

**ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ  
ІМЕНІ СЕМЕНА КУЗНЕЦЯ**

**ФАКУЛЬТЕТ ЕКОНОМІЧНОЇ ІНФОРМАТИКИ**

**КАФЕДРА ІНФОРМАЦІЙНИХ СИСТЕМ**

## **Пояснювальна записка**

до дипломної роботи

**МАГІСТРА**

на тему: «Аналіз та оптимізація інформаційної системи протоколювання спортивних змагань з уніфікованим алгоритмом розподілу спортсменів»

Виконав: студент 2 року навчання,  
за освітнім ступенем «магістр»  
зі спеціальності 126 «Інформаційні системи та  
технології»

Партола А.А.

Керівник: к.т.н., проф. Щербаков О.В.

Харків – 2019 рік

## РЕФЕРАТ

Пояснювальна записка до дипломної роботи: 81 сторінка, 39 рисунків, 6 таблиць, 2 додатки, 54 джерела.

Мета роботи – оптимізація інформаційної системи протоколювання Харківської обласної федерації контактних єдиноборств.

Завданням роботи є аналіз можливих способів розподілу спортсменів за турнірною сіткою протоколів спортивних змагань та розроблення уніфікованого алгоритму розподілу спортсменів.

Об'єктом дослідження є існуючий процес організації спортивних змагань Харківської обласної федерації контактних єдиноборств.

Предметом дослідження є процес розподілу спортсменів за протоколами спортивних змагань.

В результаті було проаналізовано можливі способи розподілу спортсменів за турнірною сіткою протоколів спортивних змагань. Також було виконано опис математичної задачі протоколювання та запропоновано алгоритм генерації протоколів, складність якого є логарифмічною та залежить від кількості учасників, що належать до відповідного протоколу спортивних змагань.

АЛГОРИТМ, СПОРТИВНІ ЗМАГАННЯ, ТУРНІР, ІНФОРМАЦІЙНА СИСТЕМА, ПРОГРАМУВАННЯ, БАЗА ДАНИХ, ОПТИМІЗАЦІЯ, JAVA, SPRING FRAMEWORK

## ABSTRACT

The diploma work report: 81 pages, 39 figures, 6 tables, 2 appendices, 54 sources.

The purpose of the work is to optimize the information system for protocols creation of the Kharkiv Regional Federation of Contact Martial Arts.

The task of the work is to analyze the possible ways of distributing participants according to the tournament grid of the protocols of sports competitions and to develop a unified algorithm for the distribution of participants.

The object of the investigation is the existing process of organizing sports competitions for the Kharkiv Regional Federation of Contact Martial Arts.

The subject of the study is the process of distribution of participants according to the protocols of sports competitions.

As a result, possible ways of allocating participants to the tournament grid of sports event protocols were analyzed. A description of the mathematical problem of protocols creation was also performed and an algorithm for generating protocols was proposed, the complexity of which is logarithmic and depends on the number of participants belonging to the corresponding protocol of sports competitions.

ALGORITHM, SPORTS, TOURNAMENT, INFORMATION SYSTEM, PROGRAMMING, DATABASE, OPTIMIZATION, JAVA, SPRING FRAMEWORK

## ЗМІСТ

ВСТУП.....	7
1 ХАРАКТЕРИСТИКА ОРГАНІЗАЦІЇ.....	9
1.1 Характеристика діяльності.....	9
1.2 Особливості менеджменту .....	18
1.3 Особливості роботи організації.....	19
2 АНАЛІЗ ІНФОРМАЦІЙНОЇ СИСТЕМИ ПРОТОКОЛЮВАННЯ .....	21
2.1 Аналіз предметної області .....	21
2.2 Опис методів та технологій, що використовувалися на підприємстві....	25
2.3 Аналіз методів та технологій, що можуть використовуватися для рішення поставлених бізнес-задач .....	31
2.4 Обґрунтування вибору методів та технологій для оптимального вирішення бізнес-задач.....	43
3 ОПТИМІЗАЦІЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ПРОТОКОЛЮВАННЯ .....	45
3.1 Опис виконаної роботи.....	45
3.2 Опис використання запропонованих методів та технологій для вирішення поставлених задач .....	59
3.3 Критичний аналіз отриманих результатів .....	70
ВИСНОВКИ.....	74
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	75
ДОДАТОК А.....	80
ДОДАТОК Б .....	81

## ВСТУП

Для ефективної роботи спортивної організації актуальним є питання оперативної організації спортивних змагань. Полегшення організації спортивних заходів є надзвичайно важливою складовою будь-якої спортивної секції, проте дедалі ця задача стає більш складнішою, адже з кожним спортсменом пов'язаний надзвичайно великий обсяг даних: фізичні показники, спортивний клуб, спортивні розділи, в яких спортсмен бере участь.

Рівень організації змагань впливає на ефективність та продуктивність усієї спортивної організації в цілому. Головною метою спортивних змагань є підвищення кваліфікації спортсменів та суддів, підтримка спортивних талантів та популяризація здорового способу життя.

Ступінь автоматизації роботи організації суттєво впливає на його продуктивність. Використання автоматизованих інформаційних систем надає цілий комплекс, придатний для зберігання даних, виконання дій та модифікацій над інформацією, проведення обчислень за допомогою спроектованого інтерфейсу користувача. В першу чергу вони призначені для ведення сучасних методів управління. Тому важливою умовою підвищення ефективності процесу даної предметної області є його автоматизація. Аналіз та оптимізація інформаційної системи протоколювання спортивних змагань дасть змогу поліпшити ведення обліку спортсменів, зменшити кількість паперових ресурсів та помилок, виникаючих у зв'язку з людським фактором, більш раціонально використовувати наявні трудові ресурси, розподіляти робочий час організаторів змагань.

Для оптимізації інформаційної системи протоколювання спортивних змагань система автоматизації повинна охоплювати аспекти, що стосуються обліку спортсменів та тренерів організації. На сьогоднішній день існує декілька подібних систем, але специфіка додатків відрізняється в залежності від конкретних особливостей діяльності підприємства, організаційної структури управління та інших факторів. Тому актуальною є задача аналізу та оптимізації

інформаційної системи протоколювання спортивних змагань з уніфікованим алгоритмом розподілу спортсменів.

Метою роботи є оптимізація інформаційної системи протоколювання Харківської обласної федерації контактних єдиноборств з можливістю розподілу спортсменів за олімпійською системою, де кількість учасників не обов'язково повинна дорівнювати ступеню двійки таким чином, щоб спортсмени з різних команд зустрічались поряд у турнірній сітці якомога рідше.

Завданням роботи є аналіз можливих способів розподілу спортсменів за турнірною сіткою протоколів спортивних змагань та розроблення уніфікованого алгоритму розподілу спортсменів.

Об'єктом дослідження є існуючий процес організації спортивних змагань Харківської обласної федерації контактних єдиноборств.

Предметом дослідження є процес розподілу спортсменів за протоколами спортивних змагань.

Під час дослідження було створено проаналізовано математичні задачі, що можуть бути застосовані для розподілу спортсменів за турнірною сіткою та створено опис математичної задачі протоколювання, а також виконано огляд очікуваних результатів створення протоколів при різній кількості учасників.

Для оптимізації був використаний метод приведення задачі розподілу спортсменів до математичної задачі розбиття графів на підграфи, де кожен підграф відповідає одній з груп турнірної таблиці, а також метод жеребкування «змійкою», коли спортсмени в першу чергу впорядковуються за рейтингом, після чого в залежності від рейтингу заповнюється турнірна сітка.

Для реалізації створеного алгоритму було розроблено веб-додаток з можливістю керування даними, що необхідні для створення протоколів спортивних змагань, а також можливістю генерування протоколів на основі запропонованого алгоритму.

## 1 ХАРАКТЕРИСТИКА ОРГАНІЗАЦІЇ

### 1.1 Характеристика діяльності

#### 1.1.1 Основні напрями та види діяльності

Об'єктом управління є Харківська обласна федерація контактних єдиноборств. Організація була створена у 2006 році у місті Харкові. З цього часу федерація стрімко нарощує свій потенціал і зарекомендувала себе надійним і вигідним організатором спортивних змагань у місті Харкові.

Обласна федерація контактних єдиноборств організує подальший розвиток і популяризацію контактних єдиноборств в м. Харкові; підвищення майстерності спортсменів та суддів; виявлення та підтримку юних талановитих спортсменів і створення умов для зростання їх спортивного потенціалу; популяризацію масових форм, організацію дитячо-юнацького та молодіжного дозвілля; профілактику і попередження наркоманії серед молодіжного середовища; розвиток і зміцнення особистих зв'язків між спортсменами різних клубів.

Харківська обласна федерація контактних єдиноборств здійснює діяльність в області спорту. Вона є неурядовою громадською організацією, що вирішує питання, стосовно фізичного виховання, спорту та фізичної культури.

Основними завданнями організації є:

- підготовка спортсменів до спортивних змагань, покращення їх фізичної форми та спортивних навичок;
- організація спортивних змагань, залучаючи, якщо це потрібно, спонсорів;
- публікація календаря змагань;
- розробка правил проведення спортивних змагань та атестацій, критерій допуску екіпірування, нормативів;
- встановлення вікових та вагових категорій для спортивних змагань;
- подальший розвиток і популяризація організації в м. Харкові;
- підвищення майстерності спортсменів та суддів;
- виявлення та підтримка юних талановитих спортсменів і створення умов для зростання їх спортивного потенціалу;

- популяризація масових форм, організація дитячо-юнацького і молодіжного дозвілля;

- профілактика і попередження наркоманії серед молодіжного середовища;

- розвиток і зміцнення особистих зв'язків між спортсменами різних клубів;

- популяризація спорту.

Одною з основних задач організації є проведення спортивних змагань. За місяць до початку змагань організація розповсюджує інформацію про розділи, вікові та вагові категорії, місце проведення і час проведення змагань. За два тижні до проведення змагань тренери та представники спортивних організацій, федерацій та клубів, які приймають участь у змаганнях, повинні подати заявку до змагань у електронному або письмовому вигляді, яка заповнена за зразком. На основі поданих заявок працівники організації виконують розподіл спортсменів, що внесені до заявок за розділами, віковими та ваговими категоріями, що встановлені у офіційному положенні змагань, та заносять їх до протоколів змагань таким чином, щоб спортсмени з одної команди зустрічалися у парі якомога рідше.

До основних послуг організації відносяться проведення тренувань, організація та проведення спортивних змагань та забезпечення спортсменів спортивним обладнанням.

Найбільш важливим напрямом діяльності Харківської обласної федерації контактних єдиноборств є система спортивних змагань.

Спортивні змагання є дуже важливою частиною всього навчального та тренувального процесу. Спортсмен не може досягти високих результатів, якщо не буде регулярно приймати участь у змаганнях.

Дуже велике виховне значення мають коректно організована підготовка до спортивних змагань та участь в цих змаганнях.

У спортивній практиці федерації широко поширені різноманітні види змагань: розіграші першостей, кубків, спартакіади тощо. Вибір того чи іншого виду змагань в першу чергу залежить від завдань, що ставляться перед змаганнями.



Спортивні змагання можуть виконувати первинні та вторинні функції. До основних особливостей проведення спортивних змагань відносяться:

- сувора регламентація діяльності учасників змагань, обслуговуючого персоналу та інших осіб;
- багатофункціональність і багатоструктурність діяльності, яка відбувається в екстремальних умовах;
- присутність в кожному змаганні боротьби, протиборства, суперництва, що виявляються в прагненні до перемоги, максимального результату;
- висока значущість для спортсмена і процесу боротьби як такої, і досягнутого результату;
- високий рівень емоційного фону діяльності, викликаного граничними фізичними і психічними зусиллями в боротьбі за результат;
- взаємодія супротивників, в ході якого кожен спортсмен прагне до досягнення переваги над суперниками;
- забезпечення рівності умов і порівнянності результатів всіх учасників змагань за допомогою певного стандарту;
- певні форми і вимоги, виконання яких дозволяє привласнити спортивне звання.

Складовими частинами будь-якого спортивного змагання є:

- спортсмени-суперники (учасники);
- предмет змагань;
- суддівство (правила проведення змагань, критерії та порядок визначення місць);
- кошти і служби забезпечення змагальної діяльності;
- вболівальники і глядачі;
- ієрархія місць учасників змагань.

Участь в спортивних змаганнях приймає не менше двох суперників. Взаємодія спортсменів в ході змагань проявляється в двох ключових аспектах:

- по відношенню до супротивника – у формі протиборства (суперництва);
- по відношенню до партнерів по команді – в формі взаємодії (співпраці).

Протиборство між суперниками або командами, що змагаються, може бути безпосереднє (жорсткий, нежорсткий, умовно-жорсткий контакт) і опосередковане (без контакту). Співпраця між партнерами теж може проявлятися в різних формах: у формі спільно взаємопов'язаних, спільно синергетичних, спільно послідовних і спільно індивідуальних дій. Ці особливості в більшості випадків зумовлюють зміст і структуру всієї змагальної діяльності спортсменів. Залежно від наявної структури зв'язків між що беруть участь в змаганні спортсменами-суперниками і спортсменами-партнерами може бути три форми ведення змагального поєдинку: індивідуальна, групова, командна і індивідуально-групова. Ці форми взаємодії проявляються в ході особистих, командних і особисто-командних змагань.

Основним організатором і керівником проведених змагань, особою, відповідальною за їх проведення є головний спортивний суддя. Він повинен бути, в першу чергу, бездоганим знавцем правил змагань, чесним, неупередженим, коректним, спокійним та людиною, яка користується авторитетом і поза змаганнями.

Об'єктивність і точність встановлення результатів залежить від великої кількості факторів:

- від психофізіологічних можливостей самого судді, які прямо чи опосередковано впливають на оцінку річних результатів;

- від наявності об'єктивних засобів і технологій реєстрації результатів в певних видах спорту;

- від суддівства – це одна з основних проблем проведення будь-якого досить великого змагання;

- від зовнішніх умов проведення змагання (освітлення, температура навколишнього середовища).

В системі спортивної підготовки Харківська обласна федерація контактних єдиноборств проводить підготовчі, контрольні, підвідні, відбіркові та головні змагання.

Підготовчі змагання. Основна їх мета – це адаптація спортсменів до всіх

умов змагань, відпрацювання валідних технічних рішень в різних ситуаціях діяльності змагань, а також становлення відповідних тактовних варіантів і розвиток здатності до їх реалізації при змагальному суперництві, придбання відповідного досвіду, вдосконалення окремих психічних якостей. Підготовчі змагання грають важливу роль як ефективний засіб інтегральної підготовки учасників.

Контрольні змагання. Вони проводяться для контролю рівня підготовленості спортсмена. У них проводиться перевірка ефективності попереднього етапу підготовки, виконується оцінка рівня розвитку фізичних якостей, технічного та тактичного професіоналізму, інтелектуального і психологічного рівня спортсмена, виявляються кращі та гірші сторони в структурі змагальної діяльності. За отриманими результатами контрольних змагань створюється програма наступної підготовки, що включає усунення знайдених недоліків для успішного виступу на відбіркових і головних змаганнях. Контрольну функцію можуть виконувати як офіційні змагання різного рівня, так і спеціально організовані контрольні змагання. Їх програма може значно відрізнитися від програми відбіркових і головних змагань і бути складена з урахуванням необхідності контролю за рівнем розвитку окремих сторін.

Підвідні змагання. У таких змаганнях покращується модель змагальної активності, що спортсмен наміряний виконати в головних змаганнях. Потреба опрацювання різних способів змагальної активності передбачає підбір спортсменів, з якими можна в найкраще підготувати той чи інший технічний або тактичний варіант. Такі змагання варто проводити при умовах, що найбільшим чином наближені до умов головних змагань. Це можуть бути як офіційні так і спеціально організовані змагання.

Відбіркові змагання. За результатами цих змагань виконується комплектація команди, відбирають спортсменів для головних змагань. Залежно від принципу, що покладено як основу комплектування складу спортсменів головних змагань, у відбіркових основне завдання учасника – завоювати певне місце, виконати необхідний норматив, що дасть можливість потрапити до складу команди і

сподіватися на успішний виступ в головних змаганнях.

Головні змагання. Мета участі в таких змаганнях – досягнення перемоги або завоювання якнайбільш високого місця. У них спортсмени орієнтуються на досягнення найвищих результатів, повну мобілізацію та прояв технічних, фізичних, психічних і тактичних можливостей.

Контрольні та підготовчі змагання займають дуже велике місце у всьому тренувальному процесі. Для досвідчених спортсменів такі змагання з програмою, що часто змінюється, організовуються зазвичай з інтервалом в один або два тижні. Підвідні змагання плануються зазвичай у другій половині підготовчого періоду і в змагальному періоді. Міський турнір і головні змагання переважно проводяться не частіше двох або трьох разів на рік.

Успішне проведення спортивних змагань дуже залежить від правильної і гарно продуманої підготовки до них. Вся підготовча робота поділяється на кілька важливих етапів.

Підготовка до змагань повинна бути розпочата після того, як федерація, яка їх організує, затвердить положення, в якому вказані назва змагань, час і місце проведення, їх мета і завдання, керівництво (хто організовує і проводить змагання), організації, строки подання заявок (попередніх і остаточних), умови проведення змагань та визначення переможців.

Основна вимога до положення про змагання полягає в тому, що всі пункти його повинні бути викладені чітко з тим, щоб виключити їх різне тлумачення.

Положення повинне бути складене відповідно до вимог чинних правил. Воно є не тільки організаційним, а й методичним документом, що визначає напрямок навчально-тренувальної роботи в період підготовки до змагань. Після затвердження повноважним організатором положення є обов'язковим для суддівської колегії та спортсменів.

Після затвердження положення про спортивні змагання створюють суддівську колегію, призначають відповідальну особу від організації, що проводить змагання, складають кошторис витрат, на підставі якої організація, що проводить ці змагання, отримує грошові кошти і матеріальні цінності. При

складанні кошторису вказують найменування змагань, терміни і місце їх проведення, кількість суддів і спортсменів (в тому числі іногородніх). У кошторисі передбачено також витрати з оренди приміщень, оформлення місця змагань, транспортні витрати, а також оплату обслуговуючого персоналу (лікарів, робітників та медичних сестер).

Положення повинно бути відомо командам, що приймають участь за 3-4 місяці до початку змагань. При підготовці до головних змагань створюють організаційний комітет. У план роботи комітету входить організація заходів, що необхідні для успішної підготовки до змагань та їх проведення (вибір і підготовка місця змагань, оформлення документації, підготовка інвентарю, обладнання, розміщення учасників, представників команд, суддів). Організаційний комітет також може залучати до своєї роботи громадський актив.

Другий, більш відповідальний та складний, період підготовки до змагань включає вибір і обладнання місця змагань; складання графіка і програми проведення змагань, попередню інформацію про змагання, підготовку афіш та плакатів.

Обладнання та місце проведення змагань повинні відповідати вимогам правил проведення змагань.

Перед головними змаганнями головний суддя, і представники організації, що проводить змагання проводять конференцію, на якій роз'яснюють положення про майбутні змагання (цілі, систему проведення, завдання та порядок визначення переможця).

До заходів організаційного та методичного характерів, що пов'язані з організацією спортивних змагань, по-перше відноситься створення календарного плану змагань, де позначається назва змагань, місце їх проведення і терміни та відповідальні за їх проведення особи.

Важливим пунктом при складанні календарного плану є традиційність заходів та термінів для команд та осіб, що беруть участь. Це підвищує спортивний інтерес до них, робить організацію навчального та тренувального процесу легше, робить змагання більш популярними для спортсменів. Такі спортивні змагання

сприяють покращенню спортивних результатів та є стимулом для регулярних тренувань, якщо вони проводяться систематично. Необхідність участі у певних змаганнях залежить від фізичної підготовки учасників.

Таким чином, календар спортивних змагань треба складати так, щоб усі змагання були якомога різноманітнішими за масштабом, складом спортсменів і умовами проведення, традиційними за термінами, складом й місцем проведення.

Інша важлива міра організаційного та методичного характеру – це створення офіційного положення змагань. У положенні вказується:

- назва змагань;
- дата та час початку;
- локація;
- мета;
- структура;
- склад спортсменів;
- спосіб визначення переможців;
- форма нагородження.

Для участі у змаганнях потрібно у визначені терміни надати заявку в чорновій формі, а потім в остаточній. Чорнова заявка містить бажання команди брати участь в змаганнях. У остаточній заявці приводяться дані про склад команд і деякі інші відомості. На підставі поданих заявок будуть складені протоколи змагань.

Для проведення змагань важливо подбати про приведення місця, обладнання та інвентарю в повну відповідність до встановлених правил, їх високій якості і необхідній кількості.

Для обслуговування спортивних змагань відповідно до їх масштабу і виду, організація призначає лікаря та інший медичний персонал.

Основним документом, на основі якого регулюється проведення змагань і впливає на їх результати, є правила змагань, у яких передбачаються умови виявлення переможців, регламентуються дії суддів і учасників і також визначаються норми поведінки учасників, міститься перелік заборонених дій, що

тягнуть за собою покарання, що ущемляє інтереси команди. Таким чином, на учасника, що порушив правила, діє не тільки рішення судді по відношенню до нього особисто, а й факт, що команда відчуває шкоди за його неправильних вчинків.

### 1.1.2 Організаційна структура

Харківська обласна федерація контактних єдиноборств налічує 3 тренери, 13 суддів, лікаря, фінансиста та менеджера, якими керує президент федерації, як показано на рис. 1.1.

Для реалізації функцій управління створюється організаційна структура управління, завдяки якій організація виживає в конкурентній боротьбі, забезпечує ефективність, окремі елементи пов'язує в єдине ціле. Помилки в структурі призводять до ослаблення всієї організаційної системи в цілому. Структура управління організації – сукупність функціональних підрозділів, взаємопов'язаних в процесі вироблення, прийняття та реалізації управлінських рішень.

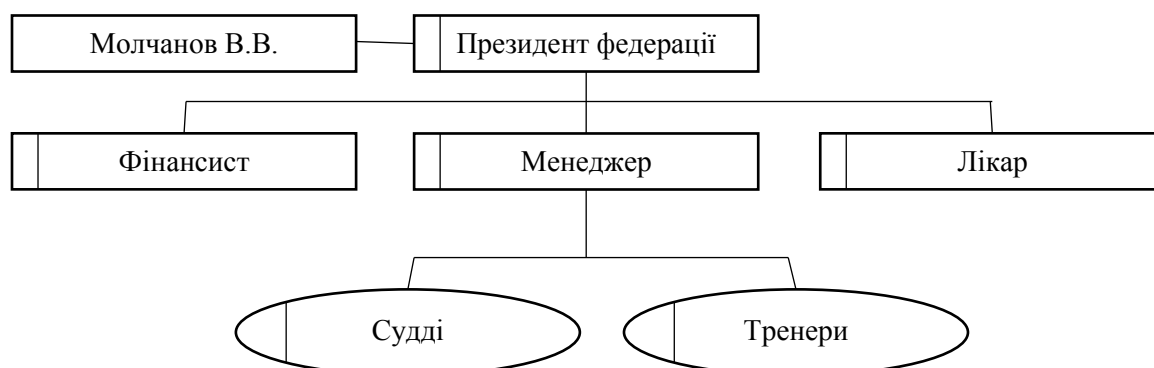


Рисунок 1.1 – Схема організаційної структури Харківської обласної федерації контактних єдиноборств

Президент федерації має в своєму підпорядкуванні інших працівників, несе всю повноту відповідальності за стан федерації та здійснює керівництво клубом. До його основних завдань відноситься забезпечення оптимальної системи взаємодій із зовнішнім середовищем, при якій федерація може успішно працювати і конкурувати. Він наділений повноваженнями вирішувати головні проблеми функціонування і розвитку федерації.

### 1.1.3 Динаміка розвитку підприємства

В майбутньому Харківська обласна федерація контактних єдиноборств планує наступні етапи розвитку підприємства:

- впровадження нових методів тренувань, що будуть мати більшу ефективність, бо будь-які зміни у методах тренування орієнтовані в першу чергу на поліпшення спортивних результатів;

- покращення умов проведення змагань, вдосконалення інвентарю, обладнання та спортивного екіпування;

- підвищення вимог до організаційного, суддівського, лікарського та інформаційного забезпечення;

- збільшення більш ніж удвічв обсягу тренувальної роботи;

- використання нових прийомів психологічної і фізіологічної спрямованості для підвищення працездатності;

- створення нових засобів відновлення спортсменів фармакологічного характеру.

Спортивні змагання крім змагальної діяльності самих учасників охоплює співучасть у змаганні та конкуренцію тренерів. Це сприяє інтенсифікації наукових розробок, пов'язаних із впровадженням нових засобів відновлення та способів підвищення ефективності загального тренувального процесу.

Крім того, потрібні значні кошти для проведення відбору та підготовки учасників, для забезпечення системи підготовки кадрів та фінансування спортивної науки й медицини.

## 1.2 Особливості менеджменту

Менеджер федерації несе пряму відповідальність за роботу виконавців. Приймає важливі рішення, що пов'язані з виконанням завдань і оптимізацією використання виділених для цього ресурсів, передає інформацію зверху вниз і знизу вгору, підтримує внутрішні і зовнішні контакти. Проаналізувавши організаційну структуру Харківської обласної федерації контактних єдиноборств,



можна сказати, що вона є лінійно-функціональною. При такій структурі управління федерацією всю повноту влади бере на себе лінійний керівник, що очолює федерацію. Йому при розробці конкретних питань і підготовці відповідних рішень, програм, планів допомагають функціональні ланки.

Менеджери малих і середніх підприємств в більшій мірі стикаються з проблемами нестабільного зовнішнього середовища, з непередбачуваними результатами діяльності. У цих умовах від підприємницьких здібностей керівників, їх вміння працювати як одна злагоджена команда, залежить ефективність управління в цілому.

У Харківській обласній федерації контактних єдиноборств переважає кланова культура, головною цінністю якої є команда. Це компанія сімейного типу, для якої характерно співучасть, взаємодопомога, згуртованість, відданість організації, тобто дружнє місце роботи, де у людей багато спільного. Керівництво сприймається як вихователі або батьки. Організація тримається разом завдяки традиції, обов'язковості та відданості співробітників. Вона робить акцент на довгостроковій вигоді вдосконалення особистості, надає значення високого ступеня згуртованості колективу і моральному клімату. Успіх визначається в орієнтації на клієнта і турботи про людей. Організація заохочує командну роботу. Метою управління кадрами у федерації є забезпечення чисельності складу, ефективне використання і консолідація інтересів.

У меншій мірі присутні ринкова культура з орієнтацією на прибуток, де головні цінності – успіх і репутація та культура, коли в довгостроковому періоді федерація, щоб залишатися конкурентною акцентується на постійному зростанні.

### **1.3 Особливості роботи організації**

Основна роль підрозділу, який було досліджено – прийом та аналіз заявок на спортивні змагання, а також створення протоколів спортивних змагань, де спортсмени будуть розподілені по турнірній сітці за олімпійською системою.

Організацією були поставлені наступні задачі:

- повний аналіз процесу організації спортивних змагань на підприємстві;

- поглиблений аналіз процесу розподілу спортсменів за протоколами спортивних змагань;

- визначення поточного алгоритму для розподілу спортсменів за турнірною сіткою протоколу спортивних змагань;

- оптимізація алгоритму таким чином, щоб була наявна можливість розподілу спортсменів не за простою олімпійською системою, а за олімпійською системою із сіяними, тобто коли кількість учасників не обов'язково повинно дорівнювати ступеню двійки;

- оптимізація алгоритму таким чином, щоб спортсмени з різних команд зустрічались поряд у турнірній сітці якомога рідше.

Також було проаналізовано загальний процес проведення змагань у Харківській обласній федерації контактних єдиноборств та організаційну структуру підприємства.

Для ознайомлення зі структурою вікових, вагових категорій а також категорій спортивних змагань Харківською обласною федерацією контактних єдиноборств було представлено приклади положень спортивних змагань, як показано на рис. А.1.

## 2 АНАЛІЗ ІНФОРМАЦІЙНОЇ СИСТЕМИ ПРОТОКОЛЮВАННЯ

### 2.1 Аналіз предметної області

Діаграма IDEF0, як показано на рис. 2.1, формалізує бізнес-процеси організації спортивних змагань [3].

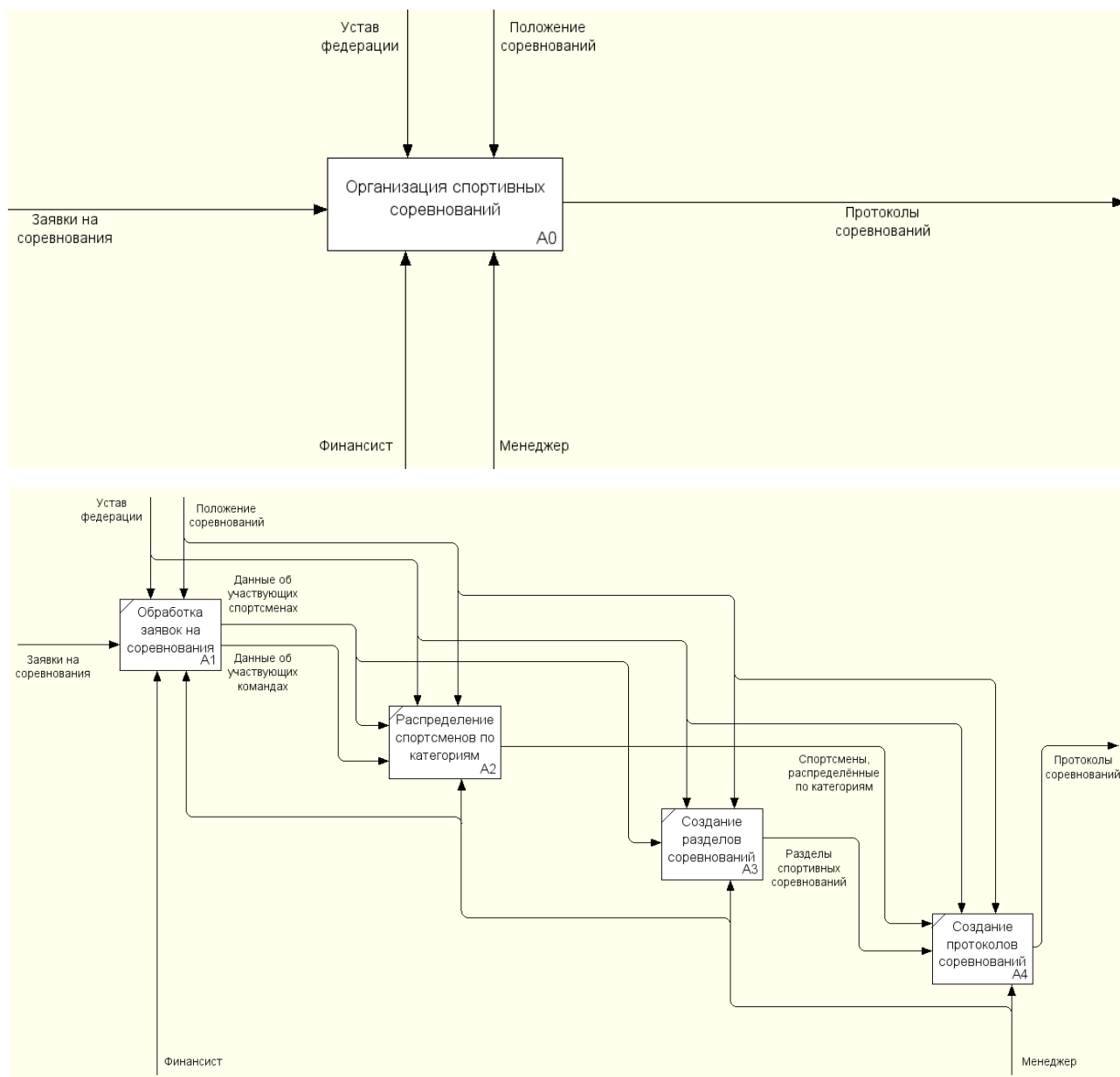


Рисунок 2.1 – Діаграма IDEF0

Бізнес-процес починається з безпосереднього надходження заяв на участь у спортивних змаганнях від тренерів. Заяви розглядаються менеджером, після чого

вносяться зміни про існуючих спортсменів. Далі інформація про нових спортсменів, тренерів та команд заноситься до бази даних. Після цього до бази даних вноситься інформація про необхідні розділи змагань, обираються команди, що приймають участь у змаганнях та відбувається зіставлення спортсменів із спортивними розділами, в яких вони приймають участь. Наступним етапом є внесення змін про вікові та вагові категорії розділів спортивних змагань за необхідністю. Після цього відбувається створення протоколів спортивних змагань шляхом розподілення спортсменів за віковими, ваговими категоріями, статтю та розділами спортивних змагань, а також розподіл кожної категорії спортсменів за олімпійською системою із сіяними по окремому протоколу змагань.

У ході проектування діаграми використання, як показано на рис. 2.2, були виявлені наступні основні варіанти використання:

- «Керування інформацією про спортсменів» – забезпечує додавання, видалення, корегування та перегляд даних про спортсменів у базі даних, їх пошук через зручний інтерфейс, вибірку даних про спортсменів з окремих команд, або окремого тренера.

- «Керування інформацією про тренерів» – забезпечує додавання, видалення, редагування та перегляд даних про тренерів у базі даних, їх пошук через зручний інтерфейс.

- «Керування інформацією про команди» – забезпечує додавання, видалення, редагування та перегляд даних про команди у базі даних, їх пошук через зручний інтерфейс, вибірку даних про команди окремих тренерів.

- «Керування інформацією про розділи змагань» – забезпечує додавання, видалення, редагування та перегляд даних про розділи змагань у базі даних, їх пошук через зручний інтерфейс, додавання, видалення та редагування даних про вікові та вагові категорії спортивних змагань, що належать окремим розділам змагань.

- «Генерація протоколів спортивних змагань» – забезпечує додавання, генерацію протоколів спортивних змагань шляхом розподілення спортсменів за відповідними категоріями.

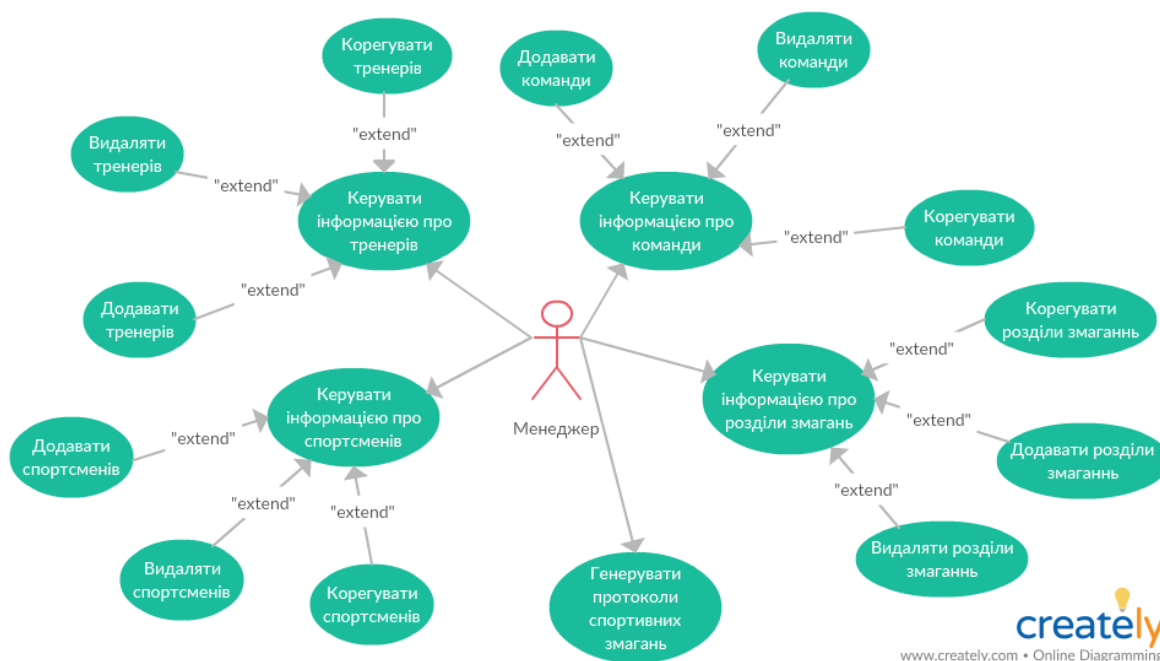


Рисунок 2.2 – Діаграма варіантів використання

Автоматизація процесу організації спортивних змагань набуває все більшої популярності, тому існує декілька реалізацій систем для вирішення проблем організації спортивних змагань різного рівня організаційного розвитку та сфер діяльності. Такі системи дозволяють оптимізувати бізнес-процеси пов'язані з організацією змагань: ведення обліку спортсменів, формування протоколів, ведення рахунку окремих поєдинків змагань, розподіл спортсменів за категоріями та багато інших задач. Найбільш поширеними програмними продуктами в Україні, які забезпечують ці функції є «Wushu Referee» та «siau.org».

Програмний продукт «Wushu Referee», інтерфейс якого показано на рис. 2.3, дозволяє автоматизувати цілий комплекс задач пов'язаний із генерацією протоколів спортивних змагань, враховуючи вимоги секцій. Серед особливостей програми можна виділити можливість керування процесом змагань під час їх проведення, а також збереження історії проведених поєдинків.

Система «siau.org», інтерфейс якої показано на рис. 2.4, також містить ряд основних функцій пов'язаних с автоматизацією організації спортивних змагань, проте функціонал системи більш обмежений. Серед переваг над іншими програмними продуктами можна виділити можливість завантаження файлу

спортсменів та категорій змагань до системи та незалежність від платформи, що забезпечує додаткову гнучкість. Порівняльний аналіз головних характеристик даних продуктів наведено у таблиці 2.1.

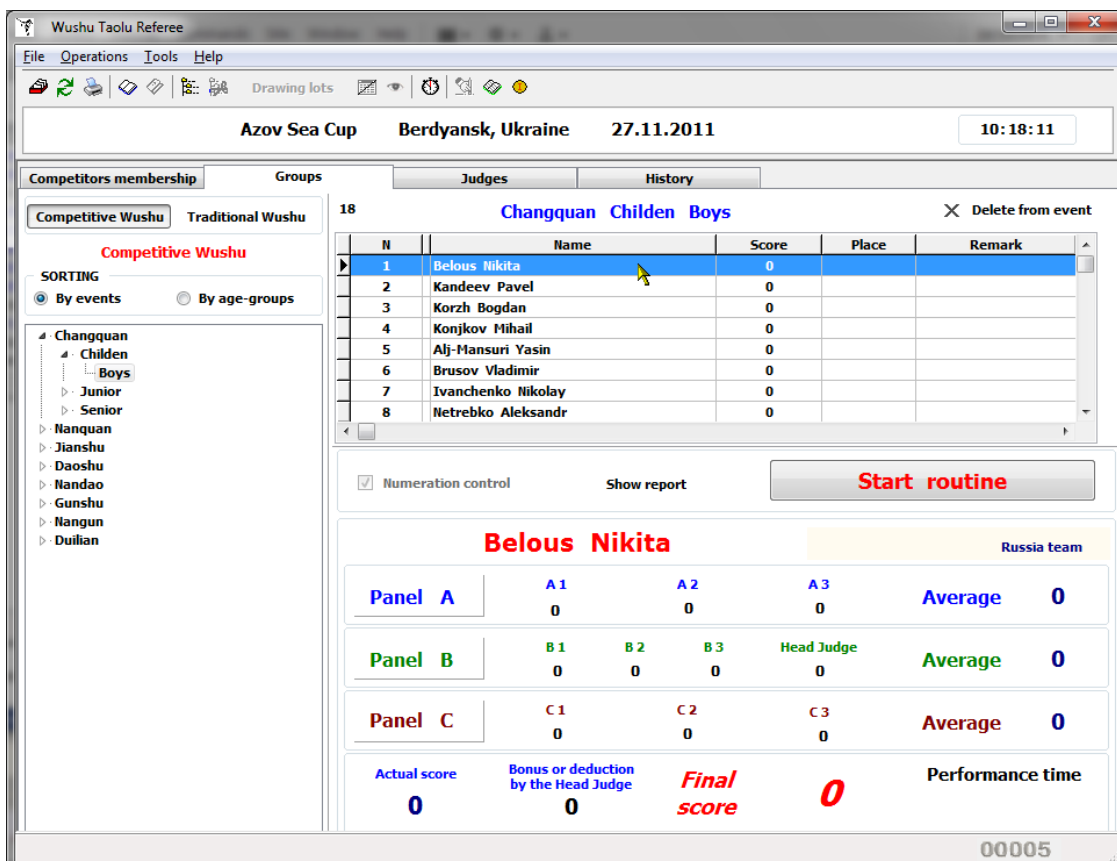


Рисунок 2.3 – Інтерфейс програми «Wushu Referee»

## Генератор протоколів змагань по Традиційному Карате

Категорії змагань:

Choose File No file chosen

Учасники змагань:

Choose File No file chosen

Код категорії (не обов'язкове поле):

Введіть код категорії

Отправить

## Рисунок 2.4 – Інтерфейс програми «siay.org»

Таблиця 2.1 - Порівняльна характеристика програмних продуктів

Фірма-розробник	Wushu Referee	Siay
Назва програмного продукту	Wushu Referee	siay.org
Версія продукту	2.1	5.2
Ліцензія	Комерційне ПЗ	Комерційне ПЗ
<b>Функціональність</b>		
Планування потреб в спортивному обладнанні	+	-
Облік спортсменів	+	+
Облік спортивних команд	+	+
Формування протоколів	+	+

Обидва програмні продукти забезпечують усі необхідні вимоги щодо функціональності бізнес-процесу, але значним недоліком є їх висока вартість.

Таким чином, рішення цих фірм-розробників було б доцільно використовувати для надання більш зручного інтерфейсу користувача у розроблюваному модулі.

## **2.2 Опис методів та технологій, що використовувалися на підприємстві**

Для проведення змагань за участю великої кількості спортсменів Харківська обласна федерація контактних єдиноборств використовувала олімпійську систему з вибуванням після першої поразки (плей-офф).

До переваг плей-офф можна віднести мінімальну кількість поєдинків, порівняно з іншими варіантами турнірів, а також «безкомпромісність» – в ньому немає ні можливості, ні сенсу в договірних нічийх. Плей-офф націлений на максимально швидке виявлення найсильнішого і забезпечує справедливе (якщо вважати силу учасників постійною і не залежить від того, хто з ким змагається) присвоєння першого місця – його займає той, хто нікому не програв, в той час як всі інші учасники турніру комусь програють.

Незручність плей-офф – в жорстких вимогах до кількості учасників. Якщо ця кількість не відповідає нормі, то єдиний вихід – за жеребом видати частині учасників технічні перемоги або технічні поразки в першому колі, що ще більше збільшує вплив випадкового фактору на результат турніру. Єдина альтернатива – випереджати турнір плей-офф серією попередніх поєдинків за вихід в основний турнір.

Плей-офф абсолютно не підходить для турнірів, де важливо забезпечити справедливий розподіл усіх місць, а не тільки першого-третього. По-перше, в плей-офф на розподіл місць, крім першого (особливо – останніх), надзвичайно сильно впливає порядок вибору пар. У разі жеребкування останні місця розподіляються практично випадково: слабкий учасник, якому жереб дає порівнянних по силі супротивників, легко може піднятися вище сильного, яким в першому ж колі дістався ще сильніший суперник.

Спроба замінити жереб на якусь осмислену систему підбору пар за рейтингами робить турнір передбачуваним. Є два варіанти такого підбору: або «сильний проти слабкого» – в кожному колі учаснику з високим рейтингом дістається противник з низьким (конкретних алгоритмів підбору може бути кілька), або «рівний з рівним» - найсильнішому дають в пару другого, третьому – четвертого і так далі. У першому випадку велика частина зустрічей виявляється передбачуваною, а тому нецікавою, у другому – половина найсильніших відсівається на перших етапах і передбачуваним виявляється фінал. Тому завжди використовують перший варіант, щоб глядач у фіналі побачив справжню гру найсильніших, а не сірий фінал нікому не цікавих команд.

Крім того, в чистому плей-офф місця, крім першого і другого, взагалі не можуть бути присвоєні (у всіх по одній поразці), і конкретне місце замінюється поняттям «вихід в етап». Якщо необхідно конкретизувати місця, зайняті учасниками, доведеться проводити додаткові гри, через що втрачається основна перевага плей-офф – швидкість.

Розклад поєдинків при цій системі складається відповідно до сітки поєдинків для заявленої кількості спортсменів. Потім проводиться жеребкування,



в результаті якого кожен спортсмен отримує певний номер, згідно з яким і включається в сітку.

Якщо кількість учасників дорівнює 4, турнірна сітка буде налічувати 2 тури, як показано на рис. 2.5.



Рисунок 2.5 – Турнірна сітка для 4 учасників

Якщо кількість учасників дорівнює 8, турнірна сітка буде налічувати 3 тури, як показано на рис. 2.6.



Рисунок 2.6 – Турнірна сітка для 8 учасників

Якщо число учасників дорівнює 6, як показано на рис. 2.7, або 10, як показано на рис. 2.8, то сітка повинна бути такою, щоб до другого туру залишилася кількість учасників, рівне ступеню двійки [4]. В даному випадку 4 або 8. Отже, згідно з жеребом, частина команд приступить до ігор у другому турі, а частина – в першому.



Рисунок 2.7 – Турнірна сітка для 6 учасників

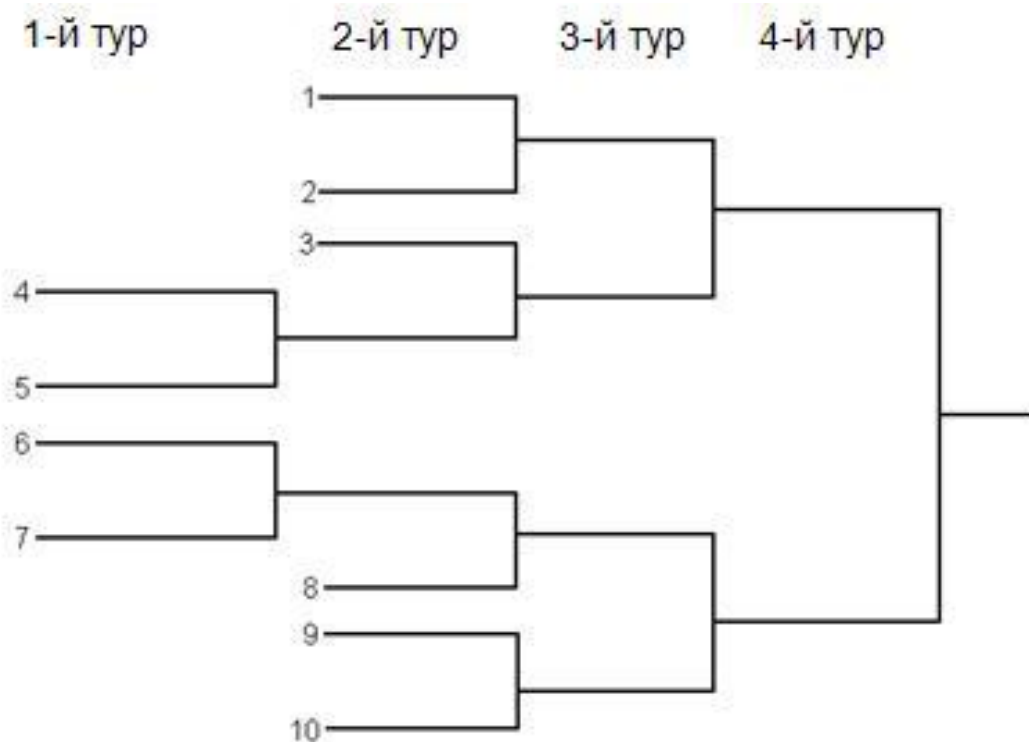


Рисунок 2.8 – Турнірна сітка для 10 учасників

При непарній кількості учасників спортсмени, котрі вступають у змагання з другого туру, розподіляються таким чином, щоб внизу сітки було на один більше.

Якщо кількість учасників дорівнює 5, як показано на рис. 2.9, або 7, як показано на рис. 2.10, турнірна сітка буде налічувати 3 тури.

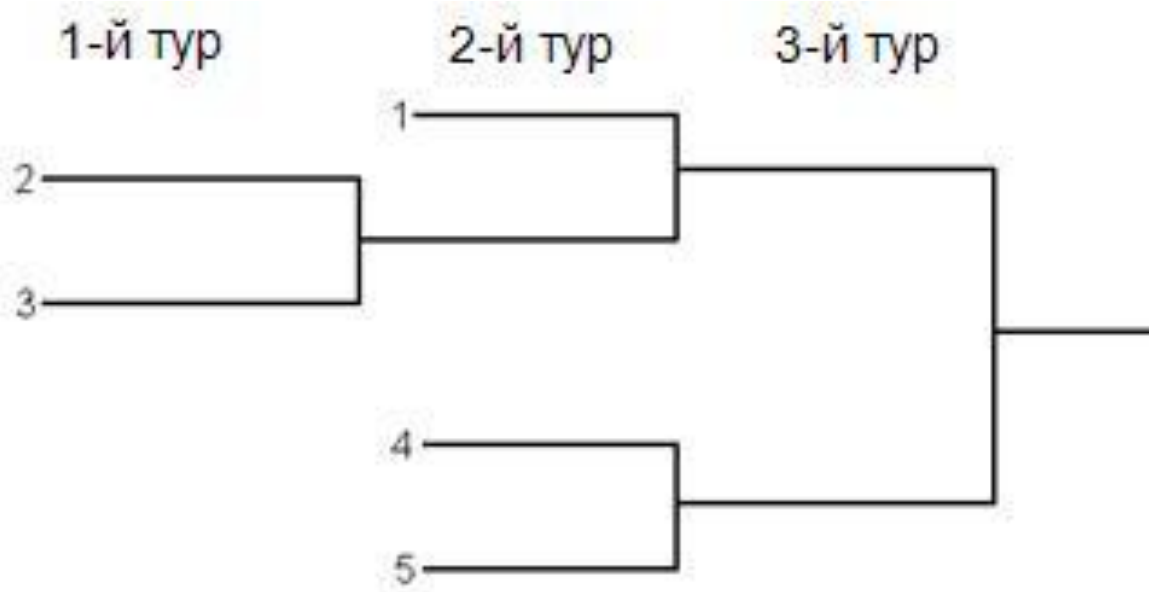


Рисунок 2.9 – Турнірна сітка для 5 учасників



Рисунок 2.10 – Турнірна сітка для 7 учасників

Якщо кількість учасників дорівнює 9, турнірна сітка буде налічувати 4 тури, як показано на рис. 2.11.



Рисунок 2.11 – Турнірна сітка для 9 учасників

Якщо кількість учасників дорівнює 19, турнірна сітка буде налічувати 5 турів, як показано на рис. 2.12.

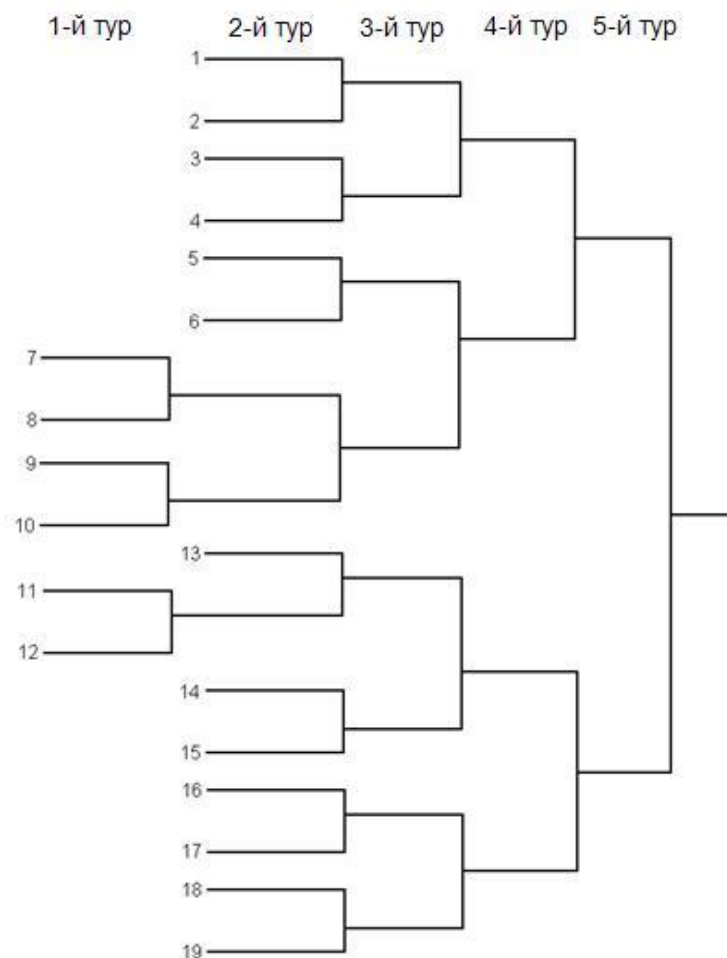


Рисунок 2.12 – Турнірна сітка для 19 учасників

Перевага системи з вибуванням полягає в можливості навіть при значній кількості спортсменів виявити переможця за мінімальний час. Однак при цьому можуть виникнути варіанти потрапляння найсильніших спортсменів в одну половину сітки і їх вибування на ранніх етапах. Щоб не допустити цього, можна штучно розсіяти сильних учасників, даючи їм за жеребом номера в різних половинах сітки, а потім провести жеребкування для решти спортсменів на вільні номери. На практиці це означає, що найсильніші спортсмени починають участь в турнірі з другого кола.

## **2.3 Аналіз методів та технологій, що можуть використовуватися для рішення поставлених бізнес-задач**

### **2.3.1 MySQL**

Система управління базами даних (СКБД) MySQL – розробка шведської

компанії MySQL AB. СУБД MySQL є програмним забезпеченням з відкритим вихідним кодом, поширюваним за ліцензією GNU (GPL) і комерційної ліцензії для ситуацій, які не підпадають під дію ліцензії GPL. MySQL підтримує реляційну модель даних, т. Е. Є реляційною СУБД [5].

Починаючи з версії 5.0, СУБД MySQL практично повністю задовольняє стандарту структурованого мови запитів SQL і, отже, сумісна з іншими базами даних.

Основні переваги СУБД MySQL:

- висока якість, бо MySQL характеризується стійкою роботою;
- поряд з Oracle, MySQL вважається однією з найшвидших СУБД в світі;
- відкритий код доступний для перегляду і модернізації, що дозволяє постійно покращувати програмний продукт;

- СУБД MySQL, розроблена з використанням мов C / C ++, протестована на багатьох платформах, серед яких Windows, Linux, FreeBSD, Mac OS X, OS / 2, Solaris;

- MySQL підтримує API (Application Programming Interface, програмний інтерфейс програми) для C, C ++, Eiffel, Java, Perl, PHP, Python, Ruby і Tcl. MySQL можна успішно застосовувати як для побудови Web-сторінок з використанням Perl, PHP і Java, так і для роботи прикладної програми, створеної з використанням Delphi, Builder C ++ або платформи .NET [6];

- СУБД MySQL надає широкий спектр типів таблиць, в тому числі і сторонніх розробників, що дає можливість реалізувати оптимальну для розв'язуваної задачі продуктивність і функціональність;

- локалізація в MySQL виконана коректно, у користувача, як правило, не виникає проблем при обробці вмісту БД з кирилицею [7].

Ще в версії MySQL 4.1 з'явилися такі важливі нововведення, як повна підтримка вкладених запитів і підтримка транзакцій. А у версії MySQL 5.0 стали доступними такі важливі механізми:

- процедури, що і функції, які об'єднують в собі цілі послідовності запитів;
- тригери – збережені процедури, прив'язані до події зміни таблиці;

- уявлення - вибірки даних, які можна уявити як повноцінні реально існуючі таблиці бази даних;

- курсори, що дозволяють циклі переглянути кожен рядок результуючої таблиці запитів;

- інформаційна схема – стерпний набір уявлень системної таблиці, в якій зберігається різноманітна внутрішня інформація;

- обробники помилок;

- безліч нових функцій.

### 2.3.2 PostgreSQL

PostgreSQL – це популярна вільна об'єктно-реляційна система управління базами даних. PostgreSQL базується на мові SQL і підтримує численні можливості.

Переваги PostgreSQL:

- підтримка БД необмеженого розміру;

- потужні і надійні механізми транзакцій і реплікації;

- розширювана система вбудованих мов програмування і підтримка завантаження C-сумісних модулів [8];

- спадкування;

- легка розширюваність.

Поточні обмеження PostgreSQL:

- немає обмежень на максимальний розмір бази даних;

- немає обмежень на кількість записів в таблиці;

- немає обмежень на кількість індексів в таблиці;

- максимальний розмір таблиці – 32 Тбайт;

- максимальний розмір запису – 1,6 Тбайт;

- максимальний розмір поля – 1 Гбайт;

- максимум полів в записі 250-1600 (в залежності від типів полів).

Функції в PostgreSQL є блоками коду, виконуваними на сервері, а не на клієнті БД. Хоча вони можуть писатися на чистому SQL, реалізація додаткової логіки, наприклад, умовних переходів і циклів, виходить за рамки власне SQL і

вимагає використання деяких мовних розширень. Функції можуть бути написані з використанням різних мов програмування. PostgreSQL допускає використання функцій, що повертають набір записів, який далі можна використовувати так само, як і результат виконання звичайного запиту. Функції можуть виконуватися як з правами їх творця, так і з правами поточного користувача. Іноді функції ототожнюються з збереженими процедурами, проте між цими поняттями є різниця.

Тригери в PostgreSQL визначаються як функції, що ініціюються DML операціями. Наприклад, операція INSERT може запускати тригер, що перевіряє доданий запис на відповідність певним умовам. При написанні функцій для тригерів можуть використовуватися різні мови програмування. Тригери асоціюються з таблицями. Множинні тригери виконуються в алфавітному порядку.

Механізм правил в PostgreSQL є механізмом створення користувацьких обробників не тільки DML операцій, а й операції вибірки. Основна відмінність від механізму тригерів полягає в тому, що правила спрацьовують на етапі розбору запиту, до вибору оптимального плану виконання і самого процесу виконання. Правила дозволяють перевизначати поведінку системи при виконанні SQL операції до таблиці [9].

Індекси в PostgreSQL наступних типів: B-дерево, хеш, R-дерево, GiST, GIN. При необхідності можна створювати нові типи індексів, хоча це далеко не тривіальний процес.

Багатоверсійність підтримується в PostgreSQL – можлива одночасна модифікація БД кількома користувачами за допомогою механізму Multiversion Concurrency Control (MVCC). Завдяки цьому дотримуються вимоги ACID, і практично відпадає потреба в блокуванні читання [10].

Розширення PostgreSQL для власних потреб можливо практично в будь-якому аспекті. Є можливість додавати власні перетворення типів, типи даних, домени (призначені для користувача типи з самого початку накладеними обмеженнями), функції (включаючи агрегатні), індекси, оператори (включаючи



перевизначення вже існуючих) і процедурні мови.

Спадкування в PostgreSQL реалізовано на рівні таблиць. Таблиці можуть успадковувати характеристики і набори полів від інших таблиць (батьківських). При цьому дані, додані в породжену таблицю, автоматично будуть брати участь (якщо це не зазначено окремо) в запитах до батьківської таблиці [11].

В розробці сайтів і веб-додатків PostgreSQL використовується не так часто, ніж MySQL, але все ж ця пара з помітним відривом випереджає за частотою використання інші системи управління базами даних. Більшість фреймворків (наприклад, Ruby on Rails, Yii, Symfony, Django) підтримують використання PostgreSQL [12].

### 2.3.3 Microsoft SQL Server

Microsoft SQL Server – система керування базами даних (РСУБД), розроблена корпорацією Microsoft. Основний використовуваний мову запитів - Transact-SQL, створений спільно Microsoft та Sybase. Transact-SQL є реалізацією стандарту ANSI/ISO структурованої мови запитів (SQL) з розширеннями. Використовується для роботи з базами даних розміром від персональних до великих баз даних масштабу підприємства; конкурує з іншими СУБД в цьому сегменті ринку [13].

Розглянемо основні можливості та переваги Microsoft SQL Server:

- SQL Server 7.0 масштабується в діапазоні від портативних комп'ютерів з Windows 95 або Windows 98 до мультипроцесорних кластерів, що працюють під управлінням Windows NT® Server Enterprise Edition;

- SQL Server 7.0 містить заново спроектований процесор запитів, який забезпечує підтримку баз даних дуже великого обсягу та обробку складних запитів, серед його нових особливостей – використання складових індексів, нові алгоритми хешування і злиття, множинні тригери, а також обробка гетерогенних, розподілених і паралельних запитів [14];

- збільшений до 8 КБ розмір сторінок сприяє швидкому вилученню даних, дозволяє використовувати такі рядки і стовпці більшого розміру, що відкриває можливість ефективного зберігання складних, докладних даних;

- менеджер блокувань динамічно адаптує алгоритм використання ресурсів у великих базах даних, що робить продукт найбільш придатним для інтерактивної обробки транзакцій (online transaction processing – OLTP) і створення сховищ даних;

- виконання багатьох рутинних завдань адміністрування тепер автоматизовано, алгоритми управління пам'яттю і блокуванням адаптуються динамічно, розмір файлів автоматично збільшується і скорочується, крім того, кошти автоматичного налаштування динамічно налаштовують алгоритми використання ресурсів в залежності від робочого навантаження [15];

- готовність до використання в Інтернеті, інтрамережі і для електронної комерції;

- SQL Server 7.0 підтримує лінгвістичний пошук, дозволяючи створювати спеціальні індекси ключових слів або фраз для обраних стовпців або таблиць;

- SQL Server пропонує широкий спектр можливостей реплікації, що забезпечують автоматичну синхронізацію змін, в тому числі і вироблених в автономному режимі, SQL Server 7.0 підтримує програми, що використовують технологію активних серверних сторінок (Active Server Pages - ASP);

- Access 2000 може безпосередньо звертатися до SQL Server, дозволяючи організувати прозору взаємодію клієнт-сервер;

- електронні таблиці, діаграми і зведені таблиці можуть бути безпосередньо пов'язані з SQL Server або службами, що надає користувачам можливості перегляду та аналізу даних за допомогою оглядача.

#### 2.3.4 Java

Мова програмування Java спеціально створювалася для розробки мережеских додатків і володіє всіма засобами для реалізації динамічного обміну даними між компонентами розподіленого додатка. Тому Java найкращим чином підходить для розробки програмного забезпечення для дистанційної діагностики, моніторингу та управління [16].

Для роботи з програмними рішеннями на мові Java, що здійснюють веб-базований доступ до технологічних даних, на комп'ютері клієнта зі

спеціального програмного забезпечення повинен бути встановлений лише стандартний веб-браузер [17]. Крім того, Java-базовані рішення мають наступні переваги:

- об'єктно-орієнтованість: в Java все є об'єктом, додаток може бути легко розширено, тому що він заснований на об'єктній моделі;

- платформонезалежність: на відміну від багатьох інших мов, включаючи C і C++, Java, коли був створений, він не компілювався в платформі конкретної машини, а в незалежному від платформи байт-кодi, цей байт код поширюється через інтернет і інтерпретується в Java Virtual Machine (JVM), на якій він в даний час працює;

- простота: процеси вивчення та введення в мову програмування Java залишаються простими, якщо є розуміння основних концепцій об'єктно-орієнтованого програмування, то Java буде простою в освоєнні;

- безпечність: методи перевірки автентичності засновані на шифруванні з відкритим ключем [18];

- архітектурна нейтральність: компілятор генерує архітектурно-нейтральні об'єкти формату файлу, що робить скомпільований код виконуваним на багатьох процесорах, з наявністю системи Java Runtime;

- портативність: не має залежності від реалізації аспектів специфікацій робить Java портативною, компілятор в Java написаний на ANSI C з чистою переносимістю, який є підмножиною POSIX [19];

- багатопоточність: можна писати програми, які можуть виконувати безліч завдань одночасно, введення в мову Java цієї конструктивної особливості дозволяє розробникам створювати налагоджені інтерактивні додатки [20];

- висока продуктивність: введення Just-In-Time компілятора, дозволило отримати високу продуктивність;

- поширеність: призначений для розподіленої середовища інтернету;

- динамічність: програмування на Java вважається більш динамічним, ніж на C або C++, так як він призначений для адаптації до мінливих умов, програми можуть виконувати велику кількість процесів обробки інформації, яка може бути

використана для перевірки і дозволу доступу до об'єктів на час виконання [21].

### 2.3.5 .Net Framework

Microsoft .NET – програмна технологія, запропонована фірмою Microsoft як платформа для створення як звичайних програм, так і веб-додатків. Багато в чому є продовженням ідей та принципів, покладених в технологію Java. Однією з ідей .NET є сумісність служб, написаних різними мовами [22]. Хоча ця можливість рекламується Microsoft як перевага .NET, платформа Java має таку саму можливість.

Кожна бібліотека (збірка) в .NET має свідчення про свою версію, що дозволяє усунути можливі конфлікти між різними версіями збірок.

.NET – крос-платформова технологія, в цей час існує реалізація для платформи Microsoft Windows, FreeBSD (від Microsoft) і варіант технології для ОС Linux в проекті Mono (в рамках угоди між Microsoft з Novell), DotGNU.

Захист авторських прав відноситься до створення середовищ виконання для програм .NET. Компілятори для .NET випускаються багатьма фірмами для різних мов вільно [23].

.NET поділяється на дві основні частини – середовище виконання (по суті віртуальна машина) та інструментарій розробки.

Середовища розробки .NET-програм: Visual Studio .NET (C++, C#, J#), SharpDevelop, Borland Developer Studio (Delphi, C#) тощо. Середовище Eclipse має додаток для розробки .NET-програм. Застосовні програми також можна розроблювати в текстовому редакторі та використовувати консольний компілятор.

Як і технологія Java, середовище розробки .NET створює байт-код, призначений для виконання віртуальною машиною [24]. Вхідна мова цієї машини в .NET називається CIL (Common Intermediate Language), також відома як MSIL (Microsoft Intermediate Language), або просто IL. Застосування байт-коду дозволяє отримати крос-платформовість на рівні скомпільованого проекту (в термінах .NET: збірка), а не на рівні сирцевого тексту, як, наприклад, в C. Перед запуском збірки в середовищі виконання (CLR) байт-код перетворюється вбудованим в

середовище JIT-компілятором (just in time, компіляція на льоту) в машинні коди цільового процесора [25].

Слід зазначити, що один з перших JIT-компіляторів для Java був також розроблений фірмою Microsoft (тепер в Java використовується досконаліша багаторівнева компіляція – Sun HotSpot). Сучасна технологія динамічної компіляції дозволяє досягнути аналогічного рівня швидкодії з традиційними «статичними» компіляторами (наприклад, C++) і питання швидкодії часто залежить від якості того чи іншого компілятора [26].

### 2.3.6 Spring Framework

Spring Framework – універсальний фреймворк з відкритим вихідним кодом для Java-платформи.

Spring Framework забезпечує комплексну модель розробки і конфігурації для сучасних бізнес-додатків на Java – на будь-яких платформах. Ключовий елемент Spring – підтримка інфраструктури на рівні програми: основна увага приділяється «водопроводу» бізнес-додатків, тому розробники можуть зосередитися на бізнес-логіці без зайвих налаштувань в залежності від середовища виконання.

Spring Framework може бути розглянутий як колекція менших фреймворків або фреймворків у фреймворку, як показано на рис. 2.13.

Кожен з компонентів Spring Framework має свою спеціалізацію та може бути використаним у багатьох поширених випадках. Більшість компонентів не залежні один від одного у архітектурному плані. Також Spring Framework має досить гарну інтеграцію з багатьма іншими сторонніми системами [27].

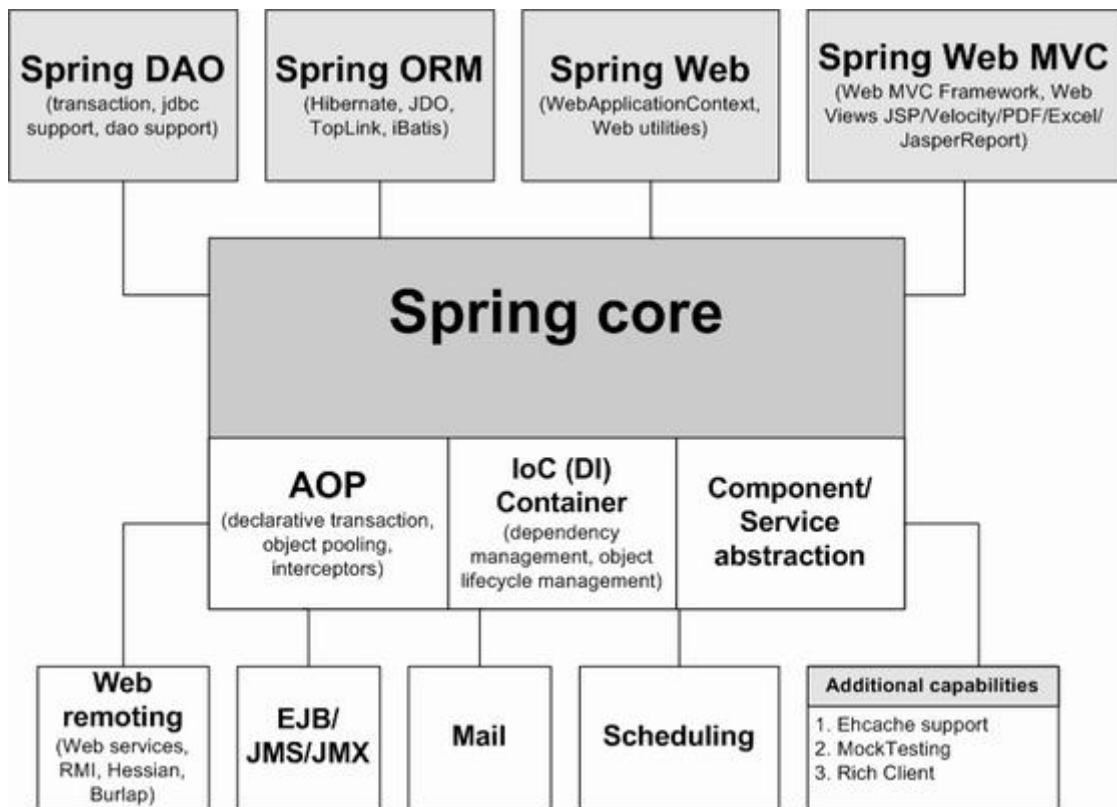


Рисунок 2.13 – Схема архітектури Spring

Більшість цих фреймворків може працювати незалежно один від одного, проте вони забезпечують більшу функціональність при спільному їх використанні. Ці фреймворки діляться на структурні елементи типових комплексних програм:

- inversion of Control-контейнер: конфігурація компонентів додатків і управління життєвим циклом Java-об'єктів;

- фреймворк аспектно-орієнтованого програмування: працює з функціональністю, яка не може бути реалізована можливостями об'єктно-орієнтованого програмування на Java без втрат [28];

- фреймворк доступу до даних: працює з системами керування базами даних на Java-платформі, використовуючи JDBC і ORM-засоби і забезпечуючи вирішення завдань, які повторюються в великому числі Java-based environments;

- фреймворк управління транзакціями: координація різних API керування транзакціями і інструментарій настроюваного управління транзакціями для об'єктів Java;

- фреймворк MVC: каркас, заснований на HTTP і сервлетах, що надає безліч можливостей для розширення та налаштування (customization);
- фреймворк віддаленого доступу: конфігурується передача Java-об'єктів через мережу в стилі RPC, підтримуюча RMI, CORBA, HTTP-based протоколи, включаючи web-сервіси (SOAP) [29];
- фреймворк аутентифікації і авторизації: конфігурується інструментарій процесів аутентифікації і авторизації, що підтримує багато популярних і стали індустріальними стандартами протоколів, інструментів, практик через дочірній проект Spring Security (раніше відомий як Aсегі);
- фреймворк віддаленого управління: конфігурується уявлення і управління Java-об'єктами для локальної або віддаленої конфігурації за допомогою JMX;
- фреймворк роботи з повідомленнями: конфігурується реєстрація об'єктів слухачів повідомлень для прозорої обробки повідомлень з черги повідомлень за допомогою JMS, поліпшена відправлення повідомлень за стандартом JMS API;
- тестування: каркас, що підтримує класи для написання модульних і інтеграційних тестів.

### 2.3.7 Hibernate

Hibernate – це ORM фреймворк для Java з відкритим вихідним кодом [30]. Ця технологія є вкрай потужною і має високі показники продуктивності.

Hibernate створює зв'язок між таблицями в базі даних і Java-класами і навпаки. Це позбавляє розробників від величезної кількості зайвої, рутинної роботи, в якій вкрай легко припуститися помилки і вкрай важко потім її знайти [31].

Переваги Hibernate:

- забезпечує простий API для запису та отримання Java-об'єктів в / з БД;
- мінімізує доступ до БД, використовуючи стратегії fetching;
- не вимагає сервера додатку;
- дозволяє не працювати з типами даних мови SQL, а мати справу зі звичними нам типами даних Java;
- піклується про створення зв'язків між Java-класами і таблицями БД за

допомогою XML-файлів не вносячи зміни в програмний код;

- якщо необхідно змінити БД, то достатньо лише внести зміни в XML-файли.

Hibernate підтримує всі основні СУБД: MySQL, Oracle, PostgreSQL, Microsoft SQL Server Database, HSQL, DB2. Hibernate також може працювати в зв'язці з такими технологіями, як Maven і J2EE [32].

### 2.3.8 Entity Framework

Платформа Entity Framework являє собою набір технологій ADO.NET, що забезпечують розробку додатків, пов'язаних з обробкою даних. Архітекторам і розробникам додатків, орієнтованих на обробку даних, доводиться враховувати необхідність досягнення двох абсолютно різних цілей. Вони повинні моделювати суті, зв'язку і логіку розв'язуваних бізнес-задач, а також працювати з ядрами СУБД, використовуваними для збереження і отримання даних. Дані можуть розподілятися за кількома системами зберігання даних, в кожній з яких застосовуються свої протоколи, але навіть в додатках, що працюють з однією системою зберігання даних, необхідно підтримувати баланс між вимогами системи зберігання даних і вимог написання ефективного і зручного для обслуговування коду програми [33].

У Entity Framework розробники отримують можливість працювати з даними, представленими у формі пов'язаних з окремими доменами об'єктів і властивостей, таких як клієнти і їх адреси, не будучи вимушеними звертатися до базових таблиць і стовпцями бази даних, де зберігаються ці дані. Entity Framework дає розробникам можливість працювати з даними на більш високому рівні абстракції, створювати і супроводжувати додатки, орієнтовані на роботу з даними, одночасно з цим скорочуючи обсяг коду, в порівнянні з традиційними програмами [34]. Так як Entity Framework входить до складу .NET Framework, Entity Framework додатки можуть виконуватися на будь-якому комп'ютері, на якому встановлена платформа .NET Framework, починаючи з версії 3.5 з пакетом оновлень 1.

Багаторічним і загальним підходом до розробки є підхід, при якому



побудова додатки або служби є його поділ на три частини: модель домену, логічну модель і фізичну модель. Модель домену визначає сутність і зв'язку в моделюється системі. Логічна модель для реляційної бази даних забезпечує нормалізацію сутностей і зв'язків з метою створення таблиць з обмеженнями зовнішнього ключа. У фізичній моделі враховуються можливості конкретної системи обробки даних шляхом визначення залежних від ядра бази даних докладних відомостей про зберігання даних, які стосуються секціонування і індексування.

Фізична модель вдосконалюється адміністраторами бази даних з метою підвищення продуктивності, але програмісти, які розробляють код додатка, в основному змушені обмежуватися роботою з логічною моделлю, готуючи SQL запити і викликаючи збережені процедури. Моделі домену в основному використовуються як інструмент для подання та обміну думками про вимоги до додатка [35], тому найчастіше служать в якості практично не змінюються схем, які розглядаються і обговорюються на ранніх стадіях проекту, після чого виходять зі сфери уваги.

## **2.4 Обґрунтування вибору методів та технологій для оптимального вирішення бізнес-задач**

Для вирішення бізнес-задач були обрані технології Java, PostgreSQL, Spring Framework та Hibernate. Такий стек технологій було обрано через те, що результатом реалізації алгоритму повинен бути веб-додаток.

Для реалізації алгоритму буде використана мова програмування Java через її платформонезалежність та об'єктно-орієнтованість, вона підходить для розробки мережевих додатків і володіє всіма засобами для реалізації динамічного обміну даними між компонентами розподіленого додатка.

Для збереження інформації про спортсменів, категорії змагань та створені протоколи змагань буде використана СУБД PostgreSQL, тому що вона має потужні і надійні механізми транзакцій і реплікації, а також легку розширюваність.

В якості ORM буде використана Hibernate, тому що вона підтримує необхідну СУБД та проста у використанні.

Для підтримки інфраструктури на рівні програми буде використано Spring Framework, тому що він забезпечує комплексну модель розробки і конфігурації для сучасних бізнес-додатків та дозволить зосередитися на реалізації бізнес-логіки алгоритму розподілу спортсменів.

## 3 ОПТИМІЗАЦІЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ ПРОТОКОЛЮВАННЯ

### 3.1 Опис виконаної роботи

Було виконано аналіз поточного алгоритму розподілу спортсменів (жеребкування).

Метою жеребкування є формування турнірної таблиці. Особливе значення це має коли в змаганнях беруть участь велика кількість спортсменів з різних команд. У цих випадках необхідно розсіяти гравців по різних групах таким чином, щоб всі групи мали приблизно однаковий сумарний рейтинг, тобто були рівними під силу, і при цьому в кожному з них входили спортсмени з різних команд. Ситуація значно ускладнюється, коли від однієї команди заявки на участь в турнірі подані кількома його представниками. Таких учасників необхідно розподілити в різні відбіркові групи.

Таким чином, задача стає двухкритеріальною: перший критерій – сумарний рейтинг груп (він повинен бути по можливості однаковим), другий критерій – розподіл учасників турніру з однієї команди за групами (воно повинно бути рівномірним). Вирішення цього завдання вже вимагає специфічних алгоритмів. Така ситуація є типовою, зокрема, для настільного турніру Харківської обласної федерації контактних єдиноборств.

Формально задача жеребкування приводиться до відомої математичної задачі розбиття графа на підграфи [36]. Кожен підграф відповідає одній з груп турнірної таблиці. Ребра графа – це відносини між спортсменами: наявність ребра означає, що відповідні спортсмени – представники однієї команди, як показано на рис. 3.1.

Такий граф симетричний, при цьому число команд дорівнює числу компонент графа [37]. Також кожна компонента є клікою графа, тобто повнозв'язним підграфом. При цьому виконується умова транзитивності: якщо  $A$  і  $B$  – спортсмени з однієї команди,  $A$  і  $C$  – спортсмени з однієї команди, то  $B$  і  $C$  також є спортсменами з однієї команди. Математична модель задачі визначається матрицею суміжності графа  $D = \|\|d_{ij}\|\|$  і вектором  $P = \{p_1, p_2, \dots, p_n\}$ , де  $p_i$  - рейтинг

спортсмена, а  $d_{ij} = 1$ , якщо спортсмени з номером  $i$  та  $j$  з однієї команди, інакше  $d_{ij} = 0$  [38]. Мета даного завдання розбиття – отримання таких непересічних підграфів (коли одна вершина належить тільки одному підграфу), в яких число внутрішніх зв'язків між вершинами було б найменшим або, відповідно, число зовнішніх зв'язків найбільшим [39].

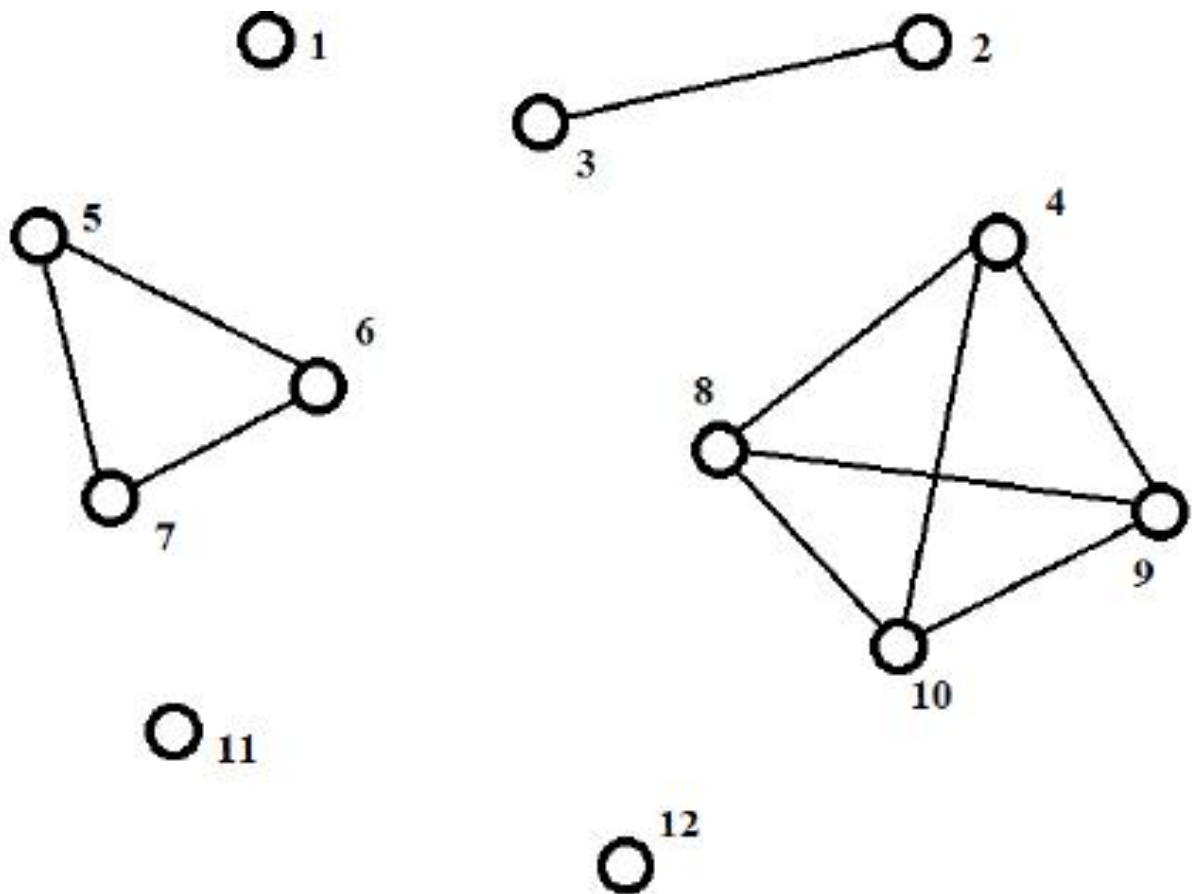


Рисунок 3.1 – Приклад графової моделі задачі турнірного жеребкування

У практиці проведення турнірів склалися певні підходи до проведення жеребкування. Як і в більшості видів спорту, при великому числі учасників спочатку проводиться відбірковий турнір. Для цього учасники розсіюються в декілька відбіркових груп. Іноді цей процес називають посівом. Посів може здійснюватися різними способами: випадково, зигзагом, змійкою.

Традиційні способи жеребкування. Найпоширенішим способом є посів змійкою. Формально він здійснюється наступним чином. Спочатку всі спортсмени упорядковуються по рейтингу. Потім, починаючи зі спортсмена з

найвищим рейтингом, послідовно у напрямку зниження рейтингу спортсменів їх атрибутами (даними про спортсменів) заповнюються перші  $m$  позицій, починаючи з 1-го по  $m$ -ї групи. Наступними за рейтингом  $m$  учасниками продовжують заповнення другої позиції у всіх групах, починаючи з  $m$ -ї групи і закінчуючи 1-ю групою. Черговими  $m$  спортсменами продовжують заповнення третьої позицій у всіх групах, знову починаючи з 1-ї групи і закінчуючи  $m$ -ю групою. Такий посів змійкою продовжують до тих пір, поки не заповняться всі позиції у всіх групах.

Формально, використовуючи поняття інверсних (зворотних) чисел [40], неважко показати, що при пониженні рейтингів учасників від максимального до мінімального на постійну величину сумарні рейтинги всіх груп будуть однаковими. Для простоти будемо вважати, що  $\epsilon$  деяка обмежена спадна послідовність натуральних чисел  $R = \{r_1, r_2, \dots, r_k\}$ . Для будь-якого числа  $r \in R$  цієї послідовності існує інверсне число  $r$ , визначається таким чином:  $r = r_1 + r_k - r$ .

Відповідно до формули будь-яку непарну і парну пару рядків турнірної таблиці з записаними в її осередки рейтингами в загальному вигляді можна представити в розгорнутому вигляді подвоєною довжини, як показано на рис. 3.2. В даному випадку величина  $m$  – число груп.

$r$	$r-\Delta$	$r-2\Delta$	$r-3\Delta$	...	$r-(2m-2)\Delta$	$r-(2m-1)\Delta$
-----	------------	-------------	-------------	-----	------------------	------------------

Рисунок 3.2 – Розгортка двох сусідніх рядків при посіві змійкою

Складаючи попарно кожне пряме і інверсне число, будемо отримувати одну і ту ж величину [41]. Аналогічним чином кожна наступна пара рядків у турнірній таблиці при складанні прямого і відповідного інверсного числа також буде давати одну і ту ж величину. У підсумку це призведе до того, що сума всіх чисел (рейтингів) у всіх шпальтах буде одна і та ж.

Найбільша застосовність посіву змійкою пояснюється тим, що, використовуючи цю евристику, вдається формувати приблизно однакові за силою групи. Це також забезпечує зниження ймовірності раннього вибування з турніру

сильних учасників і подальшого просування до фіналу слабких.

Нетрадиційні способи жеребкування. Традиційні способи жеребкування не виключають застосування інших способів. При малому числі спортсменів можна використовувати метод, заснований на повному переборі варіантів.

Такий спосіб є математично точним, оскільки з його допомогою розглядаються всі можливі варіанти, з яких можна вибрати ті, що задовольняють вибраним критеріям якості сформованих груп. Число таких розподілів представлятиме величину:

$$Q(n, m) = n!/m! \prod_{i=1}^m (k_i)!, \quad (3.1)$$

де  $n$  - число всіх спортсменів;

$m$  - кількість груп;

$k_i$  - кількість спортсменів в  $i$ -й групі.

Наприклад, відповідно до формули, при розподілі 6 спортсменів по 3 в 2 групи отримуємо  $Q(6, 2)=10$ , а по 2 учасники в 3 групи –  $Q(6, 3)=15$ . При такому числі спортсменів число можливих варіантів їх розподілу за групами представляє малу величину і завдання можна вирішити точно повним перебором [42]. Однак при розподілі реального числа спортсменів, наприклад 16 за 4 групами, число варіантів різко збільшується і становить величину  $Q(16, 4)=2627625$ .

Для такого числа варіантів повний перебір ще є можливим. Але для іншого, також реального випадку, коли 24 спортсмени розподіляються в 4 групи по 6 спортсменів у кожній, отримуємо дуже велику кількість варіантів –  $Q(24, 4) > 9,6 \cdot 10^{10}$ . Здійснити повний перебір такого числа варіантів навіть на суперсучасному комп'ютері неможливо. Ситуацію не рятують і паралельні обчислення як з економічних, так і з технічних причин, наштовхуючись на обмеження відповідно до закону Амдала [43]. Звідси випливає, що тільки в разі, якщо число учасників турніру не перевищує 16, завдання можна вирішувати

повним перебором.

Однак, як зазначалося вище, в загальному випадку задача жеребкування є двухкритеріальною. Якщо перший критерій – сумарний рейтинг спортсменів групи – легко підраховується, то другий – рівномірність розподілу спортсменів, які представляють одну команду, вимагає деяких роздумів. Логічно припустити, що одним з найгірших випадків буде такий, коли в групі потрапляють спортсмени з однієї команди, а найкращим – коли спортсмени з однієї команди розосереджені по різних групах.

Виходячи з цих крайніх випадків, цей критерій (назвемо його критерієм рівномірності,  $K_p$ ) пропонується розраховувати наступним чином: числа однакових команд в групі зводяться в квадрат і підсумовуються. Отримані таким же чином результати у всіх групах також підсумовуються, і загальний результат всіх підсумовувань ділиться на число груп. Таким чином, формула підрахунку критерію рівномірності матиме вигляд:

$$A = \sum_i \sum_{j=1}^m \frac{(a_{ij})^2}{m}, \quad (3.2)$$

де  $a_{ij}$  – число  $i$ -ї команди в групі  $j$ .

Число  $a_{ij}$  зводиться в квадрат, оскільки тільки в цьому випадку підсумовані різні команди в групі дають найменшу величину, рівну  $n$ . Якщо ж однієї і тієї ж команди в групі буде більше однієї, зведення таких чисел в квадрат і їх підсумовування призводять до перевищення цієї величини [44].

Так, наприклад, якщо в кожен з  $m$  груп входять  $n$  спортсменів з однієї команди (найгірший випадок), то  $A_1 = mn^2$ . Якщо ж в кожен групу входять спортсмени з різних команд (кращий випадок), то  $A_2 = mn < A_1$ .

Таким чином, той варіант турнірної таблиці, для якого значення  $A$  приймає найменше значення, буде найкращим. Приклад з найгіршим випадком критерію

одномірності наведений в таблиці 3.1. Приклад з кращим випадком критерію одномірності наведений в таблиці 3.2.

Таблиця 3.1 - Найгірший випадок критерію одномірності

Гр. 1	Гр. 2	Гр. 3
1	2	3
1	2	3
1	2	3
9	9	9

Таблиця 3.2 - Кращий випадок критерію одномірності

Гр. 1	Гр. 2	Гр. 3
1	1	1
2	2	2
3	3	3
3	3	3

Цифри в рядках 1-3 цих таблиць означають номери команд, цифри в їх останніх рядках визначають критерій рівномірності розподілу спортсменів в групах. Для кожної групи вони виходять як сума квадратів числа однакових команд. Так, в таблиці 1 в кожній групі всі три спортсмени представляють одну команду, тому рівномірність в усіх трьох групах дорівнює 3.

У таблиці 2 кожну групу представляють спортсмени з різних команд, тому рівномірність кожної групи дорівнює 3. Очевидно, що турнірна таблиця 1 з точки зору рівномірності розподілу спортсменів представляє найгірший випадок, а таблиця 2 – найкращий. Зрозуміло й те, що про рівномірності потрібно судити не по одній групі, а по всьому, тобто використовувати усереднене значення по всіх групах.

Для повноти аналізу критерію рівномірності в таблиці 3 наведемо деякі приклади можливих розподілів в турнірній таблиці спортсменів з різних команд.



Візуальний аналіз не суперечить наведеним підрахунками: при критерії рівномірності  $K_p = 11,5$  в двох групах спортсмени тільки з однієї команди, а при критерії рівномірності  $K_p = 6,5$  спортсмени по групах розподілені в таблиці істотно рівномірніше. Показники турнірної таблиці при умовах рівномірності  $K_p = 7,5$  і  $K_p = 9$  займають проміжне положення.

Побудова компромісною адитивною цільовою функції. Для вирішення двухкритеріального завдання формування турнірних таблиць повним перебором (за умови невеликого числа спортсменів) потрібно мати компромісну адитивну цільову функцію (ЦФ), що містить обидва ці критерії [45]. До компромісних цільових функцій пред'являється ряд вимог. Вхідні в цільову функцію параметри мають, як правило, різні розмірності, тому їх необхідно привести до безрозмірним величинам. У адитивну цільову функцію кожен параметр повинен входити з відповідним ваговим коефіцієнтом важливості, визначаючим компроміс ЦФ. Поліпшуючі ЦФ параметри повинні входити в неї зі знаком плюс, що погіршують – зі знаком мінус. Турнірні таблиці з різними критеріями рівномірності представлені у таблиці 3.3.

Безрозмірне значення параметра  $r_k$  забезпечується введенням нормуючої величини  $r_{кн}$ , а ступінь компромісу призначається за допомогою коефіцієнтів  $\alpha_k \leq 1$ . Значення нормуючої величини  $r_{кн}$  можна призначити різними способами. Кращим є спосіб, якщо нормуючу величину можна визначити апріорно як найкраще її значення з усіх можливих.

Якщо параметр  $r_k$  входить в ЦФ як зменшуючий її значення, в якості нормуючої величини цього параметра потрібно вибирати теоретично мінімальне його значення. Якщо ж параметр  $r_k$  входить в ЦФ як збільшуючий її значення, в якості нормуючої величини цього параметра потрібно вибирати теоретично максимальне його значення.

Пропонується наступний вигляд компромісної ЦФ:

$$F = \alpha_1 \frac{K_{p.\text{тек.}}}{K_{p.\text{мін.}}} + \alpha_2 \frac{R_{\text{max}} - R_{\text{мін}}}{0,1\bar{R}} \rightarrow \min \quad (3.3)$$

де  $K_{p.\text{тек.}}$  – поточний критерій рівномірності;

$K_{p.\text{мін.}}$  – мінімальний номінуючий критерій рівномірності, який визначається апіорно;

$R_{\text{max}}$  – найбільший сумарний рейтинг груп;

$R_{\text{мін}}$  – найменший сумарний рейтинг груп;

$\bar{R}$  – середнє значення сумарного рейтингу груп;

$\alpha_1$  і  $\alpha_2$  – важливість критеріїв, що встановлюються експертами.

Наведену цільову функцію можна використовувати при побудові оптимальної турнірної таблиці методом повного перебору варіантів двома способами. При першому (стандартному) способі після формування чергового варіанту турнірної таблиці за формулою підраховується цільова функція і її значення порівнюється з попереднім значенням цільової функції, що зберігаються в пам'яті.

Якщо чергове значення цільової функції менше попереднього, то її значення і відповідний їй варіант турнірної таблиці заміщають в пам'яті попередній варіант турнірної таблиці. Після перебору всіх варіантів турнірної таблиці в пам'яті буде збережено кращий її варіант.

При другому способі один зі складових критеріїв компромісної цільової функції потрібно зробити обмеженням, тобто коефіцієнт ваги при цьому критерії прирівнюємо до нуля. Тоді цільова функція стане однокритерійною і при переборі варіантів турнірної таблиці оптимізація буде здійснюватися тільки за іншим критерієм, але за умови, що значення першого критерію не виходитиме за встановлені межі [46].

Для доведення придатності побудованої цільової функції до оптимізації турнірних таблиць розглянемо два їх гіпотетичних варіанти. Нехай число гравців  $N = 16$  і представляють вони 4 різних команди, які в турнірних таблицях будемо

позначати цифрами 1, 2, 3 і 4 через кому після рейтингу спортсмену.

Розглянемо два крайніх випадку турнірної таблиці. Приклад найкращого гіпотетичного випадку наведено у таблиці 3.4. Гірший гіпотетичний випадок наведено у таблиці 3.5.

Таблиця 3.3 – Турнірні таблиці з різними критеріями рівномірності

Гр.1	Гр.2	Гр.3	Гр.4
$K_p = 6,5$			
1	3	4	2
2	1	4	3
3	4	3	4
3	2	4	4
6	4	10	6
$K_p = 7,5$			
1	1	2	3
2	4	4	3
3	4	3	3
4	4	4	2
4	10	6	10
$K_p = 9$			
1	4	4	2
2	4	3	2
3	4	3	3
4	4	3	1
4	16	10	6
$K_p = 11,5$			
3	4	3	2
3	4	4	2

3	4	4	1
3	4	2	1
16	16	6	8

Таблиця 3.4 – Кращий гіпотетичний випадок

Гр.1	Гр.2	Гр.3	Гр.4
40,1	36,1	33,1	30,1
23,1	24,1	27,1	29,1
20,1	18,2	17,2	14,2
8,3	10,3	12,4	13,4
91,10	88,6	89,6	86,6

Таблиця 3.5 – Гірший гіпотетичний випадок

Гр.1	Гр.2	Гр.3	Гр.4
40,1	23,1	18,2	8,3
36,1	24,1	17,2	10,3
33,1	27,1	14,2	12,4
30,1	29,1	20,1	13,4
139,16	103,16	69,1	43,8

В останніх рядках таблиць для кожної групи зазначений сумарний рейтинг, а через кому – критерій рівномірності. Вважаючи вагові коефіцієнти важливості обох критеріїв рівними  $\alpha_1 = \alpha_2 = 0,5$ , розрахуємо ЦФ  $F_1$  для таблиці кращого випадку і  $F_2$  для таблиці гіршого випадку. Для таблиці кращого випадку  $\Delta = 5$ , а мінімальний і розрахунковий (поточний) критерії рівномірності збігаються, тобто  $K_{p.тек.} = K_{p.мін.} = 28/4 = 7$ .

Для таблиці гіршого випадку  $\Delta = 96$ , а поточний критерій рівномірності буде дорівнювати  $K_{p.тек.} = (16 + 16 + 10 + 8) / 4 = 12,5$ .

Підставивши ці значення до формули компромісної ЦФ, отримаємо

значення  $F_1 = 0,78$ ,  $F_2 = 6,29$ . Бачимо, що для кращого варіанта турнірної таблиці ЦФ мінімальна, а для гіршого випадку максимальна. Таким чином, як і передбачалося, побудована двухкритеріальна ЦФ є правильною, тобто відповідає своєму призначенню.

Інший спосіб нетрадиційної жеребкування можна використовувати, спираючись на роботу [47], в якій розглянуто послідовний алгоритм розподілу питань дисципліни, що вивчається за тестовими завданнями. Число тестових завдань можна розглядати як число груп в нашому випадку, а число питань в тестовому завданні – як число спортсменів в групі.

Спочатку обчислюється середній сумарний рейтинг, загальний для всіх груп, за формулою:

$$R = \sum_{i=1}^n \frac{r_i}{m} \quad (3.4)$$

Після чого починається послідовне формування груп на основі такої евристики. Вважаючи, що в перші позиції кожної групи посіяні спортсмени з максимальними рейтингами, в чергову групу на  $i$ -му кроці розподіляється той спортсмен, рейтинг якого найменше відрізняється від величини, яка визначається за формулою:

$$r_i = \frac{(R - g_{i-1})m}{N - m(i - 1)} \quad (3.5)$$

де  $N$  – загальна кількість спортсменів;

$g_{i-1}$  – сума рейтингів спортсменів, вже розподілених в розглянуту групу.

Сума рейтингів спортсменів розраховується за формулою:

$$g_{i-1} = \sum_{r_j \in R_{i-1}} r_j, \quad (3.6)$$

де  $R_{i-1}$  – множина спортсменів з рейтингами, розподілених в групу до (i-1)-го кроку.

Таким чином, значення величини  $\tilde{r}_i$  визначає того спортсмена, рейтинг якого найбільш підходить для включення в сформовану на i-му кроці групу.

Одним з важливих питань, що вирішуються до проведення турніру, є питання про кількість відбіркових груп і числа учасників в групі. Це залежить від загального числа спортсменів. Оскільки в групах кожен учасник турніру змагається з кожним, число поєдинків в групі визначається як число поєднань з n по 2, тобто за формулою:

$$Q = C_n^2 = \frac{n!}{(n-2)!}, \quad (3.7)$$

де n – число спортсменів в групі.

Наприклад, при загальній кількості спортсменів  $N = 24$  найбільш типовою є турнірна таблиця, що складається з 6 груп по 4 спортсмена в кожній групі або з 4 груп по 6 спортсменів. У першому випадку необхідно провести всього 6 поєдинків в кожній групі, у другому – 15. Виходячи з наведеного вище концептуального опису можливих варіантів жеребкування, було прийнято рішення про доцільність побудови алгоритму і його програмної реалізації, спираючись на евристику змійки.

Нехай задано множину  $S = \{s_1, s_2, \dots, s_n\}$  спортсменів і число груп m. У всі групи входить однакове число спортсменів, кожен спортсмен ідентифікований рейтингом і командою, яку він представляє. Необхідно розробити алгоритм і відповідний програмний код, що розподіляє множину спортсменів S по m групам

таким чином, щоб сумарний рейтинг кожної групи мав мінімальне відхилення від середнього значення [48]. Крім того, якщо деякі команди представлені кількома спортсменами, то вони повинні бути розподілені по позиціях груп по можливості рівномірно. Під позицією розуміється місце в турнірній таблиці.

Першим етапом алгоритму жеребкування є сортування спортсменів по спадаючій їх рейтингу. Для забезпечення рівномірності розподілу декількох спортсменів з однієї і тієї ж команди на кожному кроці розподілу чергового спортсмену порівнюються команди спортсменів усіх вищих позицій з командою даного спортсмену. Якщо число команд в вищих позиціях перевищує певну величину, то даний спортсмен пропускає цю позицію, і для нього аналогічним чином шукаються наступні позиції.

Як показали попередні експерименти (ручного розподілу при малому числі спортсменів), ця стратегія призводить до більш рівномірного розподілу сумарного рейтингу за всіма групами [49]. Тому вона і була обрана для програмної реалізації алгоритму, який показано на рис. 3.3.

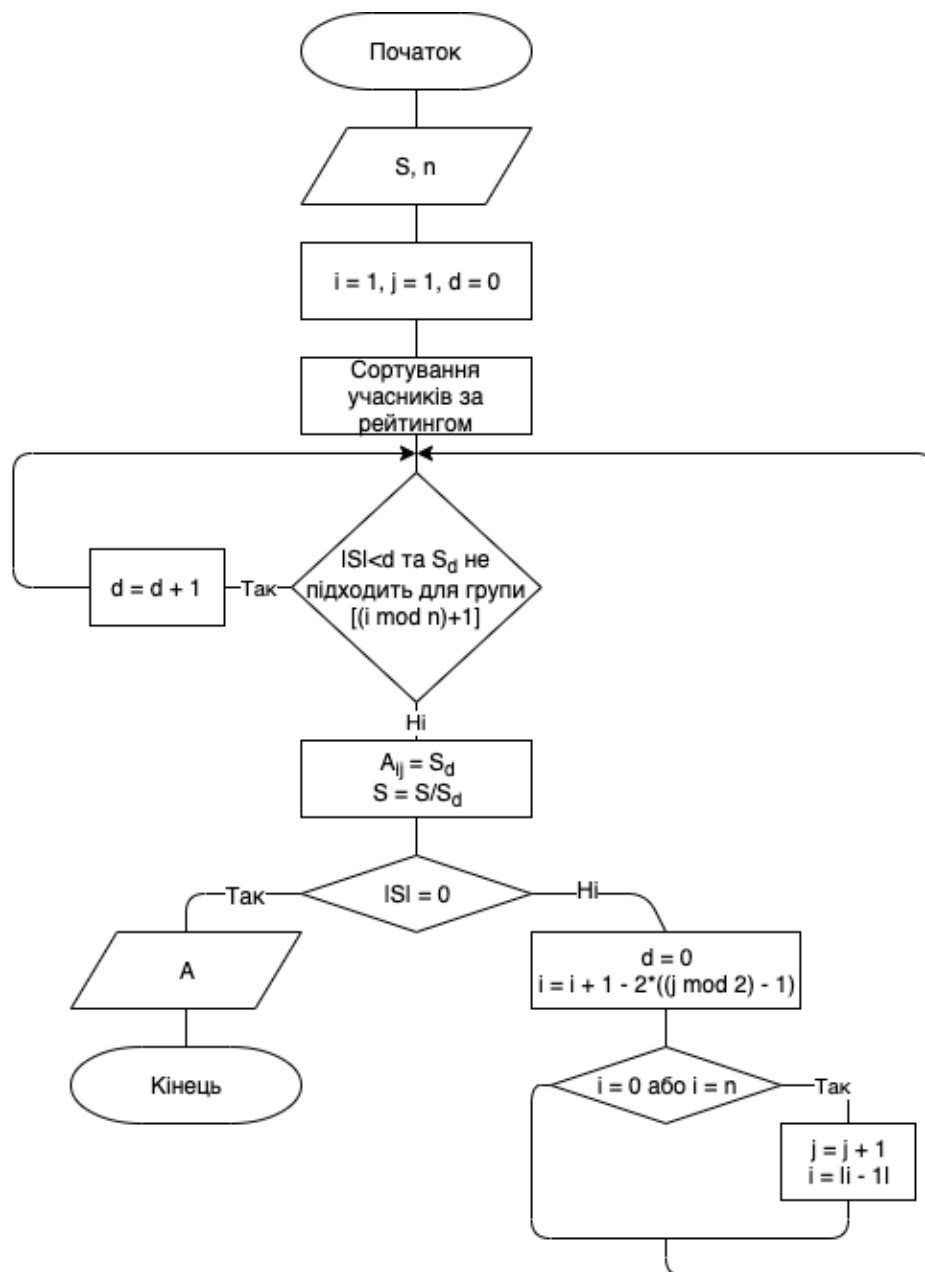


Рисунок 3.3 – Блок-схема алгоритму жеребкування

Для реалізації алгоритму буде використано структуру даних `HashMap<UUID, Integer>`. Це складна структура, що дозволяє забезпечити доступ до елементів по ключу. Ключем її є тип даних `UUID` – ідентифікатор команди. Головна властивість `HashMap` – швидкий пошук на основі ключів. Ключі перетворюються в хеш, асоціюючи хеш-код із значенням [50]. Це дозволяє додавати нові елементи без накладних витрат продуктивності програми, пов'язаних з необхідністю усунення наступних елементів в пам'яті. Аналіз наведеної блок-схеми дає наступну оцінку часової складності алгоритму (ЧСА):  $O(n) = n \log n$ , де  $n$  – число учасників змагання. Така оцінка ЧСА незначно



відрізняється від лінійної, а алгоритми з лінійної ЧСА є найбільш швидкодіючими [51].

### **3.2 Опис використання запропонованих методів та технологій для вирішення поставлених задач**

Для збереження інформації була спроектована модель бази даних, яка показана на рис. 3.4, що складається з 10 таблиць:

- `weight_category` – містить інформацію про вагові категорії учасників спортивних змагань;

- `age_category` – містить інформацію про вікові категорії учасників спортивних змагань;

- `competition_category` – містить інформацію про категорії спортивних змагань;

- `participant` – містить інформацію про учасників спортивних змагань;

- `competition_category_participant` – суміжна таблиця між учасниками та категоріями змагань;

- `team` – містить інформацію про команди;

- `coach` – містить інформацію про тренерів команд;

- `competition` – містить інформацію про конкретні спортивні змагання;

- `protocol` – містить інформацію про створені протоколи спортивних змагань;

- `protocol_participant` – містить інформацію про учасників протоколів спортивних змагань.

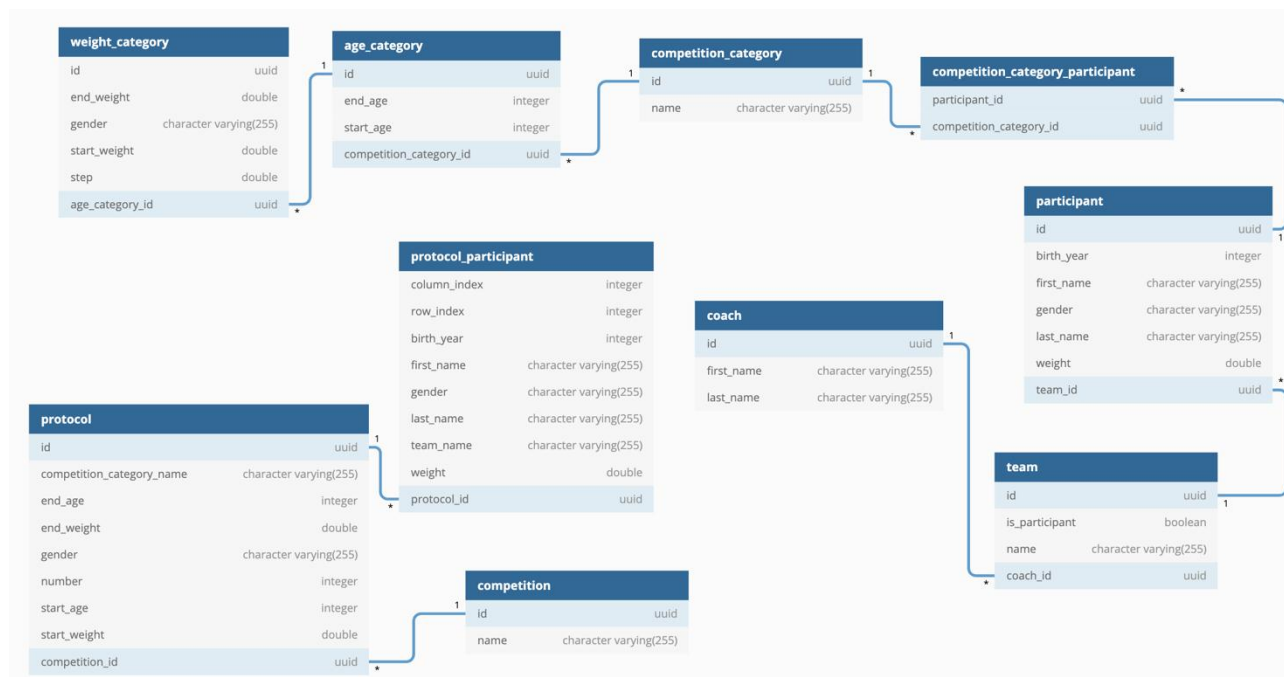


Рисунок 3.4 – Модель спроектованої бази даних

Відношення таблиць «age\_category» та «weight\_category» – один до багатьох, тому що одна вікова категорія може налічувати багато вагових категорій, але вагова категорія може належати тільки до одної вікової категорії [52].

Відношення таблиць «competition\_category» та «age\_category» – один до багатьох, тому що одна категорія змагань може налічувати багато вікових категорій, але вікова категорія може належати тільки до однієї категорії змагань.

Відношення таблиць «competition\_category» та «participant» – багато до багатьох, тому що один спортсмен може приймати участь у багатьох категоріях змагань та одна категорія змагань може налічувати багато спортсменів. Для вирішення зв'язку «багато до багатьох» була створена таблиця «competition\_category\_participant».

Відношення таблиць «team» та «participant» – один до багатьох, тому що одна команда може налічувати багато спортсменів, але спортсмен може належати тільки до однієї команди.

Відношення таблиць «coach» та «team» – один до багатьох, тому що один тренер може тренувати декілька команд, але команда може належати тільки до

одного тренера.

Відношення таблиць «protocol» та «protocol\_participant» – один до багатьох, тому що один протокол може налічувати декілька учасників, але учасник може належати тільки до одного протоколу.

Відношення таблиць «competition» та «protocol\_participant» – один до багатьох, тому що одні змагання можуть налічувати багато протоколів, але протокол може належати тільки до одного змагання.

Дуплікація даних у таблицях «participant» та «protocol\_participant» була створена для того, щоб була можливість змінювати інформацію про спортсменів без змінення інформації про цих спортсменів у вже існуючих протоколах спортивних змагань. При створенні протоколів у таблицю «protocol\_participant» буде записана інформація про спортсменів, що існують на час створення протоколів. Ця інформація не може бути змінена після того, як протоколи змагань були створені.

Для можливості керування даними було створено веб-додаток, що складається з чотирьох основних сторінок: «Панель управління», «Учасники», «Разделы» та «Соревнования».

Сторінка «Панель управління», як показано на рис. 3.5, містить наступні основні блоки:

- блок з інформацією про поточну кількість спортсменів, що приймають участь у змаганнях та про загальну кількість спортсменів;
- блок з інформацією про поточну кількість команд, що приймають участь у змаганнях та про загальну кількість команд;
- блок з інформацією про загальну кількість змагань;
- блок з інформацією про загальну кількість категорій спортивних змагань;
- блок з переліком існуючих команд з можливістю додавання команд, як показано на рис. 3.6, редагування, як показано на рис. 3.7, видалення, і пошуку;
- блок з переліком існуючих тренерів команд з можливістю додавання тренерів, редагування тренерів та видалення тренерів, а також пошуку по тренерам;

- блок з переліком існуючих категорій спортивних змагань з можливістю додавання, редагування та видалення категорій, а також пошуку по категоріям.

The screenshot displays the administrative interface for UFCF. At the top, the user is logged in as 'Администратор' (Administrator) and can click 'ВЫЙТИ' (Logout). The dashboard features four summary cards:

- Участников** (Participants): 12 (Всего спортсменов: 12)
- Команд Участвует** (Teams Participating): 3 (Всего команд: 3)
- Всего Соревнований** (Total Competitions): 2 (Обновлено только что)
- Всего Разделов** (Total Divisions): 4 (Обновлено только что)

Below the dashboard are three main sections:

- Команды** (Teams): A table with columns for ID, Name, Coach, Number of Athletes, Number of Participants, and Participation status.
 

№	НАЗВАНИЕ	ТРЕНЕР	КОЛИЧЕСТВО СПОРТСМЕНОВ	КОЛИЧЕСТВО УЧАСТНИКОВ	УЧАСТИЕ
1	Strong	Молчанов Вячеслав	3	3	✓
2	Атлант	Васин Александр	5	5	✓
3	Липцы	Васин Александр	4	4	✓
- Тренеры** (Trainers): A table with columns for ID, Family Name, and Teams.
 

№	ФАМИЛИЯ, ИМЯ	КОМАНДЫ
1	Васин Александр	Атлант, Липцы
2	Молчанов Вячеслав	Strong
- Разделы** (Divisions): A table with columns for ID, Name, and Number of Participants.
 

№	НАЗВАНИЕ	КОЛИЧЕСТВО УЧАСТНИКОВ
1	Mix-Fight	6
2	Лайт	5
3	Сабмишн	7
4	Семи	4

Each table includes a search bar and a 'ДОБАВИТЬ' (Add) button. The footer indicates '© 2019 Created by Andrii Partola'.

Рисунок 3.5 – Користувачський інтерфейс сторінки «Панель управления»

### Добавить команду ✕

Название

Тренер

Участие

ЗАКРЫТЬ
ДОБАВИТЬ

Рисунок 3.6 – Модальне вікно додавання інформації про команду

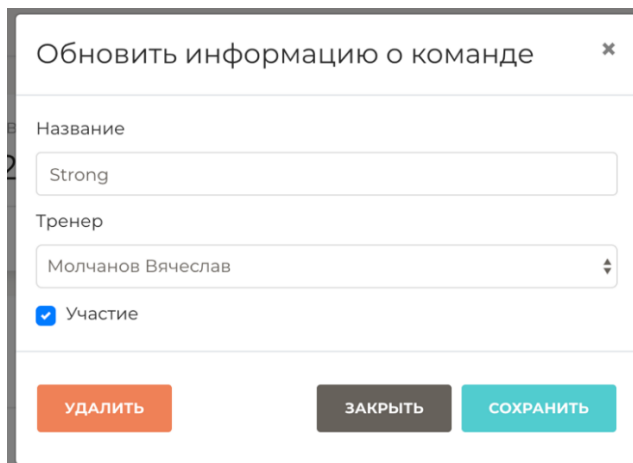
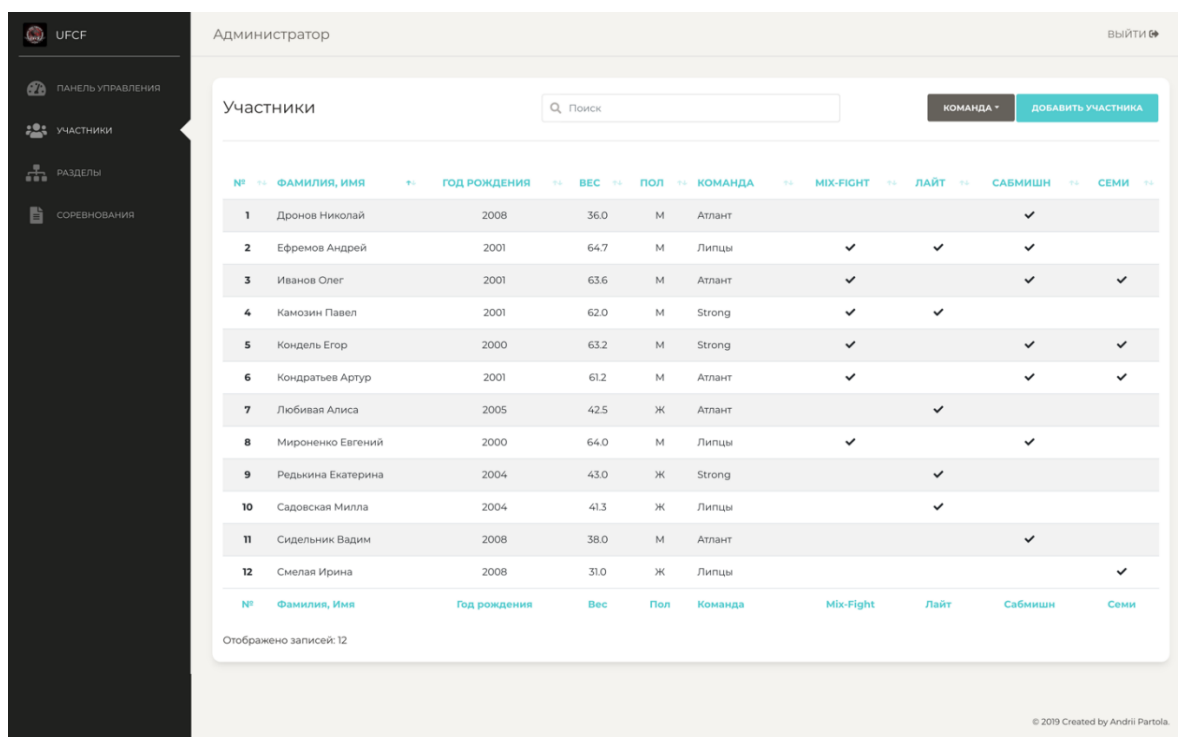


Рисунок 3.7 – Модальне вікно редагування інформації про команду

Сторінка «Участники» містить інформацію про спортсменів та категорії спортивних змагань, в яких вони приймають участь, як показано на рис. 3.8 з можливістю пошуку по учасникам, їх сортування за усіма параметрами та вибірки спортсменів, що належать до конкретної команди, а також додавання інформації про нових спортсменів, як показано на рис. 3.9, редагування інформації про існуючих спортсменів, як показано на рис. 3.10, та видалення інформації про спортсменів.



№	ФАМИЛИЯ, ИМЯ	ГОД РОЖДЕНИЯ	ВЕС	ПОЛ	КОМАНДА	МИХ-ФИГТ	ЛАЙТ	САБМИШН	СЕМИ
1	Дронов Николай	2008	36.0	М	Атлант			✓	
2	Ефремов Андрей	2001	64.7	М	Липцы	✓	✓	✓	
3	Иванов Олег	2001	63.6	М	Атлант	✓		✓	✓
4	Камозин Павел	2001	62.0	М	Strong	✓	✓		
5	Кондель Егор	2000	63.2	М	Strong	✓		✓	✓
6	Кондратьев Артур	2001	61.2	М	Атлант	✓		✓	✓
7	Любивая Алиса	2005	42.5	Ж	Атлант		✓		
8	Мироненко Евгений	2000	64.0	М	Липцы	✓		✓	
9	Редькина Екатерина	2004	43.0	Ж	Strong		✓		
10	Садовская Милла	2004	41.3	Ж	Липцы		✓		
11	Сидельник Вадим	2008	38.0	М	Атлант			✓	
12	Смелая Ирина	2008	31.0	Ж	Липцы				✓

Рисунок 3.8 – Користувацький інтерфейс сторінки «Учасники»

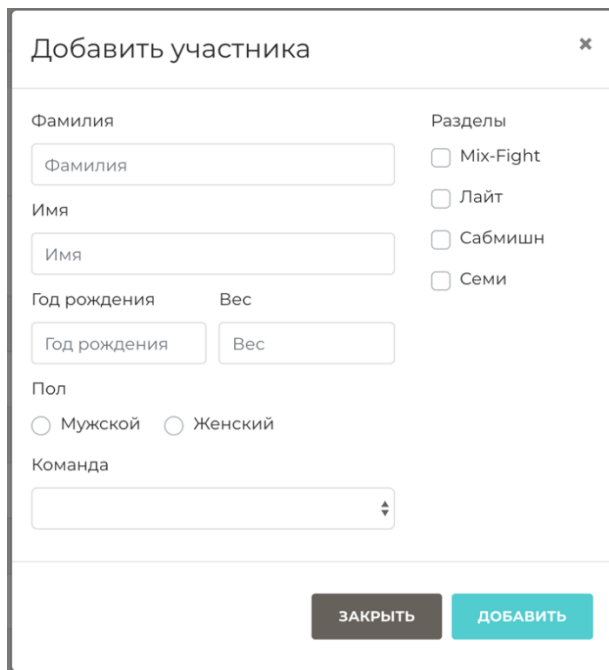


Рисунок 3.9 – Модальне вікно додавання інформації про спортсмена

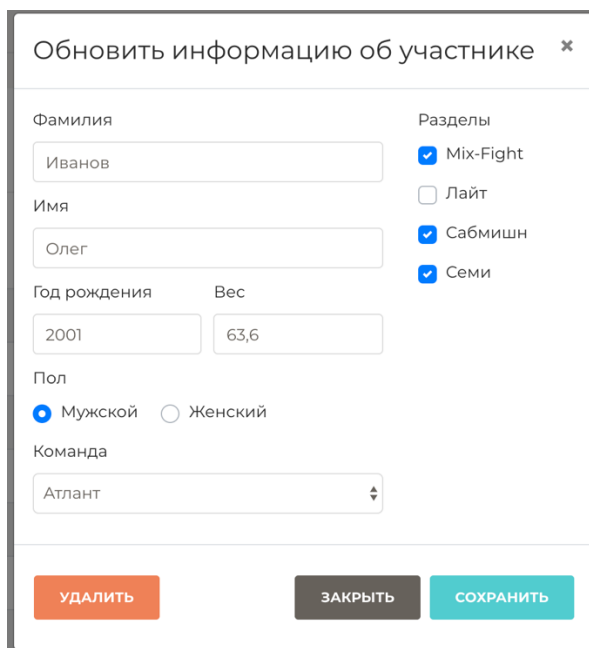


Рисунок 3.10 – Модальне вікно редагування інформації про спортсмена

Сторінка «Разделы» містить інформацію про вікові категорії, що належать до розділів, а також вагові категорії, що належать до вікових категорій, як показано на рис. 3.11, з можливістю додавання нових вікових категорій до розділів, редагування та видалення вікових категорій, додавання нових вагових

категорій, як показано на рис. 3.12, редагування, як показано на рис. 3.13, та видалення вагових категорій.

The screenshot displays the 'Разделы' (Divisions) page in the UFCF administrator interface. The interface is in Ukrainian and shows a list of weight categories for four different divisions: Mix-Fight, Лайт (Light), Сабмишн (Submission), and Семи (Seven). Each division has a table of age groups and weight categories, with a 'ДОБАВИТЬ' (Add) button for each row. The 'Mix-Fight' division has 8 rows, 'Лайт' has 8 rows, 'Сабмишн' has 8 rows, and 'Семи' has 8 rows. The 'Сабмишн' division is unique as it includes both male (М) and female (Ж) categories for each age group. The 'Лайт' and 'Семи' divisions also include both male and female categories. The 'Mix-Fight' division only shows male categories. The interface includes a sidebar with navigation options: ПАНЕЛЬ УПРАВЛЕНИЯ, УЧАСТНИКИ, РАЗДЕЛЫ, and СОРЕВНОВАНИЯ. The user is logged in as 'Администратор' and can click 'Выйти' (Logout).

ВОЗРАСТ	ПОЛ	ВЕСОВЫЕ КАТЕГОРИИ	ДОБАВИТЬ
8 - 9	М	-20, 20, 25, 30, 35, 40, 45, 45+	ДОБАВИТЬ
10 - 11	М	-20, 20, 25, 30, 35, 40, 45, 50, 50+	ДОБАВИТЬ
12 - 13	М	-25, 25, 30, 35, 40, 45, 50, 55, 60, 60+	ДОБАВИТЬ
14 - 15	М	-30, 30, 35, 40, 45, 50, 55, 60, 65, 70, 70+	ДОБАВИТЬ
16 - 17	М	-35, 35, 40, 45, 50, 55, 60, 65, 70, 75, 80, 80+	ДОБАВИТЬ
18 - 19	М	-45, 45, 50, 55, 60, 65, 70, 75, 80, 85, 90, 95, 95+	ДОБАВИТЬ
20 - 35	М	-55, 55, 60, 65, 70, 75, 80, 85, 90, 95, 100, 100+	ДОБАВИТЬ

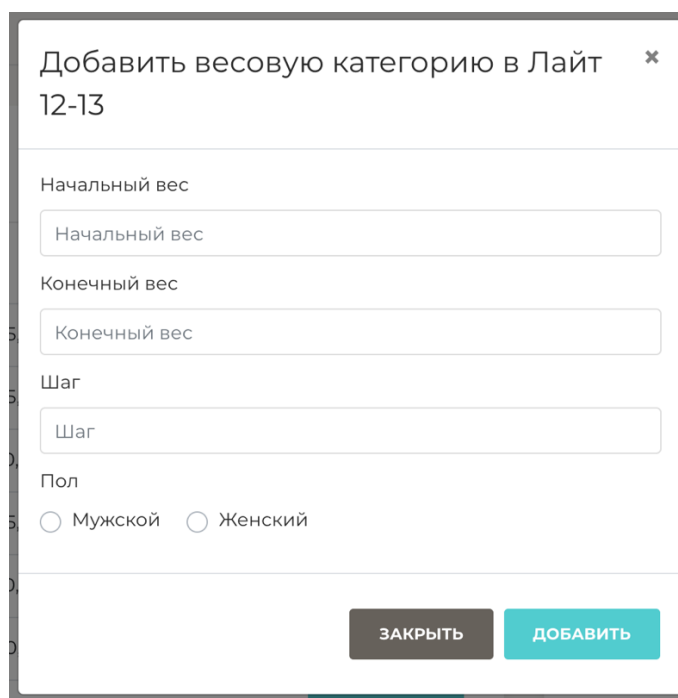
ВОЗРАСТ	ПОЛ	ВЕСОВЫЕ КАТЕГОРИИ	ДОБАВИТЬ
8 - 9	М	-20, 20, 25, 30, 35, 40, 45, 50, 50+	ДОБАВИТЬ
8 - 9	Ж	-20, 20, 25, 30, 35, 40, 45, 45+	ДОБАВИТЬ
10 - 11	М	-25, 25, 30, 35, 40, 45, 50, 55, 55+	ДОБАВИТЬ
10 - 11	Ж	-25, 25, 30, 35, 40, 45, 50, 50+	ДОБАВИТЬ
12 - 13	М	-30, 30, 35, 40, 45, 50, 55, 60, 65, 65+	ДОБАВИТЬ
12 - 13	Ж	-30, 30, 35, 40, 45, 50, 55, 55+	ДОБАВИТЬ
14 - 15	М	-35, 35, 40, 45, 50, 55, 60, 65, 70, 75, 75+	ДОБАВИТЬ
14 - 15	Ж	-35, 35, 40, 45, 50, 55, 60, 60+	ДОБАВИТЬ
16 - 17	М	-40, 40, 45, 50, 55, 60, 65, 70, 75, 80, 85, 90, 90+	ДОБАВИТЬ
16 - 17	Ж	-40, 40, 45, 50, 55, 60, 65, 70, 70+	ДОБАВИТЬ
18 - 35	М	-50, 50, 55, 60, 65, 70, 75, 80, 85, 90, 95, 100, 100+	ДОБАВИТЬ
18 - 35	Ж	-50, 50, 55, 60, 65, 70, 75, 80, 80+	ДОБАВИТЬ

ВОЗРАСТ	ПОЛ	ВЕСОВЫЕ КАТЕГОРИИ	ДОБАВИТЬ
8 - 9	М	-20, 20, 25, 30, 35, 40, 45, 45+	ДОБАВИТЬ
8 - 9	Ж	-20, 20, 25, 30, 35, 40, 45, 45+	ДОБАВИТЬ
10 - 11	М	-30, 30, 35, 40, 45, 50, 55, 55+	ДОБАВИТЬ
10 - 11	Ж	-25, 25, 30, 35, 40, 45, 50, 55, 55+	ДОБАВИТЬ
12 - 13	М	-35, 35, 40, 45, 50, 55, 60, 65, 65+	ДОБАВИТЬ
12 - 13	Ж	-30, 30, 35, 40, 45, 50, 55, 60, 65, 65+	ДОБАВИТЬ
14 - 15	М	-45, 45, 50, 55, 60, 65, 70, 75, 75+	ДОБАВИТЬ
14 - 15	Ж	-40, 40, 45, 50, 55, 60, 65, 70, 70+	ДОБАВИТЬ
16 - 17	М	-50, 50, 55, 60, 65, 70, 75, 80, 85, 85+	ДОБАВИТЬ
16 - 17	Ж	-45, 45, 50, 55, 60, 65, 70, 75, 75+	ДОБАВИТЬ
18 - 35	М	-55, 55, 60, 65, 70, 75, 80, 85, 90, 95, 100, 100+	ДОБАВИТЬ
18 - 35	Ж	-50, 50, 55, 60, 65, 70, 75, 80, 80+	ДОБАВИТЬ

ВОЗРАСТ	ПОЛ	ВЕСОВЫЕ КАТЕГОРИИ	ДОБАВИТЬ
8 - 9	М	-20, 20, 25, 30, 35, 40, 45, 45+	ДОБАВИТЬ
8 - 9	Ж	-20, 20, 25, 30, 35, 40, 40+	ДОБАВИТЬ
10 - 11	М	-25, 25, 30, 35, 40, 45, 50, 50+	ДОБАВИТЬ
10 - 11	Ж	-25, 25, 30, 35, 40, 45, 45+	ДОБАВИТЬ
12 - 13	М	-30, 30, 35, 40, 45, 50, 55, 60, 60+	ДОБАВИТЬ
12 - 13	Ж	-30, 30, 35, 40, 45, 50, 55, 55+	ДОБАВИТЬ
14 - 15	М	-35, 35, 40, 45, 50, 55, 60, 65, 70, 75, 75+	ДОБАВИТЬ
14 - 15	Ж	-35, 35, 40, 45, 50, 55, 60, 60+	ДОБАВИТЬ
16 - 17	М	-40, 40, 45, 50, 55, 60, 65, 70, 75, 80, 85, 90, 90+	ДОБАВИТЬ
16 - 17	Ж	-40, 40, 45, 50, 55, 60, 65, 70, 70+	ДОБАВИТЬ
18 - 35	М	-50, 50, 55, 60, 65, 70, 75, 80, 85, 90, 95, 100, 100+	ДОБАВИТЬ
18 - 35	Ж	-50, 50, 55, 60, 65, 70, 75, 80, 80+	ДОБАВИТЬ

© 2019 Created by Andrii Partola.

Рисунок 3.11 – Користувачський інтерфейс сторінки «Разделы»



Добавить весовую категорию в Лайт 12-13

Начальный вес  
Начальный вес

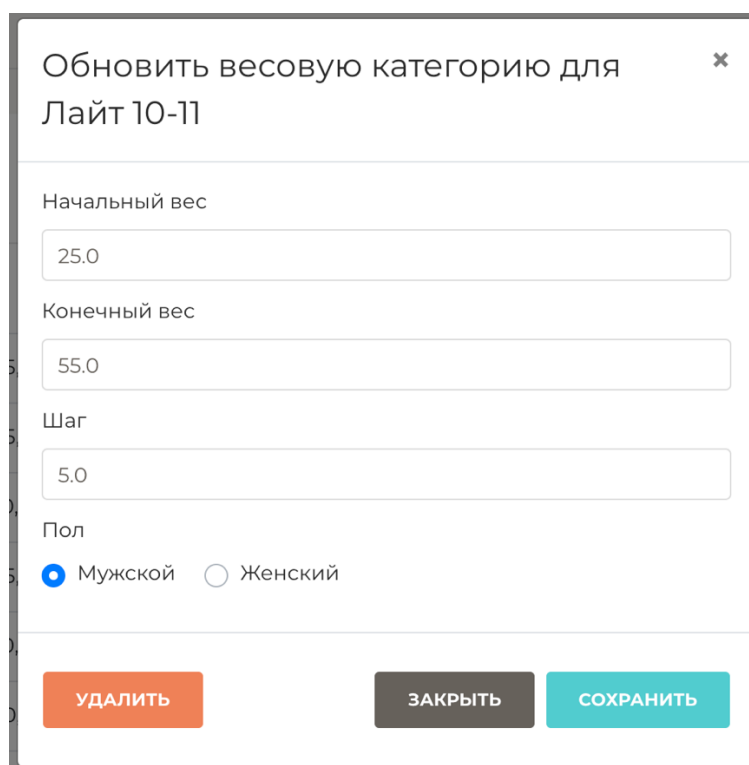
Конечный вес  
Конечный вес

Шаг  
Шаг

Пол  
 Мужской  Женский

ЗАКРЫТЬ ДОБАВИТЬ

Рисунок 3.12 – Модальное вікно додавання інформації про вагову категорію



Обновить весовую категорию для Лайт 10-11

Начальный вес  
25.0

Конечный вес  
55.0

Шаг  
5.0

Пол  
 Мужской  Женский

УДАЛИТЬ ЗАКРЫТЬ СОХРАНИТЬ

Рисунок 3.13 – Модальне вікно редагування інформації про вагову категорію

Сторінка «Соревнования» містить інформацію про змагання, як показано на рис. 3.14, з можливістю створювати нові змагання, як показано на рис. 3.15.



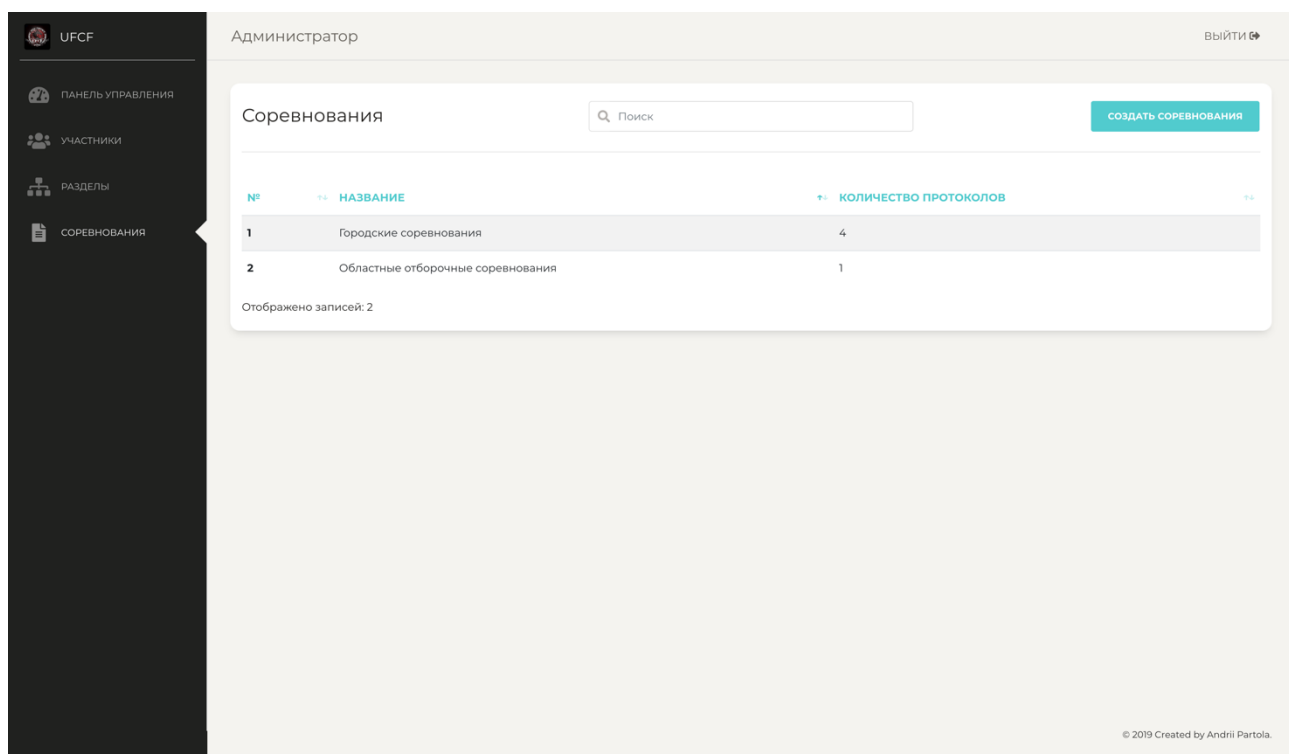


Рисунок 3.14 – Користувацький інтерфейс сторінки «Соревнования»

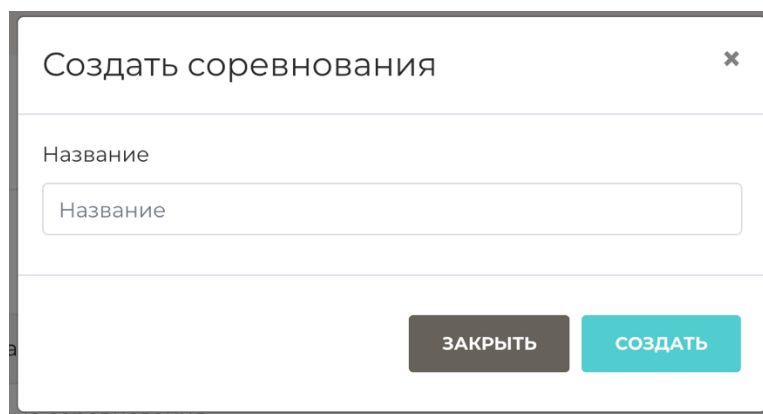


Рисунок 3.15 – Модальне вікно створення змагань

Після створення змагань користувач має можливість переглянути створені протоколи змагань, як показано на рис. 3.16.

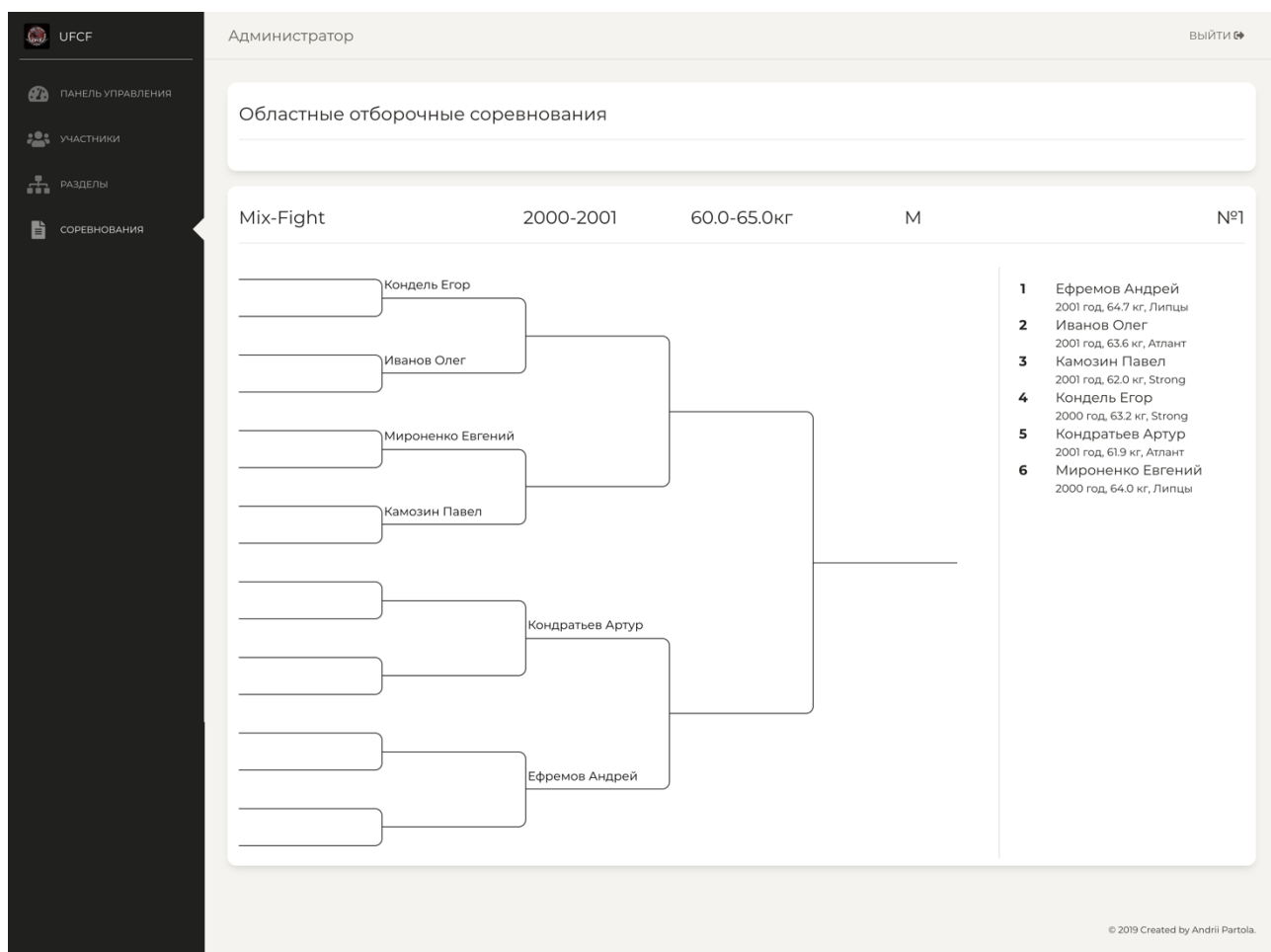


Рисунок 3.16 – Результат генерації протоколів спортивних змагань

Спортсмени у кожному протоколі відібрані за віковими, ваговими категоріями, статтю та категоріями змагань, а також розподілені по турнірній сітці відповідно до описаного алгоритму.

Для доступу до даних у пакеті «model» для кожної сутності були створені Java класи-моделі, як показано на рис. 3.17, на основі яких Hibernate створює таблиці у базі даних.

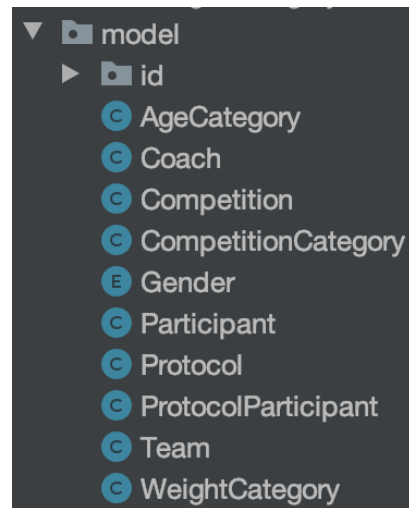


Рисунок 3.17 – Структура пакету «model»

Для доступу до бази даних у пакеті «dao» були створені інтерфейси Java Persistence API репозиторіїв, як показано на рис. 3.18.

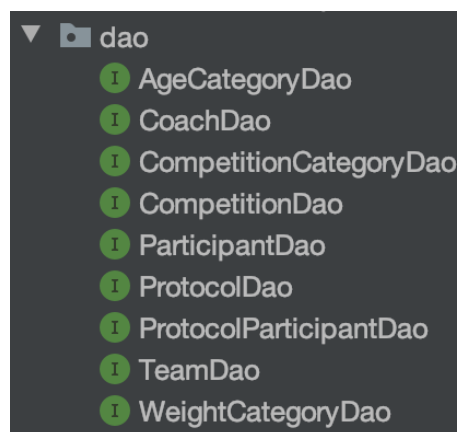


Рисунок 3.18 – Структура пакету «dao»

Для обробки даних у пакеті «service» були створені інтерфейси та класи сервісів, як показано на рис. 3.19, де кожен сервіс обробляє дані для конкретної сутності [53]. Клас-сервіс «ProtocolService», лістинг якого наведений у додатку Б, виконує роль генерації протоколів спортивних змагань.

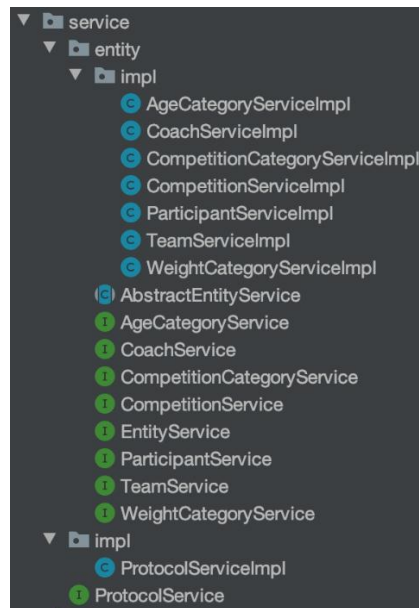


Рисунок 3.19 – Структура пакету «service»

Для представлення даних у пакеті «controller» були створені класи-контролери, як показано на рис. 3.20. Підпакет «rest» містить REST-контролери, що служать для передачі стану представлення [54].

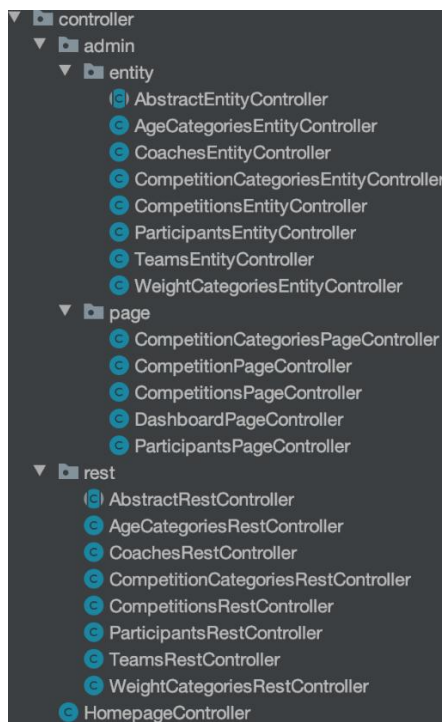


Рисунок 3.20 – Структура пакету «controller»

### 3.3 Критичний аналіз отриманих результатів

Було проаналізовано результати роботи алгоритму розподілу спортсменів за

турнірною сіткою, коли кількість учасників дорівнює двом, як показано на рис. 3.21, трьом, як показано на рис. 3.22, п'яти, як показано на рис. 3.23, та шести, як показано на рис. 3.24. Для всіх перевірених комбінацій учасників отримані результати збігаються з очікуваними, тобто спортсмени займають правильне положення у турнірній сітці (колонка та рядок), а також спортсмени з однієї команди знаходяться у різних частинах турнірної сітки та зустрічаються у парі у останню чергу. У кожному протоколі представлені тільки ті спортсмени, що належать до вікової категорії (вік учасника знаходиться у діапазоні між найменшим та найбільшим віком протоколу), вагової категорії (вага учасника знаходиться у діапазоні між найменшою та найбільшою вагою протоколу), статі, а також категорії спортивних змагань, для якої було створено протокол. Для випадків, коли за жодним критерієм протоколу не було знайдено жодного відповідного учасника, протоколи не були створені.

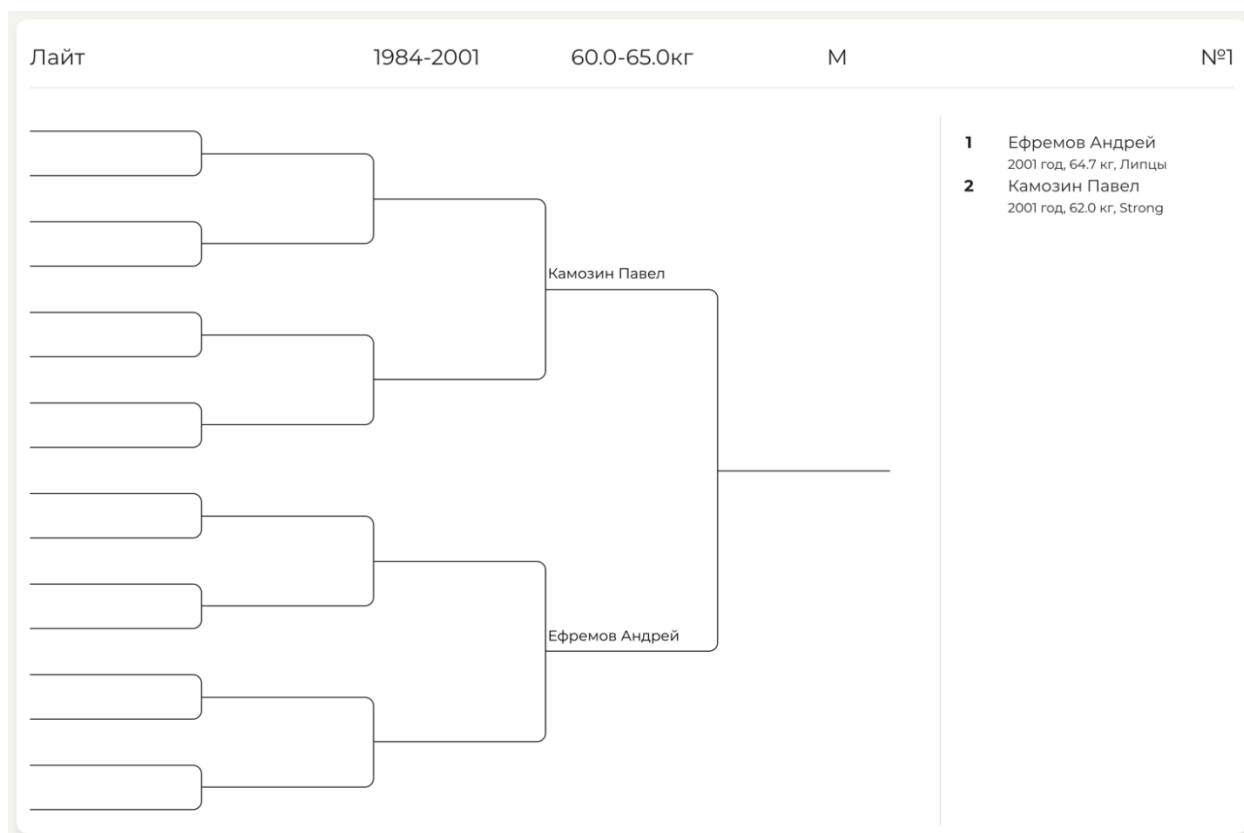


Рисунок 3.21 – Результат генерації протоколів для двох учасників

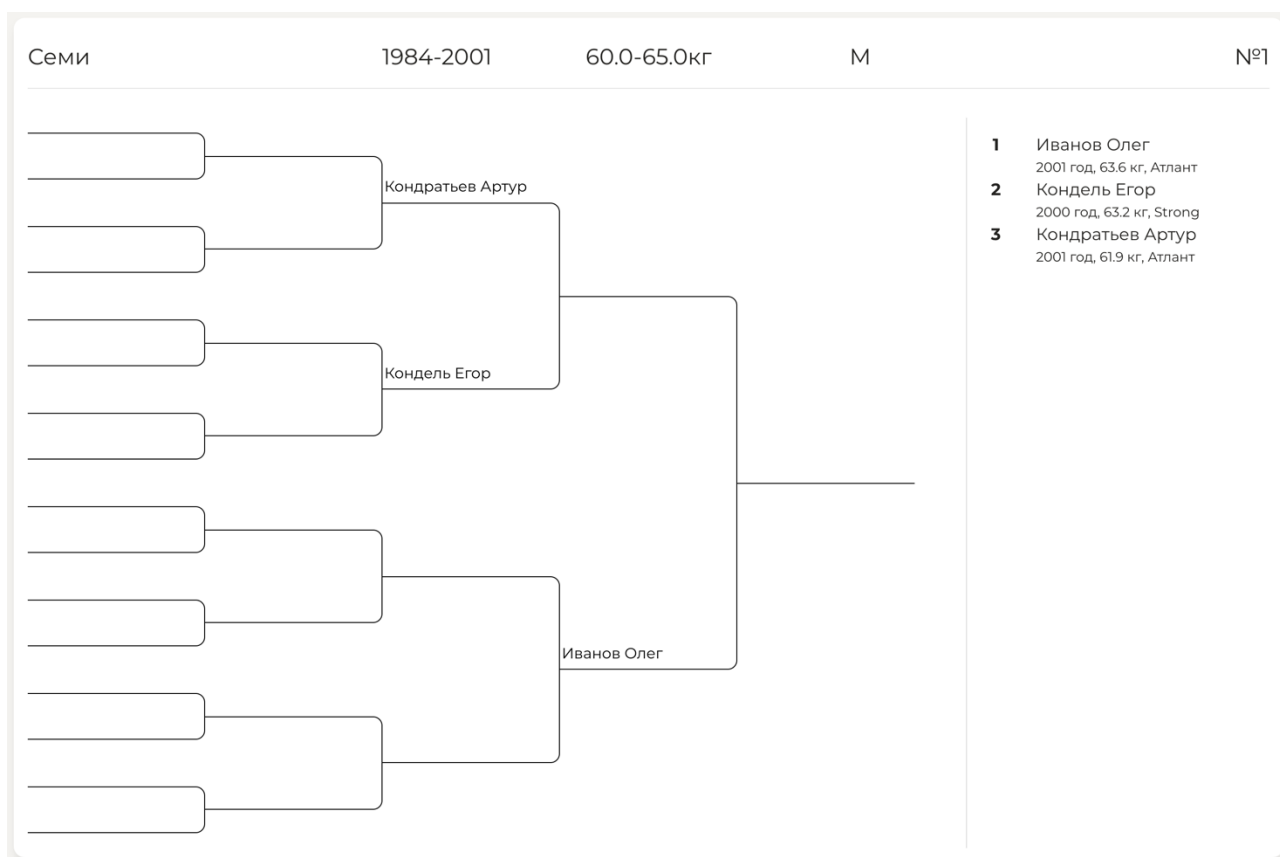


Рисунок 3.22 – Результат генерації протоколів для трьох учасників

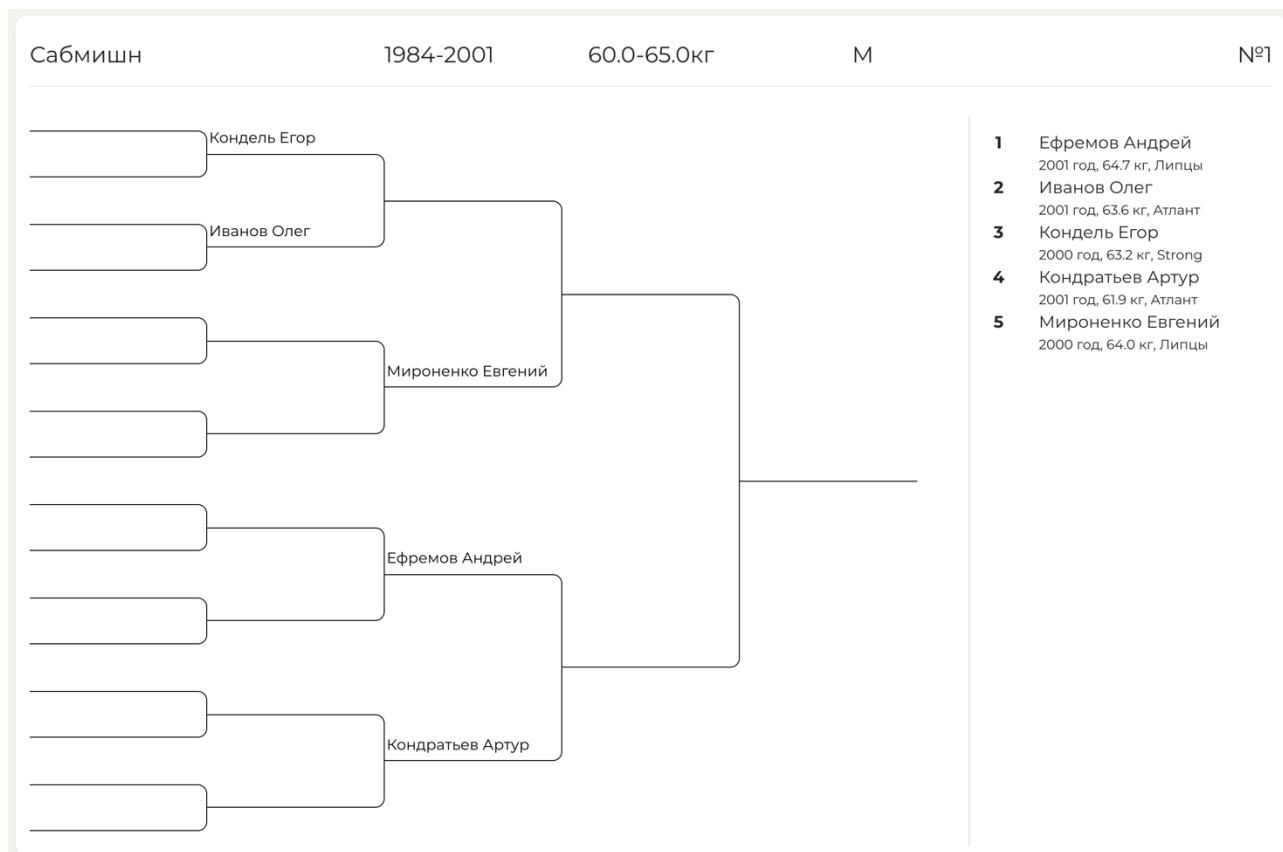


Рисунок 3.23 – Результат генерації протоколів для п'яти учасників

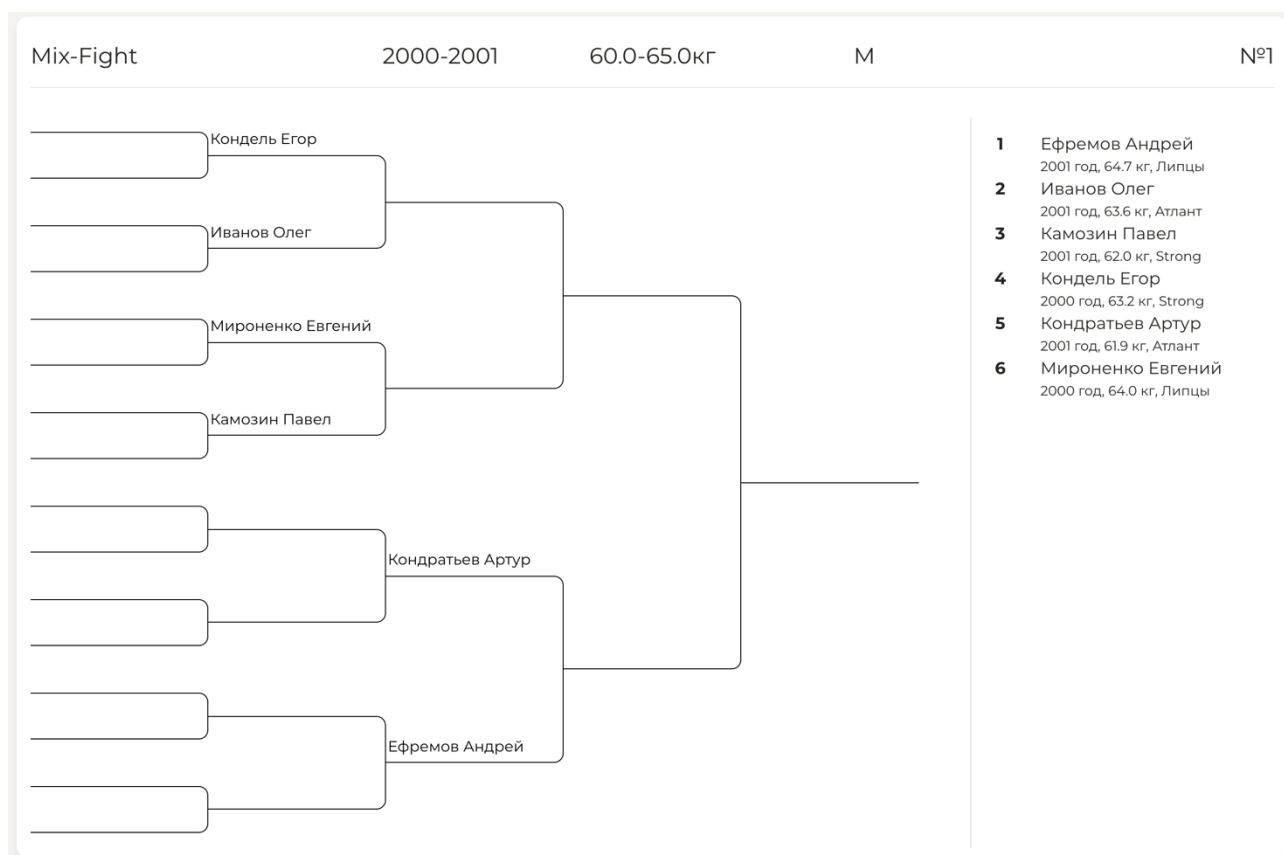


Рисунок 3.24 – Результат генерації протоколів для шести учасників

Недоліком є відсутність можливості візуального відображення протоколу, коли кількість учасників перевищує 16. Впровадження даної функціональності планується у наступних версіях програмного продукту.

## ВИСНОВКИ

Під час виконання дипломної роботи було проаналізовано та оптимізовано інформаційну систему протоколювання спортивних змагань з уніфікованим алгоритмом розподілу спортсменів.

Було проаналізовано можливі способи розподілу спортсменів за турнірною сіткою протоколів спортивних змагань. Також було виконано опис математичної задачі протоколювання та запропоновано алгоритм генерації протоколів, складність якого є логарифмічною та залежить від кількості учасників, що належать до відповідного протоколу спортивних змагань.

Для реалізації алгоритму, а також можливості керування даними було створено web-додаток на мові Java з використанням технологій Hibernate та Spring Framework. Для збереження даних було спроектовано оптимальну модель бази даних.

Було проаналізовано результати генерації протоколів спортивних змагань для різної кількості учасників. Для усіх проаналізованих випадків отримані результати збігаються з очікуваними.

Для вдосконалення процесу організації спортивних змагань Харківської обласної федерації контактних єдиноборств були розроблені наступні пропозиції:

- розширення додатку таким чином, щоб кількість учасників могла перевищувати 16;
- можливість змінювати результати генерації протоколів за потребою;
- можливість проведення спортивних змагань на основі створених протоколів у реальному часі, тобто ведення рахунку змагань у створених протоколах та визначення переможців;
- можливість друкування оброблених протоколів;
- можливість для учасників відстежувати статус проведення спортивних змагань, використовуючи веб-додаток;
- можливість для тренерів створювати та відсилати заявки до спортивних змагань, використовуючи веб-додаток.



## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. ДСТУ 1.5:2015 Національна стандартизація Правила розроблення, викладання та оформлення нормативних документів. Київ: ДП "УкрНДНЦ", 2015. 65с.
2. Вимоги до оформлення курсових і дипломних проектів: методичні рекомендації для студентів галузей знань 12 "Інформаційні технології" та 18 "Виробництво та технології" / уклад. Гаврилова А.А., Євсєєв С.П., Коц Г.П., Пушкар О. І, Руденко О. Г. Харків: Вид. ХНЕУ ім. С. Кузнеця, 2018. 49 с.
3. Елиферов В. Г. Бизнес-процессы: Регламентация и управление. Москва: ИНФРА-М, 2014. 320 с.
4. Глушань В.М. Критериальные особенности оптимального формирования турнирных таблиц // Конгресс по интеллектуальным системам и информационным технологиям «AIS– IT'14»: сб. тр. В 4-х т. М.: Физматлит, 2015. Т. 1. С. 506–512.
5. MySQL руководство администратора. // М.: Вильямс, 2016. 621 с.
6. Аткинсон MySQL. Библиотека профессионала / Аткинсон, Леон. // М.: Вильямс, 2015. 624 с.
7. Яргер, Р.Дж. MySQL и mSQL: Базы данных для небольших предприятий и Интернета / Р.Дж. Яргер, Дж. Риз, Т. Кинг. // М.: СПб: Символ-Плюс, 2017. 560 с.
8. Уорсли, Дж. PostgreSQL. Для профессионалов / Дж. Уорсли, Дж. Дрейк. // М.: СПб: Питер, 2011. 496 с.
9. Стоунз PostgreSQL. Основы / Стоунз, Мэттью Ричард; Нейл. // М.: СПб: Символ-Плюс, 2012. 640 с.
10. Саймон, Ригс Администрирование PostgreSQL 9. Книга рецептов / Ригс Саймон. // М.: ДМК Пресс, 2018. 182 с.
11. Уорсли, Дж. PostgreSQL. Для профессионалов (+ CD) / Дж. Уорсли, Дж. Дрейк. // М.: СПб: Питер, 2012. 496 с.
12. Стоунз PostgreSQL. Основы / Стоунз, Мэттью Ричард; , Нейл. // М.: СПб:

Символ-Плюс, 2012. 640 с.

13. SQL Server 2005 шаг за шагом. Практическое руководство. // М.: Эком, 2011. 463 с.

14. Бирн Microsoft SQL Server 6.5. Руководство администратора / Бирн, Джеффри. // М.: ЛОРИ, 2018. 211 с.

15. Нильсен SQL Server 2005. Библия пользователя / Нильсен, Пол. // М.: Вильямс, 2015. 499 с.

16. Эккель, Брюс Философия Java / Брюс Эккель. // М.: Питер, 2016. 809 с.

17. Савитч, Уолтер Язык Java. Курс программирования / Уолтер Савитч. // М.: Вильямс, 2015. 928 с.

18. Шилдт, Герберт Java 8. Руководство для начинающих / Герберт Шилдт. // М.: Вильямс, 2015. 720 с.

19. Шилдт Г. Полный справочник по Java. // М.: Вильямс, 2017. 1040 с.

20. Хорстманн К.С., Корнелл Г. Java 2. Основы, Т.1. // М.: Вильямс, 2017. 898 с.

21. Хемраджани А. Гибкая разработка приложений на Java с помощью Spring, Hibernate и Eclipse. // М.: Вильямс, 2018. 352 с.

22. Троелсен Э. С# та платформа .NET. Библиотека програміста. Санкт - Петербург: Санкт-Петербург, 2014. 796 с.

23. Лутай С., Байдачный С.Е Windows 8 для С# разработчиков. Москва: СОЛОН-Пресс, 2012. 278 с.

24. Рихтер CLR via C#. Программирование на платформе Microsoft .NET Framework 2.0 на языке C# / Рихтер, Джеффри. // М.: Питер, 2017. 656 с.

25. Чакраборти, Ангшуман Microsoft .NET Framework. Разработка профессиональных проектов / Ангшуман Чакраборти и др. // М.: БХВ-Петербург, 2013. 884 с.

26. Джон, Скит C#. Программирование для профессионалов / Скит Джон. // М.: Диалектика / Вильямс, 2011. 563 с.

27. Johnson R. Professional Java Development with the Spring Framework / R. Johnson, J. Hoeller, A. Arendsen, et al. // Wrox Press, 2015. 672 p.

28. Алгазинов Е.К. Анализ и компьютерное моделирование информационных процессов и систем / Е.К. Алгазинов, А.А. Сирота. // М.: Диалог МИФИ, 2012. 414 с.
29. Черушева Т. В. Проектирование программного обеспечения / Т. В. Черушева. // Пенза : Изд-во ПГУ, 2014. 172 с.
30. Бауэр, К. Java Persistence API и Hibernate / К. Бауэр. // М.: ДМК Пресс, 2017. 797 с.
31. Рассел, Джесси Hibernate (библиотека) / Джесси Рассел. // М.: VSD, 2013. 390 с.
32. Агафонов, В.Н. Логическое программирование / В.Н. Агафонов. // М.: ДМК, 2017. 579 с.
33. Г. Буч. Объектно-ориентированный анализ и проектирование / Г. Буч // СПб.: Издательство Бинум, Невский диалект, 2018. 560 с.
34. Соммервиль И. Инженерия программного обеспечения / И. Соммервиль. // К.: Вильямс, 2015. 680 с.
35. Вигерс К. Разработка требований к программному обеспечению / К. Вигерс. // М.: Русская Редакция, 2014. 580 с.
36. Гладков Л.А., Шкурко В.А. Решение задачи разбиения графа на подграфы на основе генетического алгоритма // Конгресс по интеллект. сист. и информ. технологиям «IS–IT’16»: сб. тр. В 4-х т. М.: Физматлит, 2016. Т. 3. С. 142–146.
37. Курейчик В.М., Кажаров А.А. Роевой интеллект в решении графовых задач // Междунар. конф. по мягким вычислениям и измерениям (SCM'2013): сб тр. СПб, 2013. С. 31–34.
38. Асанов, М.О. Дискретная математика: графы, матроиды, алгоритмы / М.О. Асанов, В.А. Баранский, В.В. Расин. // М.: Дрофа, 2011. 662 с.
39. Майника, Э. Алгоритмы оптимизации на сетях и графах / Э. Майника. // М.: МАИ, 2011. 359 с.
40. Большаков А.А. Методы обработки многомерных данных и временных рядов / А.А. Большаков, Р.Н. Каримов. // М.: Горячая линия-Телеком, 2017. 522 с.

41. Раскин Л.Г. Нечеткая математика: моногр. / Л.Г. Раскин, О.В. Серая. // Х.:Парус, 2011. 352 с.
42. Гладков Л.А., Гладкова Н.В., Бричеева Ю.В. Решение задачи компоновки схем ЭВА на основе выделения клик // Конгресс по интеллект. сист. и информ. технологиям «IS–IT 13»: сб. тр. В 4-х т. М.: Физматлит, 2013. Т. 3. С. 151–153.
43. Воеводин В.В., Воеводин Вл.В. Параллельные вычисления. СПб: БХВ-Петербург, 2011. 608 с.
44. Курейчик В.М., Глушань В.М., Щербаков Л.И. Комбинаторные аппаратные модели и алгоритмы в САПР. М.: Радио и связь, 2016. 216 с.
45. Калиткин Н.Н. Численные методы / Н.Н. Калиткин. // М.: Наука, 2018. 512 с.
46. Кормен, Томас Алгоритмы. Построение и анализ / Томас Кормен и др. // М.: Вильямс, 2012. 975 с.
47. Глушань В.М., Липало Н.Н., Малютин В.А. Оптимизация тестовых заданий при контроле знаний // Вестн. Таганрогского гос. пед. ин-та: Естественные науки. 2012. No 1. С. 72–76.
48. Кендалл М. Теория распределений: Пер. с англ. / М. Кендалл, А. Стюарт. // М.: Наука, 2011. 588 с.
49. Бахвалов Н.С. Численные методы / Н.С. Бахвалов. М.: Наука, 2012. 631 с.
50. Раскин Л.Г. Математические методы исследования операций и анализа сложных систем / Л.Г. Раскин. // Х.: ВИРТА, 2016. 178 с.
51. Гашков, С. Б. Арифметика. Алгоритмы. Сложность вычислений / С.Б. Гашков, В.Н. Чубариков. // М.: Дрофа, 2015. 320 с.
52. Конноли Т. Базы данных: проектирование, реализация и сопровождение. Москва : Издательский дом "Вильямс", 2013. 1120 с.
53. Басс Л. Архитектура программного обеспечения на практике / Л. Басс, П. Клементс, Р. Кацман. // СПб: Питер, 2016. 576 с.
54. Биберштейн Н. Компас в мире сервис-ориентированной архитектуры