

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ ЕКОНОМІКИ І ТЕХНОЛОГІЙ

ННІ/факультет Інформаційних технологій
Кафедра Інформатики і прикладного програмного забезпечення
Спеціальність Інженерія програмного забезпечення
Форма навчання Денна

**КВАЛІФІКАЦІЙНА
БАКАЛАВРСЬКА РОБОТА**

Хромичкіна Анастасія Олександрівна
(прізвище, ім'я, по батькові здобувача)

на тему «Розробка програмного забезпечення для жеребкування
учасників ігор з шахів»
(повна назва теми)

за матеріалами праць провідних спеціалістів з розробки ПЗ та
проектування БД

(повна назва бази дослідження)

науковий керівник д.т.н., професор Зеленський О.С.
(наук. ступінь, вчене звання) (підпис) (прізвище, ініціали)

Робота допущена до захисту в ЕК

Протокол засідання кафедри
від 11.06.2025 р. № 12

Завідувач кафедри

(підпис)

д.т.н., професор
Наук. ступень, вчене звання

Зеленський О.С.
Ініціали, прізвище

Кривий Ріг – 2025

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ ЕКОНОМІКИ І ТЕХНОЛОГІЙ

ННІ/факультет	Інформаційних технологій
Кафедра	Інформатики і прикладного програмного забезпечення
Спеціальність	Інженерія програмного забезпечення
Форма навчання	Денна

«ЗАТВЕРДЖУЮ»

Завідувач кафедри _____ Зеленський О.С.
(підпис) (Прізвище, ініціали)
« 11 » червня 2025 року

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ БАКАЛАВРСЬКУ РОБОТУ**

1. Тема роботи «Розробка програмного забезпечення для жеребкування учасників ігор з шахів»

Керівник роботи д.т.н., професор Зеленський О.С.
затвержені наказом закладу вищої освіти від «04» квітня 2025 р. № 222-ст

2. Строк подання здобувачем роботи до «09» червня 2025 р.

3. Зміст кваліфікаційної роботи, об'єкт, предмет та мета дослідження:

Розділ 1. Постановка задачі

Розділ 2. Розробка алгоритму розв'язання задачі

Розділ 3. Організація інформаційного забезпечення

Розділ 4. Розробка програмного забезпечення

Об'єкт дослідження: процес організації шахових турнірів

Предмет дослідження: алгоритми жеребкування учасників шахових турнірів

Мета кваліфікаційної роботи: розробка програмного забезпечення для автоматизованого жеребкування учасників шахових турнірів

5. Дата видачі завдання «04» квітня 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів МДР	Строк виконання етапів роботи	Відмітка керівника про виконання етапів (дата, підпис)
1	Підготовка розділу 1	04.04.2025-13.04.2025	
2	Підготовка розділу 2	14.04.2025-26.04.2025	
3.	Підготовка розділу 3	27.04.2025-15.05.2025	
4	Підготовка розділу 4	16.05.2025-08.06.2025	
5	Реєстрація завершеної кваліфікаційної роботи	09.06.2025	Реєстраційний № ____ «09»червня 2025 р.
6	Отримання відгуку від наукового керівника	10.06.2025	
7	Подання кваліфікаційної роботи на перегляд завідувачу кафедри	11.06.2025	
8	Отримання зовнішньої рецензії	12.06.2025	
9	Попередній захист кваліфікаційної роботи на кафедрі	13.06.2025	
10	Підготовка до захисту в ЕК	16.06.2025-21.06.2025	

Завдання підготував науковий керівник

(підпис)

Зеленський О.С.

(прізвище та ініціали)

Завдання одержав

(підпис)

Хромичкіна А.О.

(прізвище та ініціали)

АНОТАЦІЯ

на кваліфікаційну бакалаврську роботу

«Розробка програмного забезпечення для жеребкування учасників ігор з шахів»

Хромичкіної Анастасії Олександрівни

Кваліфікаційна бакалаврська робота на здобуття освітньо-кваліфікаційного рівня бакалавра зі спеціальності 121 «Інженерія програмного забезпечення» – Державний університет економіки і технологій – Кривий Ріг, 2025.

У бакалаврській дипломній роботі розроблено програмне забезпечення для автоматизованого жеребкування учасників шахових турнірів. Програма дозволяє формувати пари гравців для кожного раунду, зберігати результати зіграних партій, фіксувати інформацію про суперників, колір фігур та набрані очки. Також реалізована можливість експорту результатів на вебсайт. Програмне забезпечення розроблено мовою C++ з використанням бібліотеки MFC для побудови графічного інтерфейсу. Візуалізація логотипу шахової фігури здійснена за допомогою технології OpenGL. Для збереження даних використовується база даних, у якій зберігається інформація про гравців, раунди та результати ігор.

Ключові слова: ШАХИ, ЖЕРЕБКУВАННЯ, АВТОМАТИЗАЦІЯ, БАЗА ДАНИХ, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

БД(база даних)	Впорядкований набір логічно взаємопов'язаних даних, що використовуються спільно та призначені для задоволення інформаційних потреб користувачів.
СУБД	Система управління базами даних.
ПЗ	Програмне забезпечення.
ID	Ідентифікатор
MFC	Microsoft Foundation Classes

ЗМІСТ

ВСТУП.....	7
РОЗДІЛ I.....	9
ПОСТАНОВКА ЗАДАЧІ.....	9
1.1. Характеристика задачі.....	9
1.2. Огляд існуючого аналогу програмного забезпечення	10
1.3. Вхідна інформація	17
1.4. Вихідна інформація	18
РОЗДІЛ II.....	21
РОЗРОБКА АЛГОРИТМУ РОЗВ’ЯЗАННЯ ЗАДАЧІ НА МОВІ ПРОГРАМУВАННЯ C++	21
РОЗДІЛ III	27
ОРГАНІЗАЦІЯ ІНФОРМАЦІЙНОГО ЗАБЕЗПЕЧЕННЯ.....	27
3.1. Загальна характеристика інформаційного забезпечення.....	27
3.2. Структура баз даних та інформаційних масивів.....	30
РОЗДІЛ IV	35
РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ЗАДАЧІ	35
4.1. Опис головного модулю програми (головного вікна) на мові C++	35
4.2. Опис інтерфейсу та алгоритму розробки сайту	54
ВИСНОВКИ.....	61
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	62
ДОДАТКИ.....	63

ВСТУП

У сучасному світі інформаційні технології активно впроваджуються в усі сфери людської діяльності, включаючи спорт та організацію змагань. Автоматизація процесів планування та проведення спортивних турнірів дозволяє значно підвищити ефективність роботи організаторів, зменшити ймовірність помилок та забезпечити справедливість змагань. Особливо актуальним це є для інтелектуальних видів спорту, таких як шахи, де точність організації турнірів має першочергове значення.

Шахи є одним з найпопулярніших інтелектуальних видів спорту у світі. Щорічно проводиться тисячі турнірів різного рівня – від шкільних змагань до міжнародних чемпіонатів. Організація шахових турнірів потребує ретельного планування, особливо на етапі жеребкування учасників, оскільки від правильного розподілу гравців залежить справедливість та якість змагань. Традиційні методи жеребкування часто є трудомісткими, займають багато часу та можуть містити людські помилки.

Метою дипломної роботи є розробка програмного забезпечення для автоматизації процесу жеребкування учасників шахових турнірів з урахуванням правил проведення змагань. Дане програмне рішення має забезпечити швидке та справедливе розподілення учасників по турах, автоматичне формування пар для гри та ведення обліку результатів змагань.

Актуальність створення подібного програмного забезпечення полягає в тому, що на сьогоднішній день існує обмежена кількість якісних та доступних рішень для організації шахових турнірів. Більшість наявних програм є комерційними продуктами з високою вартістю або мають складний інтерфейс, що ускладнює їх використання організаторами турнірів різного рівня. Розроблене програмне забезпечення дозволить організаторам шахових змагань у школах, клубах та спортивних федераціях ефективно проводити турніри без значних фінансових витрат.

Головні завдання програмного забезпечення, яке створюватиметься: автоматичне жеребкування учасників турніру з урахуванням їх рейтингових показників; формування пар гравців для кожного туру змагань; ведення обліку результатів ігор; збереження та завантаження даних турніру; формування звітів про результати змагань; забезпечення зручного та інтуїтивного користувацького інтерфейсу для організаторів турнірів.

Об'єктом дослідження є процеси організації та проведення шахових турнірів. Предметом дослідження є методи та алгоритми жеребкування учасників шахових змагань та автоматизація ведення турнірної документації.

Програмний продукт розроблятиметься за допомогою інструментальних засобів розробки мови програмування C++, середовища розробки Microsoft Visual Studio та системи управління базами даних для зберігання інформації про турніри та учасників.

РОЗДІЛ I

ПОСТАНОВКА ЗАДАЧІ

1.1. Характеристика задачі

Сучасні шахові турніри потребують ефективного програмного забезпечення для автоматизації процесів організації змагань. Розробка спеціалізованого програмного рішення для жеребкування учасників шахових ігор дозволить значно спростити роботу організаторів турнірів та підвищить якість проведення змагань.

Основною метою розробки є створення комплексної системи, яка забезпечить повну автоматизацію турнірних процесів від реєстрації учасників до підведення підсумків змагання. Програмне забезпечення повинно бути зручним у використанні, надійним та функціональним.

Для досягнення поставленої мети необхідно вирішити наступні задачі. Перш за все потрібно створити унікальний логотип у вигляді тривимірної шахової фігури, який стане візуальним символом програми. Важливою складовою є організація надійного зберігання інформації в базі даних, що забезпечить збереження всіх турнірних даних.

Система повинна забезпечувати повноцінне управління учасниками турніру, включаючи можливості додавання нових гравців, видалення існуючих записів та відображення інформації про всіх зареєстрованих учасників. Необхідно реалізувати функціонал регулювання кількості раундів відповідно до специфіки конкретного турніру.

Ключовою функцією системи є автоматичний пошук опонентів для кожного раунду, що забезпечить справедливе та збалансоване жеребкування учасників. Програма повинна наочно відображати всі партії турніру.

Важливим аспектом є можливість виставлення результатів кожної гри та зручний перегляд як окремих партій, так і загального стану турніру. Система має

підтримувати функції збереження поточного турніру та його подальшого завантаження для продовження змагання.

Програмне забезпечення повинно автоматично підраховувати бали учасників за результатами їх ігор та формувати підсумковий список гравців, відсортований за кінцевими результатами. Додатково необхідно реалізувати веб-компонент для відображення результатів та турнірної інформації онлайн, що дозволить учасникам та глядачам отримувати актуальні дані в режимі реального часу.

Реалізація всіх зазначених задач забезпечить створення комплексного рішення для організації шахових турнірів, яке значно спростить роботу організаторів та покращить досвід учасників змагань.

1.2. Огляд існуючого аналогу програмного забезпечення

На сучасному етапі розвитку цифрових технологій зростає потреба у спеціалізованому програмному забезпеченні для автоматизації проведення турнірів у різних видах спорту, зокрема у шахах. Ведення шахового турніру вручну або за допомогою загального призначення програм (наприклад, електронних таблиць) вимагає значних зусиль з боку організатора, підвищує ймовірність помилок та ускладнює оперативне внесення змін. У зв'язку з цим усе більшого значення набувають програмні продукти, здатні автоматизувати процес реєстрації учасників, формування турів, обробки результатів, нарахування очок, виведення турнірних таблиць тощо.

У світі існує низка комерційних рішень для таких задач, однак більшість з них є платними або мають суттєві обмеження у безкоштовних версіях. Безкоштовне або умовно безкоштовне програмне забезпечення з відкритим вихідним кодом, яке б повноцінно підтримувало процеси організації шахових турнірів відповідно до міжнародних стандартів, на даний момент майже відсутнє. Тому при створенні власного програмного продукту доцільно

орієнтуватися на наявні рішення, зокрема веб-додатки, які забезпечують потрібну функціональність без необхідності встановлення локального ПЗ.

У рамках даного підрозділу буде розглянуто Chess-Results – один із найвідоміших веб-додатків для організації шахових турнірів, який використовується на міжнародному рівні. Платформа Chess-Results зарекомендувала себе як надійний інструмент для публікації результатів турнірів, рейтингових списків, жеребкування та турнірних таблиць. Завдяки онлайн-доступу, учасники та глядачі можуть оперативно переглядати результати, а організатори – швидко оновлювати інформацію про перебіг турніру.

Огляд можливостей Chess-Results дозволить виявити ключові функції, які користувачі очікують від сучасного програмного забезпечення, а також визначити недоліки чи обмеження, що будуть враховані під час розробки власного рішення. Серед важливих аспектів огляду – аналіз інтерфейсу користувача, доступного функціоналу, гнучкості налаштувань та відповідності міжнародним правилам проведення турнірів (зокрема, системі Швейцарського жеребкування).

Таким чином, аналіз Chess-Results слугуватиме відправною точкою для визначення технічних вимог до власного програмного забезпечення та дозволить уникнути поширених помилок при проєктуванні архітектури майбутнього додатку.

Головна сторінка сайту Chess-Results.com (рис. 1.1.) надає доступ до великої бази даних шахових турнірів і результатів. У верхній частині сторінки розміщена панель навігації, де можна вибрати мову інтерфейсу (доступні десятки мов, зокрема англійська, німецька тощо). Навігаційне меню містить посилання на головну сторінку, базу даних турнірів, розділ "ЧАВО", онлайн-реєстрацію, сторінки Swiss-Manager, ÖSB, FIDE та контактну інформацію.

Chess-Results.com
in close cooperation with the
Administrations-and Pairing-
program **Swiss-Manager**

Logged on: Gast Svertime 04.06.2025 14:55:03
 Arabic ARM AZE BIH BUL CAT CHN CRO CZE DEN ENG ESP FAI FIN FRA GER GRE INA ITA JPN MKD LTU NED POL POR ROU RUS SRB SVK SWE TUR UKR VIE FontSize:11pt Login Logout

[Главная](#) [Турнирная база данных](#) [AUT](#) [Фото](#) [ЧАВО](#) [Онлайн](#) [Регистрация](#) [Swiss-Manager](#) [ÖSB](#) [FIDE](#)

[Important tournaments](#) [Календарь](#) [Ссылки](#) [Technical News](#) [Контактная информация](#)

Добро пожаловать в турнирную базу данных chess-results.com

Chess-Results.com - это мощный специализированный сервер шахматных результатов.

Архив chess-results.com скоро будет насчитывать более 905.000 турниров со всего мира, от клубных до Европейских Чемпионатов и Олимпиад. Это стало возможным благодаря Swiss-Manager, программе администрирования шахматных турниров, которая работает в тесном взаимодействии с chess-results.com.

Поиск турнира

Поиск турниров а также отображение/печать /экспорт в Excel итоговых результатов, пар туров, кросс-таблиц и т.п.

Поиск игрока

Поиск всех турниров, в которых игрок принимал участие, а также отображение/печать/экспорт в Excel всех данных этого игрока и этих турниров.

Поиск партии

Поиск партий, просмотр в онлайн или же загрузка в PGN формате.

Контактная информация

Вопросы, предложения и замечания высылайте мне (Heinz Herzog) посредством E-mail на адрес : contact@chess-results.com (Please in English or German)

Running tournaments

06.06 - 09.06 17. Internationales Salzkammergut Schachopen 2025

Выбор федерации

ACF AFG AHO ALB ALG AND ANG ANT ARG ARM ARU AUS AUT (Wien NÖ Bgld OÖ Sbg Tirol Vbg Stmk Knt ÜRe) AZE BAH BAN BAR BDI BEL BEN BHU BIH BLR BOL BOT BRA BRN BRU BUL BUR CAF CAM CAN CAT CAY CCA CHI CHN CIV CMR COL CPV CRC CRO CUB CYP CZE DEN DJI DOM ECU ECU EGY ENG ESA ESP EST ETH FAI FIDE FJ FIN FRA GAB GAM GCI GEO GEQ GER GHA GIN GRE GUA GUM GUY HAI HKG HON HUN INA IND IOM IRI IRL IRQ ISL ISR ISV ITA IVB JAM JCI JOR JPN KAZ KEN KGZ KOR KOS KSA KUW LAO LAT LBA LBN LBR LCA LES LIE LTU LUX MAC MAD MAR MAS MAW MDA MDV MEX MGL MKD MLI MLT MNC MNE MOZ MRI MTN MYA NAM NCA NED NEP NGR NIG NOR NRU NZL OMA ONL PAK PAN PAR PER PHI PLE PLW PNG POL POR PUR QAT ROU RSA RUS RWA SCG SCO SEN SEY SGP SLE SLO SMR SOL SOM SRB SRI SSD STP SUD SUI SUR SVK SWE SWZ SYR TAN THA TJK TKM TLS TOG TON TPE TTO TUN TUR UAE UGA UKR URU USA UZB VAN VEN VIE WFCCWLS YEM ZAM ZIM

Выбор турнира

Все турниры

Сортировка в соответствии с

Последней загрузкой (по убыванию)

Ном.	Турнир ● Not started ● Playing ● Finalized	ФЕД	Последнее обновление
1	Первенство Удмуртской Республики по быстрым шахматам юноши до 17 лет	RUS	● Rp 0 Мин.
2	[K7XQ] K7 XIANGQI TRAINING (04.06.2025)	MAS	● Bz 0 Мин.
3	JIFMS-2025-XADREZ FEMININO	BRA	● Rp 0 Мин.
4	Квалификационный турнир на 4 разряд 03-04.06.2025, Жезказган	KAZ	● St 1 Мин.
5	2nd SXCCAA Rapid Rating Chess Tournament 2025@St. Xavier's College	IND	● Rp 1 Мин.
6	Первенство Приморского края по шахматам среди девочек до 11 лет (2016 г.р. и моложе)	RUS	● St 2 Мин.
7	Открытое первенство МАОУ СШ № 9 по шахматам среди мальчиков и девочек, юношей и девушек	RUS	● St 2 Мин.

Рис. 1.1. Головна сторінка сайту Chess-Results.com

У центральній частині розміщено привітальне повідомлення, яке пояснює, що Chess-Results.com є потужним спеціалізованим сервером шахових результатів, який містить понад 905 000 турнірів з усього світу. Сайт тісно

пов'язаний із програмою Swiss-Manager, яка використовується для адміністрування турнірів.

Нижче розміщені функціональні блоки. Розділ "Пошук турніра" дозволяє знайти турнір за назвою або експортувати його результати в Excel. Розділ "Пошук игрока" допомагає знайти гравця і переглянути всі турніри, в яких він брав участь. Також доступний "Пошук партії" — це функція для перегляду партій у форматі PGN.

Нижче розташований алфавітний список скорочень шахових федерацій світу, які ведуть до сторінок, присвячених турнірам відповідної країни.

У нижній частині сторінки представлена таблиця з останніми оновленнями турнірів. У таблиці вказано номер турніру, його назву, статус (позначений кольором: зелений — турнір йде, жовтий — у процесі, сірий — завершено), федерацію, а також час останнього оновлення.

Таким чином, головна сторінка Chess-Results є зручною точкою входу для перегляду шахових результатів, пошуку гравців, партій, а також доступу до всієї структури міжнародних змагань.

Перейдемо на сторінку турніру, наприклад: Півфінал 75-го чемпіонату Кривого Рогу з класичних шахів (рис. 1.2.). На сторінці вгорі зазначено дату та час останнього оновлення, а також ім'я користувача, який виконав завантаження даних.

У верхній частині також доступні різні параметри для перегляду та навігації по турніру. Є можливість відобразити додаткову інформацію або перейти за посиланням на календар турнірів. Доступні списки включають стартовий список (який відкрито на цій сторінці), список гравців за алфавітом, статистику учасників і розклад турів. Також є посилання на остаточне положення після 9 турів, стартову таблицю учасників, пари по дошках у кожному турі, а також варіант перегляду "без пари".

Додатково є розділ із положенням після кожного туру, починаючи з першого і до дев'ятого, а також опції експорту даних у форматах Excel або PDF-файл. Є також можливість згенерувати QR-коди.

Chess-Results.com
in close cooperation with the
Administrations-and Pairing-
program **Swiss-Manager**

Logged on: Gast Servertime 04.06.2025 14:49:52

Arabic ARM AZE BIH BUL CAT CHN CRO CZE DEN ENG ESP FAI FIN FRA GER GRE INA ITA JPN MKD LTU NED POL POR ROU RUS SRB SVK SWE TUR UKR UKR VIE FontSize:11pt Login Logout

Главная Турнирная база данных **AUT** **Фото** **ЧАВО** **Онлайн** **Регистрация** **Swiss-Manager** **ÖSB** **FIDE**

Півфінал 75-го чемпіонату Кривого Рогу з класичних шахів

Последнее обновление: 30.03.2025 12:53:44, Автор/Последняя загрузка: Kazmiruk Vadim

Выбор параметров [Отобразить дополнительную информацию](#), [Ссылка на турнирный календарь](#)

Списки [Стартовый список](#), [Список игроков по алфавиту](#), [Статистика](#), [Расписание](#)
[Итоговое положение после 9 туров](#), [Стартовая таблица участников](#)

Пары по доскам [Тур1](#), [Тур2](#), [Тур3](#), [Тур4](#), [Тур5](#), [Тур6](#), [Тур7](#), [Тур8](#), [Тур9/9](#), [без пары](#)

Положение после [Тур1](#), [Тур2](#), [Тур3](#), [Тур4](#), [Тур5](#), [Тур6](#), [Тур7](#), [Тур8](#), [Тур9](#)

Excel и Печать [Экспорт в Excel \(.xlsx\)](#), [Экспорт в PDF-файл](#), [QR-Codes](#)

Search for player [Поиск](#)

Стартовый список

Ном.	Имя	код FIDE	ФЕД.	Рейт.
1	Арутюнов, Степан	14144638	ДНІ	2081
2	Десятник, Денис	34170898	ДНІ	2017
3	Соколов, Віктор	34101900	ДНІ	1992
4	Герус, Євген	34150790	ДНІ	1983
5	Сергєєв, Сергій	5249	ДНІ	1982
6	Хомуха, Геннадій	34189602	ДНІ	1955
7	Новицький, Андрій	34101870	ДНІ	1915
8	Дзюба, Олег	529008760	ДНІ	1900
9	Журавльов, Артур	34193944	ДНІ	1900
10	Штефюк, Володимир		ДНІ	1900
11	Грищак, Сергій	34101810	ДНІ	1862
12	Швагер, Сергій	34196730	ДНІ	1821
13	Гуменюк, Крістіна	34191070	ДНІ	1809
14	Гінул, Олексій		ДНІ	1800
15	Жураківський, Петро		ДНІ	1800
16	Карапетян, Ерік		ДНІ	1800
17	Куделя, Олег	5955	ДНІ	1800
18	Посесор, Сергій		ДНІ	1800
19	Пришляк, Ганна		ДНІ	1800
20	Харкавенко, Дмитро		ДНІ	1800
21	Житков, Олександр	34101934	ДНІ	1790
22	Покровський, Олексій	5969	ДНІ	1780

сервер Chess-Tournament-Results © 2006-2025 Heinz Herzog, CMS-Version 04.06.2025 12:16
условия использования

Рис. 1.2. Сторінка турніру «Півфінал 75-го чемпіонату Кривого Рогу з класичних шахів»

Під цими параметрами знаходиться стартовий список учасників турніру. У таблиці зазначені номер, ім'я гравця, його код FIDE, шахова федерація, а також рейтинг.

Якщо обрати в меню «Тур 1», можна побачити конкретну інформацію про перший тур змагань (рис. 1.3.). Таблиця містить дані про десять ігрових пар. У

кожному рядку зазначено номер столу, імена гравців (білими та чорними фігурами), їхній рейтинг, кількість очок, здобутих у партії, і загальний результат.

Chess-Results.com
in close cooperation with the
Administrations-and Pairing-
program **Swiss-Manager**

Logged on: Gast Servertime 04.06.2025 15:10:06

Arabic ARM AZE BIH BUL CAT CHN CRO CZE DEN ENG ESP FAI FIN FRA GER GRE INA ITA JPN MKD LTU NED POL POR ROU **RUS** SRB SVK SWE TUR UKR VIE FontSize:11pt Login Logout

Главная Турнирная база данных **AUT** Фото ЧАВО Онлайн Регистрация **Swiss-Manager** **OSB** **FIDE**

Півфінал 75-го чемпіонату Кривого Рогу з класичних шахів
Последнее обновление 30.03.2025 12:53:44, Автор/Последняя загрузка: Kazmiruk Vadim

Выбор параметров [Отобразить дополнительную информацию, Ссылка на турнирный календарь](#)

Списки [Стартовый список, Список игроков по алфавиту, Статистика, Расписание](#)
[Итоговое положение после 9 туров, Стартовая таблица участников](#)

Пары по доскам [Тур1, Тур2, Тур3, Тур4, Тур5, Тур6, Тур7, Тур8, Тур9/9, без пары](#)

Положение после [Тур1, Тур2, Тур3, Тур4, Тур5, Тур6, Тур7, Тур8, Тур9](#)

Excel и Печать [Экспорт в Excel \(.xlsx\), Экспорт в PDF-файл, QR-Codes](#)

Search for player [Поиск](#)

Пары/Результаты
1. Тур on 2025/03/12 в 15 00

Во. Ном.	White	Рейт	Очки	Результат	Очки	Black	Рейт	Ном.
1 12	Швагер, Сергій	1821	0	0 - 1	0	Арутюнов, Степан	2081	1
2 2	Десятник, Денис	2017	0	0 - 1	0	Гуменюк, Крістіна	1809	13
3 14	Гінул, Олексій	1800	0	0 - 1	0	Соколов, Віктор	1992	3
4 4	Герус, Євген	1983	0	1 - 0	0	Жураківський, Петро	1800	15
5 16	Карапетян, Ерік	1800	0	0 - 1	0	Сергеев, Сергій	1982	5
6 6	Хомуха, Геннадій	1955	0	1 - 0	0	Куделя, Олег	1800	17
7 18	Посесор, Сергій	1800	0	0 - 1	0	Новицький, Андрій	1915	7
8 8	Дзюба, Олег	1900	0	1 - 0	0	Пришляк, Ганна	1800	19
9 20	Харкавенко, Дмитро	1800	0	1 - 0	0	Журавльов, Артур	1900	9
10 10	Штефюк, Володимир	1900	0	0 - 1	0	Житков, Олександр	1790	21
11 22	Покровський, Олексій	1780	0	½ - ½	0	Грищак, Сергій	1862	11

сервер Chess-Tournament-Results © 2006-2025 Heinz Herzog, CMS-Version 04.06.2025 12:16
условия использования

Рис. 1.3. Таблиця з інформацією про перший тур змагань

Ця сторінка турніру забезпечує повний доступ до всіх основних даних змагання — від складу учасників до результатів турів, що дозволяє легко слідкувати за перебігом шахового чемпіонату.

Перейдемо на сторінку гравця, натиснувши на будь-яке прізвище, наприклад, Арутюнов (рис. 1.4.). Ця сторінка відображає персональну інформацію та результати виступів гравця на турнірі «Півфінал 75-го чемпіонату Кривого Рогу з класичних шахів». Згори знову вказано назву турніру, дату останнього оновлення, час та відповідального за внесення даних. Присутня навігаційна панель, яка дозволяє перейти до інших розділів, таких як список

гравців, статистика, розклад, підсумкова таблиця, пари по турах і варіанти експорту результатів.

Chess-Results.com
in close cooperation with the
Administrations-and Pairing-
program **Swiss-Manager**

Logged on: Gast Servertime 04.06.2025 15:14:06

Arabic ARM AZE BIH BUL CAT CHN CRO CZE DEN ENG ESP FAI FIN FRA GER GRE INA ITA JPN MKD LTU NED POL POR ROU **RUS** SRB SVK SWE TUR UKR VIE FontSize: 11pt Login Logout

[Главная](#) [Турнирная база данных](#) [AUT](#) [Фото](#) [ЧАВО](#) [Онлайн](#) [Регистрация](#) [Swiss-Manager](#) [ÖSB](#) [FIDE](#)

Півфінал 75-го чемпіонату Кривого Рогу з класичних шахів
 Последнее обновление: 30.03.2025 12:53:44, Автор/Последняя загрузка: Kazmiruk Vadim

Выбор параметров [Отобразить дополнительную информацию, Ссылка на турнирный календарь](#)

Списки [Стартовый список, Список игроков по алфавиту, Статистика, Расписание](#)
[Итоговое положение после 9 туров, Стартовая таблица участников](#)

Пары по доскам [Тур1, Тур2, Тур3, Тур4, Тур5, Тур6, Тур7, Тур8, Тур9/9, без пары](#)

Положение после [Тур1, Тур2, Тур3, Тур4, Тур5, Тур6, Тур7, Тур8, Тур9](#)

Excel и Печать [Экспорт в Excel \(.xlsx\), Экспорт в PDF-файл, QR-Codes](#)

Инфо игрока

Имя [Арутюнов, Степан](#)

Стартовое место [1](#)

Рейтинг [2081](#)

Нац.рейтинг [2081](#)

Междун. рейтинг [2081](#)

Рейтинговый перфоманс [2062](#)

FIDE рейт +/- [0](#)

Очки [6,5](#)

Место [1](#)

Федерация [ДНІ](#)

Идент.Номер [0](#)

код FIDE [14144638](#)

Год рождения [1965](#)

Тур	Во.	Ст.ном.	Имя	Рейт.	ФЕД.	Очки	Рез.
1	1	12	Швагер, Сергій	1821	ДНІ	6	■ 1
2	1	7	Новицький, Андрій	1915	ДНІ	4,5	□ ½
3	2	20	Харкавенко, Дмитро	1800	ДНІ	3,5	■ ½
4	4	9	Журавльов, Артур	1900	ДНІ	5,5	□ 1
5	2	4	Герус, Євген	1983	ДНІ	6,5	■ 1
6	1	11	Гришак, Сергій	1862	ДНІ	5	□ 1
7	1	5	Сергеев, Сергій	1982	ДНІ	6	■ ½
8	1	3	Соколов, Віктор	1992	ДНІ	5,5	□ ½
9	1	13	Гуменюк, Крістіна	1809	ДНІ	5	■ ½

*) Rating difference of more than 400. It was limited to 400.

сервер **Chess-Tournament-Results** © 2006-2025 Heinz Herzog, CMS-Version 04.06.2025 12:16
 условия использования

Рис. 1.4. Сторінка гравця «Арутюнов Степан»

У центральній частині сторінки подано детальну інформацію про гравця: його стартовий номер у турнірі, поточний рейтинг і національний рейтинг, а також міжнародний рейтинг та іншу особисту інформацію.

Нижче представлено таблицю з результатами всіх дев'яти турів, у яких брав участь Арутюнов. Таблиця містить інформацію про тур, номер столу, ім'я суперника, його рейтинг, федерацію, набрані ним очки та результат партії.

Серед основних плюсів сайту Chess-Results: зручна навігація, підтримка багатьох мов, автоматизоване формування турів за швейцарською системою, швидке оновлення результатів, доступність для учасників та глядачів у режимі онлайн, сумісність із Swiss-Manager.

До мінусів можна віднести: застарілий інтерфейс, обмежену гнучкість у налаштуваннях, відсутність мобільного застосунку, складність інтеграції в інші системи та відсутність відкритого вихідного коду.

1.3. Вхідна інформація

Вхідна інформація — це дані, які формуються на початковій стадії розробки програмного забезпечення. Вони являють собою сукупність первинних та довідкових даних, необхідних для реалізації процесу автоматизованого жеребкування учасників, ведення турніру та обліку результатів шахових партій [4, с.36]. Ця інформація є основою для подальшої обробки, збереження та виведення результатів.

До складу вхідних даних належать:

1. Ім'я та прізвище учасника.
2. Кількість раундів.
3. Назва турніру.
4. Завантаження гри.
5. Результат гри.

Ці дані є необхідними для формування пар гравців, відстеження результатів, збереження інформації про зіграні партії та формування турнірної таблиці.

Задачі, на вирішення яких спрямоване проектування програмного забезпечення та бази даних представлені у табл. 1.1.

Таблиця 1.1

Задачі програмного забезпечення

№	Найменування задачі	Призначення задачі	Періодичність виконання
1	Ведення учасників	Додавання і видалення даних про гравців	Один раз на початку
2	Введення параметрів турніру	Встановлення назви турніру та кількості раундів	Один раз на початку
3	Формування пар	Автоматичне жеребкування учасників для кожного раунду	Перед кожним раундом
4	Занесення результатів	Фіксація результатів зіграних партій	Після кожного раунду
5	Виведення інформації	Відображення поточних результатів, пар, статистики	Постійно
6	Експорт даних на вебсайт	Передача підсумкових результатів на зовнішній ресурс	Після завершення турніру

Дані про учасників, параметри турніру та сформовані пари зберігаються у базі даних, що забезпечує цілісність та доступність інформації впродовж усього турніру.

Окрім цього, експорт результатів турніру на вебсайт здійснюється безпосередньо з бази даних, що дозволяє оперативно оновлювати інформацію в онлайн-режимі без дублювання чи додаткового введення даних.

1.4. Вихідна інформація

Вихідна інформація є результатом обробки вхідних даних та запитів до бази даних. Вона містить як поточні, так і підсумкові результати турніру, а також інформацію для візуалізації та подальшого аналізу. У даному програмному забезпеченні вихідною інформацією є табличне представлення списків учасників, сформованих партій, результатів раундів, а також графічне відображення логотипу.

Також до вихідної інформації належать збережені ігри, підсумкові результати турніру та експортовані на веб-сайт дані.

Згідно з табл. 1.2, можемо переглянути перелік і опис вихідних повідомлень.

Таблиця 1.2

Перелік і опис вихідних повідомлень

№	Назва вихідного повідомлення	Призначення	Форма представлення	Термін і частота надходження
1	Логотип у вигляді 3D-фігури	Візуальне оформлення інтерфейсу програми	Графічна	Постійно
2	ID гравця	Унікальне позначення учасника у базі даних	Текстова	При кожному запиті
3	Список учасників	Відображення зареєстрованих гравців турніру	Текстова	При запуску програми або за запитом
4	Список партій (пар учасників)	Показ сформованих пар на раунд	Текстова	Перед кожним раундом
5	Номер поточного раунду	Відображення етапу турніру	Числова	Перед початком кожного раунду
6	Поточні результати	Список результатів зіграних партій	Текстова	Після завершення кожного раунду
7	Збереження гри	Архівування інформації про поточний стан турніру	Базована на СУБД	Після кожного раунду
8	Кінцеві результати турніру	Підсумковий рейтинг учасників	Текстова	Після завершення турніру
9	Дані на веб-сайті	Публікація турнірної інформації онлайн	Таблична (у вигляді HTML-сторінки)	Після завершення або за запитом

Таким чином, вихідна інформація в розробленому програмному забезпеченні відіграє ключову роль у відображенні актуального стану турніру, включаючи сформовані пари учасників, результати зіграних партій, поточні турнірні таблиці та підсумкові результати. Уся інформація формується в результаті запитів до бази даних і відображається у зручній табличній або графічній формі. Збереження стану гри реалізоване через базу даних, що дозволяє здійснювати резервне копіювання та подальше відновлення даних. Крім того, передбачена можливість експорту даних безпосередньо з бази до

вебсайту, що забезпечує оперативне інформування учасників та організаторів турніру.

Висновки до розділу 1

У даному розділі було проаналізовано основні задачі, які постають при розробці програмного забезпечення для автоматизованого проведення шахових турнірів, оглянуто аналог веб-додатку, а також визначено вхідні та вихідні дані, необхідні для реалізації функціоналу системи.

Було встановлено, що ефективне програмне забезпечення повинно охоплювати весь цикл організації турніру — від введення основних параметрів до підведення підсумків та публікації результатів. Зокрема, до ключових задач належать: ведення списку учасників, формування пар для кожного раунду, реєстрація результатів партій, обчислення підсумкових балів і формування рейтингових таблиць.

Вхідна інформація представлена персональними даними учасників, параметрами турніру та результатами ігор, які зберігаються у базі даних для забезпечення цілісності та доступності. Вихідна інформація включає у себе як текстові, так і графічні дані, зокрема списки учасників, пари гравців, поточні та фінальні результати, а також логотип у вигляді 3D-фігури. Особливу увагу приділено збереженню гри та можливості експорту результатів на вебсайт у зручному форматі.

Таким чином, сформульовані вимоги до вхідних і вихідних даних дозволяють чітко визначити структуру та функціональні особливості програмного забезпечення, що забезпечить його надійність, зручність у використанні та ефективність в організації шахових турнірів.

РОЗДІЛ II

РОЗРОБКА АЛГОРИТМУ РОЗВ'ЯЗАННЯ ЗАДАЧІ НА МОВІ ПРОГРАМУВАННЯ C++

Розробка алгоритмічного забезпечення є ключовим етапом у створенні програмного продукту. На цьому етапі відбувається побудова логічної структури програми, визначення основних функціональних блоків, їх взаємозв'язків, а також розробка ефективних способів обробки вхідних і виведення вихідних даних.

Для реалізації алгоритмів використовується мова програмування C++ із застосуванням бібліотеки MFC, що забезпечує створення зручного графічного інтерфейсу користувача. Також до проєкту інтегровано OpenGL для візуального представлення 3D-логотипу, а базу даних використовують для зберігання та обробки інформації про гравців, турніри, результати партій.

Розроблений алгоритм має бути не лише функціональним, але й оптимізованим з точки зору продуктивності, гнучкості та масштабованості, що дозволить ефективно використовувати програмне забезпечення як у невеликих, так і в масштабних змаганнях.

Керування програмою здійснюється за допомогою елементів головного меню (див. рис. 2.1).

Щоб увійти в діалогові вікна «Реєстрація учасників» або «Формування раундів» - натискаємо на меню «Жеребкування» (рис. 2.2.).

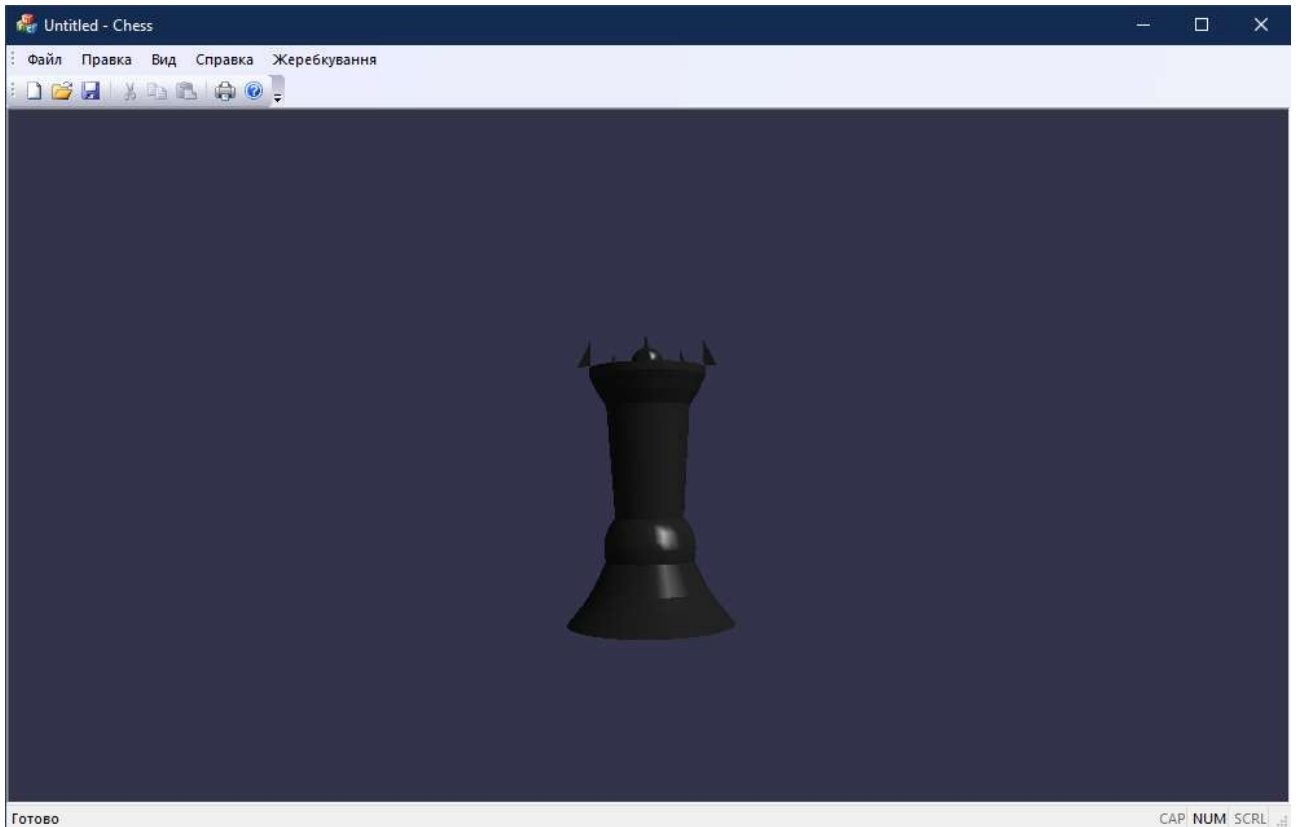


Рис. 2.1. Вигляд меню програми

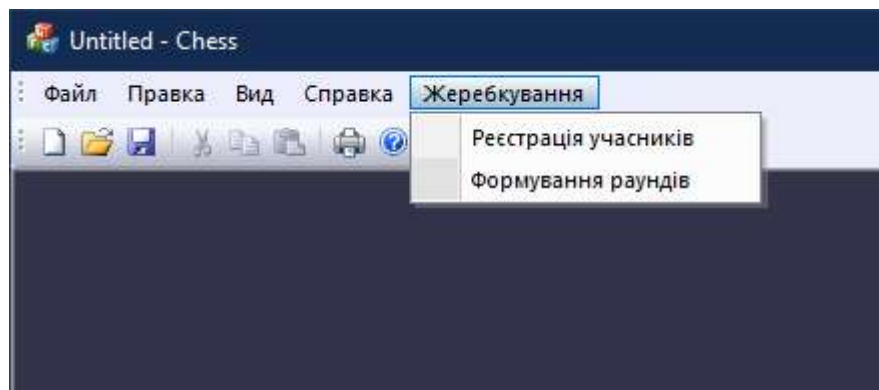


Рис. 2.2. Меню «Жеребкування»

Складемо алгоритм розв'язання задачі (рис. 2.3.), що дозволить реалізувати логіку роботи програмного забезпечення. Алгоритм охоплює основні етапи функціонування програми, зокрема реєстрацію учасників, формування пар, збереження та обробку результатів турніру.

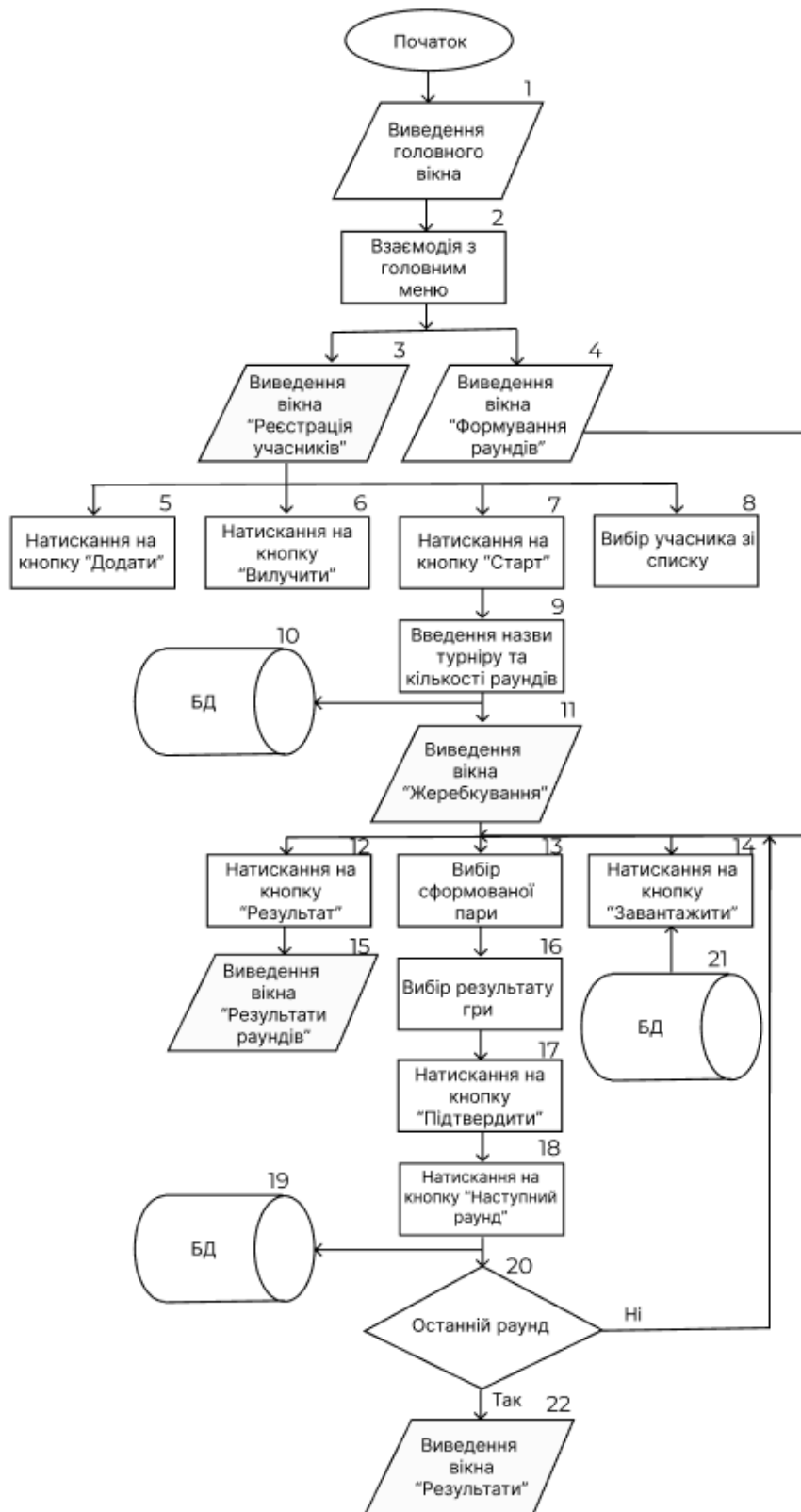


Рис. 2.3. Алгоритм роботи програми

Алгоритм описує послідовність роботи користувача з програмним забезпеченням для організації шахового турніру. Нижче подано опис етапів його виконання:

1. Початок роботи. Програма запускається, після чого відбувається виведення головного вікна (крок 1).

2. Користувач починає взаємодію з головним меню (крок 2), що дозволяє перейти до різних функціональних вікон:

- Вікно "Реєстрація учасників" (крок 3).
- Вікно "Формування раундів" (крок 4).

Реєстрація учасників:

3. У вікні реєстрації можливі такі дії:

- Натискання кнопки "Додати" (крок 5) — додає нового учасника.
- Натискання кнопки "Вилучити" (крок 6) — видаляє вибраного учасника зі списку.

• Вибір учасника зі списку (крок 8) — необхідно для перегляду даних про учасника або взаємодії з ним.

- Дані зберігаються у базі даних (крок 10).

Параметри турніру:

4. У вікні формування раундів:

- Натискається кнопка "Старт" (крок 7).
- Користувач вводить назву турніру та кількість раундів (крок 9).
- Виводиться вікно "Жеребкування" (крок 11), де генеруються пари для гри.

Проведення раундів:

5. Далі користувач може:

• Натиснути кнопку "Результат" (крок 12) — виводиться вікно результатів поточного раунду (крок 15).

- Обрати сформовану пару (крок 13), після чого:

a) Обрати результат гри (крок 16).

b) Натиснути "Підтвердити" (крок 17) — результат зберігається.

- с) Натиснути "Наступний раунд" (крок 18).
 - Натиснути "Завантажити" (крок 14), щоб відновити збережену гру з бази даних (крок 21).
6. Після завершення кожного раунду дані зберігаються в базу даних (крок 19).

Завершення турніру:

7. Відбувається перевірка — чи останній це раунд (крок 20):
- Якщо ні — починається новий раунд (повторення з жеребкування).
 - Якщо так — виводиться вікно з кінцевими результатами турніру (крок 22).

Таким чином, алгоритм забезпечує повний цикл проведення шахового турніру: від реєстрації учасників і формування раундів до занесення результатів та завершення змагання з відображенням підсумків. Усі ключові дані зберігаються в базі даних, що дозволяє зберігати та відновлювати стан турніру.

Висновки до розділу 2

У результаті розробки алгоритмічного забезпечення було створено чітку логічну структуру функціонування програмного забезпечення для проведення шахового турніру. Побудований алгоритм охоплює всі основні етапи взаємодії користувача з програмою — від запуску головного вікна до завершення турніру з виведенням підсумкових результатів.

Реалізація алгоритму на мові C++ з використанням бібліотеки MFC забезпечує зручний інтерфейс для реєстрації учасників, організації раундів та обробки результатів. Інтеграція з базою даних дозволяє надійно зберігати інформацію про хід турніру, а також забезпечує можливість відновлення попередніх станів системи.

Застосування OpenGL дозволило реалізувати візуальну частину, зокрема 3D-логотип програми, що підвищує привабливість інтерфейсу.

Таким чином, розроблений алгоритм є повноцінною основою для реалізації програмного продукту, який відповідає вимогам функціональності, наочності, стабільності та масштабованості. Це дозволяє ефективно використовувати програму як у невеликих локальних турнірах, так і в більш складних багатокористувацьких змаганнях.

РОЗДІЛ III

ОРГАНІЗАЦІЯ ІНФОРМАЦІЙНОГО ЗАБЕЗПЕЧЕННЯ

3.1. Загальна характеристика інформаційного забезпечення

Інформаційне забезпечення є однією з найважливіших складових будь-якої автоматизованої системи управління. Воно являє собою сукупність єдиної системи класифікації та кодування інформації, уніфікованих систем документації, схем інформаційних потоків, що циркулюють в організації, а також методології побудови баз даних. Ефективність функціонування інформаційної системи безпосередньо залежить від якості організації інформаційного забезпечення [8, с.134].

Інформація у контексті інформаційних систем визначається як сукупність відомостей, даних та знань, які сприймаються з навколишнього середовища, обробляються, зберігаються та передаються в межах системи для прийняття управлінських рішень. Інформація має властивості актуальності, достовірності, повноти, релевантності та своєчасності. Ці характеристики визначають якість інформаційного забезпечення та ефективність його використання [9, с.35].

У структурі інформаційного забезпечення програмного комплексу для жеребкування шахових турнірів можна виділити кілька основних компонентів. Вхідна інформація включає первинні дані про учасників турніру, параметри турніру такі як кількість раундів та назва. Ця інформація надходить від організаторів турніру та самих учасників на етапі реєстрації та налаштування змагання.

Проміжна інформація формується в процесі роботи системи та включає дані про поточний стан турніру, результати завершених партій, турнірну таблицю, статистичні показники учасників. Ця інформація постійно оновлюється протягом проведення турніру та служить основою для прийняття рішень щодо жеребкування наступних турів.

Вихідна інформація представляє собою результати роботи системи, які надаються користувачам у вигляді звітів, турнірних таблиць, розкладу партій, статистичних аналізів. Ця інформація може бути представлена у різних форматах залежно від потреб користувачів та може виводитися як на екран, так і в друкованому вигляді або експортуватися в зовнішні системи.

Організація інформаційного забезпечення базується на принципах системності, достовірності, своєчасності, повноти та економічності [13, с.54]. Принцип системності передбачає розгляд інформаційного забезпечення як єдиного цілого з урахуванням всіх взаємозв'язків між його елементами. Достовірність інформації забезпечується через контроль введення даних, валідацію інформації та використання надійних джерел.

Своєчасність інформаційного забезпечення означає надання необхідної інформації в потрібний момент часу для прийняття управлінських рішень. У контексті шахових турнірів це особливо важливо, оскільки затримка в обробці результатів може призвести до порушення розкладу змагань.

Повнота інформаційного забезпечення передбачає наявність всієї необхідної інформації для ефективного функціонування системи. Економічність означає оптимальне співвідношення між витратами на створення та підтримку інформаційного забезпечення та отриманим ефектом від його використання.

Класифікація інформації в системі жеребкування шахових турнірів може здійснюватися за різними критеріями. За джерелом походження інформація поділяється на внутрішню, яка генерується самою системою, та зовнішню, що надходить від користувачів або інших систем. За ступенем обробки розрізняють первинну інформацію, яка вводиться безпосередньо користувачами, та похідну, що формується в результаті обробки первинних даних.

За характером використання інформація може бути постійною, яка зберігається протягом тривалого часу та використовується багаторазово, та змінною, що регулярно оновлюється відповідно до поточного стану системи. За ступенем важливості виділяють критично важливу інформацію, втрата якої може

призвести до серйозних наслідків, та допоміжну інформацію, що використовується для покращення функціональності системи.

Кодування інформації в системі здійснюється з метою забезпечення однозначності ідентифікації об'єктів та їх характеристик. Система кодування повинна бути простою для розуміння, економічною за обсягом та гнучкою для розширення. У контексті шахових турнірів використовуються коди для ідентифікації учасників, партій, турнірів та інших об'єктів системи.

Захист інформації є критично важливим аспектом інформаційного забезпечення. Він включає фізичний захист обладнання та носіїв інформації, логічний захист через системи автентифікації та авторизації, криптографічний захист для забезпечення конфіденційності даних та організаційний захист через розробку та дотримання відповідних процедур.

Система резервного копіювання забезпечує збереженість інформації у випадку збоїв обладнання або програмного забезпечення. Регулярне створення резервних копій та їх тестування є обов'язковими компонентами надійного інформаційного забезпечення.

Контроль якості інформації здійснюється на всіх етапах її життєвого циклу. Це включає контроль введення даних через валідацію полів, контроль обробки через перевірку алгоритмів та контроль виведення через аналіз результатів. Автоматичні засоби контролю доповнюються ручними процедурами перевірки критично важливої інформації.

Стандартизація інформаційного забезпечення передбачає використання загальноприйнятих форматів даних, протоколів обміну інформацією та методів організації баз даних. Це забезпечує сумісність з іншими системами та полегшує процес розробки та супроводу програмного забезпечення.

Моніторинг інформаційного забезпечення включає постійне спостереження за станом інформаційних ресурсів, аналіз ефективності їх використання та виявлення потенційних проблем. Система моніторингу повинна забезпечувати своєчасне попередження про критичні ситуації та надавати інструменти для їх усунення.

Інтеграція з зовнішніми системами дозволяє розширити функціональність інформаційного забезпечення та забезпечити обмін даними з іншими програмними комплексами. Це може включати в себе бази даних шахових федерацій.

Метадані описують структуру та властивості інформаційних ресурсів, що полегшує їх пошук, використання та управління. Система метаданих повинна бути достатньо детальною для ефективного управління інформацією, але не надто складною для практичного використання.

3.2. Структура баз даних та інформаційних масивів

У розроблюваному програмному забезпеченні для жеребкування шахових турнірів використовується система управління базами даних MySQL. MySQL - це відкрита реляційна система управління базами даних, яка розроблена шведською компанією MySQL AB у 1995 році. Вона належить до найпопулярніших СУБД у світі завдяки своїй надійності, високій продуктивності, простоті використання та безкоштовності для некомерційного використання [14, с.135].

MySQL підтримує стандарт SQL (Structured Query Language) та забезпечує всі основні можливості реляційних баз даних, включаючи транзакції, зовнішні ключі, представлення, збережені процедури та тригери. СУБД працює за архітектурою клієнт-сервер, що дозволяє організувати централізоване зберігання даних з можливістю одночасного доступу множини користувачів.

Основними перевагами MySQL є її кросплатформність, підтримка різних механізмів зберігання даних (InnoDB, MyISAM, MEMORY та інші), масштабованість, безпека та інтеграція з різними мовами програмування. MySQL підтримує реплікацію даних, кластеризацію та має вбудовані засоби резервного копіювання та відновлення.

Для потреб системи жеребкування створюється база даних "Chess", яка включає чотири основні таблиці. Таблиця "Tournament" (табл. 3.1) призначена

для зберігання інформації про турніри, включаючи назву та кількість раундів. Таблиця "Participant" (табл. 3.2) містить дані про учасників турнірів, їх персональну інформацію.

Таблиця "Result" (табл. 3.3) зберігає результати всіх партій, включаючи інформацію про учасників, результат гри та номер туру. Таблиця "TablesConnection" (табл. 3.4) реалізує зв'язок типу "багато до багатьох" між учасниками та турнірами, оскільки кожен учасник може брати участь у декількох турнірах, а кожен турнір може включати багато учасників.

Таблиця 3.1

Опис структури таблиці «Tournament»

Найменування поля	Поле	Тип поля	Довжина	Кількість знаків після коми	Первинний ключ	Умова на значення	Обов'язкове поле	Індексне поле
Номер турніру	id_tournament	Лічильник	-	-	+	-	+	+
Назва турніру	name_tournament	Текстовий	100	-	-	-	-	-
Кількість раундів	round_quantity	Цілочисловий	-	-	-	-	-	-

Таблиця 3.2

Опис структури таблиці «Participant»

Найменування поля	Поле	Тип поля	Довжина	Кількість знаків після коми	Первинний ключ	Умова на значення	Обов'язкове поле	Індексне поле
Номер учасника	id_participant	Лічильник	-	-	+	-	+	+
Ім'я	firstname	Текстовий	20	-	-	-	-	-
Прізвище	lastname	Текстовий	20	-	-	-	-	-

Таблиця 3.3

Опис структури таблиці «Result»

Найменування поля	Поле	Тип поля	Довжина	Кількість знаків після коми	Первинний ключ	Умова на значення	Обов'язкове поле	Індексне поле
Номер результату	id_result	Лічильник	-	--	+	-	+	+
Номер туру	tour	Цілочисловий	-	-	-	-	-	-
Учасник-білими	participant_white	Цілочисловий	-	-	-	-	-	-
Учасник-чорними	participant_black	Цілочисловий	-	-	-	-	-	-
Результат білих	result_white	Цілочисловий	-	-	-	$0 \leq \text{значення} \leq 2$	-	-
Номер турніру	id_tournament	Цілочисловий	-	-	-	FOREIGN KEY	+	-

Таблиця 3.4

Опис структури таблиці «TablesConnection»

Найменування поля	Поле	Тип поля	Довжина	Кількість знаків після коми	Первинний ключ	Умова на значення	Обов'язкове поле	Індексне поле
Номер турніру	id_tournament	Цілочисловий	-	-	+(разом)	-	+	+
Номер учасника	id_participant	Цілочисловий	-	-	+(разом)	-	+	+

Структура бази даних побудована на чотирьох взаємопов'язаних таблицях, які забезпечують зберігання інформації про турніри, учасників, результати партій та їх зв'язки. Така модель дозволяє ефективно реалізувати логіку

шахового турніру, забезпечуючи цілісність, масштабованість і зручність доступу до даних.

Висновки до розділу 3

Інформаційне забезпечення є фундаментальною складовою системи жеребкування шахових турнірів, що забезпечує повний цикл обробки даних — від введення первинної інформації до формування результатів змагань та їх збереження.

Інформація в системі класифікується за джерелом, рівнем обробки, характером використання та ступенем важливості, що дозволяє ефективно керувати різними видами даних та забезпечувати їх якісну обробку.

Основними принципами організації інформаційного забезпечення є:

- системність (узгодженість усіх компонентів);
- достовірність (валідація і контроль введення даних);
- своєчасність (оперативне оновлення та обробка);
- повнота (відсутність прогалин в даних);
- економічність (ефективне використання ресурсів).

Для зберігання інформації використовується СУБД MySQL, яка є надійною, продуктивною та підтримує транзакції, реплікацію, масштабування і резервне копіювання, що особливо важливо для стабільної роботи під час змагань.

База даних "Chess" має чітку структуровану модель, що включає:

- таблицю "Tournament" для зберігання загальної інформації про турніри;
- таблицю "Participant" з персональними даними гравців;
- таблицю "Result", яка фіксує результати кожної партії;
- таблицю "TablesConnection", що реалізує зв'язок між гравцями і турнірами за принципом "багато до багатьох", запобігаючи дублюванню даних.

Стандартизація та моніторинг інформаційних ресурсів сприяють підвищенню сумісності з іншими системами, забезпечують контроль якості

інформації протягом усього її життєвого циклу, а також своєчасне виявлення і усунення можливих проблем.

РОЗДІЛ IV

РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ЗАДАЧІ

4.1. Опис головного модулю програми (головного вікна) на мові C++

Головне вікно програми — це основне вікно, яке відображається після запуску застосунку і є центральним елементом його інтерфейсу. У більшості випадків саме це вікно визначає логіку завершення роботи програми: коли користувач його закриває, додаток повністю завершує роботу. У середовищі Windows головне вікно може бути як звичайним вікном (наприклад, у WinAPI чи MFC з використанням CMainFrame), так і діалоговим вікном у діалогових застосунках. У MFC головне вікно зазвичай задається через змінну `m_pMainWnd`, і саме воно отримує фокус, обробляє повідомлення, а також може виступати батьківським для інших дочірніх елементів інтерфейсу.

Головне вікно нашої програми можна побачити на рис. 4.1.

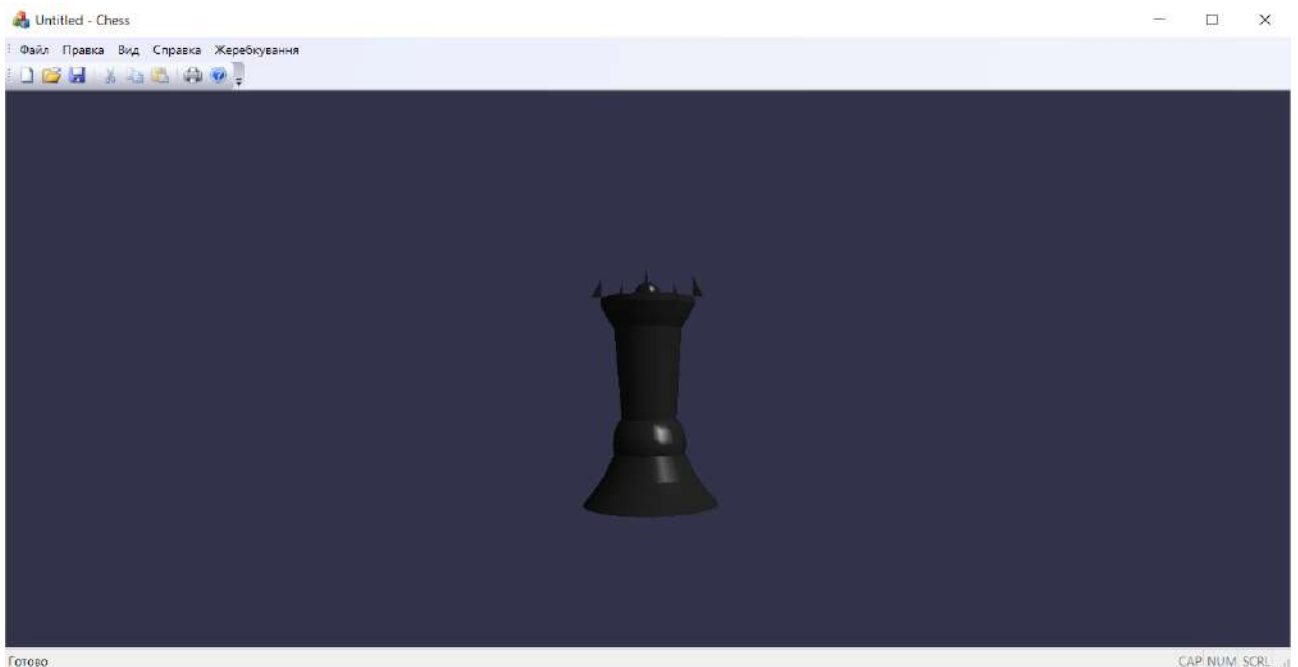


Рис. 4.1. Головне вікно програми

На рисунку представлена тривимірна (3D) шахова фігура, змодельована з використанням бібліотеки OpenGL, яка є потужним інструментом для побудови

графіки в реальному часі. Фігура має реалістичну форму, об'єм, світло та тіні, що дозволяє користувачу отримати візуально привабливе уявлення про її вигляд у віртуальному просторі. У проєкті реалізовано можливість інтерактивної взаємодії з об'єктом: за допомогою натискання клавіш клавіатури фігура може змінювати своє положення у тривимірному просторі — переміщатися по осі X, Y чи Z.

Це дозволяє наочно дослідити її геометрію з різних ракурсів, а також додає динаміки візуалізації. Такий підхід є корисним при створенні візуалізацій для ігор, навчальних програм або демонстраційних моделей. Код, який відповідає за обробку натискань клавіш і зміну координат фігури у просторі, наведено у додатку А. Він демонструє, як за допомогою простих інструкцій можна керувати об'єктами в OpenGL, забезпечуючи плавність руху та інтерактивність сцени.

Зверху в меню ми бачимо пункт «Жеребкування», який відкриває нам наступні підменю: «Реєстрація учасників» та «Формування раундів» (рис. 4.2.).

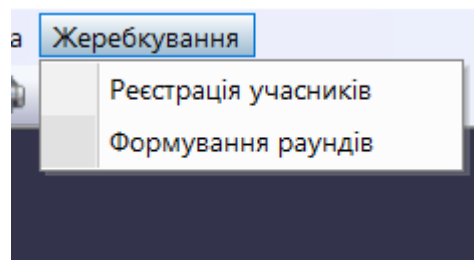


Рис. 4.2. Підменю «Жеребкування»

При відкритті діалогового вікна «Реєстрація учасників» (рис. 4.3.) бачимо інформацію про учасників. Це вікно призначене для введення, редагування та керування списком учасників, які беруть участь у шаховому турнірі.

У лівій частині вікна зображений список (ListBox), де відображені прізвища зареєстрованих учасників, відсортовані за алфавітом. Поточним учасником обрано першого гравця – Бабанського.

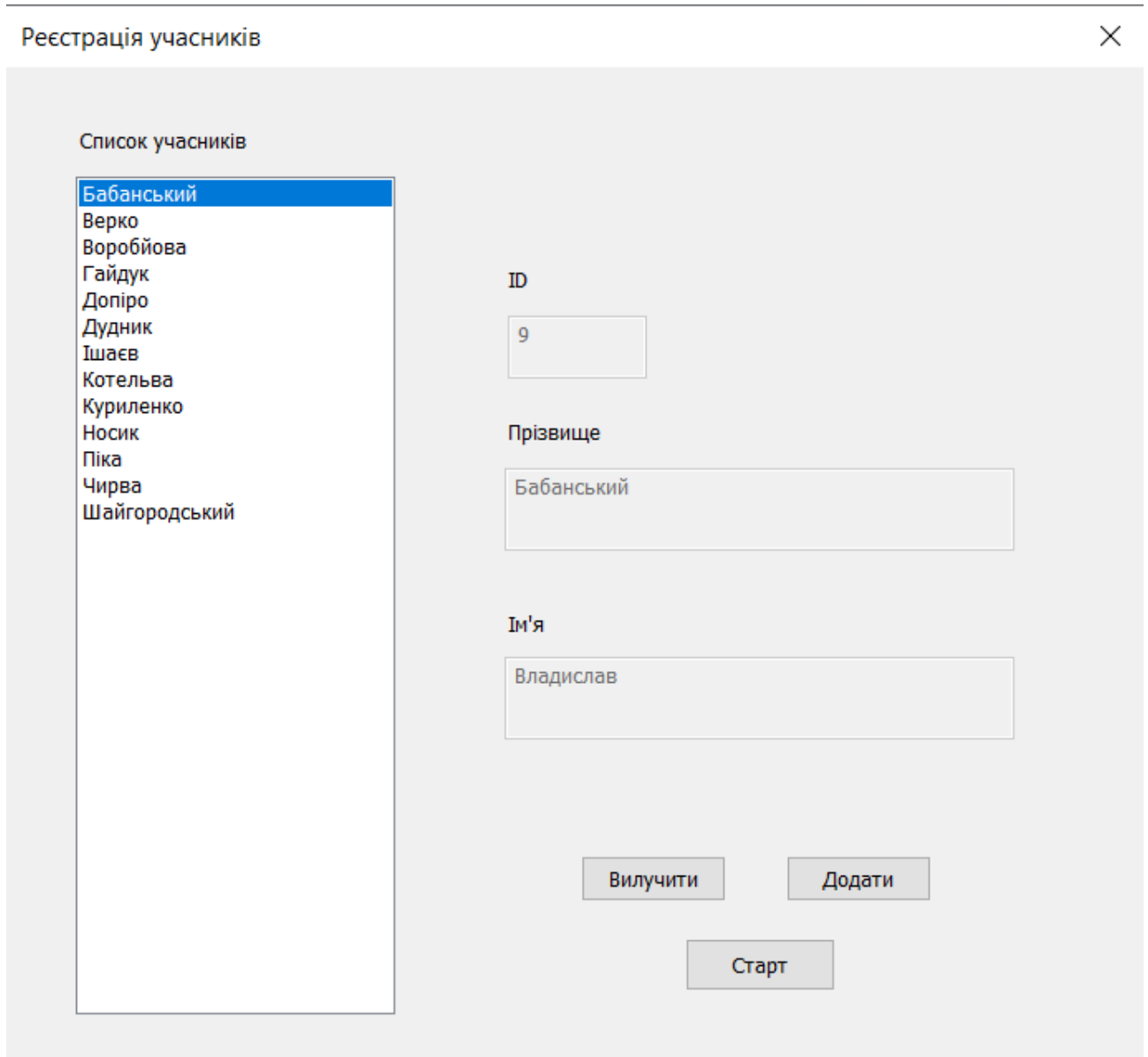


Рис. 4.3. Діалогове вікно «Реєстрація учасників»

У правій частині вікна можна побачити наступні елементи (поля введення або виводу):

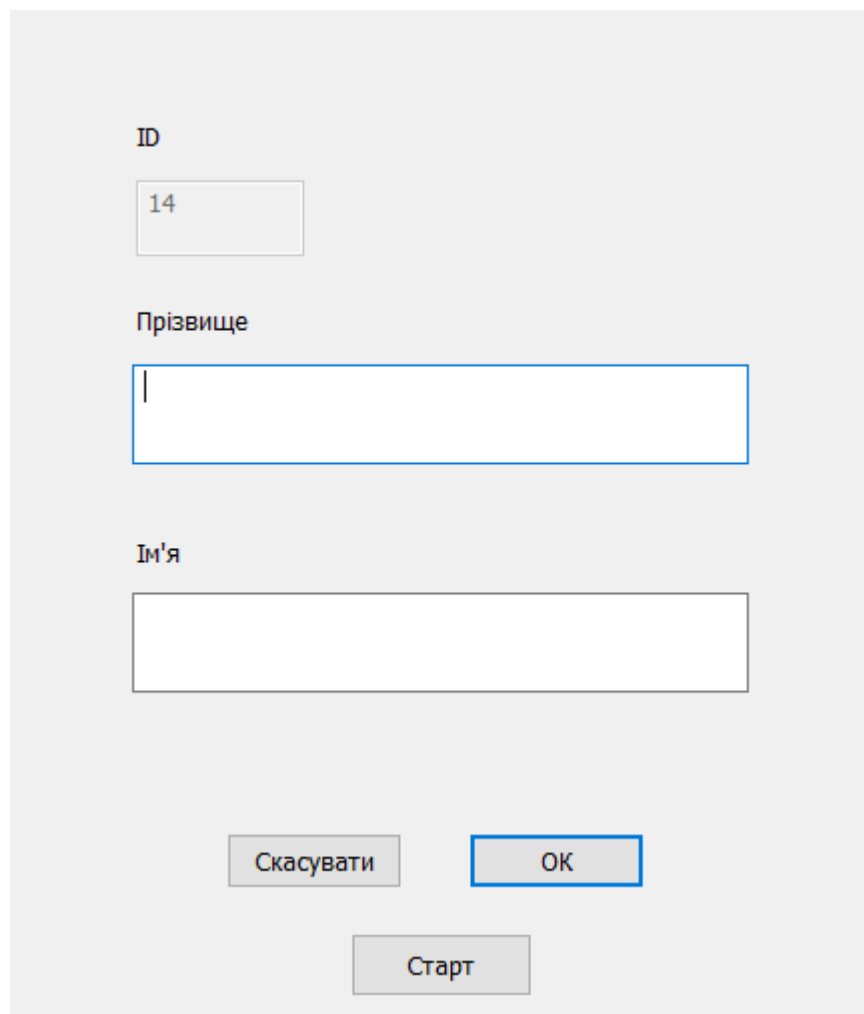
1. ID — цифрове поле, що відображає унікальний ідентифікатор користувача. Воно є заблокованим для редагування, оскільки генерується автоматично.
2. Прізвище — текстове поле для введення або відображення прізвища учасника.
3. Ім'я — текстове поле для введення або відображення імені учасника.

В нижній частині правої панелі бачимо кнопки керування, якими можна редагувати даний список:

1. «Вилучити» — видаляє вибраного учасника зі списку.
2. «Додати» — додає нового учасника на основі заповнених полів «Ім'я» та «Прізвище».
3. «Старт» — запускає наступний етап формування раундів для початку змагання.

Загалом це вікно є зручним інтерфейсом для попередньої реєстрації гравців і базового редагування інформації про них перед початком турніру.

Додамо нового учасника до списку, натиснувши кнопку «Додати». Бачимо, як в правій частині вікна поля для імені та прізвища очистились, тепер їх можна заповнювати новим учасником (рис. 4.4.).

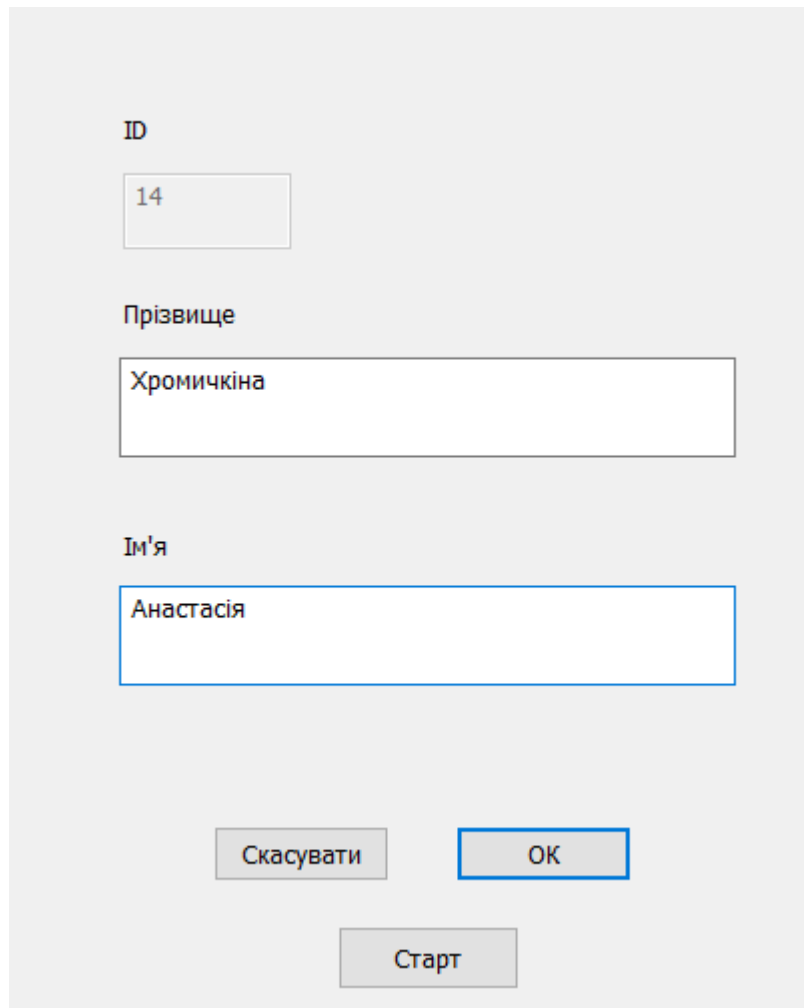


The image shows a registration form with the following elements:

- ID:** A text input field containing the number '14'.
- Прізвище:** A text input field that is currently empty, with a blue border indicating it is the active field.
- Ім'я:** A text input field that is currently empty.
- Buttons:** Three buttons are located at the bottom: 'Скасувати' (Cancel), 'ОК' (OK), and 'Старт' (Start). The 'ОК' button has a blue border, indicating it is the active button.

Рис. 4.4. Очистка полів при додаванні нового учасника

Наше ID додалось автоматично і змінювати його не можна. Заповнимо доступні поля новим учасником (рис. 4.5.).



The image shows a registration form with the following fields and values:

- ID: 14
- Прізвище: Хромичкіна
- Ім'я: Анастасія

At the bottom of the form, there are three buttons: "Скасувати" (Cancel), "ОК", and "Старт" (Start). The "ОК" button is highlighted with a blue border.

Рис. 4.5. Заповнення полів новим учасником

З рис. 4.5. бачимо, що тепер у нас є можливість зберегти дані про учасника, натиснувши кнопку «ОК», або ж не додавати ці дані, натиснувши кнопку «Скасувати».

При натисненні кнопки «ОК» дані записуються в тар та додаються в наш список до усіх інших гравців (рис. 4.6.). При цьому, поля «Прізвище» та «Ім'я» стають знову неактивними.

Також, можемо видалити будь-якого гравця, наприклад, Бабанського, натиснувши на кнопку «Вилучити». Після цього отримуємо повідомлення про зроблену операцію (рис. 4.7.). На рис. 4.8. бачимо список учасників вже без гравця «Бабанський».

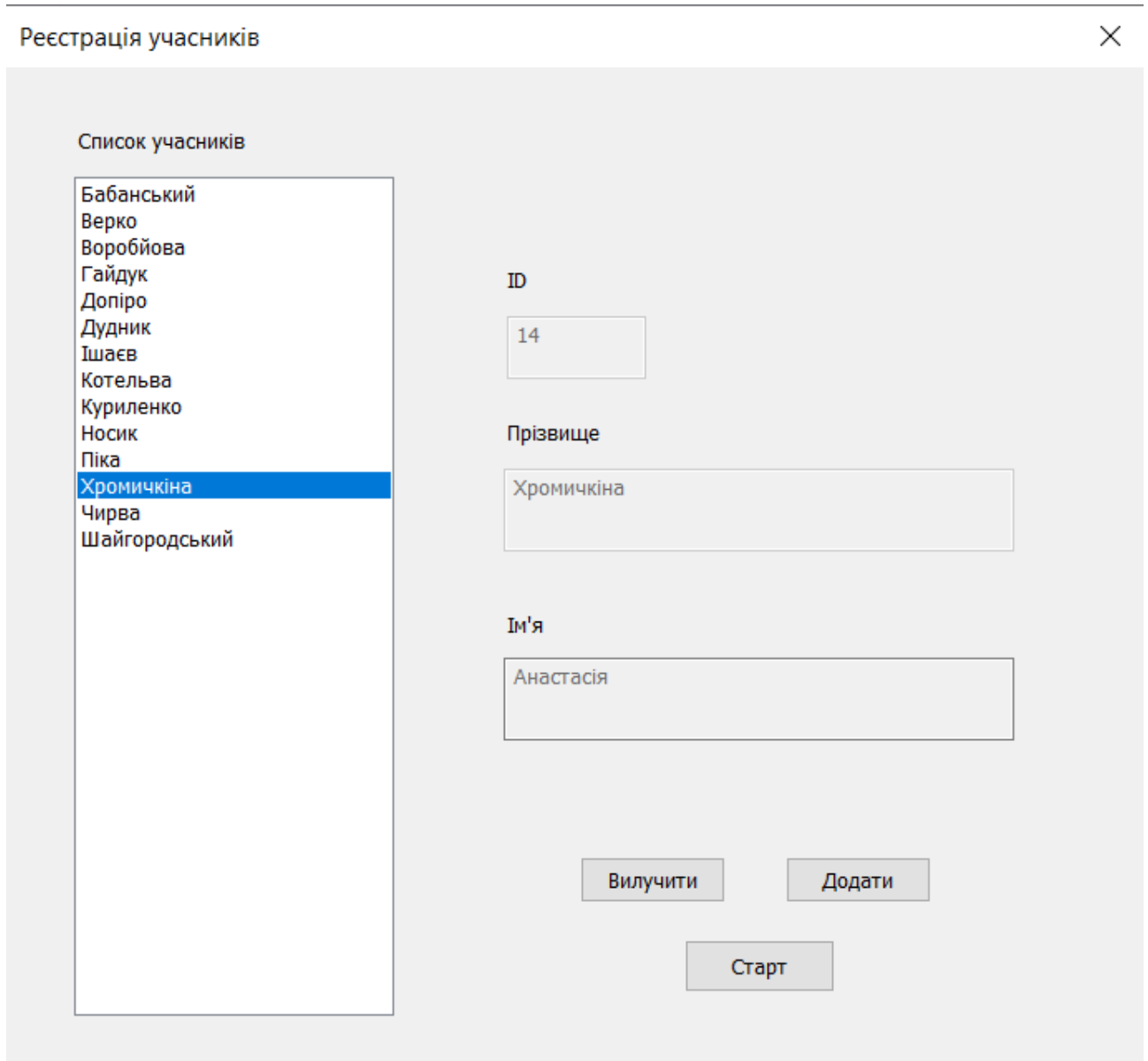


Рис. 4.6. Додавання нового учасника в мар та до списку гравців

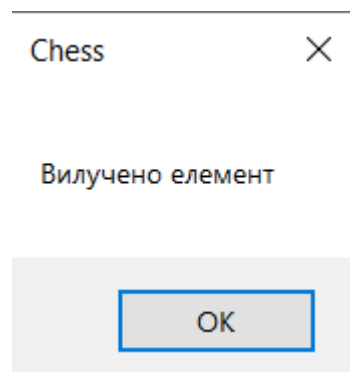


Рис. 4.7. Повідомлення про вилучення елемента

Реєстрація учасників ×

Список учасників

- Верко
- Воробйова
- Гайдук
- Допіро
- Дудник
- Ішаєв
- Котельва
- Куриленко
- Носик
- Піка
- Хромичкіна
- Чирва
- Шайгородський

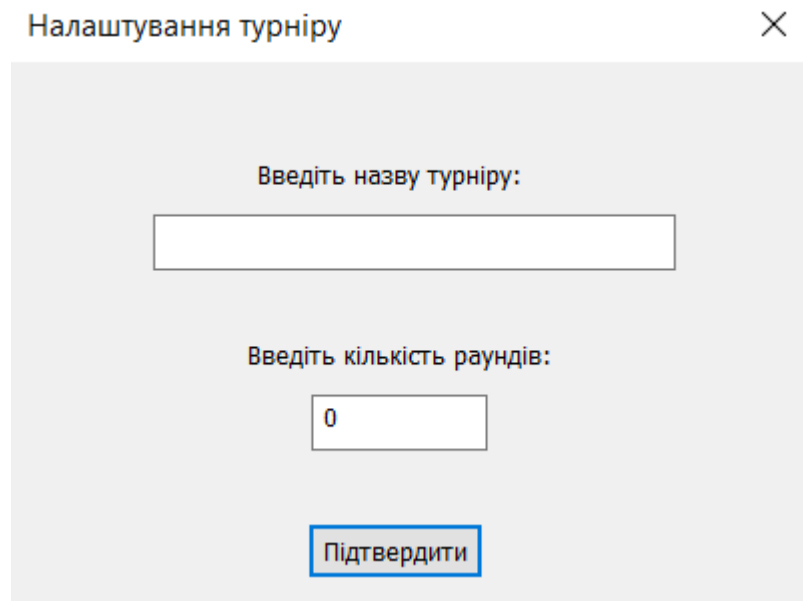
ID

Прізвище

Ім'я

Рис. 4.8. Список учасників без гравця «Бабанський»

Натиснемо на кнопку «Старт» і перейдемо до наступного вікна налаштування турніру, яке зображено на рис. 4.9. В цьому діалоговому вікні учасник повинен ввести 2 параметри: назву турніру та кількість необхідних раундів.



Налаштування турніру

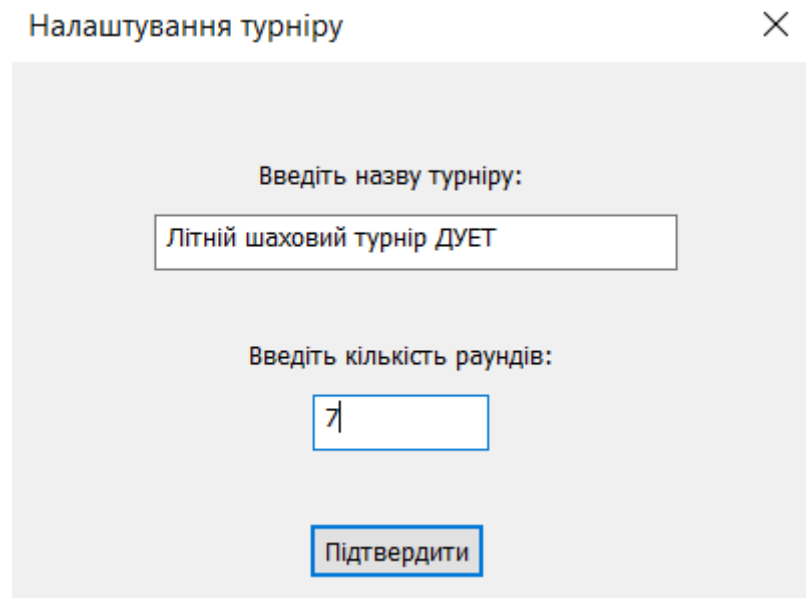
Введіть назву турніру:

Введіть кількість раундів:

Підтвердити

Рис. 4.9. Налаштування турніру

Введемо назву турніру: «Літній шаховий турнір ДУЕТ». І введемо кількість раундів: 7. На рис. 4.10. бачимо наші введені параметри для шахового турніру.



Налаштування турніру

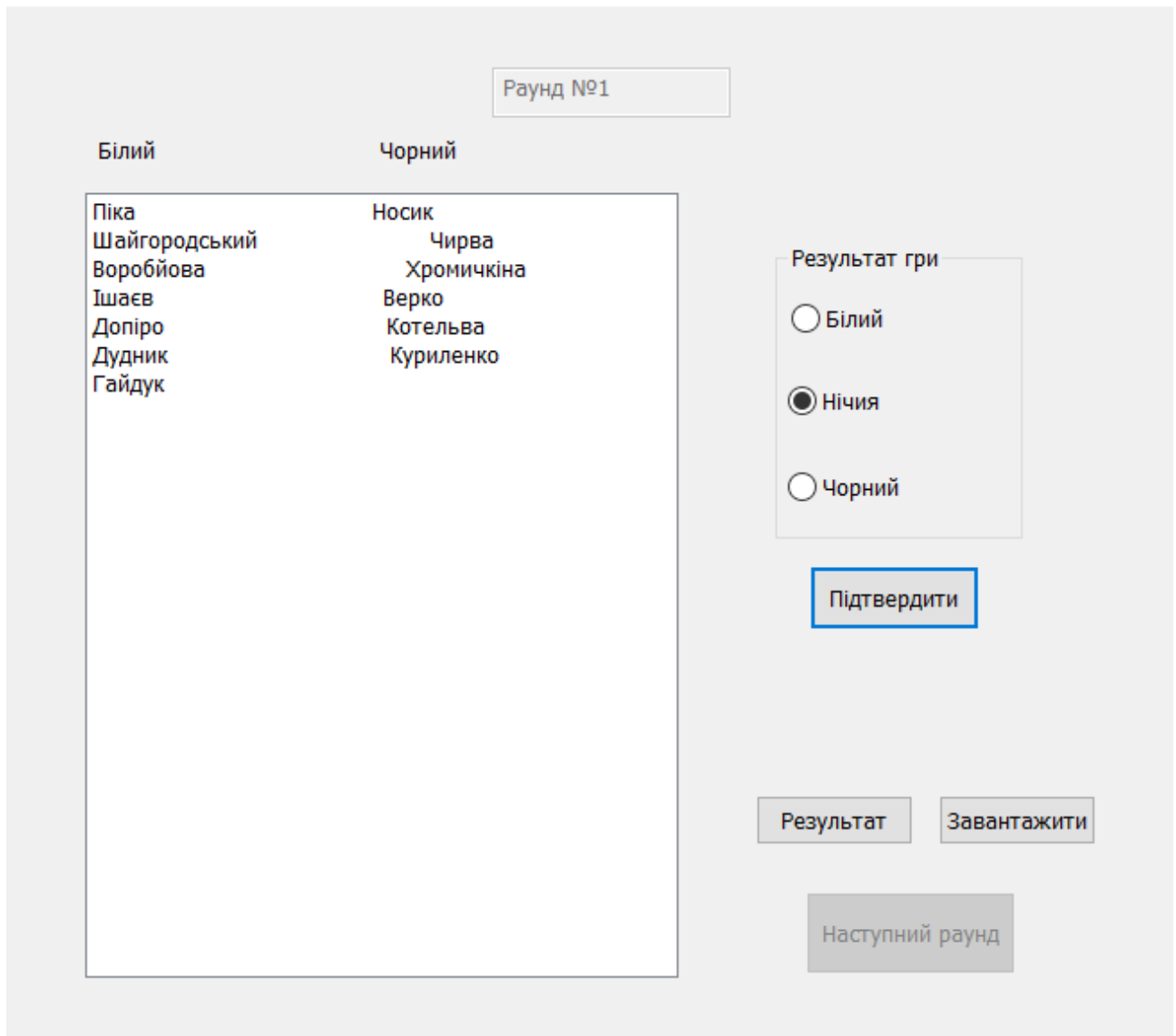
Введіть назву турніру:

Введіть кількість раундів:

Підтвердити

Рис. 4.10. Введені параметри для шахового турніру

На рис. 4.11. показано діалогове вікно з назвою "Жеребкування", яке відкрилось одразу після натискання кнопки "Підтвердити".



Раунд №1

Білий	Чорний
Піка	Носик
Шайгородський	Чирва
Воробйова	Хромичкіна
Ішаєв	Верко
Допіро	Котельва
Дудник	Куриленко
Гайдук	

Результат гри

Білий

Нічия

Чорний

Підтвердити

Результат Завантажити

Наступний раунд

Рис. 4.11. Діалогове вікно "Жеребкування"

У верхній частині вікна розміщене текстове поле з написом "Раунд №1", яке недоступне для редагування. Центральну частину займає таблиця, поділена на дві колонки: у лівій колонці вказані прізвища гравців, які гратимуть білими фігурами, а в правій — гравців, які гратимуть чорними. Таким чином, методом рандому формується список пар учасників для першого раунду. У додатку Б наведений код реалізації формування першого раунду.

Праворуч знаходиться блок "Результат гри", у якому можна обрати підсумок партії для кожної пари: перемогу білого, нічию або перемогу чорного.

За замовчуванням встановлений варіант "Нічия". Під блоком результатів розташована кнопка "Підтвердити", яка слугує для збереження обраного результату поточної гри. Гравці, які залишаються без пари, отримують перемогу в будь-якому випадку.

У нижній частині вікна розміщено дві кнопки: "Результат" та "Завантажити", які призначені для перегляду підсумків або завантаження даних. Кнопка "Наступний раунд" наразі неактивна (відображена сірим кольором), що вказує на те, що вона стане доступною лише після завершення фіксації результатів усіх матчів поточного раунду.

Розставимо результати партії для усіх пар та перевіримо результати першого раунду (рис. 4.12.).

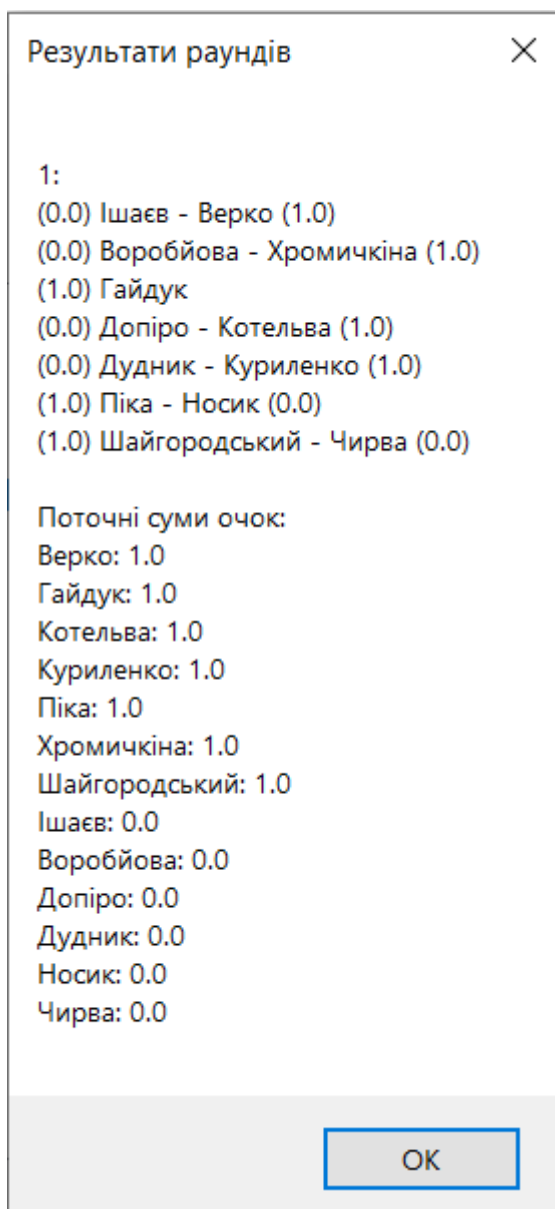


Рис. 4.12. Результати першого раунду

В діалоговому вікні «Результати раундів» бачимо результати першого раунду та поточні суми очок за всі попередні раунди. Натиснемо на кнопку «ОК» і перейдемо до наступного другого раунду (рис. 4.13.).

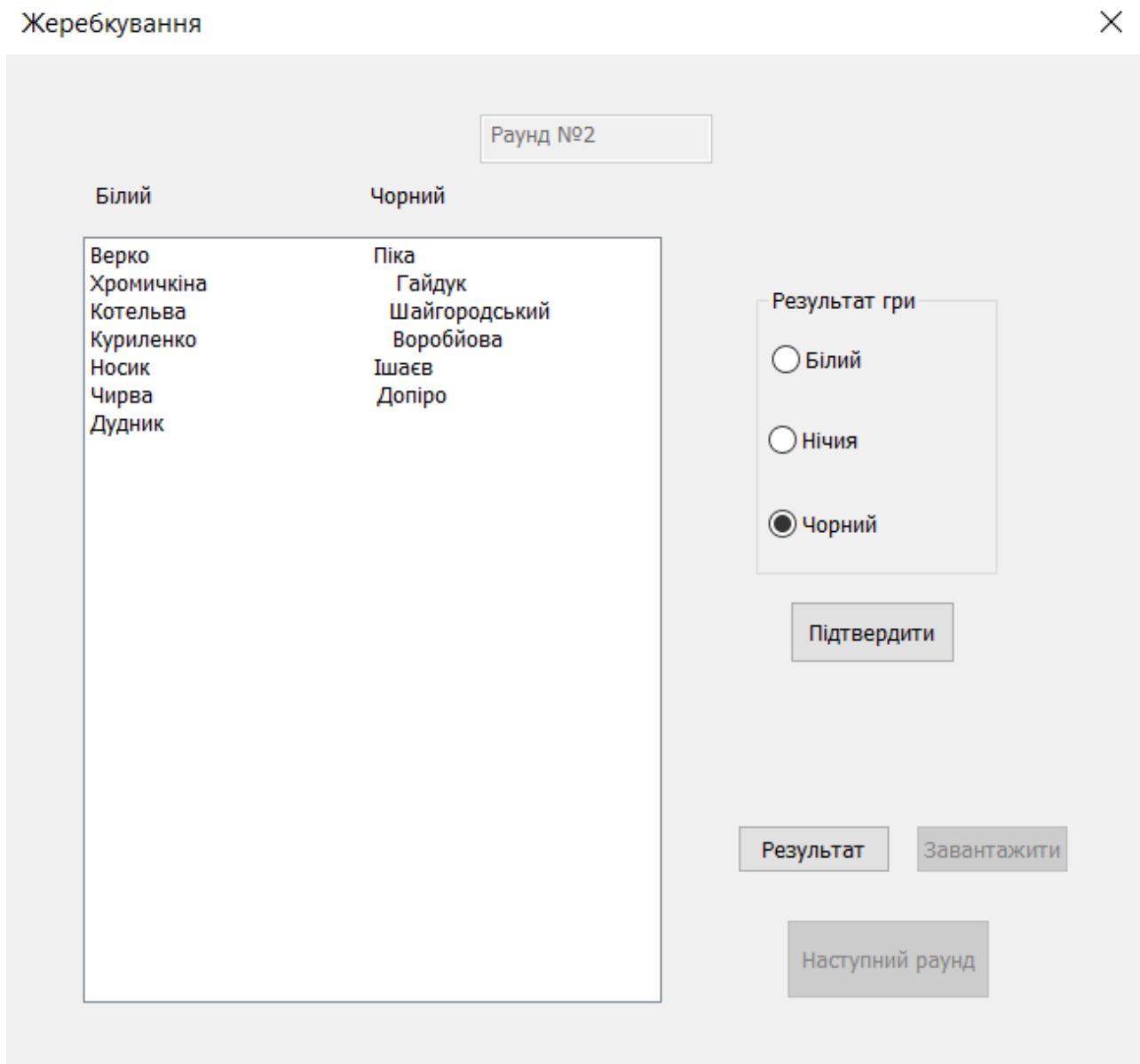


Рис. 4.13. Вікно з наступним другим раундом

У другому та наступних раундах жеребкування відбувається алгоритмічними та систематизованими методами, які можна переглянути у додатку В.

При закритті програми, уся інформація про учасників, параметри турніру та результати партій зберігаються у БД, тож користувач зможе продовжити турнір у будь-який час. Для наочності, закриємо усі діалогові вікна та в головному меню оберемо пункт «Формування раундів» (рис. 4.14.).

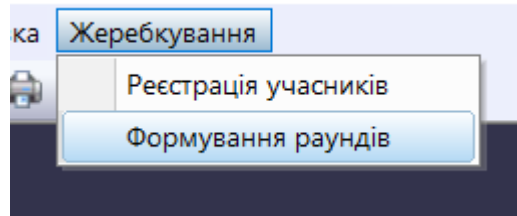


Рис. 4.14. Пункт «Формування раундів» у головному меню

Відкривається діалогове вікно «Жеребкування» (рис. 4.15.), в якому ListBox за замовчуванням пустий.

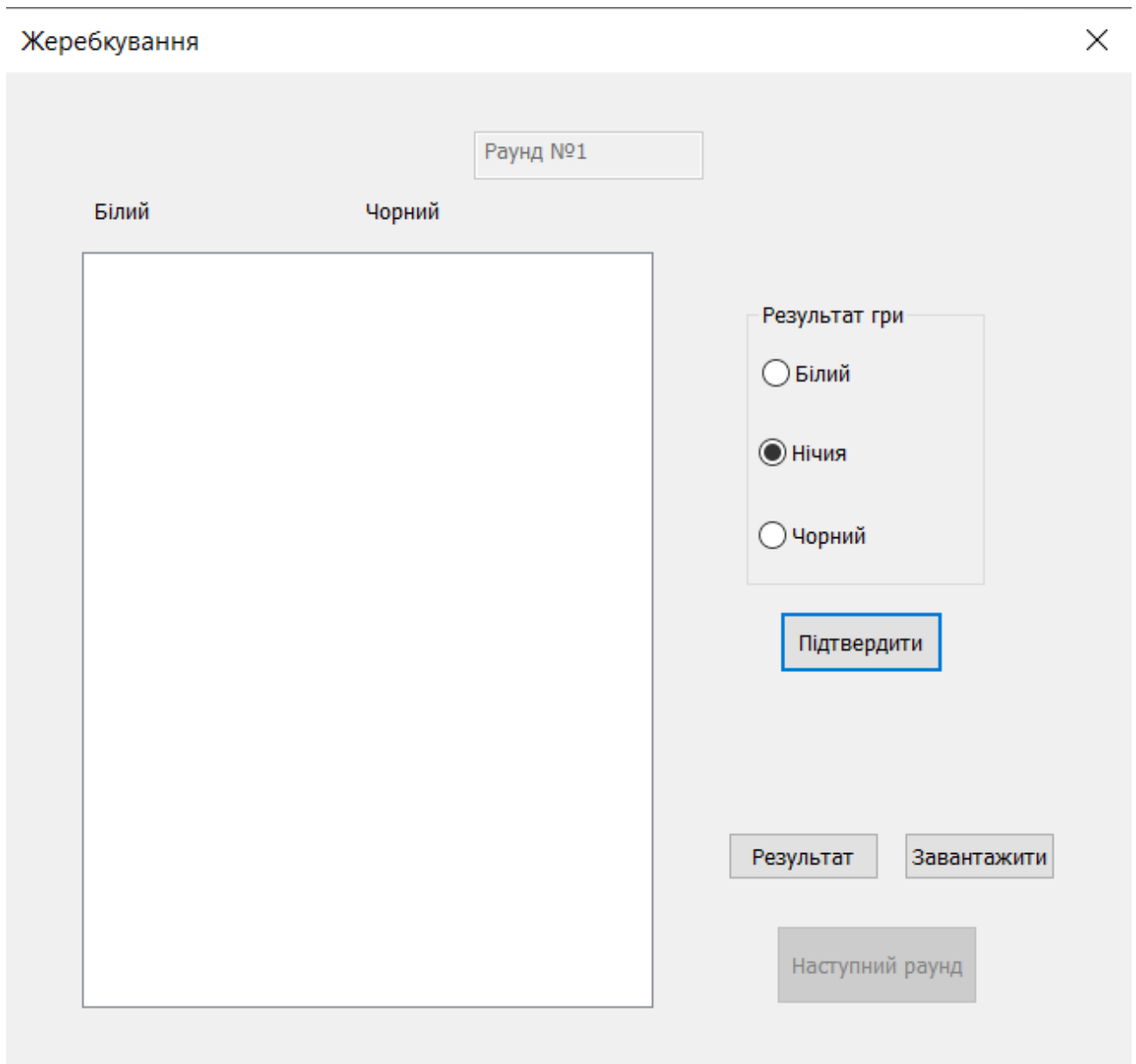


Рис. 4.15. Діалогове вікно «Жеребкування» з пустим ListBox

Натискаємо на кнопку «Завантажити» і на рис. 4.16. бачимо віконце «Завантаження турніру».

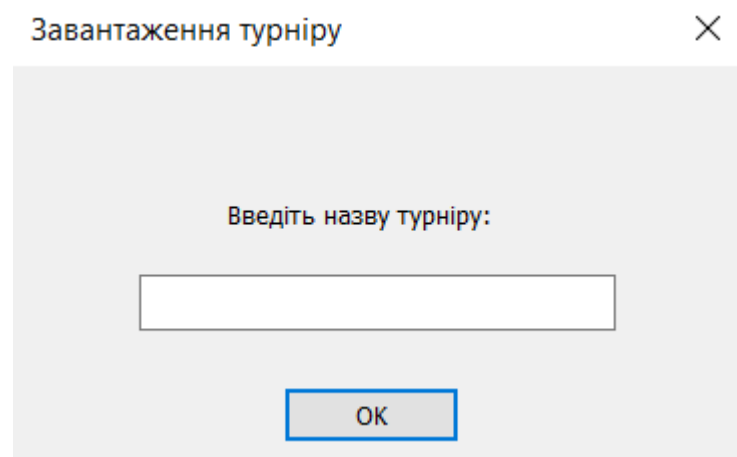


Рис. 4.16. Віконце «Завантаження турніру»

Вводимо назву попереднього турніру – «Літній шаховий турнір ДУЕТ», і натискаємо кнопку «ОК» (рис. 4.17.).

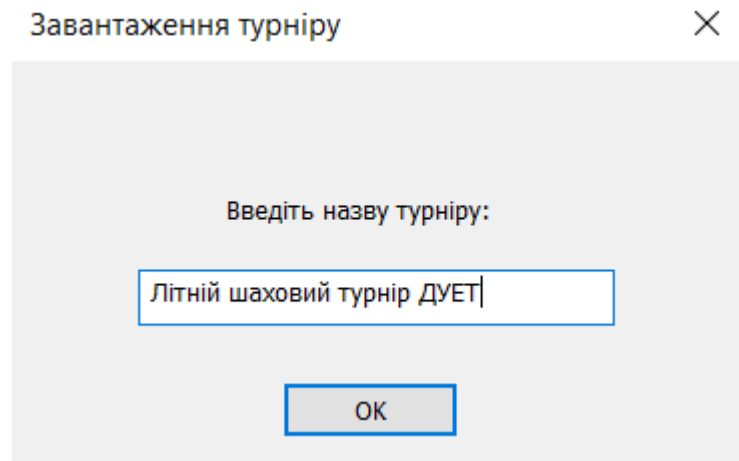


Рис. 4.17. Введення назви попереднього турніру

На рис. 4.18. бачимо, що відкрилось діалогове вікно «Жеребкування» одразу на другому раунді, на якому ми й закрили програму.

Виставимо гравцям результати включно до 7 раунду і перевіримо результати усіх раундів (рис. 4.19.).

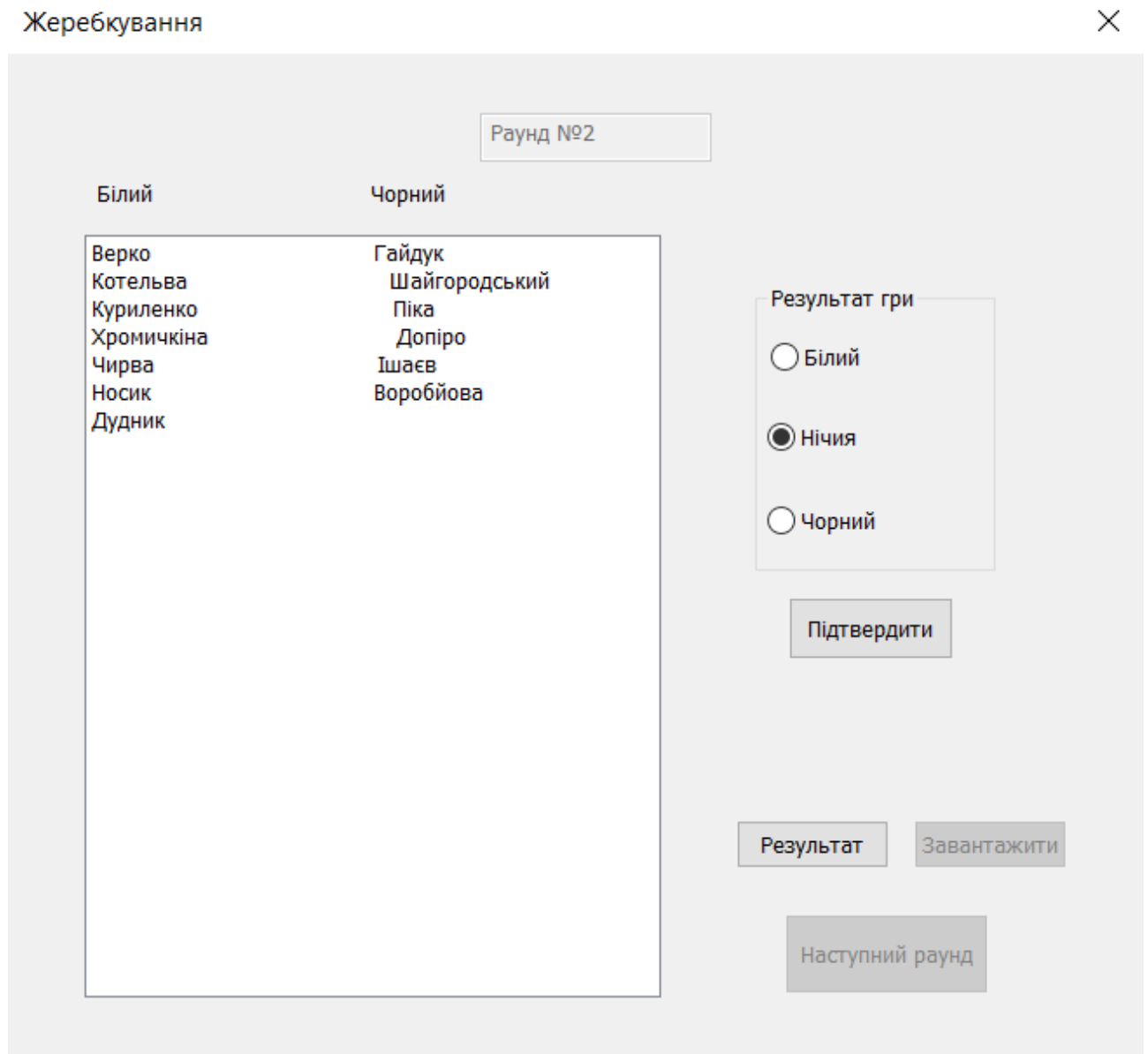


Рис. 4.18. Продовження турніру «Літній шаховий турнір ДУЕТ»

У діалоговому вікні «Результати раундів» бачимо результати гравців у перших трьох партіях і поточні суми очок по цим трьом раундам.

При натисненні кнопки «ОК» виводяться наступні три раунди (рис. 4.20.). І так як у нас раундів 7, то при ще одному натисненні кнопки «ОК» - бачимо результати останнього, сьомого раунду і поточні суми очок по всім раундам (рис. 4.21.).

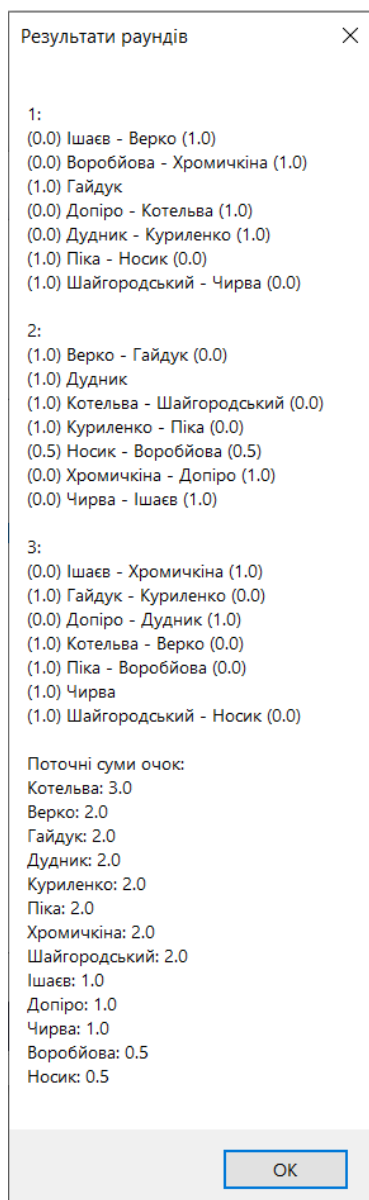


Рис. 4.19. Вікно з результатами

На зображенні показано діалогове вікно з назвою "Результати раундів". У ньому наведено підсумки зіграних партій першого, другого та третього раундів шахового турніру, а також поточні суми очок усіх учасників.

Результати кожного раунду представлені у вигляді списку партій. У дужках вказано результат гри у форматі (очки білого – очки чорного), після чого зазначено прізвища учасників: першим завжди йде гравець білими, другим — чорними. Наприклад, у раунді 3 гра "(0.0) Ішаєв – Хромичкіна (1.0)" означає, що Хромичкіна, граючи чорними, перемогла Ішаєва.



Рис. 4.20. Продовження виводу результатів

Окремим блоком після раундів виведено зведену таблицю з поточними сумами очок кожного учасника. Список сформовано у порядку спадання за кількістю набраних очок.

При натисненні кнопки «ОК» діалогове вікно з результатами закривається і ми повертаємось до вікна «Жеребкування». Натискаємо на кнопку «Наступний раунд» і так як 7 раунд був останнім – виводиться вікно з поточними результатами за всі раунди (рис. 4.22.). В цьому вікні так само сформовано список у порядку спадання за кількістю набраних очок.

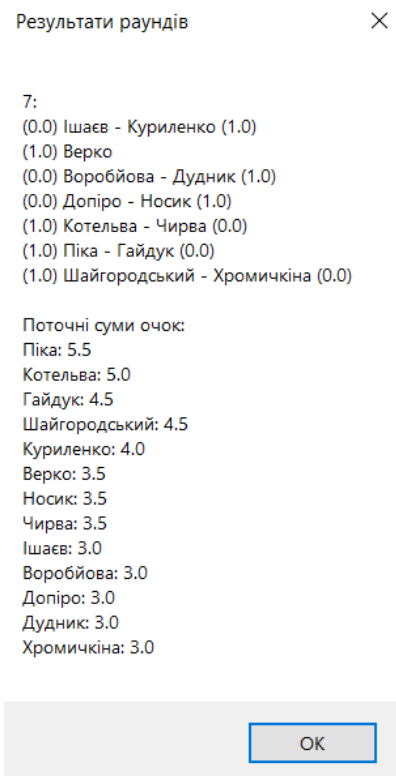


Рис. 4.21. Вивід результатів останнього раунду

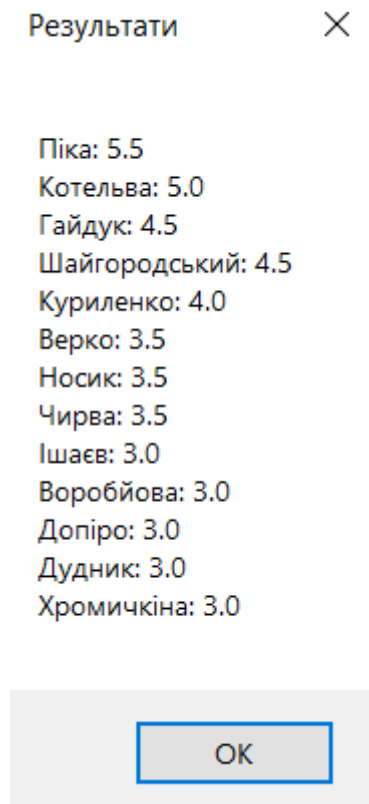


Рис. 4.22. Вікно з поточними результатами за всі раунди

При натисненні на кнопку «ОК» відкривається наступне вікно з назвою "Результати" (рис. 4.23.). У цьому вікні показано підсумкову таблицю шахового турніру з трьома ключовими показниками для кожного гравця:

1. Очки — загальна кількість набраних очок у турнірі.
2. Прогрес — сума очок, набраних учасником у кожному раунді, з наростанням (так званий коефіцієнт прогресу).
3. Бухгольц — сума очок усіх суперників, з якими грав учасник, тобто коефіцієнт, що враховує силу суперників.

У додатку Г наведений приклад коду для рахування очок, прогресу і коефіцієнту Бухгольца. Учасники перераховані у порядку спадання за кількістю очок. Лідером є Піка з 5.5 очками, прогресом 20.5 і Бухгольцем 27.5. Далі йдуть Котельва (5.0 очок) і Шайгородський з Гайдуком (по 4.5 очка).

Внизу вікна розташована кнопка "ОК", яка закриває діалогове вікно після ознайомлення з результатами.

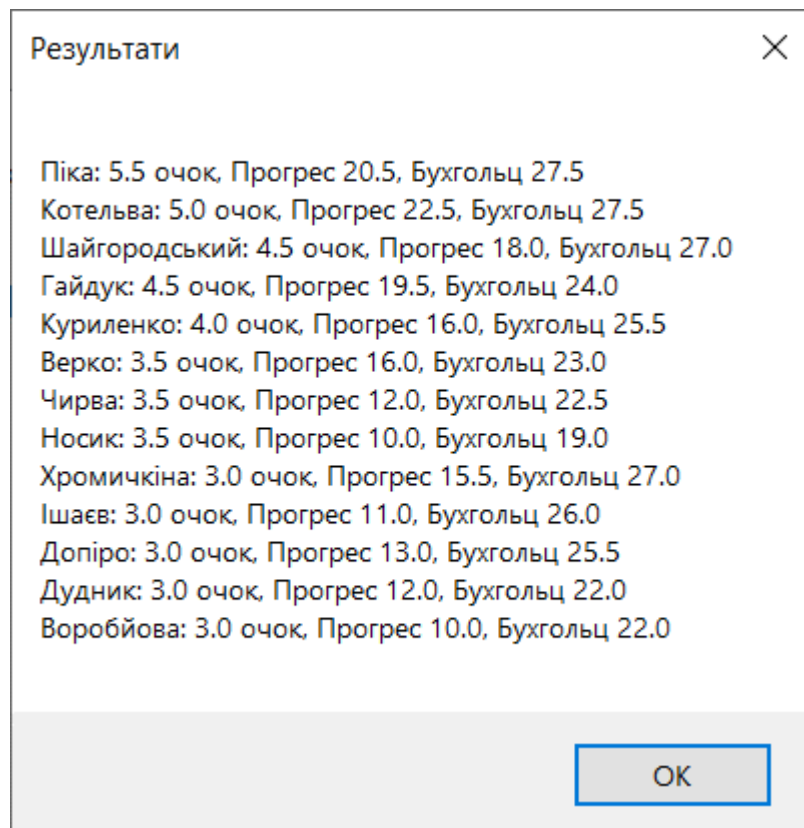


Рис. 4.23. Діалогове вікно "Результати"

На цьому весь функціонал програми було повністю розглянуто. Усі ключові етапи роботи — від жеребкування, фіксації результатів партій, перегляду поточних турів до підбиття підсумкових результатів із розрахунком очок, прогресу та коефіцієнта Бухгольца — реалізовані та продемонстровані.

4.2. Опис інтерфейсу та алгоритму розробки сайту

Наш WEB-сайт призначений для відображення інформації про турніри, їх учасників, а також результатів, що зберігаються у базі даних. Розробка сайту здійснюється із використанням базових та водночас потужних вебтехнологій, зокрема HTML (HyperText Markup Language) для структурування вебсторінок, CSS (Cascading Style Sheets) для оформлення зовнішнього вигляду, JavaScript для інтерактивності та динамічних елементів, а також PHP для серверної логіки й взаємодії з базою даних.

Основною метою розроблюваного сайту є забезпечення користувачів зручним і швидким доступом до актуальної інформації про всі турніри, які були проведені. Сайт дозволяє переглядати перелік турнірів, бачити детальний склад учасників кожного турніру, а також ознайомлюватися з підсумковими результатами та статистикою. Уся інформація динамічно отримується з бази даних, що гарантує її актуальність та точність.

У цьому підрозділі буде розглянуто структуру інтерфейсу користувача, описано основні елементи взаємодії, а також логіку роботи сторінок.

На рис. 4.24. зображена головна сторінка сайту. Вітальний заголовок чітко повідомляє мету порталу — це сайт, присвячений турнірним таблицям і аналітиці шахових змагань на основі даних із бази Chess. Користувачам пропонується можливість стежити за прогресом гравців, аналізувати загальну ситуацію у турнірах та оперативно виявляти лідерів, що робить цей ресурс незамінним інструментом як для шахістів, так і для тренерів, організаторів і просто зацікавлених осіб.

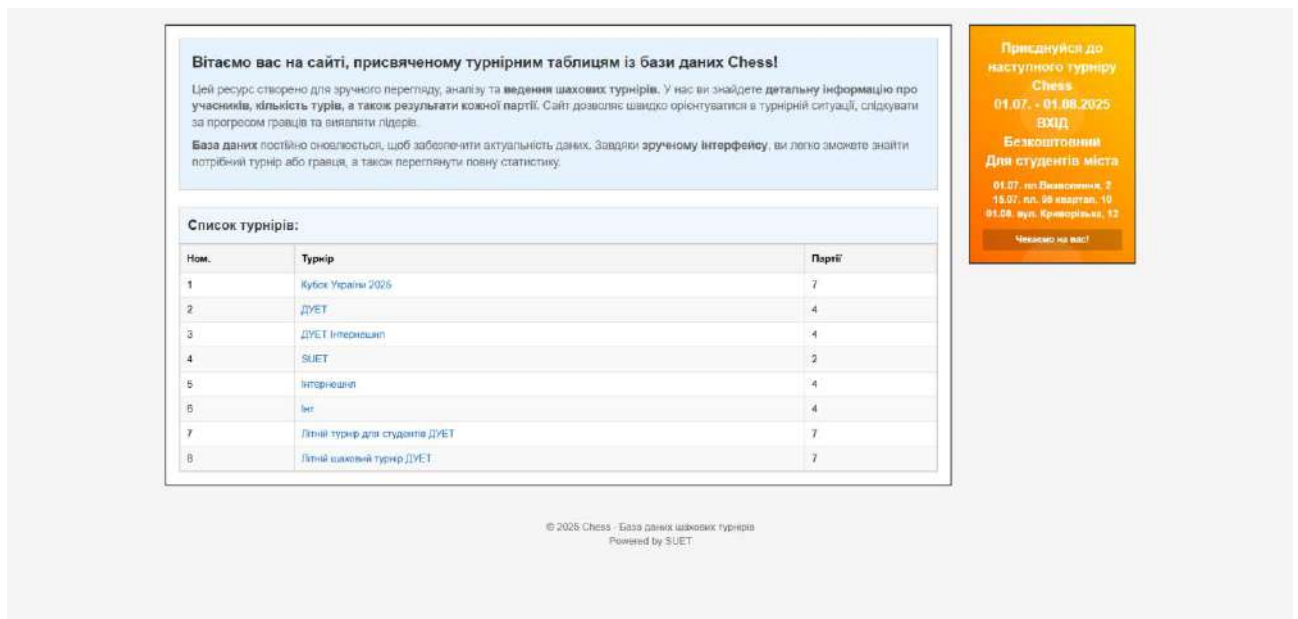


Рис. 4.24. Головна сторінка сайту

Особливу увагу приділено актуальності й надійності інформації — база даних оновлюється постійно, що гарантує достовірність та точність відображених даних. Важливо й те, що сайт побудований з урахуванням принципів зручності: завдяки простому й інтуїтивно зрозумілому інтерфейсу кожен може легко знайти потрібну інформацію — будь то конкретний турнір або статистика окремого гравця.

Центральне місце займає таблиця зі списком турнірів. У таблиці наведені записи, де зазначено порядковий номер, назву турніру та кількість зіграних партій.

Праворуч, у яскравому помаранчевому блоці, привертає увагу інформація про майбутній шаховий турнір.

Загалом, сторінка створює дуже позитивне враження: все виглядає організовано, чисто, професійно. Цю сторінку можна вважати вдалим прикладом того, як зручно подати складну інформацію у простому вигляді.

Перейдемо до сторінки турніру, наприклад: Кубок України 2025 (рис. 4.25.). На цій сторінці користувача зустрічає чітка й структурована інформація, що дозволяє швидко зорієнтуватися у деталях події. Візуально сторінка оформлена у тому ж лаконічному та функціональному стилі, що й

головна, завдяки чому зберігається відчуття цілісності й логічної послідовності в усьому вебресурсі.

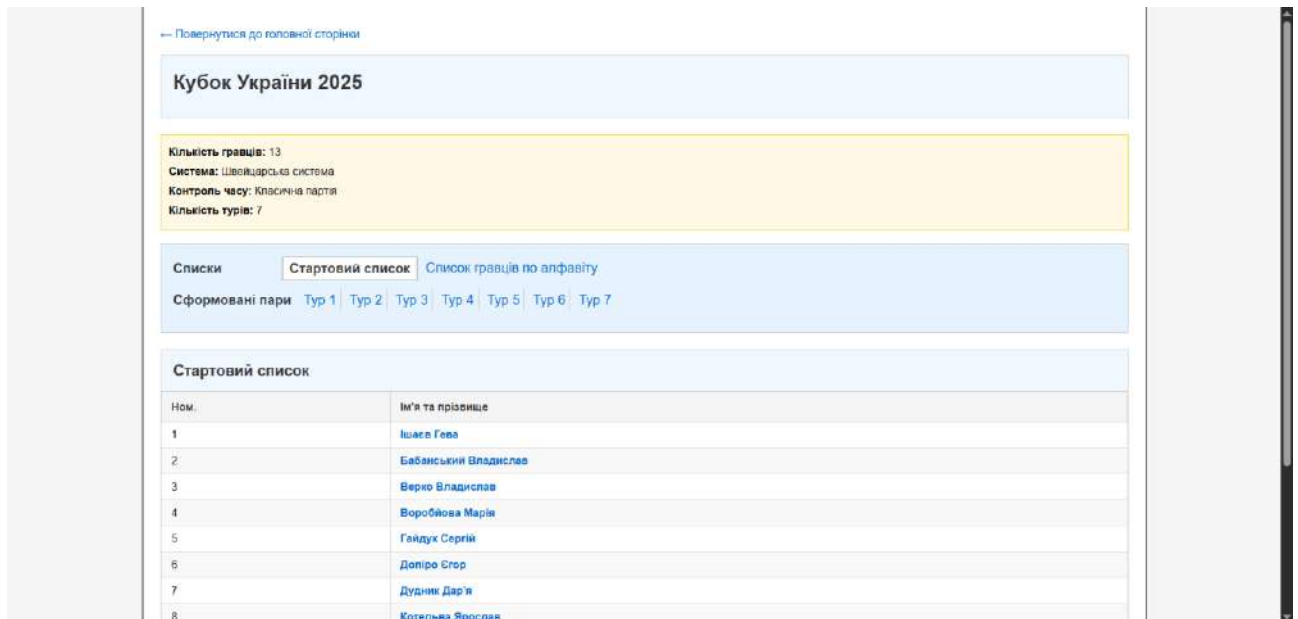


Рис. 4.25. Сторінка турніру «Кубок України 2025»

У верхній частині сторінки, одразу під посиланням на повернення до головної, розташовано великий заголовок із назвою турніру — «Кубок України 2025», що акцентує увагу користувача й задає тему вмісту. Під цим заголовком знаходиться жовтий блок, де подано основну довідкову інформацію про турнір. У цьому блоці вказано кількість учасників, систему проведення, тип контролю часу, а також загальну кількість турів у турнірі. Такий стислий, але змістовний виклад дозволяє одразу скласти загальне уявлення про формат змагання.

Далі сторінка пропонує два варіанти перегляду списків: стартовий список і список гравців в алфавітному порядку. Це корисна функціональність, яка дає змогу зручно знаходити потрібного учасника — залежно від того, чи вас цікавить послідовність за стартовими номерами, чи просто пошук за прізвищем.

Окремо, у вигляді навігаційних посилань, доступні сформовані пари для кожного з турів (рис. 4.26.). Це дозволяє глибше зануритися у перебіг змагання й дізнатися, хто з ким грав на кожному етапі.

Нижче надано таблицю зі стартовим списком учасників. У таблиці вказано порядковий номер гравця та його ім'я і прізвище.

— Повернутися до головної сторінки

Кубок України 2025

Кількість гравців: 13
Система: Швейцарська система
Контроль часу: Класична партія
Кількість турів: 7

Списки [Стартовий список](#) [Список гравців по алфавіту](#)

Сформовані пари [Тур 1](#) [Тур 2](#) [Тур 3](#) [Тур 4](#) [Тур 5](#) [Тур 6](#) [Тур 7](#)

Пари 1 туру

Ном.	Білий	Очки	Результат	Очки	Чорний
1	Ішася Гева	0	1-0		без пари
2	Бабанський Владислав	0	1-0	0	Гайдук Сергій
3	Верко Владислав	0	0-1	0	Воробйова Марія
4	Дудник Дар'я	0	1-0	0	Котельва Ярослав
5	Носик Дмитро	0	1-0	0	Допіро Єгор
6	Чирва Назар	0	1-0	0	Куриленко Іван
7	Шайгородський Богдан	0	1-0	0	Піва Антон

Рис. 4.26. Сформовані пари для кожного з турів

В розділі «Пари 1 туру», представлено детальну таблицю, яка відображає, як саме були сформовані гравці для першого раунду змагань. Цей блок є надзвичайно важливим для учасників та глядачів, оскільки дозволяє стежити за перебігом кожної окремої партії, її результатом, а також бачити, хто саме грав білими, а хто чорними фігурами.

Таблиця розпочинається з нумерації — це умовні порядкові номери ігор у турі, що полегшує навігацію та орієнтацію. Далі йде розділ «Білий», у якому вказано імена гравців, які грали білими фігурами в цьому турі. У сусідній колонці — їхні початкові очки (на момент до партії — всі по нулях), а далі — результат партії (він поданий у форматі 1-0 або 0-1, що відповідає перемозі одного з учасників).

Наступна колонка — це очки супротивника (теж усюди по нулях, оскільки тур лише стартував), і, нарешті, колонка «Чорний» містить імена гравців, які грали чорними фігурами.

Із таблиці видно, що в турі було зіграно 6 повноцінних партій і один учасник — Гайдук Сергій — отримав пропуск у зв'язку з непарною кількістю гравців (13), про що свідчить позначка «без пари» навпроти його імені. Це

характерна особливість швейцарської системи, коли в першому турі хтось може залишитись без суперника.

Уся інформація на сторінці подана максимально чітко, без зайвої візуальної перевантаженості, що дозволяє зосередитися на змісті. Ресурс виглядає професійно і надійно, а структура сторінки сприяє зручній навігації як для гравців, так і для організаторів чи глядачів.

Якщо натиснути на будь-якого гравця, наприклад, на Ішаєва Геву – побачимо сторінку профіля гравця шахових турнірів (рис. 4.27.).

← Повернутися до головної сторінки

Ішаєв Гева

Останнє оновлення: 03.05.2025 19:27:29

Інфо гравця

Ім'я: [Ішаєв Гева](#)

Стартове місце: 1

Очки: 19

Місце: 7

Зіграно партій: 35

№	Тур	Ном. гравця	Ім'я	Очки	Рез.
1	1	--	Без партії	--	1
2	1	3	Верко Владислав	18	0
3	1	--	Без партії	--	1
4	1	11	Гіва Антон	20.5	1
5	1	12	Черва Назар	21	1/2
6	1	5	Гайдух Сергій	19	1
7	1	3	Верко Владислав	18	0
8	2	4	Воробієва Марія	18	0
9	2	13	Шейкороосієвич Богдан	26	0
10	2	12	Черва Назар	21	1

Рис. 4.27. Сторінка профіля гравця шахових турнірів

Сторінка відображає профіль гравця з повною інформацією про його участь у турнірі. На ній представлено як загальні статистичні дані, так і деталізований список зіграних партій.

У верхній частині сторінки розміщено заголовок з іменем гравця. Безпосередньо під ним вказується дата і час останнього оновлення даних.

Основний блок містить таблицю з такими параметрами:

1. Ім'я гравця (клікабельне посилання).
2. Стартове місце — позиція гравця у списку на початку турніру.
3. Кількість очок, набраних у турнірі.
4. Поточне місце у турнірній таблиці.

5. Загальна кількість зіграних партій.

Нижче наведена таблиця з історією ігор. Вона складається з таких колонок:

- номер запису (№);
- номер туру, у якому відбулась партія;
- номер суперника у списку учасників;
- ім'я суперника;
- очки суперника на момент гри;
- результат партії (позначення: 1 — перемога, 0 — поразка, $\frac{1}{2}$ — нічия).

Висновки до розділу 4

У цьому розділі було детально розглянуто головне вікно програми та ключові компоненти графічного інтерфейсу, що реалізують логіку проведення шахового турніру. Було показано, як за допомогою бібліотеки OpenGL можливо створити інтерактивне тривимірне зображення шахової фігури з можливістю її переміщення у просторі, що демонструє гнучкість графічного середовища програми.

Особливу увагу приділено діалоговим вікнам, які забезпечують керування турніром: від реєстрації учасників до формування пар у раундах та фіксації результатів партій. Інтерфейс дозволяє зручно додавати нових гравців, редагувати список учасників, зберігати та відновлювати стан турніру з бази даних, що забезпечує повноцінну підтримку багатораундових змагань.

Також було реалізовано алгоритми жеребкування та обрахунку основних показників для підсумкової таблиці: загальної кількості очок, коефіцієнту прогресу та Бухгольца. Це дає можливість формувати об'єктивні результати турніру на основі зафіксованих партій.

Загалом, програмне забезпечення продемонструвало стабільну роботу, інтуїтивно зрозумілий інтерфейс і гнучкість у використанні, а реалізовані

алгоритми дозволяють ефективно автоматизувати процес проведення шахового турніру.

Також було розглянуто процес створення веб-сайту, який слугує доповненням до десктопного застосунку та забезпечує зручний онлайн-доступ до інформації про шахові турніри. Основною метою розробки сайту стало відображення списків учасників, результатів партій, таблиць з очками, а також можливість ознайомлення з турнірною статистикою через будь-який сучасний браузер.

Інтерфейс веб-сайту було реалізовано з урахуванням принципів зручності, доступності та мінімалізму. Для реалізації використовувалися сучасні вебтехнології (HTML, CSS, JavaScript, PHP), а також зв'язок з базою даних, у якій зберігається уся інформація про турнір. Забезпечено динамічне завантаження та відображення даних із БД, що дозволяє оперативно оновлювати та переглядати результати в режимі реального часу.

У результаті розробки створено ефективну, зрозумілу, зручну програму та веб-інструмент, які забезпечують прозорість, відкритість і публічність турніру, а також розширюють аудиторію завдяки онлайн-доступу до результатів.

ВИСНОВКИ

У результаті виконання курсової роботи було повністю реалізовано програмне забезпечення для автоматизованого проведення шахових турнірів. Розроблена система охоплює всі етапи організації та супроводу змагання — від введення учасників до виведення підсумкових результатів, що підтверджує її функціональну завершеність та практичну придатність до реального використання.

У першому розділі було проаналізовано специфіку задач, які виникають під час організації шахових турнірів, та визначено вхідні і вихідні дані, необхідні для реалізації програмного забезпечення. На основі цього було сформульовано вимоги до системи, що лягли в основу логіки її побудови.

У другому розділі було розроблено алгоритмічне забезпечення, яке повністю охоплює цикл проведення турніру. Реалізація на мові програмування C++ із використанням бібліотеки MFC забезпечила зручний інтерфейс, а вбудована підтримка з боку OpenGL дозволила реалізувати 3D-елементи для візуального оформлення програми.

Третій розділ присвячено інформаційному забезпеченню, зокрема проектуванню бази даних. Створено чітку логічну модель з використанням СУБД MySQL, яка забезпечує надійне зберігання даних, ефективну обробку результатів та масштабованість для можливого розширення функціоналу.

У четвертому розділі реалізовано ключові компоненти інтерфейсу користувача та механізми керування даними. Особливу увагу приділено можливостям динамічного формування пар, збереження стану турніру, підрахунку очок і відображення інформації у зручному форматі. Крім того, розроблено веб-сайт, який надає відкритий доступ до даних турніру через Інтернет.

Загалом розроблене рішення є функціональним, стабільним, інтуїтивно зрозумілим і відповідає вимогам автоматизації шахових турнірів.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Страуструп Б. Язык программирования C++. Базовый курс / Страуструп Б. – М. : Вильямс, 2015. – 1136 с.
2. Дейтел П. Как программировать на C++ / Дейтел П., Дейтел Х. – М. : Бином, 2016. – 1456 с.
3. Шилдт Г. МFC. Основы программирования / Шилдт Г. – М. : Вильямс, 2001. – 384 с.
4. Круглински Д. Программирование на Visual C++ / Круглински Д. – М. : Русская редакция, 1997. – 864 с.
5. Ангел Э. Интерактивная компьютерная графика. Вводный курс на базе OpenGL / Ангел Э. – М. : Вильямс, 2001. – 592 с.
6. Wright R. OpenGL SuperBible / Wright R., Lipchak B. – Indianapolis : Sams Publishing, 2007. – 1200 с.
7. Когаловский М. Р. Энциклопедия технологий баз данных / Когаловский М. Р. – М. : Финансы и статистика, 2002. – 800 с.
8. Кузнецов С. Д. Основы современных баз данных / Кузнецов С. Д. – М. : Интернет-университет информационных технологий, 2007. – 484 с.
9. Дейт К. Введение в системы баз данных / Дейт К. – М. : Вильямс, 2005. – 1328 с.
10. Коннолли Т. Базы данных. Проектирование, реализация и сопровождение / Коннолли Т., Бегг К. – М. : Вильямс, 2003. – 1440 с.
11. Гарсиа-Молина Г. Системы баз данных. Полный курс / Гарсиа-Молина Г., Ульман Д., Уидом Д. – М. : Вильямс, 2003. – 1088 с.
12. Блаха М. Объектно-ориентированное моделирование и проектирование / Блаха М., Румбо Д. – СПб. : Лори, 2001. – 542 с.
13. Карпова Т. С. Базы данных: модели, разработка, реализация / Карпова Т. С. – СПб. : Питер, 2001. – 304 с.
14. Хансен Г. Базы данных. Разработка и управление / Хансен Г., Хансен Д. – М. : БИНОМ, 1999. – 704 с.

ДОДАТКИ

Обробка натискань клавіш і зміна координат фігури у просторі

```

void CChessView::RenderScene() {
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    glLoadIdentity();

    // Відступ для кращого огляду фігури
    glTranslatef(0.0f, 0.0f, -10.0f);

    // Обертання сцени для інтерактивності
    glRotatef(turnX, 1.0f, 0.0f, 0.0f);
    glRotatef(turnY, 0.0f, 1.0f, 0.0f);

    // Відображення шахової фігури
    DrawChessQueen();

    // Обмін буферів
    SwapBuffers(m_pDC->GetSafeHdc());
}

void CChessView::DrawChessQueen() {
    GLfloat queenColor[] = { 0.1f, 0.1f, 0.1f, 1.0f }; // темно-сірий ферзь як на зображенні
    glMaterialfv(GL_FRONT, GL_DIFFUSE, queenColor);

    glRotatef(-85.0f, 1.0f, 0.0f, 0.0f);
    glRotatef(20.0f, 0.0f, 1.0f, 0.0f);
    glRotatef(20.0f, 0.0f, 0.0f, 1.0f);

    // Встановлюємо щільнішу сітку примітивів для кращої деталізації
    int slices = 32; // кількість сегментів по колу
    int stacks = 16; // кількість сегментів по висоті

    // 1. База (основа) фігури
    glPushMatrix();
    glTranslatef(0.0f, -2.0f, 0.0f);
    glRotatef(-90.0f, 1.0f, 0.0f, 0.0f);
    gluDisk(m_quadric, 0.0, 1.0, slices, 4);
    gluCylinder(m_quadric, 1.0, 0.8, 0.25, slices, stacks / 4);
    glTranslatef(0.0f, 0.0f, 0.25f);
    glPopMatrix();

    // 2. Нижній диск основи
    glPushMatrix();
    glTranslatef(0.0f, -1.75f, 0.0f);
    glRotatef(-90.0f, 1.0f, 0.0f, 0.0f);
    gluCylinder(m_quadric, 0.8, 0.7, 0.15, slices, stacks / 4);
    glPopMatrix();

    // 3. Нижня частина стовбура
    glPushMatrix();
    glTranslatef(0.0f, -1.6f, 0.0f);
    glRotatef(-90.0f, 1.0f, 0.0f, 0.0f);
    gluCylinder(m_quadric, 0.7, 0.5, 0.4, slices, stacks / 2);
    glPopMatrix();

    // 4. Розширена середня частина (шарик)
    glPushMatrix();
    glTranslatef(0.0f, -1.1f, 0.0f);
    glRotatef(-90.0f, 1.0f, 0.0f, 0.0f);

```

Продовження додатку А

```

gluSphere(m_quadric, 0.55, slices, stacks);
glPopMatrix();

// 5. Середня частина стовбура
glPushMatrix();
glTranslatef(0.0f, -0.9f, 0.0f);
glRotatef(-90.0f, 1.0f, 0.0f, 0.0f);
gluCylinder(m_quadric, 0.4, 0.5, 1.5, slices, stacks);
glPopMatrix();

// 6. Верхній диск (під короною)
glPushMatrix();
glTranslatef(0.0f, 0.6f, 0.0f);
glRotatef(-90.0f, 1.0f, 0.0f, 0.0f);
gluCylinder(m_quadric, 0.5, 0.6, 0.15, slices, stacks / 4);
glPopMatrix();

// 7. Нижня частина корони
glPushMatrix();
glTranslatef(0.0f, 0.75f, 0.0f);
glRotatef(-90.0f, 1.0f, 0.0f, 0.0f);
gluCylinder(m_quadric, 0.6, 0.7, 0.2, slices, stacks / 4);
glPopMatrix();

// 8. Зубці корони
glPushMatrix();
glTranslatef(0.0f, 0.95f, 0.0f);
glRotatef(-90.0f, 1.0f, 0.0f, 0.0f);

// Основа корони
gluCylinder(m_quadric, 0.7, 0.7, 0.1, slices, stacks / 4);

// Зубці - 5 гострих виступів
const int numTeeth = 5;
float toothHeight = 0.3f;
float toothBaseRadius = 0.15f;
float crownRadius = 0.7f;

for (int i = 0; i < numTeeth; i++) {
    float angle = (float)i * 2.0f * (float)M_PI / numTeeth;
    float x = crownRadius * cos(angle);
    float z = crownRadius * sin(angle);

    glPushMatrix();
    glTranslatef(x, z, 0.1f);

    // Трикутник для зубця
    glBegin(GL_TRIANGLES);
    glNormal3f(0.0f, 0.0f, 1.0f);
    glVertex3f(0.0f, 0.0f, 0.0f);
    glVertex3f(toothBaseRadius * cos(angle - M_PI / 10), toothBaseRadius * sin(angle - M_PI / 10),
0.0f);
    glVertex3f(0.0f, 0.0f, toothHeight);

    glVertex3f(0.0f, 0.0f, 0.0f);
    glVertex3f(0.0f, 0.0f, toothHeight);
    glVertex3f(toothBaseRadius * cos(angle + M_PI / 10), toothBaseRadius * sin(angle + M_PI / 10),
0.0f);

    glEnd();

    glPopMatrix();
}

```

Продовження додатку А

```

}

glPopMatrix();

// 9. Колька на вершині фігури
glPushMatrix();
glTranslatef(0.0f, 1.1f, 0.0f);
gluSphere(m_quadric, 0.2, slices, stacks / 2);
glPopMatrix();

//10. Верхній диск
glPushMatrix();
glTranslatef(0.0f, 0.95f, 0.0f);
glRotatef(-90.0f, 1.0f, 0.0f, 0.0f);
gluDisk(m_quadric, 0.0, 0.7, slices, 4);
glPopMatrix();
}

void CChessView::OnKeyDown(UINT nChar, UINT nRepCnt, UINT nFlags)
{
    switch (nChar) {
    case VK_LEFT:
        turnY -= 5.0f;
        break;
    case VK_RIGHT:
        turnY += 5.0f;
        break;
    case VK_UP:
        turnX -= 5.0f;
        break;
    case VK_DOWN:
        turnX += 5.0f;
        break;
    default:
        CView::OnKeyDown(nChar, nRepCnt, nFlags);
        return;
    }

    if (turnX >= 360.0f) turnX -= 360.0f;
    if (turnX < 0.0f) turnX += 360.0f;

    if (turnY >= 360.0f) turnY -= 360.0f;
    if (turnY < 0.0f) turnY += 360.0f;

    Invalidate(false);

    CView::OnKeyDown(nChar, nRepCnt, nFlags);
}

```

Метод рандому формування списку пар учасників для першого раунду

```
void RoundsDlg::CreatePair() {
    //створення рандому
    if (!strings.empty()) {
        strings.clear();
    }

    for (map<CString, StructGame>::const_iterator it = mapGame.begin(); it != mapGame.end(); ++it) {
        strings.push_back(it->first);
    }

    random_device rd;
    mt19937 gen(rd());

    shuffle(strings.begin(), strings.end(), gen);

    //формування рядків з прізвищами
    CString str;
    int idx = 0;
    for (int i = 0; i < strings.size(); i += 2) {
        if (i + 1 < strings.size()) {
            str.Format(L"%-30s    %s", strings[i], strings[i + 1]);
            idx = txtListBox2.AddString(str);
            txtListBox2.SetItemData(idx, arrayIndex);
            arrayIndex++;
        }
        else {
            idx = txtListBox2.AddString(strings[i]);
            txtListBox2.SetItemData(idx, arrayIndex);
            arrayIndex++;
        }
    }
    txtRoundsNum.SetWindowText(L"Раунд №1");
}
```

Жеребкування учасників алгоритмічними та систематизованими

МЕТОДАМИ

```

void RoundsDlg::WorkWithNextRound() {
    float sum = 0.0;
    CString sRound;
    int r = 0;

    vector<CString> batchProblem(mapGame.size());
    for (int i = 0; i < batchProblem.size(); i++) {
        batchProblem[i] = L"";
    }

    //рахування результатів в останньому раунді
    if (round >= n) {

        CString line;
        CString message;
        vector<pair<CString, float>> keySumPairs;

        for (map<CString, StructGame>::const_iterator pair = mapGame.begin(); pair != mapGame.end();
++pair) {
            sum = 0.0;
            for (int i = 0; i < 100; ++i) {
                if (pair->second.points[i] == -1.0f) {
                    break;
                }
                sum += pair->second.points[i];
            }
            keySumPairs.emplace_back(pair->first, sum);
        }
        sort(keySumPairs.begin(), keySumPairs.end(), [](const pair<CString, float>& a, const pair<CString,
float>& b) {
            return a.second > b.second;
        });

        for (vector<pair<CString, float>>::iterator pair = keySumPairs.begin(); pair != keySumPairs.end();
++pair) {
            line.Format(L"%s: %.1f\n", pair->first, pair->second);
            message.Append(line);
        }

        MessageBox(message, L"Результати", MB_OK);

        ShowTournamentResults();

        return;
    }

    sRound.Format(L"Раунд №%d", round + 1);
    txtRoundsNum.SetWindowText(sRound);
    mapPoint.clear();
    for (map<CString, StructGame>::iterator it = mapGame.begin(); it != mapGame.end(); ++it) {
        for (int i = 0; i < mapGame.size() - 1; ++i) {
            it->second.futureRival[i].Empty();
        }
    }
    //1 частина

```

Продовження додатку В

```

y rival
// цикл для копіювання всіх рядків із векторних рядків у FutureRival, перевірка рядків, які вже є
for (int i = 0; i < strings.size(); ++i) {
    // перевіряємо, чи є поточний рядок strings[i] у масиві rival
    bool isAlreadyInRival = false;
    for (int j = 0; j < mapGame.size(); ++j) {
        if (it->second.rival[j] == strings[i] || it->first == strings[i] || "" == strings[i]) {
            isAlreadyInRival = true;
            break;
        }
    }
    if (!isAlreadyInRival) {
        for (r = 0; r < mapGame.size() - 1; ++r) {
            if (it->second.futureRival[r].IsEmpty()) {
                it->second.futureRival[r] = strings[i];
                break;
            }
        }
    }
}
//перевірка супротивника за кольором
CString arrCString(it->second.color);
if (arrCString.GetLength() >= 2 && (arrCString.Right(2) == "ww" || arrCString.GetLength() >= 4
&& arrCString.Right(4) == "wwbw")) {
    int k = 0;
    while (k <= r) {
        iter = mapGame.find(it->second.futureRival[k]);
        if (strcmp(iter->second.color, "ww") == 0 || strcmp(iter->second.color, "wwbw") ==
0) {
            it->second.futureRival[k].Empty();
            for (int i = 0; i < mapGame.size() - 1; ++i) {
                if (iter->second.futureRival[i] == it->first) {
                    iter->second.futureRival[i].Empty();
                    break;
                }
            }
        }
        k++;
    }
}
else if (arrCString.GetLength() >= 2 && (arrCString.Right(2) == "bb" || arrCString.GetLength() >= 4
&& arrCString.Right(4) == "bbwb")) {
    int k = 0;
    while (k <= r) {
        iter = mapGame.find(it->second.futureRival[k]);
        if (strcmp(iter->second.color, "bb") == 0 || strcmp(iter->second.color, "bbwb") == 0)
{
            it->second.futureRival[k].Empty();
            for (int i = 0; i < mapGame.size() - 1; ++i) {
                if (iter->second.futureRival[i] == it->first) {
                    iter->second.futureRival[i].Empty();
                    break;
                }
            }
        }
        k++;
    }
}
}

```

Продовження додатку В

```

        }
    }

    //2 частина, розподілення гравців по групах
    for (int i = 0; i < round; ++i) {
        sum += it->second.points[i];
    }
    itp = mapPoint.find(sum);
    if (itp != mapPoint.end()) {
        itp->second.push_back(it->first);
    }
    else {
        vector<CString> surnames;
        surnames.push_back(it->first);
        mapPoint.emplace(sum, surnames);
    }
    sum = 0.0;
}
CalculateMatches(batchProblem);
round += 1;

GetDlgItem(IDCNEXTROUND)->EnableWindow(FALSE);
GetDlgItem(IDC_UPLOAD)->EnableWindow(FALSE);
}

int RoundsDlg::FindIdeal(map<CString, StructGame>::iterator it, vector<CString>& batch, int nUsers, int& a, int& h)
{
    iter = mapGame.find(it->second.futureRival[a]);
    for (int v = 0; v < nUsers; v++) {
        if (iter->first == batch[v]) {
            return -1;
        }
        else if (batch[v] == "") {
            break;
        }
    }
    //запис ідеальної пари до batch
    if (iter->second.color[round - 1] != it->second.color[round - 1]) {
        for (int j = 0; j < nUsers; j += 2) {
            if (batch[j].IsEmpty()) {
                if (it->second.color[round-1] == 'w') {
                    batch[j] = iter->first;
                    batch[j + 1] = it->first;
                }
                else
                {
                    batch[j] = it->first;
                    batch[j + 1] = iter->first;
                }
                break;
            }
        }
    }
    else {
        return a;
    }
    return -2;
}
}

```

Продовження додатку В

```

void RoundsDlg::FindNotIdeal(map<CString, StructGame>::iterator& it, vector<CString>& batch, int last_iteration, int
nUsers) {
    iter = mapGame.find(it->second.futureRival[last_iteration]);
    for (int j = 0; j < nUsers; j += 2) {
        if (batch[j].IsEmpty()) {
            if (strcmp(iter->second.color, "bb") == 0 || strcmp(iter->second.color, "bbwb") == 0) {
                batch[j] = iter->first;
                batch[j + 1] = it->first;
                break;
            }
            else if (strcmp(iter->second.color, "ww") == 0 || strcmp(iter->second.color, "wwbw") == 0) {
                batch[j] = it->first;
                batch[j + 1] = iter->first;
                break;
            }
            else {
                if (it->second.color[round - 1] == 'w') {
                    batch[j] = iter->first;
                    batch[j + 1] = it->first;
                    break;
                }
                else {
                    batch[j] = it->first;
                    batch[j + 1] = iter->first;
                    break;
                }
            }
        }
    }
}

void RoundsDlg::FindForProblemPlayer(std::map<CString, StructGame>::iterator& it, int nUsers, vector<CString>&
batchProblem)
{
    map<float, vector<CString>>::reverse_iterator riterp = ritp;

    bool reverseBool = false;

    bool isOdd;
    bool isEmpty;
    int nRival = 0;
    int last_iteration = -1;
    for (int i = nUsers - 1; i >= 0; --i) {
        if (!it->second.futureRival[i].IsEmpty()) {
            nRival = i + 1;
            break;
        }
    }
    for (int a = 0; a < nRival; a++) {
        if (it->second.futureRival[a] == "" && a != nRival - 1) {
            continue;
        }
        //прокручуємо групу (по балам), де знаходиться гравець
        for (int h = 0; h < riterp->second.size(); h++) {
            if (riterp->second[h] == it->first && a != nRival - 1 && h != riterp->second.size() - 1) {
                continue;
            }
            isOdd = (nUsers & 1) == 1; //побітова операція на визначення непарної к-ті гравців
            isEmpty = !batchProblem[nUsers - 2].IsEmpty(); //для непарної к-ті гравців (чи всі,
окрім останнього, знайшли собі ворога)
            //перевірка останнього гравця без пари

```

Продовження додатку В

```

if (isOdd && isEmpty) {
    batchProblem[nUsers - 1] = it->first;
    a = nUsers;
    break;
}
//пошук "ідеальної" пари
if (it->second.futureRival[a] == riterp->second[h]) {
    last_iteration = FindIdeal(it, batchProblem, nUsers, a, h);
    if (last_iteration == -2) {
        a = nUsers;
        break;
    }
    else if (last_iteration == -1 && a != nRival - 1) {
        break;
    }
}
//h - останній в групі, a - останній в futureRival, last_iteration - неідеальний ворог з
однаковим кольором
if (h == riterp->second.size() - 1 && a == nRival - 1 && last_iteration != -1) { //пошук
неідеального ворога
    FindNotIdeal(it, batchProblem, last_iteration, nUsers);
    a = nUsers;
    break;
}
else if (h == riterp->second.size() - 1 && a == nRival - 1) { //якщо гравцю не підійшли
супротивники в його групі
    //переходимо до наступ групи (по очкам)
    if (!reverseBool) {
        riterp++;
    }
    else {
        riterp--;
    }
    if (riterp != mapPoint.rend()) {
        a = -1;
        h = -1;
        break;
    }
    else if (!reverseBool) {
        riterp = ritp;
        riterp--;
        reverseBool = true;
        a = -1;
        h = -1;
        break;
    }
    else {
        //...
    }
}
}
}
CalculateMatches(batchProblem);
}

```

Рахування очок, прогресу і коефіцієнту Бухгольца

```

void RoundsDlg::ShowTournamentResults() {
    vector<tuple<CString, float, float, float>> results; // name, sum, progress, buchholz
    map<CString, StructGame>::iterator opp;
    CString name;
    StructGame game;
    float progress;
    float cur;
    float buchholz;
    float opp_sum;
    for (map<CString, StructGame>::iterator it = mapGame.begin(); it != mapGame.end(); ++it) {
        name = it->first;
        game = it->second;

        progress = 0.0f;
        cur = 0.0f;

        for (int i = 0; i < 100 && game.points[i] != -1.0f; ++i) {
            cur += game.points[i];
            progress += cur;
        }

        // Підрахунок Бухгольца
        buchholz = 0.0f;
        for (int i = 0; i < 100; ++i) {
            opp = mapGame.find(game.rival[i]);
            if (opp != mapGame.end()) {
                opp_sum = 0.0f;
                for (int j = 0; j < 100 && opp->second.points[j] != -1.0f; ++j) {
                    opp_sum += opp->second.points[j];
                }
                buchholz += opp_sum;
            }
        }

        results.emplace_back(name, cur, progress, buchholz);
    }

    // Сортуння за сумою очок, далі за Бухгольцем, далі за прогресом
    sort(results.begin(), results.end(), [](const auto& a, const auto& b) {
        if (get<1>(a) != get<1>(b)) return get<1>(a) > get<1>(b);
        if (get<3>(a) != get<3>(b)) return get<3>(a) > get<3>(b);
        return get<2>(a) > get<2>(b);
    });

    // Вивід результатів
    CString message;
    CString line;
    for (const vector<tuple<CString, float, float, float>>::iterator::value_type result : results) {
        line.Format(L"%s: %.1f очок, Прогрес %.1f, Бухгольц %.1f\n",
            get<0>(result), // name
            get<1>(result), // sum
            get<2>(result), // progress
            get<3>(result)); // buchholz
        message.Append(line);
    }

    MessageBox(message, L"Результати", MB_OK);
}

```

Продовження додатку Г

```

void RoundsDlg::SetMap(map<CString, StructGame>& inputGame, map<CString, StructGame>::iterator iter) {
    mapGame = inputGame;
    iter = iter;
}

void RoundsDlg::RoundResults() {
    float rivalPoints;
    CString result;

    const int roundsQuantity = 3;

    for (int i = 0; i < round; i += roundsQuantity) {
        result.Empty(); // Очистити результат перед новою порцією

        // Виводимо по 3 раунди або менше, якщо кінець
        for (int r = i; r < i + roundsQuantity && r < round; ++r) {
            result.AppendFormat(L"%d:\n", r + 1);

            for (map<CString, StructGame>::iterator it = mapGame.begin(); it != mapGame.end(); ++it)
            {
                if (it->second.points[r] == -1.0f || it->second.color[r] == 'b')
                    continue;

                // Розрахунок балів суперника
                if (it->second.points[r] == 1.0f) rivalPoints = 0.0f;
                else if (it->second.points[r] == 0.5f) rivalPoints = 0.5f;
                else rivalPoints = 1.0f;

                if (it->second.rival[r].IsEmpty()) {
                    result.AppendFormat(L"(%0.1f) %s\n", it->second.points[r], (LPCTSTR)it->first);
                }
                else {
                    result.AppendFormat(L"(%0.1f) %s - %s (%0.1f)\n", it->second.points[r],
                    (LPCTSTR)it->first, (LPCTSTR)it->second.rival[r], rivalPoints);
                }
            }

            result.Append(L"\n");
        }

        // Додаємо підрахунок очок до поточного раунду
        result.Append(L"Поточні суми очок:\n");
        vector<pair<CString, float>> keySumPairs;
        float sum;
        for (map<CString, StructGame>::iterator it = mapGame.begin(); it != mapGame.end(); ++it) {
            sum = 0.0f;
            for (int k = 0; k < i + roundsQuantity && k < round; ++k) {
                if (it->second.points[k] == -1.0f) break;
                sum += it->second.points[k];
            }
            keySumPairs.emplace_back(it->first, sum);
        }

        // Сортуємо по спаданню
        sort(keySumPairs.begin(), keySumPairs.end(), [](const std::pair<CString, float>& a, const
        std::pair<CString, float>& b) {
            return a.second > b.second;
        });
    }
}

```

Продовження додатку Г

```
for (const vector<pair<CString, float>>::iterator::value_type pair : keySumPairs) {
    CString line;
    line.Format(L"%s: %.1f\n", (LPCTSTR)pair.first, pair.second);
    result.Append(line);
}

MessageBox(result, L"Результати раундів", MB_OK);
}
}
```