.TryParse(row[i].ToString(), out double score)) {  
                sum += score;  
                count++;  
            }  
        }  
        double avg = count > 0 ? sum / count : 0;  
        averages.Add((subject, avg));  
    }  
    var sorted = averages.OrderBy(x => x.Average).ToList();  
    int total = sorted.Count;  
    int oneThird = total / 3;  
    var paragraph = new Paragraph();  
    paragraph.Inlines.Add(new Run("Середні бали по дисциплінах:\n") { FontWeight  
    = FontWeights.Bold });  
  
    for (int i = 0; i < total; i++)  
    {  
        var item = sorted[i];
```

```

SolidColorBrush color = Brushes.Goldenrod;
if (i < oneThird) color = Brushes.Red;
else if (i >= 2 * oneThird) color = Brushes.Green;
paragraph.Inlines.Add(new Run($"{item.Subject}: {item.Average:F2}\n") {
Foreground = color });
}
UpdateReport(paragraph);
}
private void RankStudents_Click(object sender, RoutedEventArgs e) {
if (_dataTable == null) return;
var results = new List<(string Name, double Average)>();
foreach (DataRow row in _dataTable.Rows)
{
string name = row[0].ToString();

double sum = 0;
int count = 0;
for (int i = 1; i < _dataTable.Columns.Count; i++)
{
if (double.TryParse(row[i].ToString(), out double score)) {
sum += score;
count++;
}
}

double avg = count > 0 ? sum / count : 0;
results.Add((name, avg));
}
var paragraph = new Paragraph();
paragraph.Inlines.Add(new Run("Рейтинг студентів:\n") { FontWeight =
FontWeights.Bold });

```

```
int rank = 1;
foreach (var student in results.OrderByDescending(s => s.Average)) {
    paragraph.Inlines.Add(new Run($"{rank++}. {student.Name} —
{student.Average:F2}\n"));
}
```

```
UpdateReport(paragraph);
}
```

```
private void UpdateReport(Paragraph paragraph)
{
    TextBoxReport.Document.Blocks.Clear();
    TextBoxReport.Document.Blocks.Add(paragraph);
```

54

```
}
```

```
private void ShowMessage(string message, SolidColorBrush color) {
    var p = new Paragraph(new Run(message) { Foreground = color });
    TextBoxReport.Document.Blocks.Clear();
    TextBoxReport.Document.Blocks.Add(p);
```

```
}
```

```
}
```

```
}
```

55

ДОДАТОК Б

Лістинг програми WPF

```
<Window x:Class="EducationStatsApp.MainWindow"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml" Title="Аналіз
якості освіти" Height="600" Width="900"> <Grid Margin="10">
    <Grid.RowDefinitions>
    <RowDefinition Height="Auto"/>
```

```

<RowDefinition Height="*/>
<RowDefinition Height="2*/>
</Grid.RowDefinitions>
<StackPanel Orientation="Horizontal" HorizontalAlignment="Left"
Margin="0,0,0,10">
  <Button Name="LoadExcel" Content="Завантажити Excel" Width="140"
Margin="0,0,10,0" Click="LoadExcel_Click"/> <Button
Name="AvgStudentScores" Content="Середній бал студентів"
Width="170" Margin="0,0,10,0" Click="AvgStudentScores_Click"/>
  <Button Name="FindProblemStudents" Content="Проблемні студенти"
Width="150" Margin="0,0,10,0" Click="FindProblemStudents_Click"/>
  <Button Name="AnalyzeSubjectDifficulty" Content="Складність
дисциплін" Width="170" Margin="0,0,10,0"
Click="AnalyzeSubjectDifficulty_Click"/>
  <Button Name="RankStudents" Content="Рейтинг студентів"
Width="140" Click="RankStudents_Click"/>
</StackPanel>

```

56

```

<DataGrid Name="DataGridStudents" Grid.Row="1"
AutoGenerateColumns="True" IsReadOnly="True" Margin="0,0,0,10" />
<RichTextBox Name="TextBoxReport" Grid.Row="2" IsReadOnly="True"
VerticalScrollBarVisibility="Auto" FontSize="14" Padding="5"/>
</Grid>
</Window>

```

РЕЦЕНЗІЯ

на кваліфікаційну роботу

випускника спеціальності: 123 «Комп'ютерна інженерія»

відділення: комп'ютерної та програмної інженерії

циклова комісія: комп'ютерних систем та мереж

Артур НАТИКАНИЙ
(ім'я, прізвище)

1. Обрана тема кваліфікаційної роботи, «Програмування додатку для статичного аналізу якості освіти засобами C# та WPF», є актуальною.
2. Кваліфікаційна робота відповідає темі, затвердженій наказом.
3. Завдання на виконання кваліфікаційної роботи виконано у повному обсязі у встановлений термін.
4. В результаті виконання кваліфікаційної роботи було додаток на мові програмування C#, який може формувати рейтинг здобувачів освіти, аналізувати освітній процес та виводити різні статистичні показники.
5. Якість виконання пояснювальної записки та ілюстративного (графічного) матеріалу відповідає вимогам державних стандартів.
6. В кваліфікаційній роботі зроблений акцент на використанні Windows Presentation Foundation в середовищі розробки Visual Studio.
7. Кваліфікаційна робота заслуговує оцінку «добре», а випускник присвоєння кваліфікації фахівця освітньо-професійного ступеня «Фаховий молодший бакалавр» спеціальності 123 «Комп'ютерна інженерія».

Рецензент К. пед. н. ; викладач
(науковий ступінь, посада)

«30» 05 2025 р. Кисл
(підпис)

Марія КИСЛОВА
(ім'я, прізвище)

З рецензією ознайомлений

у/м
(підпис)

Артур НАТИКАНИЙ
(ім'я, прізвище)