

Центральноукраїнський національний технічний університет
Механіко-технологічний факультет
Кафедра кібербезпеки та програмного забезпечення

”Допущено до захисту”
Завідувач кафедри кібербезпеки
та програмного забезпечення
д.т.н., професор
_____ Олексій СМІРНОВ
“ ____ ” _____ 2024 р.

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за другим (магістерським) рівнем вищої освіти
на тему
“Дослідження та програмна реалізація системи генерування
складних динамічних структур”

Виконав здобувач вищої освіти
II курсу, групи КН-23М
ОПП «Комп’ютерні науки»
спеціальності 122 «Комп’ютерні науки»
_____ Чирва М.В.
« ____ » _____ 2024 р.

Керівник проекту
кандидат технічних наук, доцент
_____ Коваленко А.С.
« ____ » _____ 2024 р.
Рецензент _____

Центральноукраїнський національний технічний університет
Факультет Механіко-технологічний
Кафедра Кібербезпеки та програмного забезпечення
Рівень вищої освіти магістр
Галузь знань 12 "Інформаційні технології"
Спеціальність 122 "Комп'ютерні науки"
Освітньо-професійна (освітньо-наукова) програма "Комп'ютерні науки"

ЗАТВЕРДЖУЮ

Завідувач кафедри

д.т.н., проф.

Олексій СМІРНОВ

« 6 » вересня 2024 року

ЗАВДАННЯ НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ РОБОТУ ЗА ДРУГИМ (МАГІСТЕРСЬКИМ) РІВНЕМ ВИЩОЇ ОСВІТИ ЗДОБУВАЧА ВИЩОЇ ОСВІТИ

Чирві Максиму Віталійовичу

(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження та програмна реалізація системи генерування складних динамічних структур

2. Керівник роботи Коваленко Анна Степанівна, канд. техн. наук, доцент

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

затверджені наказом вищого навчального закладу № 18-13 від 07.08.2024 року

3. Строк подання студентом роботи до захисту 2.12.2024 р.

4. Мета та завдання випускної кваліфікаційної роботи: Метою розробки є дослідження та програмна реалізація системи генерування складних динамічних структур

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

1. Призначення та область використання.

6. Наукова новизна.

2. Перегляд аналогічних існуючих систем.

7. Маркетингове та економічне обґрунтування ІТ-проєкту.

3. Опис і обґрунтування проектних рішень.

8. Заходи з охорони праці та техніки безпеки.

4. Етапи програмування системи.

9. Висновки.

5. Впровадження системи в промислову експлуатацію

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

Наукова новизна

1 аркуш

Структурна схема системи

1 аркуш

Функціональна схема системи

1 аркуш

Діаграма процесів

1 аркуш

Блок-схема алгоритму роботи додатку

2 аркуша

Показники економічної ефективності

1 аркуш

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Економічний	Доренська А.О.	05.10.2024	14.11.2024
Охорона праці	Марченко К.М., к.т.н., доцент	06.10.2024	16.11.2024

7. Дата видачі завдання « 6 » вересня 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти	Строк виконання етапів випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти	Примітка
1.	Аналіз існуючих систем	10.10.2024 р.	
2.	Постановка задачі, оформлення ТЗ	15.10.2024 р.	
3.	Розробка моделі компонента	20.10.2024 р.	
4.	Розробка структур даних	25.10.2024 р.	
5.	Розробка алгоритмів зв'язку та відображення	30.10.2024 р.	
6.	Програмування алгоритмів	10.11.2024 р.	
7.	Розрахунок економічної ефективності	13.11.2024 р.	
8.	Розрахунки з охорони праці та техніки безпеки	15.11.2024 р.	
9.	Оформлення ПЗ	17.11.2024 р.	
10.	Попередній захист роботи	2.12.2024 р.	

Дата видачі завдання
« 6 » вересня 2024 р.

Підпис керівника

(прізвище та ініціали)Завдання прийнято до виконання
« 6 » вересня 2024 р.

Підпис здобувача

(прізвище та ініціали)

АНОТАЦІЯ

Чирва М.В. Дослідження та програмна реалізація системи генерування складних динамічних структур. 122 Комп'ютерні науки. Центральноукраїнський національний технічний університет. Кропивницький. 2024.

В даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи генерування складних динамічних структур.

Метою розробки є дослідження та програмна реалізація системи генерування складних динамічних структур.

Об'єктом дослідження є процес генерування складних динамічних структур.

Предметом дослідження є методи генерування складних динамічних структур.

Методи дослідження базуються на методах теорії складних систем, методах математичної статистики, методах розробки програмного забезпечення.

Результат роботи – програмна реалізація системи генерування складних динамічних структур.

В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.

Розроблено зручний інтерфейс користувача. Наведені інструкції по роботі з програмними засобами.

Програма може використовуватися на ПЕОМ з ОС Windows 10/11.

Програму розроблено в середовищі PHP фреймворку Yii2.

Ключові слова: Комп'ютерні науки генерування складних динамічних структур

ABSTRACT

Chirva M.V. Research and software implementation of the system for generating complex dynamic structures. 122 Computer Science. Central Ukrainian National Technical University. Kropyvnytskyi. 2024.

In this graduation thesis for the second (master's) level of higher education, software was developed, which is intended for the system of generating complex dynamic structures.

The purpose of the development is the research and software implementation of the system for generating complex dynamic structures.

The object of research is the process of generating complex dynamic structures.

The subject of research is methods of generating complex dynamic structures.

Research methods are based on the methods of theory of complex systems, methods of mathematical statistics, methods of software development.

The result of the work is the software implementation of the system for generating complex dynamic structures.

In the process of working on the software model, an analysis of existing hardware and software was performed. All components of the developed software are fully described.

A convenient user interface has been developed. Instructions for working with software tools are provided.

The program can be used on a PC with Windows 10/11 OS.

The program was developed in the PHP environment of the Yii2 framework.

Keywords: Computer science of generating complex dynamic structures

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ	3
ВСТУП.....	4
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ	7
1.1 Призначення системи.....	7
1.2 Область застосування.....	7
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ	9
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.....	9
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування.....	14
2.3 Розгорнута постановка завдання	18
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ	20
3.1 Опис функціонування системи	20
3.2 Розробка структурної схеми.....	22
3.3 Розробка функціональної схеми	24
3.4 Розробка діаграми процесів.....	27
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....	29
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	29
4.2 Захист розробленого програмного забезпечення.....	42
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ	44
6 НАУКОВА НОВИЗНА	48

						ВКРМ-122.24.0019.00.00.ПЗ		
Вим.	Арк.	№ докум.	Підп.	Дата				
Розроб.	Чирва М.В.				Дослідження та програмна реалізація системи генерування складних динамічних структур	Літ.	Аркуш	Аркушів
Перев.	Коваленко А.С.					М	1	75
Н.контр.	Коваленко А.С.					ЦНТУ КН-23М		
Затв.	Смірнов О.А.							

7	МАРКЕТИНГОВЕ ТА ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ ІТ-ПРОЄКТУ	49
7.1	Визначення цільової аудиторії кінцевого готового продукту	49
7.2	Оцінка привабливості шляхом застосування методів експертних оцінок ...	50
7.3	Вибір методу оцінки вартості ПЗ	52
7.4	Розрахунок економічної ефективності від впровадження реалізованого ПЗ як фактору його привабливості.....	52
7.5	Пропозиція алгоритму просування проєкту розробки ПЗ	54
7.6	Оптимізація каналів збуту та шляхів реалізації ПЗ	55
7.7	Визначення ключових факторів успіху конкретного проєкту.....	56
8	ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ	58
8.1	Вступ.....	58
8.2	Характеристика умов праці програміста	60
8.3	Розробка заходів з умов поліпшення охорони праці.....	64
8.4	Розрахункова частина	65
8.5	Висновки до розділу.....	66
9	ОСНОВНІ ВИСНОВКИ.....	67
	СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	69

КБПЗ-2024

					ВКРМ-122.24.0019.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		2

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

ЕОМ	– електронна обчислювальна машина
КС	– комп'ютерна система
КТЗ	– комплекс технічних засобів
МЕТОЗ	– методичне забезпечення
ОС	– операційна система
ОРЗ	– організаційне забезпечення
ПЕОМ	– персональна обчислювальна машина
ПЗ	– програмне забезпечення
ПК	– персональний комп'ютер
ПП	– програмний продукт
SVG	– Scalable Vector Graphics
HTML	– HyperText Markup Language
MVC	– Model View Controller
БД	– база даних
DLL	– бібліотека динамічної компоновки

					ВКРМ-122.24.0019.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		3

ВСТУП

Актуальність теми. На сьогодні немає жодної сфери діяльності людини, яка б не використовувала можливості сучасних інформаційних технологій або не могла б бути автоматизована за допомогою комп'ютерів. Одними з таких є спорт та різноманітні змагання. Важко уявити спортивну чи кіберспортивну дисципліну у якій не проводилися б турніри. Якісна організація подібних змагань потребує немало зусиль, уваги та знань. Тому автоматизація такого процесу надає можливість зручного вирішення подібних проблем.

Метою магістерського проекту є розробка програмного забезпечення генерування складних динамічних структур інформації на прикладі турнірних таблиць результатів змагань.

Об'єктом дослідження є процес автоматизованого створення та адміністрування турнірних таблиць на базі двійкових дерев.

Предметом дослідження є механізми роботи веб-сервісів та розробка програмного забезпечення, яке генерує інформаційні структури на основі повних двійкових дерев та виконує подальші дії для забезпечення зручного адміністрування, такі як планування, внесення і обробка даних, їх публікація, ведення обліку.

Методи дослідження базуються на основі вивчення існуючих систем та аналізу методів роботи з веб-сервісами, двійковими деревами а також роботи з базами даних і архітектурою MVC.

Практичне значення одержаних результатів полягає в тому, що розроблене програмне забезпечення може бути використане в наступних галузях:

- створення та проведення змагань з використанням олімпійських систем турніру double elimination та single elimination;
- розробка ігор;
- комбінаторика;

					ВКРМ-122.24.0019.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

- теорія ігор;
- екстенсивна або розгорнута форма гри;
- задачі з використанням повного бінарного дерева.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи генерування складних динамічних структур.

Для досягнення поставленої мети визначена програма дослідження, що складається з наступних завдань:

- Огляд існуючих систем генерування складних динамічних структур.
- Дослідження системи генерування складних динамічних структур.
- Програмна реалізація системи генерування складних динамічних структур.

Об'єктом дослідження є процес генерування складних динамічних структур.

Предметом дослідження є методи генерування складних динамічних структур.

Методи дослідження базуються на методах теорії складних систем, методах математичної статистики, методах розробки програмного забезпечення.

Наукова новизна отриманих результатів. У процесі рішення завдань, обумовлених цілями дослідження, отримані наступні результати:

- Удосконалено метод генерування складних динамічних структур.
- Розроблено вітчизняний продукт генерування складних динамічних структур, який має більш широкі можливості, на відміну від існуючих аналогів.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі генерування складних динамічних структур.

Достовірність наукових результатів підтверджена теоретичними викладеннями, даними комп'ютерного моделювання, коректними дослідженнями параметрів на функціонуючій обчислювальній мережі, а також відповідністю отриманих результатів окремим результатам, наведеним у науковій літературі.

					ВКРМ-122.24.0019.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

Робота апробована на LVII Науково-технічній конференції здобувачів вищої освіти LV науково-технічної конференції «Наука в ЦНТУ: основні досягнення та перспективи розвитку» (2024 р.), основні положення випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти надруковані у статті збірника праць молодих науковців ЦНТУ, випуск №15.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи генерування складних динамічних структур, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

КБПЗ – 2024

					ВКРМ-122.24.0019.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Програмне забезпечення, що розробляється в даній магістерській роботі, призначене для:

- забезпечення регіональних федерацій, місцевих об'єднань, спортивних клубів, кіберспортивних та аматорських організацій зручним та простим у користуванні програмним комплексом, необхідними для організації турнірних змагань;
- створення турнірної таблиці як для повного так і не повного списку учасників;
- введення та обробки результатів;
- планування та публікацію результатів.

1.2 Область застосування

Організація та проведення будь-яких турнірних змагань незалежно від їх рівня включає у себе створення турнірної таблиці, реєстрацію учасників, жеребкування, обробку і публікацію даних, та багато інших дій. Виконання усіх цих дій вручну може потребувати багато людських ресурсів. Окрім того, секретар чи людина, відповідаюча за ведення, турніру має бути компетентна у нюансах його проведення.

На сьогоднішня великого поширення набули різного плану системи автоматизації технічних та організаційних процесів, які раніше доводилось виконувати вручну. Наприклад, різноманітні прогнозуючі, регулюючі, проектуючі, експертні системи тощо, що замінюють досвідчених фахівців у самих різних областях науки, техніки, бізнесу. До подібних процесів можна віднести і створення турнірної таблиці. Сучасні технології мають у своєму

					ВКРМ-122.24.0019.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

розпорядженні засоби, що надають можливість найкращим чином автоматизувати будь-який процес, в тому числі і організаційний.

Розроблюваний програмний комплекс, представлений в даній магістерській роботі, покликаний знайти своє застосування у діяльності різноманітних спортивних, кіберспортивних, юнацьких та аматорських об'єднань і клубів, потребуючих зручного програмного засобу для проведення змагань у олімпійських системах турніру (single elimination, double elimination) а також деякі модулі та алгоритми можуть бути використані у розробці програмних симуляторів, включаючих у себе елементи турнірних змагань.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи генерування складних динамічних структур, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

КБПЗ - 2024

					VKPM-122.24.0019.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

2.1 Огляд існуючих систем, технологій, архітектур, програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти

Предметом розробки магістерського проекту (МП) є механізми генерування динамічних турнірних таблиць для проведення змагаль у Олімпійській та інших системах. Різноманітні комп'ютерні системи проведення змагань реалізують цей механізм, хоч і в більшості, лише для повного списку учасників (тобто відповідному кількості учасників 4, 8, 16 і т.д.).

До таких програмних комплексів можна віднести такі системи автоматизації турнірів як Tournament Planner, CAT Tournament Management Solution[1].

Tournament Planner – розроблена американською фірмою Visual Reality програма призначена для управління різноманітними видами турнірів і спеціалізується на таких видах спорту як теніс, сквош, дартс, та ін. Для кожного з них має окремий особливо налаштований клієнт, реалізуючий різноманітні нюанси кожного. (Рисунок 2.1). Екпортувати дані можна у HTML та Excel. Підтримується як ручне так і автоматизоване жеребкування. До недоліків можна віднести відсутність реалізації системи з вибуванням після двох поразок (Double elimination), наявність тільки англійського інтерфейсу, обмеженість декількома видами спорту та велику вартість покупки програмного пакету.

CAT – одна з найстаріших у світі програм для автоматизації проведення спортивних змагань, перша версія якої була розроблена ще на початку 80х років минулого сторіччя. Програма орієнтована лише на теніс та сквош. Експорт даних доступний у графічні формати зображень. Недоліками є відсутність підтримки системи до трьох поразок, яка популярна у грі в сквош; обмеженість кількості

					ВКРМ-122.24.0019.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

матчів у один день максимум двадцятьма, наявність лише англomовного інтерфейсу.

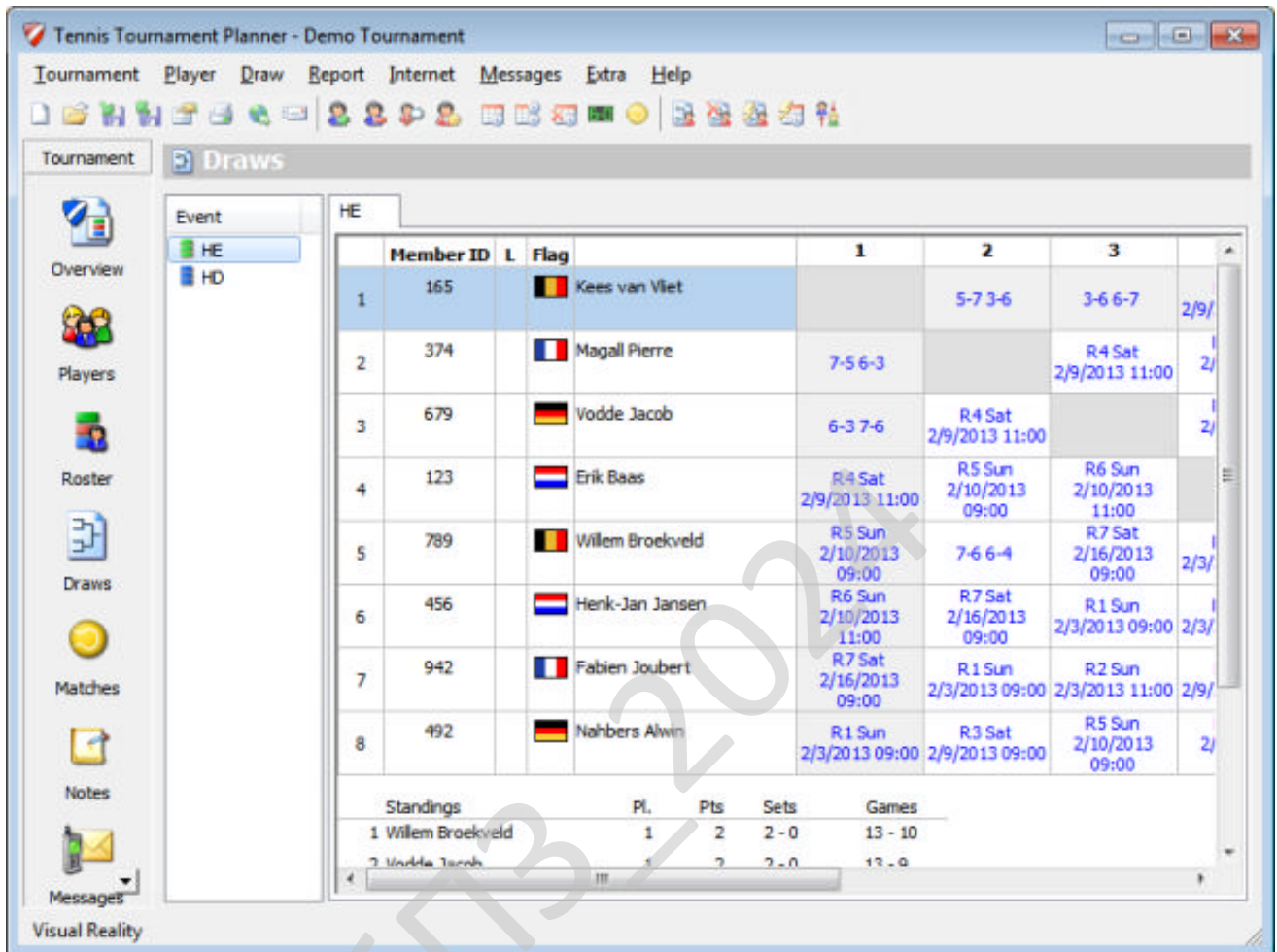


Рисунок 2.1 – Tournament Planner

Tourney master – клієнтська програма призначена для проведення та планування турнірів і змагань. (Рисунок 2.2).

В числі особливостей програми можна виділити:

- створення та оновлення турнірної сітки;
- жеребкування;
- створення розкладів;
- сортування та протоколювання ігор;
- експортування результатів в html або xls формати.

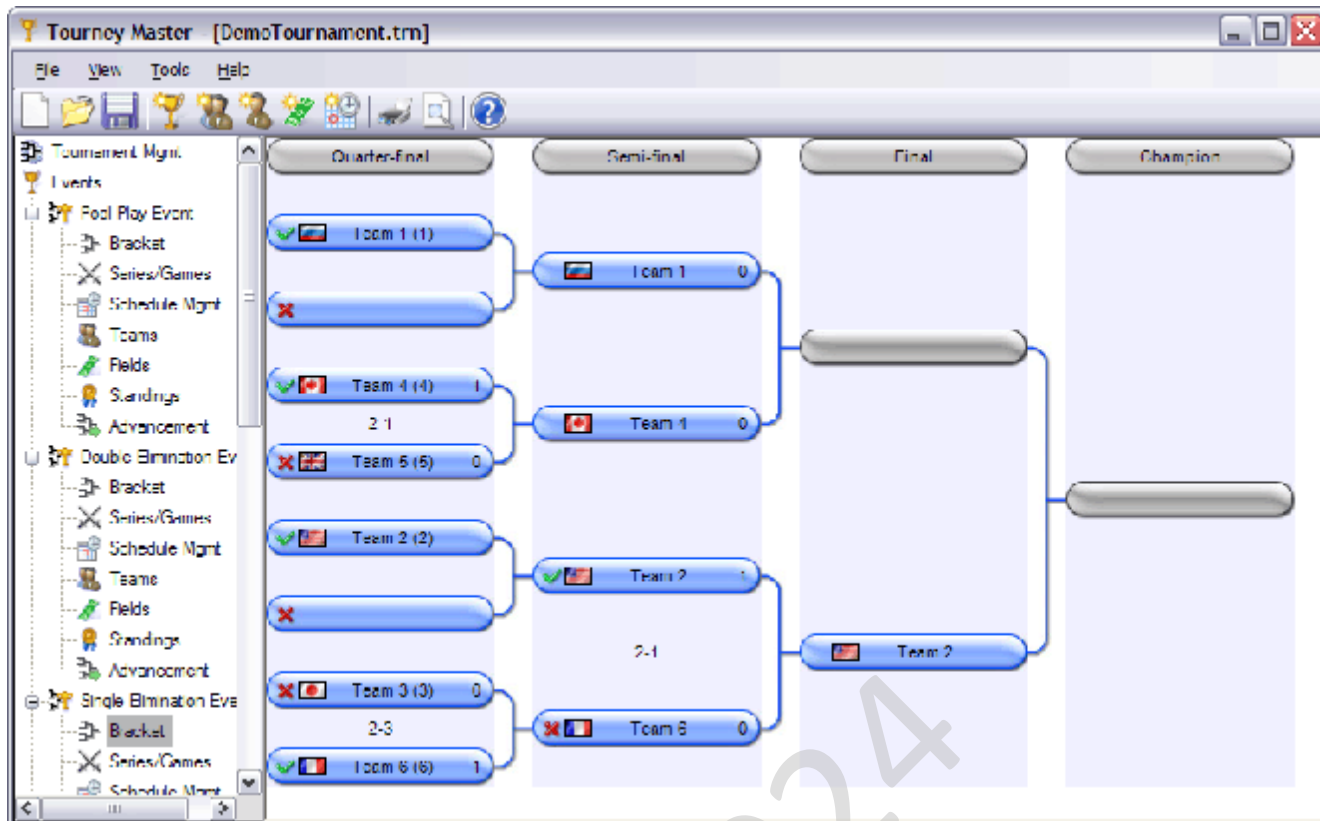


Рисунок 2.2 – Скріншот вікна програми Tourney master

Програма підтримує такі формати турнірів як олімпійський, олімпійський на вибування, раунд. Також вона дозволяє зв'язувати турніри один з одним, що дозволяє проводити їх у декілька раундів. До недоліків можна віднести обмеженість інтерфейсу тільки англійською мовою та складність у освоєнні.

Окрім великих програмних комплексів існує невелика кількість плагінів, створених для генерації турнірних таблиць олімпійської системи. Написані вони в більшості на мові JavaScript з використанням бібліотеки JQuery та використанням технологій SVG і HTML5.

Серед плагінів можна виділити такі як:

- Bracket World (Рисунок 2.3);
- Customizable Tournament, Brackets.js (Рисунок 2.4).

Нажаль, більшість подібних плагінів тільки англomовні, не мають документації та технічної підтримки і, що головне, не реалізують механізму генерації турнірної сітки для кількості учасників відмінної від 2 у степені n .

Крім великих комерційних клієнтських програм та веб-плагінів слід звернути увагу на веб-сервіси для створення та ведення турнірів та турнірних таблиць. Зазвичай вони більш зручні за рахунок доступності з будь-яких пристроїв незалежно від програмних вимог, реалізують багатомовний інтерфейс, дозволяють своєчасно слідкувати за змінами у турнірній таблиці.

Challonge – один з найпопулярніших веб-сервісів у сфері створення турнірних таблиць (Рисунок 2.5).

Challonge надає можливість створювати турніри у таких системах як олімпійська, олімпійська на вибування, раунд-робін, швейцарський турнір, двухсходова система. Створювати можна як вручну так із файлів. Максимальна кількість учасників – 256. Підтримується реєстрація учасників та запрошення до турніру через електронну пошту.

Проект був запущений Адамом Даррахом та Девідом Корнеліусом у 2009му році [2]. Створений як власна розробка не претендуюча на комерційність, проект швидко зайняв центральну нішу у своїй сфері. Серед партнерів сервісу є багато спортивних та кіберспортивних організацій таких як Team Decerto, ProjectX, LevelUp, Brackets for Good, EVG Tournaments [3].

Сервіс розроблено на мові PHP. Клієнтська частина використовує CSS фреймворк Twitter Bootstrap та JS бібліотеку JQuery.

До мінусів можна віднести обмежений функціонал безкоштовної версії та ненадання технічної підтримки, відсутність експорту прогресу та результатів у файл, наявність реклами.

В PHP вбудовані бібліотеки для роботи з MySQL, PostgreSQL, mSQL, Oracle, dbm, Hyperware, та іншими СУБД.

Мова PHP здаватиметься знайомою програмістам, що працюють в різних областях. Багато конструкцій мови запозичені з C, Perl. Код PHP дуже схожий на той, який зустрічається в типових програмах на C або Pascal. Це помітно знижує початкові зусилля при вивченні PHP. PHP – мова, що поєднує переваги Perl і C і спеціально спрямована на роботу в Інтернеті, мова з універсальним і зрозумілим синтаксисом. І хоча PHP є досить молодою мовою, вона здобула таку популярність серед web-програмістів, що в наш час є мало не найпопулярнішою мовою для створення веб-застосунків (скриптів).

Ефективність є дуже важливим чинником у програмуванні для середовищ розрахованих на багато користувачів, до яких належить і web. Важливою перевагою PHP є те, що ця мова належить до інтерпретованих. Це дозволяє обробляти сценарії з достатньо високою швидкістю. Проте хоч би що робили розробники PHP, виконавчі файли, отримані за допомогою компіляції, працюватимуть значно швидше – в десятки, а іноді і в сотні разів. Але продуктивність PHP достатня для створення цілком серйозних веб-додатків.[4]

MVC (або Модель-вид-контролер, англ. Model-view-controller, MVC) – архітектурний шаблон, який використовується під час проектування та розробки програмного забезпечення. Цей шаблон поділяє систему на три частини: модель даних, вигляд даних та керування.

Мета шаблону – гнучкий дизайн програмного забезпечення, який повинен полегшувати подальші зміни чи розширення програм, а також надавати можливість повторного використання окремих компонентів програми. Крім того використання цього шаблону у великих системах призводить до певної впорядкованості їх структури і робить їх зрозумілішими завдяки зменшенню складності.

У тріаді до обов'язків компоненту Модель (Model) входить зберігання даних і забезпечення інтерфейсу до них. Вигляд (View) відповідальний за

					ВКРМ-122.24.0019.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		16

представлення цих даних користувачеві. Контролер (Controller) керує компонентами, отримує сигнали у вигляді реакції на дії користувача, і повідомляє про зміни компоненту Модель. Така внутрішня структура в цілому поділяє систему на самостійні частини і розподіляє відповідальність між різними компонентами.

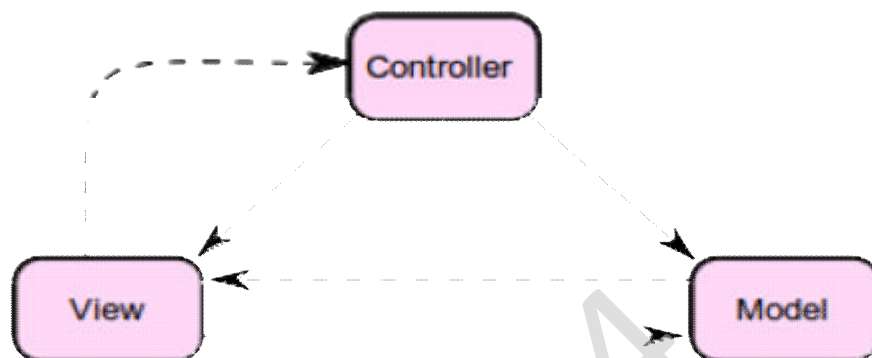


Рисунок 2.8 – Схема MVC

Модель інкапсулює ядро даних і основний функціонал з їх обробки. Також компонент Модель не залежить від процесу введення або виведення даних. Компонент виводу Вигляд може мати декілька взаємопов'язаних областей, наприклад, різні таблиці і поля форм, в яких відображається інформація. У функції Контролера входить моніторинг за подіями, що виникають в результаті дій користувача (зміна положення курсора миші, натиснення кнопки або введення даних в текстове поле).

Зареєстровані події транслюються в різні запити, що спрямовуються компонентам Моделі або об'єктам, відповідальним за відображення даних. Відокремлення моделі від вигляду даних дозволяє незалежно використовувати різні компоненти для відображення інформації. Таким чином, якщо користувач через Контролер внесе зміни до Моделі даних, то інформація, подана одним або декількома візуальними компонентами, буде автоматично відкоригована відповідно до змін, що відбулися.[5]

Yii2 – це високопродуктивний та швидкодіючий веб-фреймворк, написаний на PHP, реалізує парадигму модель-вид-контролер. Yii – скорочення від «Yes It Is!».

Головними перевагами і можливостями фреймворку є:

- Реалізація архітектурного шаблону Модель-вид-контролер.
- Інтерфейси DAO та Active Record для роботи з базами даних.
- Підтримка інтернаціоналізації.
- Кешування сторінок та окремих фрагментів.
- Перехоплення та обробка помилок.
- Введення та валідація веб-форм.
- Генерація базового PHP-коду для CRUD-операцій .
- Використання AJAX та інтеграція з jQuery.
- Міграції бази даних.
- Можливість підключення сторонніх бібліотек [6].

Спільнота розробників, використовуючих фреймворк, активно розвивається та розробляє все більше нових корисних компонентів, які легко можна підключити і використовувати при роботі з фреймворком.

2.3 Розгорнута постановка завдання

Згідно з технічним завданням на випускні кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, реалізації підлягає програмне забезпечення, яке призначено для системи генерування складних динамічних структур.

В процесі розробки випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти необхідно виконати наступний обсяг роботи:

а) провести аналіз існуючих систем-аналогів для виявлення їх позитивних і негативних якостей. Результати аналізу врахувати в подальших розробках;

б) вибрати та обґрунтувати методіку побудови системи контролю роботи технологічного обладнання на виробництві в автоматизованому режимі.

					ВКРМ-122.24.0019.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		18

Розробити функціональну та структурну схеми системи;

в) розробити програмне забезпечення системи, що дозволить реалізувати поставлену технічним завданням задачу. Побудувати блок-схеми алгоритмів програми та підпрограми;

г) організувати інтерфейс користувача з метою формування та виводу на екран ЕОМ повідомлень про некоректні дії користувача та нестандартні ситуації в роботі технологічного обладнання;

д) розробити рекомендації по організаційних та методичних заходах, які забезпечать впровадження системи в промислову експлуатацію та її подальшу успішну експлуатацію;

е) провести розрахунки по визначенню економічної ефективності розробленої системи;

ж) розробити заходи по охороні праці при впровадженні та експлуатації системи, а також розробити заходи з цивільного захисту;

з) сформуванати висновки про виконаний обсяг робіт та одержані результати.

КБПЗ - 2024

					VKPM-122.24.0019.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		19

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

Програмне забезпечення для динамічної генерації складних структур представлених у вигляді турнірних таблиць, яке розроблене у процесі виконання магістерського проекту, є складною багатофункціональною системою, що призначена для швидкого та зручного створення турнірних таблиць, їх управління та публікації .

Виходячи з висновків, зроблених під час розгляду вже існуючих систем, було вирішено розробити програмний комплекс у вигляді веб-сервісу, що дозволяє просто та зручно створювати та керувати турнірними таблицями.

Було обрано архітектурний шаблон MVC (Модель-Вид-Контролер) та його варіацію у PHP фреймворку Yii2. Моделі – це класи, що представляють собою дані, бізнес логіку і бізнес правила. Вигляд відповідає за представлення та відображення інформації, в тому числі і на основі даних, отриманих із моделі. Контролери приймають вхідні дані та запити від користувача та перетворюють їх в зрозумілий для моделі формат та команди, а також відповідають за відображення конкретних представлень, тобто виду. Окрім тріади основних сутностей, веб-додаток у даній реалізації складається з таких важливих сутностей як вхідний скрипт, об'єкт застосунку, компоненти, модулі, фільтри та віджети.

До функціональної частини бази даних можна віднести 5 таблиць (де зберігаються дані про користувачів, турніри, їх журналювання, турнірні сітки у вигляді двійкових дерев, та їх вузли).

Мова запитів SQL майже не використовувалась напряму і знайшла своє застосування тільки в запитах на створення таблиць.

					ВКРМ-122.24.0019.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20

Наприклад запит на створення таблиць турніру та вузла дерева виглядає

наступним чином:

```
CREATE TABLE `tcr_tournament` (  
  `id` int(10) unsigned NOT NULL AUTO_INCREMENT,  
  `title` varchar(255) NOT NULL,  
  `info` mediumtext,  
  `participants` text,  
  `three_id` int(10) unsigned DEFAULT NULL,  
  `owner_id` int(10) unsigned DEFAULT NULL,  
  `type` enum('single','double') NOT NULL DEFAULT 'single',  
  `created` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE  
CURRENT_TIMESTAMP,  
  
  PRIMARY KEY (`id`),  
  CONSTRAINT `FK_tourney_three` FOREIGN KEY (`three_id`) REFERENCES  
`tcr_three` (`id`) ON DELETE SET NULL ON UPDATE CASCADE,  
  CONSTRAINT `FK_tourney_owner` FOREIGN KEY (`owner_id`) REFERENCES  
`tcr_user` (`id`) ON DELETE SET NULL ON UPDATE CASCADE  
  ) ENGINE=InnoDB DEFAULT CHARSET=utf8;  
  
CREATE TABLE `tcr_node` (  
  `id` int(100) unsigned NOT NULL AUTO_INCREMENT,  
  `left` varchar(255),  
  `right` varchar(255),  
  `score_left` int(4) unsigned DEFAULT NULL,  
  `score_right` int(4) unsigned DEFAULT NULL,  
  `three_id` int(10) unsigned DEFAULT NULL,  
  `array_index` int(10) unsigned DEFAULT NULL,  
  `type` enum('active','passive') NOT NULL DEFAULT 'active',  
  
  PRIMARY KEY (`id`),  
  CONSTRAINT `FK_node_three` FOREIGN KEY (`three_id`) REFERENCES `tcr_three`  
(`id`) ON DELETE SET NULL ON UPDATE CASCADE  
  ) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

Різноманітні запити на вибірку, перевірку та додавання даних виконуються завдяки наданими фреймворком засобами. Yii2 надає такі можливості для комунікації з базою даних:

– Об'єкти доступу до даних (DAO): побудовані поверх PDO (PHP Data Objects – стандартні засоби мови для надання інтерфейсу доступу до БД) вони

					ВКРМ-122.24.0019.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		21

забезпечують об'єктно-орієнтований підхід для доступу до реляційних баз даних. Вони є основою для інших, більш просунутих, методів доступу до баз даних таких як будівник запитів та Active Record;

– Будівник запитів: побудований поверх DAO дозволяє конструювати SQL вирази у програмованому і незалежному від СУБД вигляді. Допомогає писати більш читаємий код та генерувати більш безпечні SQL запити;

– Active Record: клас, що забезпечує об'єктно-орієнтований інтерфейс для доступу і маніпулювання даними, що зберігаються у БД.

Було обрано останній метод для маніпулювання даними. Клас Active Record відповідає таблиці у БД, його об'єкт відповідає рядку у таблиці а його атрибут представляє собою значення окремого стовбчика рядка. Розширюючи свої моделі від класу Active Record можна отримати зручний механізм управління даними і абстрагуватись від мови запитів, виконуючи дії над даними за допомогою методів класу, наприклад створення нового запису у таблиці турнірів, присвоєння автора та збереження результату у БД виглядатиме так:

```
$tournament = new Tournament();  
$tournament->owner = 'Victor';  
$tournament->save();
```

Нижче представлені структурна та функціональна схеми, а також діаграма процесів, які нададуть наочне уявлення про принципи функціонування основних частин програмного забезпечення та взаємодію між ними.

3.2 Розробка структурної схеми системи

Структурна схема – це схема, яка визначає основні функціональні частини програми, їх взаємозв'язки та призначення. Структурна схема зображена на рисунку 3.1 відображає основні вузли програмного забезпечення та їх взаємодію між собою. Веб-сервіс, що розробляється, побудований з використанням шаблону проектування MVC.

					ВКРМ-122.24.0019.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22

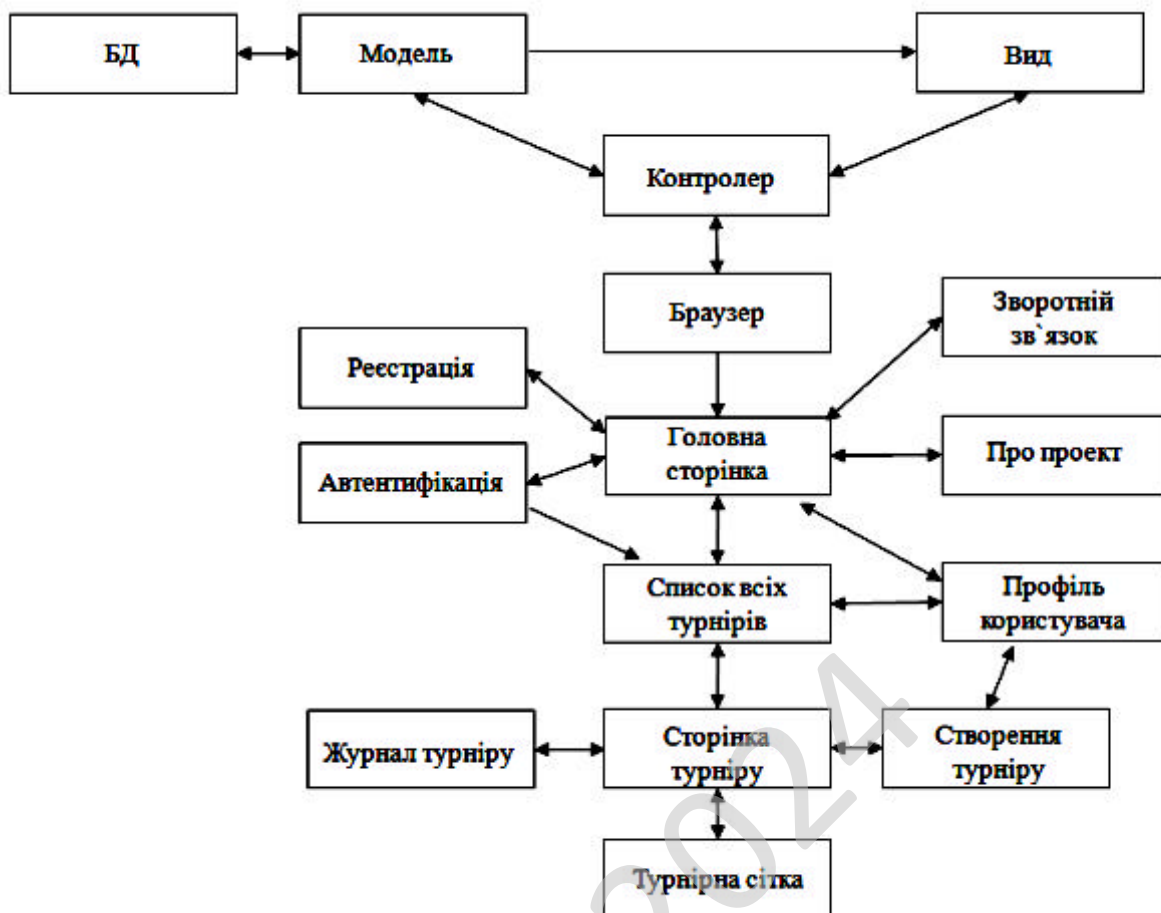


Рисунок 3.1 – Структурна схема системи

Структурна схема зображає основні блоки системи та дає уявлення про їх взаємодію між собою. Головним модулем є головна сторінка веб-сервісу, з якої при натисненні відповідних кнопок та посилань можна виконувати різноманітні дії. При виконанні будь якій дії система буде використовувати базові частини архітектури – контролер, модель, взаємодіючи з БД, та вид.

Окрім раніше описаних базових частин MVC парадигми, застосунок також має такі внутрішні елементи:

- Вхідні скрипти: це PHP скрипти, котрі доступні напряму кінцевому користувачу застосунку. Вони відповідають за запуск і обробку вхідних запитів.

- Об'єкт застосунку: це глобально доступний об'єкт, котрий здійснює коректну роботу різноманітних компонентів програми і їх координацію для обробки запиту.

- Компоненти: це об'єкти, зареєстровані у застосунку, що надають різноманітні можливості для обробки поточного запиту.
- Модулі: самодостатні пакети, котрі включають у себе повністю усі засоби для MVC. Застосунок може бути організований за допомогою декількох модулів.
- Фільтри: код, що має бути виконаний до і після обробки запиту контролером.
- Віджети: це об'єкти, що можуть включати у себе частини представлень. Можуть містити в собі різноманітну логіку і бути використані у видах.

3.3 Розробка функціональної схеми системи

Функціональна схема – це схема, яка роз'яснює різноманітні процеси, що відбуваються у певних функціональних ланцюгах програмного забезпечення чи у програмному забезпеченні в цілому.

Функціональна схема містить інформацію про способи реалізації функцій програмою. За нею можна визначити, як здійснюються різноманітні дії і які для цього необхідні функціональні елементи. Кожен функціональний елемент містить лише ті входи і виходи, які необхідні для його коректної роботи. Дана схема розробляється на основі структурної схеми для кожного блоку, в результаті з окремих функціональних елементів складається загальна функціональна схема програмного забезпечення.

Таким чином, визначивши структуру майбутньої системи та функції, які будуть виконувати її складові, отримаємо всі необхідні дані для побудови функціональної схеми системи, яка представлена на рисунку 3.2 .

					ВКРМ-122.24.0019.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		24

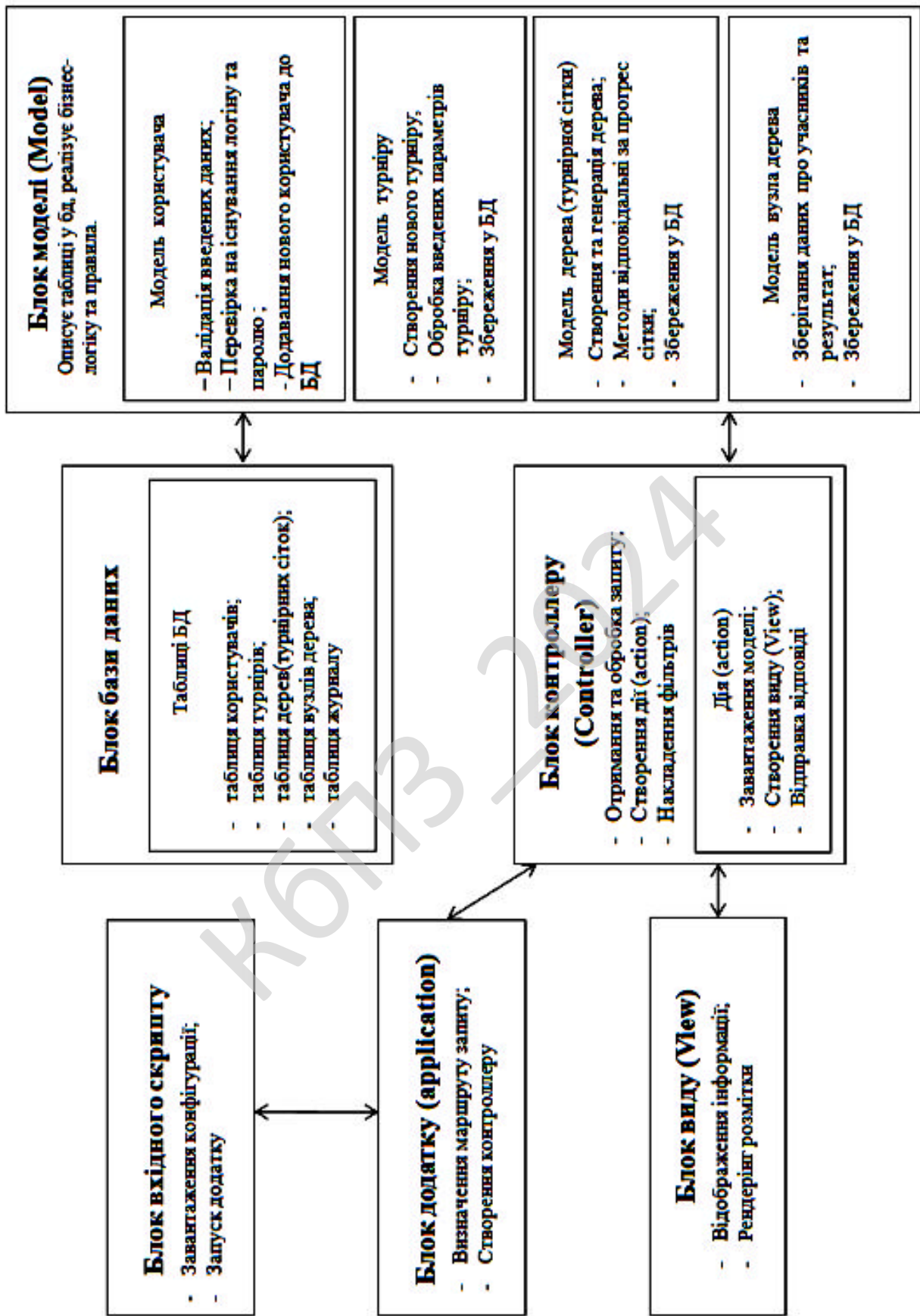


Рисунок 3.2 – Функціональна схема системи

Виходячи з наведеної схеми, життєвий цикл запиту користувача виглядатиме наступним чином:

1. Користувач створює запит до вхідного скрипту у `web/index.php`.
2. Вхідний скрипт завантажує конфігурацію та створює екземпляр класу застосунку для обробки запиту.
3. Об'єкт застосунку визначає маршрут, що був запитаний користувачем, за допомогою компонента `request`.
4. Об'єкт застосунку екземпляр контролера для обробки запиту.
5. Контролер створює екземпляр дії і виконує фільтри для нього.
6. При невдалому виконанні будь-якого фільтру, дія не буде виконуватись.
7. При успішному виконанні фільтрів виконується дія.
8. Дія завантажує модель даних, можливо, і з бази даних.
9. Дія генерує та відображає вигляд, надає йому ресурси (зображення, js та css файли) і передає йому об'єкт моделі даних.
10. Результат згенерованого вигляду передається у компонент `response`.
11. Компонент `response` відсилає готові дані користувачу.

Наприклад користувач запитує головну сторінку. Запит відбудеться через вхідний скрипт, котрий завантажить конфігурацію, підключить необхідні файли та створить екземпляр класу застосунку:

```
defined('YII_DEBUG') or define('YII_DEBUG', true);
defined('YII_ENV') or define('YII_ENV', 'dev');
require(__DIR__ . '/../vendor/autoload.php');
require(__DIR__ . '/../vendor/yiisoft/yii2/Yii.php');
$config = require(__DIR__ . '/../config/web.php');
(new yii\web\Application($config))->run();
```

Далі додаток парсером компоненту `request` визначить маршрут запиту, створить екземпляр визначеного класу `SiteController`, який виконає фільтри та передасть управління методу `actionIndex`. Призначенням даного методу є лише відображення головної сторінки, тому запитів до моделей і бази даних не відбувається. Метод генерує вид `index.php`, котрий лежить у відповідному до

					ВКРМ-122.24.0019.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

імені контролера та дії каталогу, та віддає його у компонент response, що надсилає готову сторінку у відповідь користувачу. Код методу виглядає наступним чином:

```
public function actionIndex()
{
    return $this->render('index');
}
```

3.4 Діаграма процесів, які відбуваються в системі

На діаграмі процесів, яка зображена на рисунку 3.3, вказано почерговість виконання дій програми та їх взаємодія між собою.

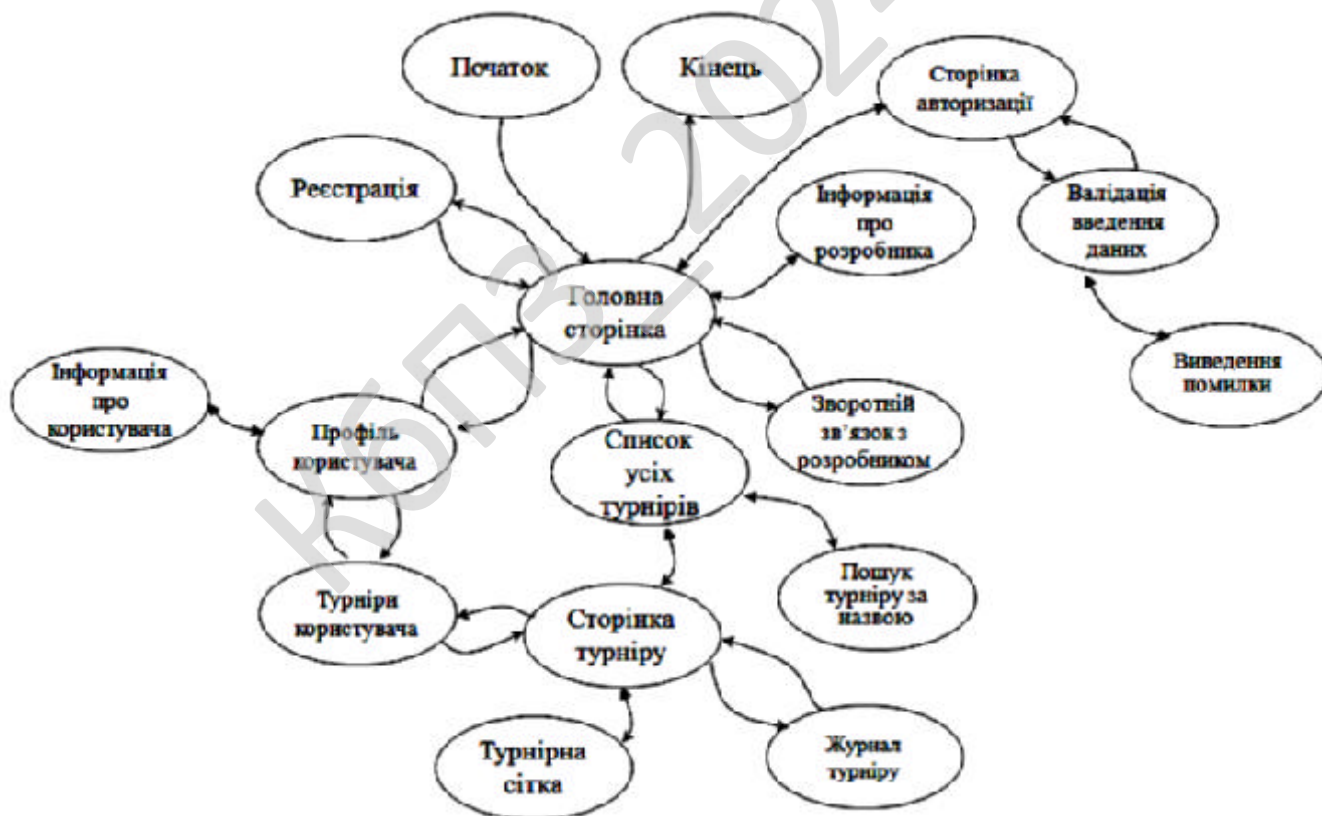


Рисунок 3.3 – Діаграма процесів системи

Потрапивши на головну сторінку, користувач може отримати список усіх існуючих у системі турнірів, переглянути окремо сторінку обраного, або

знайденого через пошук за назвою турніру, інформацію про нього, турнірну сітку, журнал турніру. Для можливості створення свого турніру користувач має пройти автентифікацію, якщо він попередньо зареєстрований, або зареєструватись, якщо ні. Після перевірки введених даних, у випадку якщо все вірно, користувача буде допущено в систему під своїм логіном. Якщо ж введені дані є невірними, буде виведено повідомлення про помилку. Після того як користувач автентифікується, він може створити свій власний турнір та адмініструвати його.

Крім того, завдяки розбиттю представлень на шаблони з метою повторного використання коду, посилання на зв'язок з автором та інформацію про проект, доступні з будь якої сторінки сервісу. Теж саме стосується і посилань на сторінки автентифікації чи вихід з системи.

Таким чином, розглянувши опис системи, структурну, функціональну схеми системи, та діаграму взаємодії процесів перейдемо до опису блок-схем основної програми, та підпрограм, які використовуються, для реалізації системи.

КБПЗ - 2024

					ВКРМ-122.24.0019.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		28


```

<?php
return [
    'class' => 'yii\db\Connection',
    'dsn' => 'mysql:host=localhost;dbname=tcr',
    'username' => 'root',
    'password' => '',
    'charset' => 'utf8',
    'tablePrefix' => 'tcr_'
];
?>

```

Основною таблицею у системі є “tcr_tournament”, що відображає сутність турнір. Вона має такі поля як:

- id: унікальний ідентифікатор;
- title: Заголовок, назва турніру;
- info: Поле для інформації про турнір;
- participants: список учасників турніру;
- three_id: зв'язок з таблицею дерева, що формує турнірну сітку;
- owner_id: зв'язок з таблицею користувачів, відображає хто створив цей турнір;
- type: тип турніру, який може бути single elimination або double elimination;
- created: дата створення турніру.

Таблиця “tcr_user” відображає користувача. Вона містить у собі такі поля:

- id: унікальний ідентифікатор;
- name: ім'я, нік або назва організації зареєстрованої на сайті;
- login: логін користувача;
- pass: пароль від аккаунту.

Логін та пароль є обов'язковими полями, що контролюється методами rules у моделях, зв'язаних з користувачем. Нижче наведено код цих методів з моделей User та Signup.

```

public function rules()
{

```

					ВКРМ-122.24.0019.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		30

```

return [
    [['login', 'pass'], 'required'],
    [['login', 'pass'], 'string', 'max' => 255],
];
}

public function rules()
{
return [
    [['login','password'],'required'],
    ['login','email'],
    ['login','unique','targetClass'=>'app\models\User'],
    ['password','string','min'=>2,'max'=>10]
];
}
}

```

Рисунок 4.2 – Сторінка автентифікації

Логін користувача має бути імейлом. Пароль під час реєстрації шифрується методом Secure Hash Algorithm 1(SHA-1) і зберігається у базі даних у

					ВКРМ-122.24.0019.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		31

зашифрованому вигляді задля безпеки. Під час автентифікації, за яку відповідна модель Login, користувач має ввести обов'язкові поля логіну та паролю. Правила валідації даних із моделі будуть перевірятись не тільки на стороні сервера, але і на стороні клієнта. Фреймворк трансформує задані моделлю правила у відповідний JavaScript код при створенні відповідної форми у представленні. Скріншот сторінки автентифікації наведено далі.

Далі на рисунку 4.3 наведено блок-схему головного функціонального модуля програми, який описує основні стадії таких операцій як автентифікація, реєстрація та створення нового турніру.

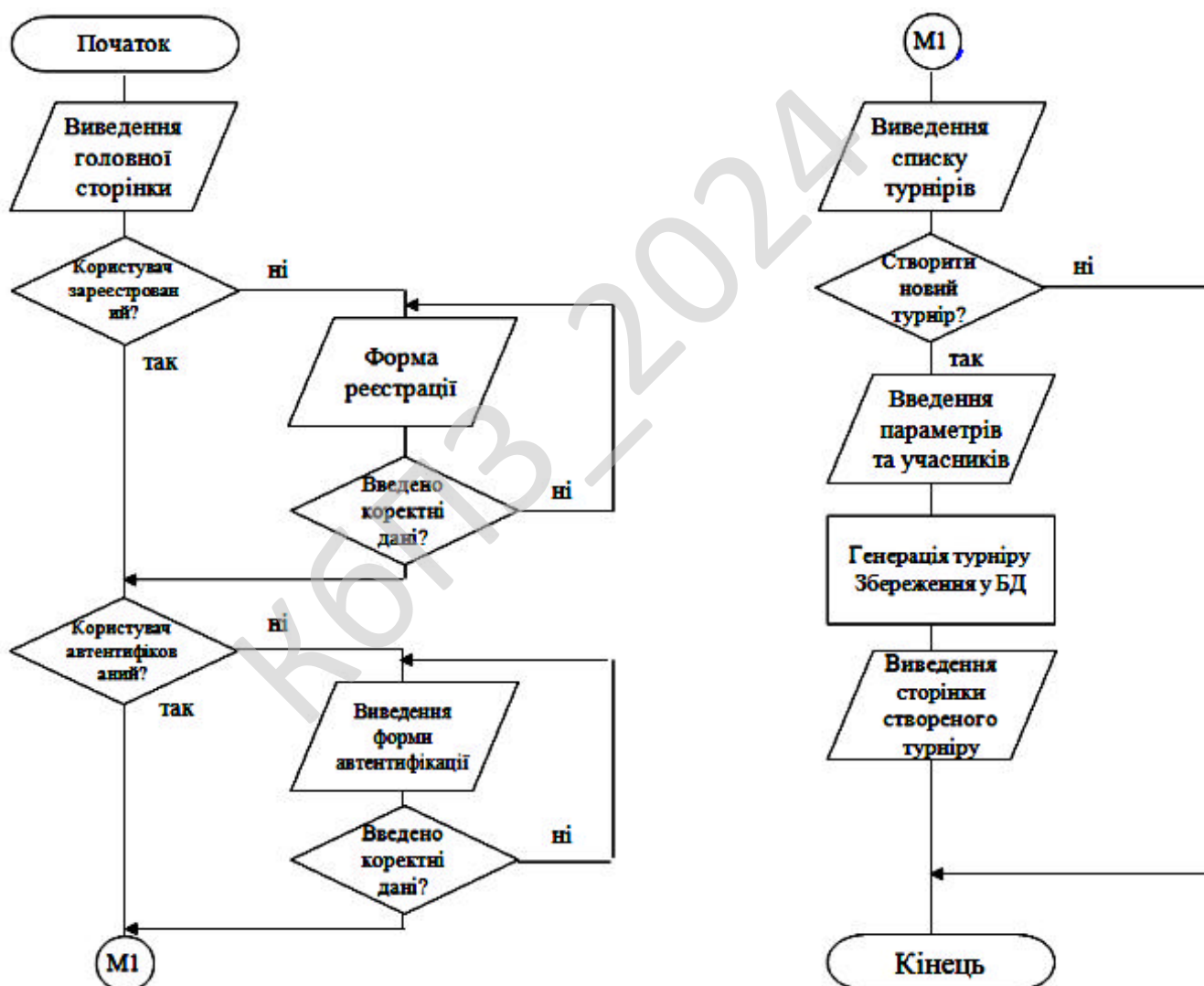


Рисунок 4.3 – Блок-схема основної програми

структури даних досить очевидний: якщо повернути будь-яку турнірну сітку(фінал якої знаходиться у правій частині) проти годинникової стрілки, то можна отримати схему двійкового дерева. Використання алгоритму, оснований на двійковому дереві, замінило собою попередньо спроектоване використання великих масивів даних та об'єктів, що значно зменшило об'єм зайвої інформації та дало можливість більш наглядно розуміти алгоритм графічно. Якщо турнірну сітку представляє собою дерево і за нього в програмі відповідає клас Three, то кожна вершина цього дерева є об'єктом класу Node. Клас Node містить таку основну інформацію як:

- учасник з лівого боку;
- учасник з правого боку;
- рахунок лівого учасника;
- рахунок правого учасника;
- ідентифікатор дерева, до якого належить вузол;
- індекс у масиві;
- тип вузла: пасивний чи активний.

Обравши для використання таку структуру даних як дерево, слід також обрати і спосіб її зберігання у програмі та базі даних. Класичним варіантом зберігання структури двійкового дерева у пам'яті є додання до об'єкта вузла посилання на його нащадків та на предка, тоді маючи вузол-вершину можна рекурсивно обійти усе дерево. Але для реалізації у програмі було обрано інший спосіб – зберігання дерева у масиві. Щоб мати таку можливість, потрібно щоб двійкове дерево було повним. Таким може називатись дерево у якого всі рівні (окрім можливо останнього) заповнені без пропусків. Подібний спосіб може знадобитись при реалізації таких структур даних як heap (купа), для дерева відрізків та інше. Дерево у масиві зберігається порівнево, починаючи з кореня, який матиме індекс 0. Елементи наступних рівнів зберігаються у масиві зліва направо, відповідно лівий дочірній елемент кореня матиме індекс 1, а правий – 2. Приклад такого дерева представлений далі на рисунку 4.4.

					ВКРМ-122.24.0019.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		34

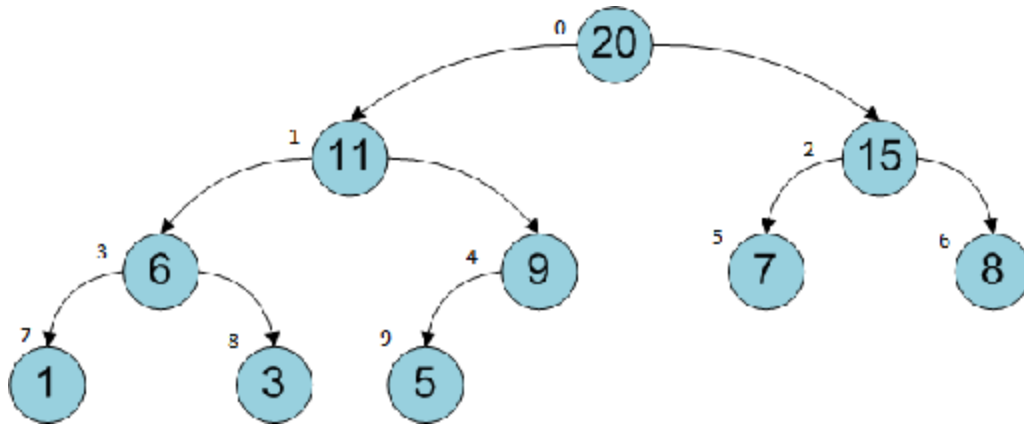


Рисунок 4.4 – Приклад повного двійкового дерева

Звільнившись від необхідності зберігання посилання на предка та нащадків, кожен вузол тепер може зберігати тільки свій індекс у масиві. Щоб із будь якого вузла дістатись до предка, слід звернутись по індексу, визначеному цілим значенням результату такого обчислення:

$$(i-1)/2, \quad (4.1)$$

де i – індекс поточного вузла у масиві. Щоб дістатись до лівого дочірнього вузла, слід, відповідно, до поточного індексу додати одиницю та помножити на два:

$$(i*2)+1, \quad (4.2)$$

А правий дочірній елемент доступний за індексом:

$$(i*2)+2, \quad (4.3)$$

Для визначення того, чи є вузол лівим чи правим дочірнім слід перевірити на парність його індекс. Лівий вузол завжди буде непарним.

Але при використанні повних двійкових дерев у якості турнірної сітки, виникає проблема при його застосуванні до турнірної сітки з не ідеальною (відмінною від степені двійки, nonPerfect) кількістю учасників, коли дерево з неповним останнім рівнем не відповідає правилам формування таких турнірних сіток. Останній рівень дерева буде заповнюватись зліва направо, що створить нерівномірність розподілення гравців та відсутність паралельності проведення ігор. Коректний варіант такої турнірної сітки наведений на рисунку нижче.



Рисунок 4.5 – Приклад турнірної сітки з десятима учасниками

Як видно, останній рівень дерева заповнений не впорядковано з однієї сторони у іншу, а рівномірно розподілено між вузлами наступного рівня, що забезпечує правильне розподілення учасників, які гратимуть у найближчому раунді і учасників, які очікують переможця.

Для вирішення даної проблеми було прийнято рішення ввести поняття активного та пасивного вузла. Дерево в будь-якому разі буде створюватись повним до кінця, із заповненням усіх рівнів для збереження цілісності структури дерева. Але усі вузли останнього рівня, що не заповнені гравцями а лише підтримують цілісність структури, матимуть статус пасивних. Вони зберігатимуться у масиві разом з іншими, проте не будуть відображені на схему турніру.

Окрім того, для досягнення виконання усіх правил формування подібних (nonPerfect) турнірних сіток, дерево формується у три етапи. Усе це відбувається у методі createThree класу Three, блок-схема якого наведена далі.

У представленій блок-схемі є деякі службові змінні, такі як:

- n – кількість учасників турніру;
- min – нижня межа учасників при nonPerfect сітці;
- max – верхня межа учасників при nonPerfect сітці;

– $pLeft$ – кількість вузлів дерева , що залишилось перетворити після створення дерева відповідного min , при створенні $nonPerfect$ сітки. При ініціалізації дорівнює різниці n і min ;

– $PMAX$ – константа, що характеризує найбільш допустиму кількість учасників для створення сітки.

КБПЗ – 2024

					VKPM-122.24.0019.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		37

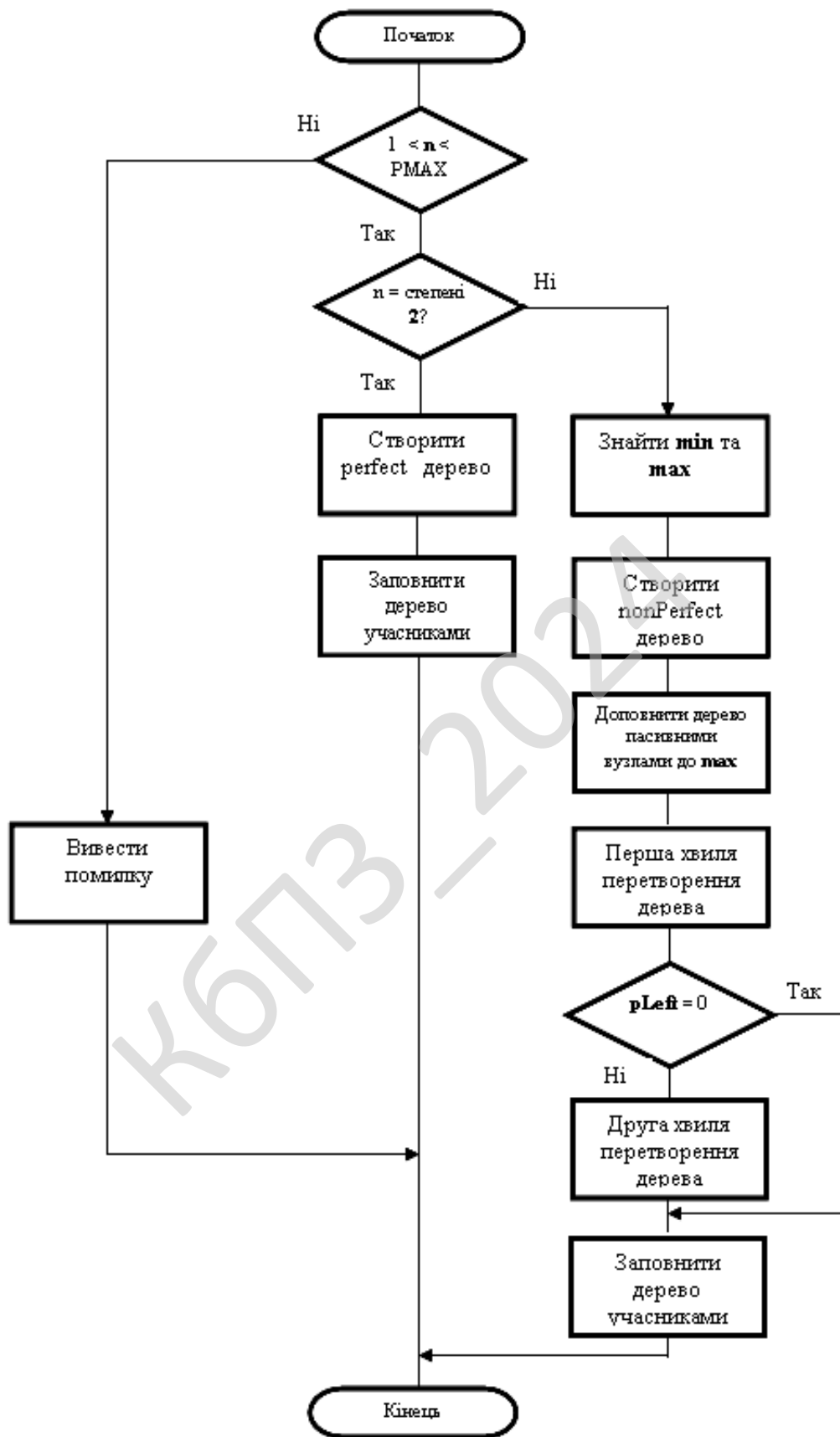


Рисунок 4.6 – Блок-схема функції createThree

Якщо ж цикл дійшов до вже активного вузла(тобто потрапив на рівень вище) або ж змінна `pLeft` стала дорівнювати нулю, то перша хвиля перетворення закінчена. Йде перевірка чи залишилися ще вузли, які потрібно активувати. Якщо так, тоді стартує друга хвиля, котра починає активувати вузли через один, починаючи з передостаннього у масиві, тобто ті, які не потрапили у першу хвилю. Виконання зупиниться коли цикл потрапить на вже активний вузол(що рівнем вище) або лічильник `pLeft` стане нулем. Після завершення другої хвилі перетворення, або лише першої, якщо її було достатньо, буде викликаний метод `fillNPThree`. Цей метод буде заповнювати вузли дерева учасниками, якщо вузол є активним. Якщо індекс вузла буде дорівнювати $(\text{min} - 2)$, тобто здійсниться перехід на рівень вище, то метод почне перевіряти дочірні вузли поточного на активність. Якщо обидва дочірні вузли активні, це значить, що до поточного вузла перейде двоє переможців з дочірніх і тому його заповнювати не потрібно. Якщо активний лише один, то метод заповнить учасником відповідне вільне місце, а місце на яке потім потрапить переможець, не чіпатиме. Якщо обидва дочірні вузли поточного вузла є пасивними, тоді він буде заповнений двома учасниками. Після цього метод `createThree` завершить роботу, а поле `three` класу `Three` міститиме у собі масив, заповнений готовим деревом, тобто турнірною сіткою.

Код описаної функції `createThree` та функції першої хвилі перетворення `firstWave` наведено нижче:

```
public function createThree($pArray) {
    $n = count($pArray);
    if($n < 2 || $n > $PMAX){
        //error
    }

    if(isPerfect($n)){
        createPerfectThree($pArray);
        fillPerfectThree();
    }else{
        $min = findMin($n);
        $max = findMax($n);
    }
}
```

					ВКРМ-122.24.0019.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		40

```

        createPerfectThree($min);
        addNodesToMax($max);
        $pLeft = $n - $min;
        firstWave($pLeft);

        if($pLeft){
            secondWave($pLeft);
        }

        fillNPThree($pArray);
    }
}

private function firstWave(&$pLeft){
    $three = $this->three;
    $i = count($three);
    $runFlag = true;
    while($pLeft && ($i >= 1) && $runFlag){
        $node = $three[$i];
        if($node->isActive()){
            $runFlag = false;
            continue;
        }
        $node->setActive();
    }
    $pLeft--;
    $i -= 2;
}
}

```

Маючи готовий до проведення турнір, його можна розпочати. Активні вузли, що мають учасників будуть, доступні до редагування результатів. Після введення результату матчу у певний вузол, буде викликаний метод `setMatchResult` класу `Three`, що згідно введеним результатам визначить переможця та запише його як учасника у вузол наступного раунду.

Усі дії CRUD (`create`, `read`, `update`, `delete`) відбуваються методами класу `Active record`, функціональність якого була описана у минулих розділах. Записи у БД відбуваються після формування дерева та запису будь якого нового прогресу у вже запущеній турнірній сітці.

					ВКРМ-122.24.0019.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		41

2. Якщо молодший значущий біт $x_0 = 1$, то $x_0 = x_0 \oplus C$. Якщо молодший значущий байт $x_3 = 0$, то $x_3 = x_3 \oplus C$.

3. $x_i = x_{i-1} \oplus x_i \oplus x_{i+1}$ для $i = 0..3$.

Всі операції з індексами виконуються по модулю 4. Операція множення на кроці 1 виконується по модулі $2^{32}-1$. Спеціальний випадок для даного алгоритму: якщо другий операнд дорівнює $2^{32}-1$, результат теж дорівнює $2^{32}-1$. В алгоритмі використовуються наступні константи:

$$C = 2\text{aaaaaaa}, c_0 = 025\text{f1cdb}, c_1 = 2 * c_0, c_2 = 2^3 * c_0, c_3 = 2^7 * c_0.$$

Константа C – «найпростіша» константа без кругової симетрії, високою трійковою вагою й нульовим молодшим значущим бітом. У константи c_0 є інші особливі характеристики. Константи c_1 , c_2 і c_3 – зрушені версії c_0 , і служать для запобігання атак, заснованих на симетрії.

Розшифрування виконується у зворотному порядку, Етапи 2 і 3 інверсні їм самим. На етапі 1 замість c_i використовується c_i^{-1} . Значення $c_0^{-1} = 0\text{dad4694}$.

КБПЗ-2024

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

Відповідно до технічного завдання, в процесі виконання магістерського проекту було розроблене програмне забезпечення генерування складних динамічних структур на прикладі турнірних таблиць.

Для успішного впровадження та інтеграції розробленого ПЗ в існуючу апаратну систему вона повинна задовольняти такі мінімальні вимоги на стороні сервера:

- підтримка PHP не нижче версії 5.4;
- підтримка серверу Apache або nginx ;
- ОС Linux, Windows або інші, що підтримують PHP 5.4+;
- обсяг оперативної пам'яті пристрою повинен становити щонайменше 512 Мб.

І також такі мінімальні вимоги на стороні клієнта:

- встановлений браузер сучасний (Google Chrome, Opera, Mozilla Firefox);
- Windows XP або більш пізніша, Mac OS X 10.6+, Ubuntu 10.04+;
- обсяг оперативної пам'яті пристрою повинен становити щонайменше 512 Мб;
- Процесор Intel Pentium 4 / Athlon 64 або більш пізніх версій з підтримкою SSE2

Програмне забезпечення, що було розроблене в процесі виконання магістерського проекту, представляє собою веб-сервіс, архітектура і структура якого побудовані на базі PHP фреймворку Yii2 та детально описані в попередніх розділах пояснювальної записки. Щоб здійснити ефективне впровадження веб-сервісу в роботу була розроблена інструкція користувача з відповідними поясненнями з приводу експлуатації системи. Методичне забезпечення (МЕТОЗ)

					ВКРМ-122.24.0019.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		44

по використанню програми було розроблено з розрахунком на користувача, що має мінімальні знання та навички роботи з програмним забезпеченням подібного класу та потребує простого у користуванні інтерфейсу.

Згідно з інструкцією користувача, для того, щоб створити обліковий запис необхідно вказати адресу електронної пошти та пароль у відповідній формі на сторінці реєстрації.

Tournament CReator Головна Про проект Зв'язатись Увійти

Реєстрація

Логін

Пароль

Підтвердити

©Tournament CReator

Рисунок 5.1 – Сторінка реєстрації користувача

Реєстрація не є обов'язковою для тих, хто не збирається створювати турніри; переглядати всі існуючі турніри та слідкувати за ходом їх розвитку можна і в якості анонімного користувача. Але інші функції неможливо зробити без реєстрації, наприклад створення турніру, його редагування та інше.

Для створення турніру зареєстрований користувач має натиснути відповідну кнопку, після чого обрати різноманітні параметри, ввести список учасників та розпочати турнір як це зображено на рисунку 5.2.

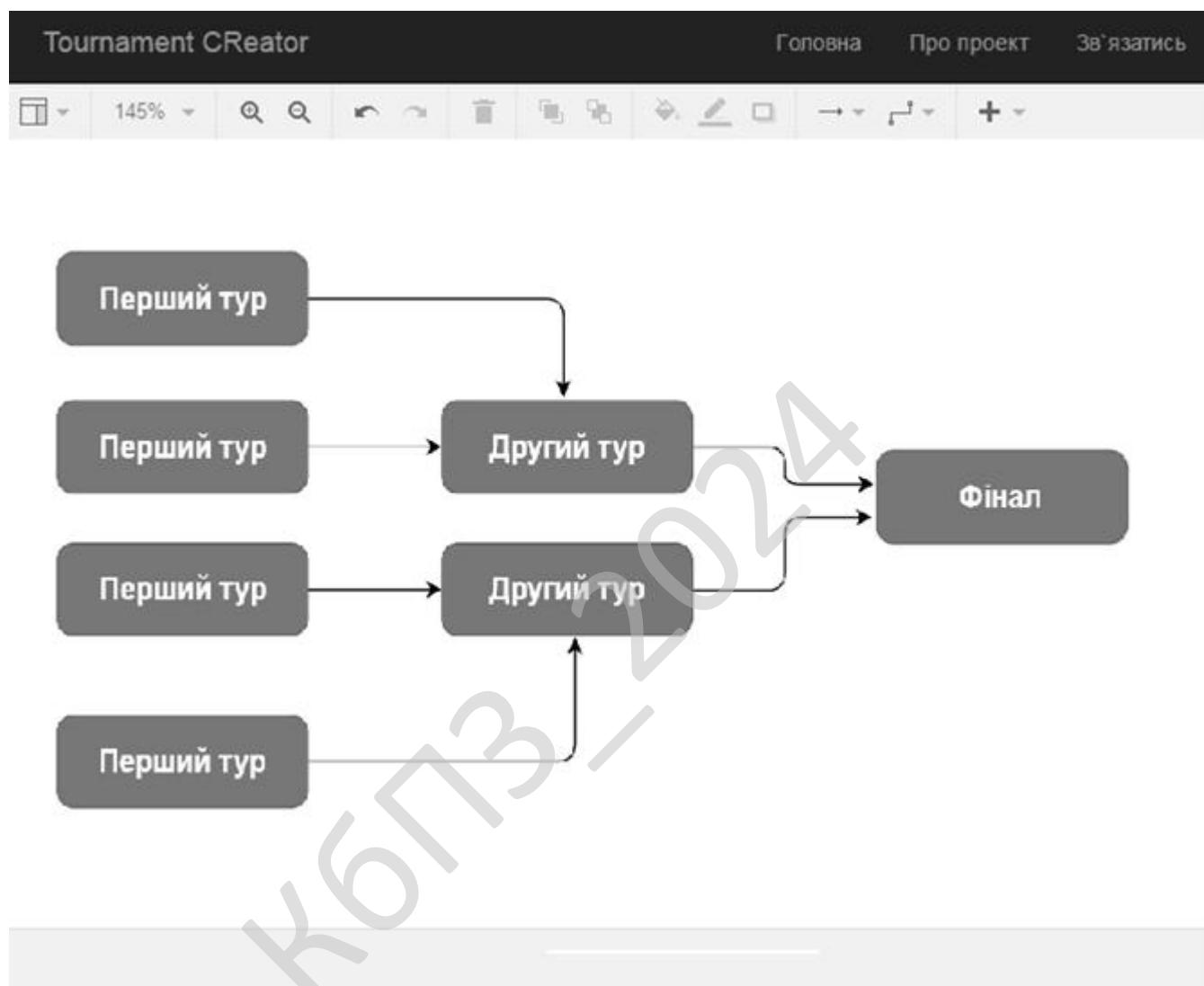


Рисунок 5.2 – Сторінка створення турніру

Перейшовши до вкладки «турнірна сітка», можна побачити згенеровану сітку турніру, готову до внесення ігрових даних. Створення сітки, журналювання дій, рух гравців по сітці відбуваються автоматично. Користувач зобов'язаний лише ввести параметри на початку та вносити результати у відповідні вузли упродовж турніру.

Отже, вказані мінімальні вимоги системи та розроблена інструкція користувача дають змогу легко та зручно користуватись створеним програмним забезпеченням. Авторське право показане на рисунку 5.3.

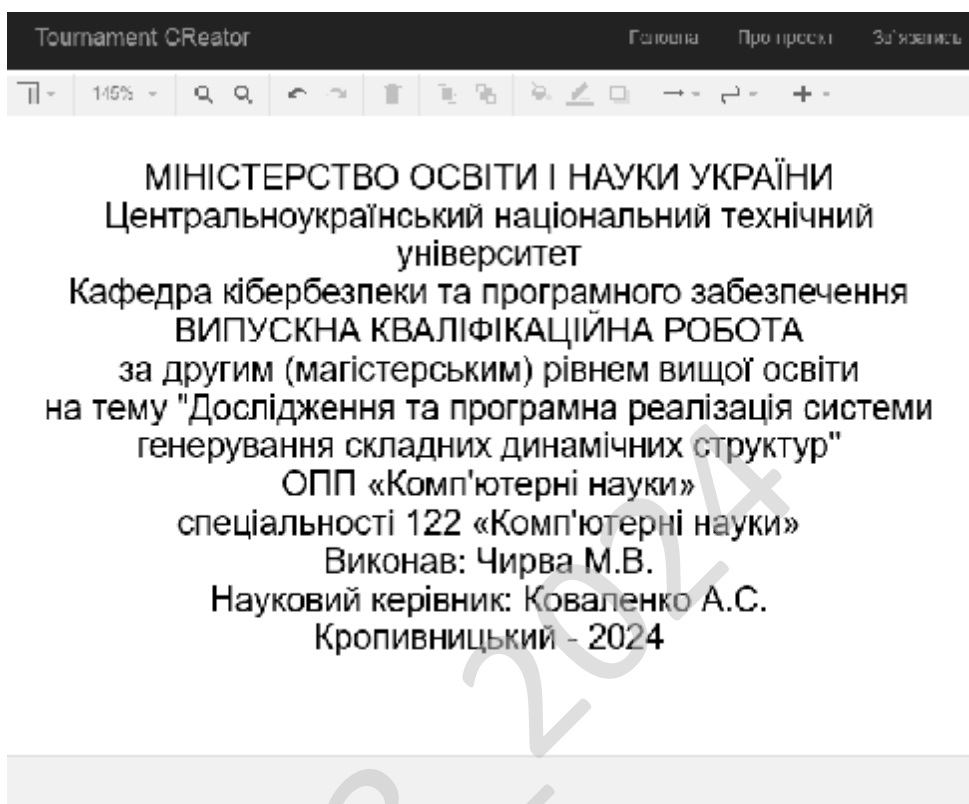


Рисунок 5.3 – Вікно розробника

Під час роботи над програмою було проведено тестування, тобто технічне дослідження, призначене для виявлення інформації про якість продукту відносно контексту, в якому воно має використовуватись.

					ВКРМ-122.24.0019.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		47

6 НАУКОВА НОВИЗНА

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи генерування складних динамічних структур.

Метою розробки є дослідження та програмна реалізація системи генерування складних динамічних структур.

Об'єктом дослідження є процес генерування складних динамічних структур.

Предметом дослідження є методи генерування складних динамічних структур.

Методи дослідження базуються на методах теорії складних систем, методах математичної статистики, методах розробки програмного забезпечення.

Наукова новизна отриманих результатів. У процесі рішення завдань, обумовлених цілями дослідження, отримані наступні результати:

- Удосконалено метод генерування складних динамічних структур.
- Розроблено вітчизняний продукт генерування складних динамічних структур, який має більш широкі можливості, на відміну від існуючих аналогів.

					VKPM-122.24.0019.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		48

7 МАРКЕТИНГОВЕ ТА ЕКОНОМІЧНЕ ОБГРУНТУВАННЯ ІТ-ПРОЄКТУ

7.1 Визначення цільової аудиторії кінцевого готового продукту

Результати дослідження та програмної реалізації системи генерування складних динамічних структур можуть бути цікаві різним групам та професіоналам(рисунок 7.1).

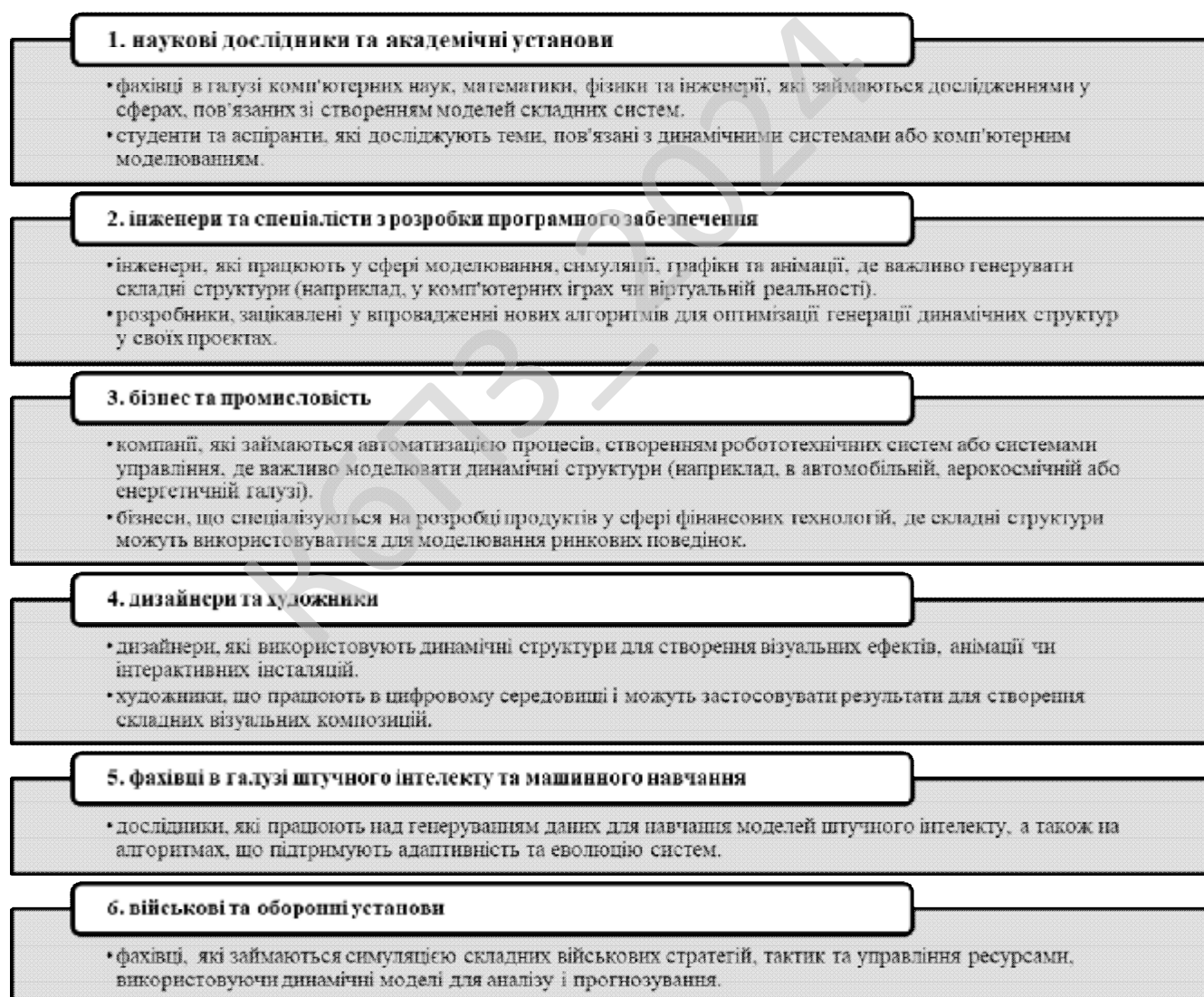


Рисунок 7.1 – Цільова аудиторія

Ці результати можуть стати основою для нових досліджень, розробок та інновацій у різних галузях, сприяючи покращенню процесів, оптимізації рішень та впровадженню нових технологій.

7.2 Оцінка привабливості шляхом застосування методів експертних оцінок

Оцінка привабливості проєкту програмної реалізації системи генерування складних динамічних структур може бути проведена за допомогою методів експертних оцінок.

Метою оцінки є оцінити привабливість проєкту для залучення інвестицій та визначення його конкурентоспроможності на ринку.

Починається сам процес з визначення основних критеріїв, які впливають на привабливість проєкту (рисунок 7.2).

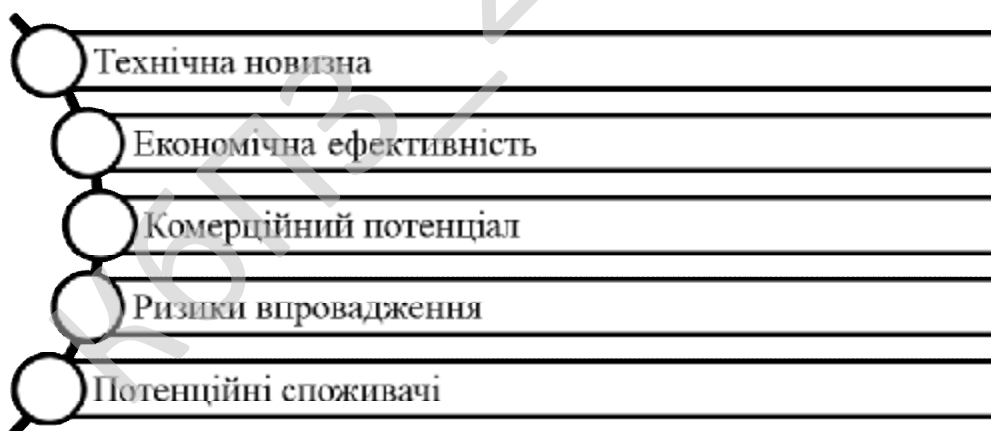


Рисунок 7.2 – Основні критерії оцінки привабливості

Наступним етапом є формування групи експертів, до якої можуть входити: спеціалісти в галузі програмування та розробки програмного забезпечення, економісти, які можуть оцінити фінансові аспекти, представники потенційних

користувачів, які можуть надати зворотний зв'язок щодо практичного застосування системи.

Далі відбувається сам процес виставлення балів експертами з використання методу бальної оцінки (наприклад, від 1 до 5, де 1 – дуже низька оцінка, а 5 – дуже висока) для оцінки кожного критерію. Експерти оцінюють кожен з критеріїв за бальною системою. Результати зводяться до таблиці 7.1.

Таблиця 7.1 – Зведені результати експертних оцінок

Критерій	Експерт			Середня оцінка
	1	2	3	
Технічна новизна	4	5	4	$(4 + 5 + 4)/3 = 4.33$
Економічна ефективність	3	4	4	$(3 + 4 + 4)/3 = 3.67$
Комерційний потенціал	5	4	5	$(5 + 4 + 5)/3 = 4.67$
Ризики впровадження	3	3	2	$(3 + 3 + 2)/3 = 2.67$
Потенційні споживачі	4	5	4	$(4 + 5 + 4)/3 = 4.33$

Обчислюємо загальну оцінку проєкту, зважуючи на кожен критерій відповідно до його важливості. Ваги критеріїв розподілені наступним чином: технічна новизна – 25%, економічна ефективність – 20%, комерційний потенціал – 30%, ризики впровадження – 15%, потенційні споживачі – 10%.

Тоді загальна оцінка буде: $= (4.33 \times 0.25) + (3.67 \times 0.20) + (4.67 \times 0.30) + (2.67 \times 0.15) + (4.33 \times 0.10) = 4.051$.

На основі отриманої загальної оцінки експерти можуть дати рекомендації щодо доцільності подальшої розробки проєкту, визначити сильні і слабкі сторони, а також запропонувати шляхи покращення, щоб підвищити привабливість проєкту.

7.3 Вибір методу оцінки вартості ПЗ

Для оцінки вартості програмної реалізації системи генерування складних динамічних структур можна використовувати кілька методів, залежно від специфіки проєкту, його масштабів і доступності даних. Рекомендовані методи представлено на рисунку 7.3.

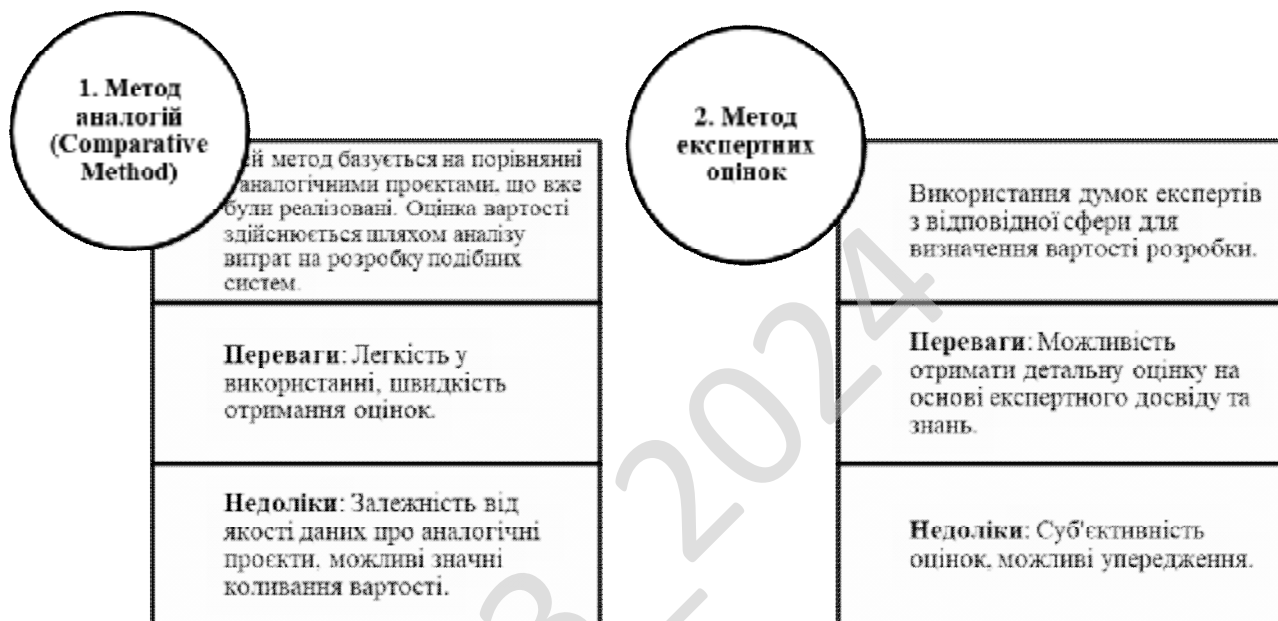


Рисунок 7.3 – Оптимальні методи оцінки вартості ПЗ

Загалом, найкращим підходом буде комбінування методів, що дозволить отримати більш точну та надійну оцінку вартості розробки системи генерування складних динамічних структур.

7.4 Розрахунок економічної ефективності від впровадження реалізованого ПЗ як фактору його привабливості

Економічна ефективність від впровадження системи генерування складних динамічних структур може бути оцінена за кількома параметрами, які демонструють позитивний вплив на бізнес чи організацію (рис. 7.4).

7.5 Пропозиція алгоритму просування проекту розробки ПЗ

Алгоритм просування проекту програмної реалізації системи генерування складних динамічних структур подано на рисунку 7.5.

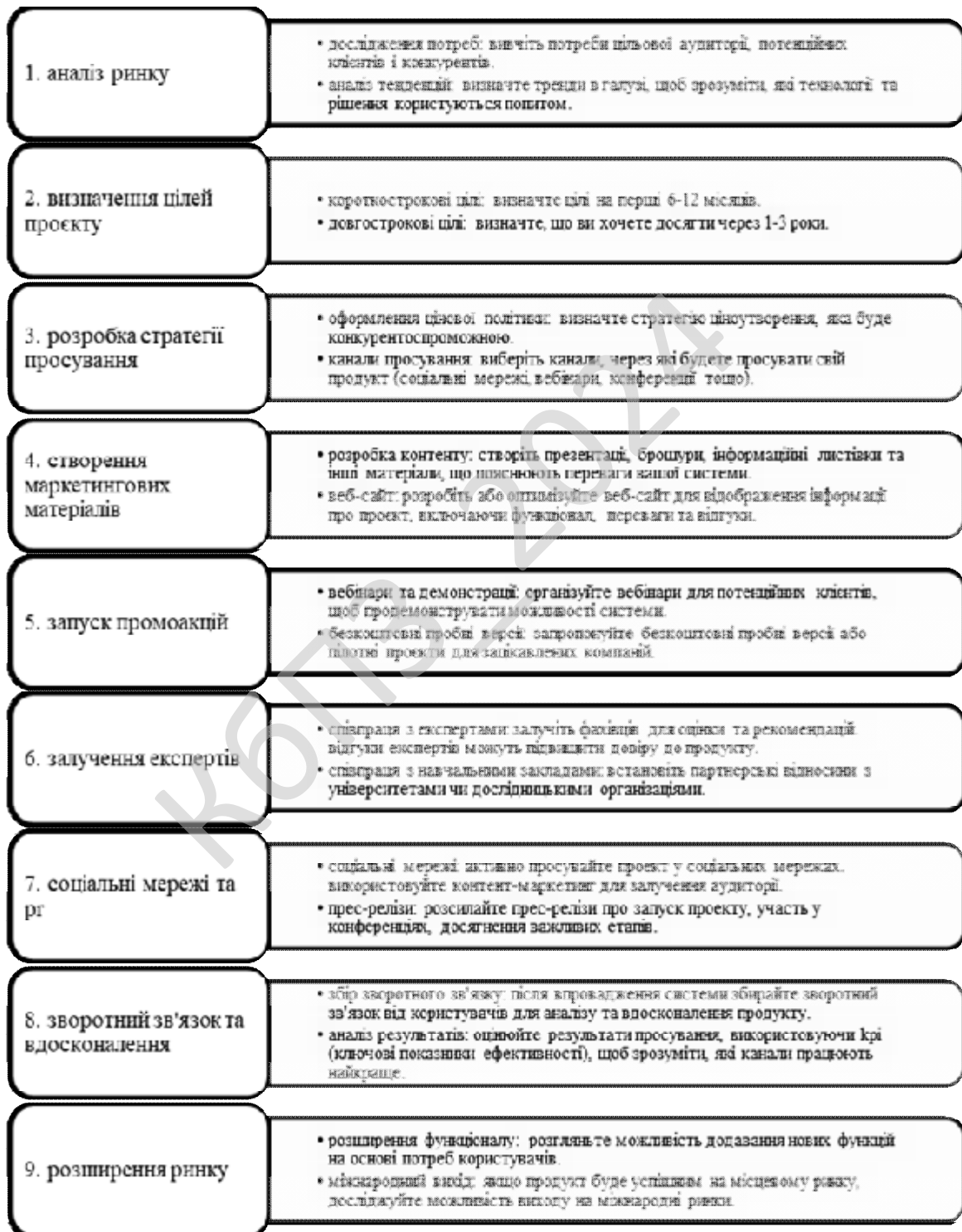


Рисунок 7.5 – Алгоритм просування проекту

Цей алгоритм просування допоможе вам систематично підходити до реалізації проєкту, визначаючи чіткі етапи та дії для досягнення успіху. Спостереження за ринком і гнучкість у стратегії просування дозволять адаптувати проєкт до вимог користувачів і швидко реагувати на зміни в середовищі.

7.6 Оптимізація каналів збуту та шляхів реалізації ПЗ

Для оптимізації каналів збуту та шляхів реалізації проєкту програмної реалізації системи генерування складних динамічних структур можна застосувати кілька стратегій і заходів (рисунок 7.6).

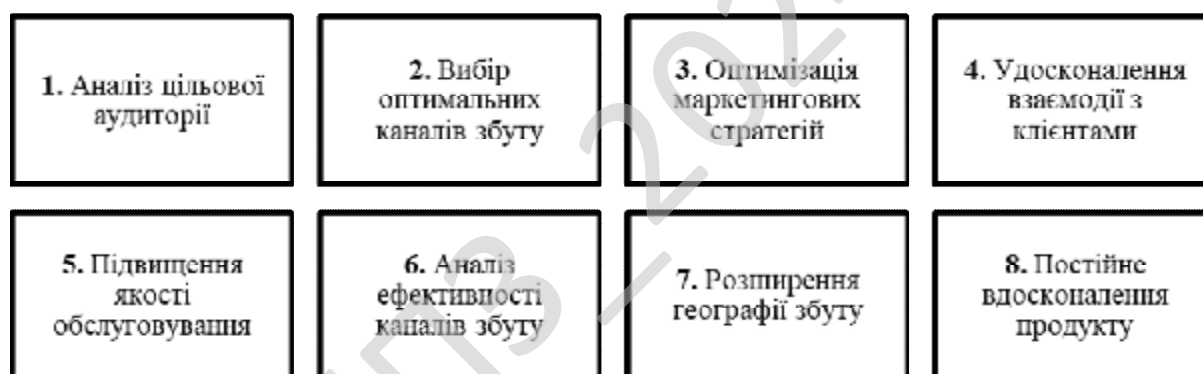


Рисунок 7.6 – Оптимізація каналів збуту та шляхів реалізації

Впровадження цих заходів дозволить оптимізувати канали збуту та реалізації проєкту, що, у свою чергу, призведе до збільшення продажів, підвищення задоволеності клієнтів та покращення фінансових показників компанії.

7.7 Визначення ключових факторів успіху конкретного проєкту

Ключові фактори успіху проєкту програмної реалізації системи генерування складних динамічних структур можуть включати в себе складові елементи, представлені схематично на рисунку 7.7.

Впровадження цих факторів у проєкт допоможе забезпечити його успіх та досягнення поставлених цілей.

КБПЗ_2024

					ВКРМ-122.24.0019.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		57

8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

8.1 Вступ

Сучасний розвиток технічного та технологічного стану виробництва передбачає постійну автоматизацію та оптимізацію виробничих процесів. Комп'ютер – невід'ємна складова сучасного життя. За допомогою обчислювальної техніки вирішують складні робочі задачі, ведуться наукові дослідження, створюються архітектурні креслення і твори мистецтва. Сьогодні, напевно, важко уявити компанію, господарська діяльність в якій здійснювалася би без використання комп'ютерної техніки.

Незважаючи на видиму безпеку та розвитку сучасних технологій, при роботі за комп'ютером є ряд чинників, які можуть вплинути на здоров'я людини. Через масовий характер робіт, що виконуються працівниками за допомогою комп'ютера, законодавством України чітко врегульовано норми та вимоги до використання комп'ютерної техніки на підприємстві, безпосередньо й охорона праці на підприємстві при роботі за комп'ютером.

Законом України “Про охорону праці” [1] регламентуються загальні положення державної політики в галузі охорони праці, а конкретизуються ці положення нормативно-правовими актами про охорону праці, зокрема Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями, затверджені наказом Мінсоцполітики від 14.02.2018р. № 207, зареєстровані в Міністерстві юстиції України 25 квітня 2018 р. за №508/31960 [2].

Робота з комп'ютером характеризується значною розумовою напругою і нервово-емоційним навантаженням операторів, високою напруженістю зорової роботи і достатньо великим навантаженням на м'язи рук при роботі з клавіатурою ЕОМ. Шкідливими факторами при роботі з персональним комп'ютером є неіонізуюче випромінювання промислової частоти, збільшене нервово-емоційне

					ВКРМ-122.24.0019.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		58

навантаження на оператора, збільшення навантаження на органи зору та дрібні стереостатичні рухи кінцівок.

У розділі даної магістерської роботи висвітлюються основні питання охорони праці працівників, робота яких пов'язана з роботою за комп'ютером, планування робочого приміщення, де працюють користувачі ПК; параметри мікроклімату, освітленість робочих місць та виробничих приміщень; шумові завади.

Правильна організація і раціональне устаткування робочого місця можливість ефективно і з якнайменшими витратами праці виконувати свої функції, плідно спілкуватися співробітниками і підлеглими, підтримувати високу працездатність і робочий настрій.

Велике значення має раціональна конструкція і розташування елементів робочого місця, що важливе для підтримки оптимальної робочої пози людини-оператора, а також необхідно дотримувати правильний режим праці і відпочинку.

Що стосується питання охорони праці людини необхідно вирішувати на всіх стадіях трудового процесу незалежно від виду професійної діяльності.

Забезпечення безпечних і здорових умов праці в значній мірі залежить від правильної оцінки небезпечних, шкідливих виробничих факторів. Однакові по складності зміни в організмі людини можуть бути викликані різними причинами. Це можуть бути фактори виробничого середовища, надмірне фізичне і розумове навантаження, нервово-емоційна напруга, а також різне сполучення цих причин. Робота працівників пов'язана з роботою за комп'ютером, тому актуальною є розгляд саме умов праці та стану охорони праці працівників які постійно працюють з комп'ютерною технікою.

Завдання даного розділу полягає у тому, щоб розробити якісний програмний продукт необхідно організувати безпеку на робочому місці програміста. Під час проектування безпеки робочому місці з ПК необхідно домагатися високої якості та надійності технічного забезпечення, але й створювати комфортні параметри довкілля для розробників.

					ВКРМ-122.24.0019.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		59

8.2 Характеристика умов праці програміста

Приміщення в якому проводиться розробка і дослідження програмного продукту за завданням наведені в таблиці 8.1

Таблиця 8.1 – Характеристика умов праці

Шкідливі та небезпечні фактори на робочому місці	Джерела утворення небезпек	Примітка (дані наведені для приміщення)
Електрична напруга вище 127В; Шум; Електромагнітні випромінювання; Статична електрика; Іонізація повітря; Пожежна безпека у приміщенні; Не якісне освітлення;	Кондиціонер 6 ПЕОМ Принтер Папір Світильники	Розміри приміщення (м) Довжина – 9 Ширина – 6 Висота – 3 Кількість працюючих – 7

Згідно Державними санітарними правилами і нормами ДСанПіН 3.3.2.007-98 [3] може перебувати 7 працівників. Мінімальна площа приміщення на 1 людину повинна складати не менше 6 м². Висновок – за умовами завдання це виконується. В приміщенні відсутні умови, які можуть створювати підвищену або особливо підвищену небезпеку, тому воно відноситься до класу звичайних приміщень згідно ПУЕ. Джерелом живлення є трифазна мережа напруги 380/220 В з глухо заземленою нейтралі, з частотою 50 Гц. За пожежо-вибухонебезпекою приміщення відноситься до класу В.

В таблиці 8.2 наведена загальна характеристика приміщення щодо вибухопожежо-небезпеки та важкістю робіт. Температура повітря в приміщенні визначається температурою зовнішнього повітря і тепловою енергією, що виділяється всередині приміщення.

Таблиця 8.2 – Загальна характеристика приміщення щодо вибухопожежонебезпеки та важкістю робіт

Характеристика приміщень за вибухопожежною категорією та класом зони	Загальна характеристика приміщення	Категорія за важкістю робіт згідно ГН 3.3.5-8.6.6.1 -2002
В – пожежонебезпечне клас П – П	Звичайне без ознак хімічного забруднення та нормальної вологості	1а.....до 139 Вт/м ² 1б.....до 140-174 Вт/м ² Клас умов праці – оптимальний

Джерелами теплоти в даному приміщенні є люди, електроустаткування, а також освітлювальні прилади в темний час доби. Зовнішнім джерелом надлишкового тепла є сонячна радіація у світлий час доби. Робота, виконувана в даному приміщенні, відноситься до категорії І-а. Людиною в цьому випадку виділяється до 120 ккал теплової енергії в годину. Вологість повітря в приміщенні визначається вологістю атмосферного і видихуваного людьми повітря, а також випарами з поверхні шкіри.

У таблиці 8.3 приведені оптимальні значення параметрів мікроклімату для категорії ваги робіт І-а, а також фактичні значення цих параметрів у розглянутому приміщенні. У приміщеннях з використанням обчислювальної техніки рекомендується застосування тільки оптимальних значень показників мікроклімату, тобто таких, при яких людина відчуває себе комфортно.

Таким чином, показники мікроклімату в приміщенні, загалом, відповідають установленим нормам. В холодний період року використовується індивідуальне опалення, завдяки якому дотримується температурний режим в приміщенні в залежності від температури повітря навколишнього середовища.

Таблиця 8.3 – Оптимальні і фактичні значення параметрів мікроклімату

Пора року	Оптимальні для Ia			Фактичні		
	Температура, °C	Воло- гість,%	Швидкість повітря, м/с	Температура, °C	Воло- гість%	Швидкість повітря, м/с
Холодна	22-24	40-60	0,1	22-24	40-55	0,12
Тепла	23-25	50-70	0,1	24-25	50-65	0,9

Для підтримки температури в літню пору встановлений кондиціонер, який має достатню потужність по холоду, а для підтримки вологості є зволожувач повітря.

Джерелами запиленості повітря в приміщенні є одяг людей і пил, що проникає з вулиці. З метою боротьби з пилом робляться регулярні вологі прибирання і провітрювання.

У приміщенні немає виділення шкідливих газів. Тому що в ньому не проводиться монтажних робіт, пайки чи інших робіт, при яких виділяються шкідливі гази.

Для нормалізації параметрів повітряного середовища також періодично здійснюється провітрювання приміщення і вологе прибирання. У всьому будинку діє встановлена загально обмінна витяжна вентиляція.

Особливістю роботи за дисплеєм ЕОМ є постійна й значна напруга функцій зорового аналізатора, обумовленого необхідністю розходження самосвітних об'єктів (символів, знаків тощо) при наявності відблисків на екрані, рядковій структурі екрана, мерехтіння зображення, недостатньою чіткістю об'єктів розходження. По характеру зорової роботи, робота відноситься до високої точності, розряд зорової роботи III, підрозряд г.

Рациональне освітлення приміщення сприяє кращому виконанню виробничого завдання і забезпеченню комфорту при роботі. Для забезпечення нормального освітлення застосовуються природне, однобічне, бічне і штучне

освітлення, а також сполучене, нормуються згідно ДБН В.2.5-28-2006 Природне і штучне освітлення [4]. Дані по нормах освітлення наведені в таблиці 8.4

Таблиця 8.4 – Норми освітлення

Мінімальний розмір об'єкта розрізнювання, мм	Фон	Контраст	Розряд, під розряд зорової роботи	Нормоване значення		
				Природне освітлення КПО, %	Штучне освітлення	
					Е _{мін.} лк	Тип ламп
Від 0,3 до 0,5	Світлий	Середній	IIIг	1,5	300	Газорозрядні

За результатами виміру освітленості величина освітленості від системи загального штучного висвітлення лежить у межах – 320 лк, що відповідає вимогам, які пред'являються до даного приміщення.

Основними джерелами шуму на робочих місцях, обладнаних відео дисплейними терміналами, є принтер, сканер факс і обладнання для кондиціонування повітря, в самих відео дисплейних терміналів – вентилятори систем охолодження і трансформатори.

Згідно ДСанПіН 3.3.2.007-98 [3] допустимий еквівалентний рівень шуму для робочого місця програміста складає 50 дБА, Допустимі параметри рівнів звуку та звукового тиску представлені в таблиці 8.5.

Таблиця 8.5 – Рівні звукового тиску від різних джерел.

Джерело шуму	Рівень шуму, дБА
Жорсткий диск	45
Вентилятор	45
Принтер	55
Сканер	50

8.3 Розробка заходів з умов поліпшення охорони праці

У комплексі заходів, що сприяють удосконаленню організації праці програміста, збереженню здоров'я і підвищенню працездатності, його велике значення має організація робочих місць. Проводячи аналіз умов праці в розглянутому приміщенні, ми одержали наступні результати:

- обсяг приміщення, що приходить на одного працюючого, відповідає нормативному значенню;
- показники мікроклімату відповідають нормативному значенню;
- акустичні умови роботи в нормі.

Важлива роль в ефективному забезпеченні праці належить моральному мікроклімату. Відносини працівників повинні ґрунтуватися на об'єктивності, доброзичливості, взаємодопомозі, глибокій повазі до кожного члена колективу, турботі про молодих співробітників.

Для забезпечення найбільш ефективного виконання обов'язків, плануючи розпорядок, слід дотримуватися таких принципів:

- для зняття втоми через кожні 1,5-2 год. роботи перерви для відпочинку на 5-10 хв.;
- для усунення монотонності виконуваних робіт чергувати характер праці.

За умови неправильної організації праці та відпочинку, втома може нагромаджуватися щоденно й переходити в перевтому або захворювання. У зв'язку з цим режим праці та відпочинку користувачів ПК необхідно будувати з урахуванням працездатності, яка змінюється протягом доби.

Виходячи із наступного можна зробити висновок, що основною причиною втомлюваності та зниження працездатності працівника, який постійно працює за комп'ютером є психофізіологічний фактор, тому основною пропозицією правильна організація робочого місця з урахуванням ергономічних вимог, а також дотримання регламентованого режиму праці та відпочинку.

					ВКРМ-122.24.0019.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		64

8.4 Розрахункова частина

Виконаємо розрахунок ризику травмування людей, зайнятих певним видом діяльності (в розрахунку за рік), якщо середньорічна кількість осіб, що займаються цією діяльністю – 600, а за останні 4 років травми одержали 9 осіб. Порівняємо обчислений рівень ризику з нормованим (прийнятим на сьогоднішній день) у світовій практиці.

1. Визначаємо середньорічну кількість травмованих осіб, для цього ділимо кількість постраждалих на кількість років, за які сталися ці трагічні події:

$$N = K/T \quad (8.1)$$

де, K – кількість постраждалих; T – період часу; підставляємо значення і отримуємо:

$$N = 9 / 4 = 2.25 \text{ осіб на рік.}$$

2. Знаходимо величину ризику за формулою

$$R = n / N, \quad (8.2)$$

де n – кількість подій, які відбулись з небажаними наслідками, N – загальна кількість подій, що може відбутися.

$$R = 2.25 / 600 = 0.00375 = 3,75 \cdot 10^{-3}.$$

Далі порівняємо отриманий результат з величиною прийнятного ризику, який визначений у світовій практиці і дорівнює $1 \cdot 10^{-6}$.

Для цього отриманий результат ділимо на прийнятий ризик:

$$\Pi = 0.003.75 / 0.000001 = 3750 \text{ разів.}$$

Відповідь: величина ризику становить $3.75 \cdot 10^{-3}$, а ризик травмування людей більший за прийнятий у 3750 разів. Крім того, можна розрахувати коефіцієнт частоти виникнення небезпечних ситуацій на 1000 чоловік, а для цього необхідно:

$$K = R \cdot 1000 \quad (8.3)$$

$$K = 0.00375 \cdot 1000 = 3.75.$$

Таким чином, коефіцієнт частоти дорівнює 3.75 людини на 1000 осіб.

					ВКРМ-122.24.0019.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		65

8.5 Висновки до розділу

У даному розділі магістерської роботи розглянуті умови праці програміста, проведено аналіз санітарно-гігієнічних умов праці працівників, які зайняті роботою з комп'ютерною технікою. Наведено приклад розрахунку ризику травмування людей, зайнятих певним видом діяльності. Можна зробити наступний висновок, що шкідливі та небезпечні виробничі фактори існують практично на будь якому робочому місці. Тільки повна усвідомленість працівника про можливі небезпеки, що можуть підстерігати його на робочому місці та дотримання вимог нормативних актів о питань охорони праці та відповідних рекомендацій фахівців, дозволять значною мірою знизити негативний вплив шкідливих та небезпечних факторів при роботі з комп'ютером на організм людини.

КБПЗ_2024

					VKPM-122.24.0019.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		66

9 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти, призначено для системи генерування складних динамічних структур.

В межах України в недостатній мірі представлені вітчизняні розробки в цій області.

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти наведені теоретичне узагальнення й рішення наукового завдання дослідження методів генерування складних динамічних структур.

Рішення даного завдання полягало у вирішенні наступних задач:

– Був проведений огляд існуючих систем генерування складних динамічних структур.

– Досліджена система генерування складних динамічних структур.

– На основі отриманих результатів досліджень створена програмна реалізація системи генерування складних динамічних структур.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання генерування складних динамічних структур.

Проведено аналіз предметної галузі в ході якого були виявлені об'єкти, взаємодія яких носить істотний характер для функціональної діяльності предметної галузі, і їхні основні характеристики; побудована алгоритм і вибраний середовище розробки.

Розроблене програмне забезпечення має простий, дружній та зручний інтерфейс користувача, що забезпечує легкість у освоєнні роботи програмного продукту, зручність у використанні, і не потребує особливих спеціальних знань.

					ВКРМ-122.24.0019.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		67

При створенні програмного забезпечення було використано об'єктно-орієнтований підхід, що відповідає сучасним тенденціям у галузі розробки комерційних програмних систем.

Програма реалізована на мові високого рівня PHP фреймворку Yii2. Дана мова програмування дозволяє найбільш ефективно обробляти дані. Це дозволило мінімізувати строк розробки програмного забезпечення, і, як слід, зменшити витрати на його розробку. Запропоноване програмне забезпечення ділиться на загальне програмне забезпечення, що поставляється із засобами обчислювальної техніки й спеціальне програмне забезпечення, що спеціально розроблене для даної конкретної системи й включає програми, що реалізують її функції.

Програма призначена для виконання під управлінням багатозадачної операційної системи Windows 10/11.

Даються необхідні рекомендації з установки розробленого програмного забезпечення.

Для підвищення рівня безпеки запропоновано застосовувати алгоритм ММВ.

В цілому створене програмне забезпечення підтверджує правильність використаних проектних рішень та повністю відповідає вимогам технічного завдання. Створене програмне забезпечення має потенційну можливість для подальшого вдосконалення і застосування у різних галузях.

Проведено маркетингове та економічне обґрунтування ІТ-проєкту, що дозволило визначити ключові фактори успіху даного проєкту.

					ВКРМ-122.24.0019.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		68

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Чирва М.В. Дослідження та програмна реалізація системи генерування складних динамічних структур // Збірник праць молодих науковців ЦНТУ. – Вип. 14. – Кропивницький: ЦНТУ, 2024.
2. Cormen T.H., Leiserson C.E., Rivest R.L., Stein C. Introduction to Algorithms, 3rd Edition (The MIT Press) 3rd Edition – The MIT Press, 2019. – 1292 p.
3. Fenner M. Machine Learning with Python for Everyone (Addison-Wesley Data & Analytics Series) 1st Edition, Kindle Edition. – Addison-Wesley Professional, 2019. – 586 p.
4. Foreman J.W. Data Smart: Using Data Science to Transform Information into Insight 1st Edition. – Wiley, 2013. – 432 p.
5. Hurbans R. Grokking Artificial Intelligence Algorithms. – Manning, 2020. – 631 p.
6. Gusfield D. Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology 1st Edition. – Cambridge University Press, 2008. – 556 p.
7. Kotu V., Deshpande B. Data Science: Concepts and Practice. – Elsevier Science, 2018. – 953 p.
8. Knowledge Base A Complete Guide – 2021 Edition // The Art of Service – Knowledge Base Publishing, 2020. – 306 p.
9. Knuth D. The Art of Computer Programming, Vol. 1: Fundamental Algorithms, 3rd Edition 3rd Edition. – Addison-Wesley Professional, 2019. – 672 p.
10. Mattmann C. Machine Learning with TensorFlow, Second Edition. – Manning, 2020. – 1124 p.
11. Mueller J.P., Massaron L. Machine Learning For Dummies. – Wiley, 2016. – 714 p.
12. Teofili T. Deep Learning for Search. – Manning, 2019. – 695 p.

					ВКРМ-122.24.0019.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		69

13. Rungta K. TensorFlow in 1 Day: Make your own Neural Network. – Publishdrive, 2019. – 587 p.
14. Weidman S. Deep Learning from Scratch: Building with Python from First Principles. – O’Reilly. 2019. – 252 p.
15. Rajasekaran S., Vijayalakshmi Pai G.A. Neural networks, fuzzy logic, and genetic algorithms: synthesis and applications (with cd-rom) Kindle Edition. – PHI, 2013. – 628 p.
16. Smirnov O., Fedorov E., Neskrodieva A., Neskrodieva T. «Intellectual Classification method of Gymnastic Elements Based on Combinations of Descriptive and Generative Approache». *CEUR Workshop Proceedings Volume 3664*, 2024, Pages 11-23.
17. Malyukov V., Bebeshko B., Lakhno V., Smirnov O., Malyukova I., Mohylnyi H. «Managing the Purchase-Sale Process of Digital Currencies Under Fuzzy Conditions». *Lecture Notes in Networks and Systems*, 2023, 729 LNNS, pp. 104–112.
18. Al-Mudhafar Aqeel, A.M., Smirnova, T., Buravchenko, K., Smirnov, O. «The method of assessing and improving the user experience of subscribers in software-configured networks based on the use of machine learning». *Advanced Information Systems*, 2023, 7(2), pp. 49-56.
19. Smirnov, O., Sydorenko, V., Aleksander, M., Zhyharevych, O., Yanchev, S. «Simulation of the cloud IoT-based monitoring system for critical infrastructures». *CEUR Workshop Proceedings*, Volume 3530, 2023, pp. 256-265.
20. Smirnov, O., Karapetyan, A., Fedorov, E., «Creating Neural Network and Single Solution Human-Based Metaheuristic Methods of Solving the Traveling Salesman Problem». *CEUR Workshop Proceedings*, Volume 3312, 2022, pp. 47-58.
21. Smirnov, O., Neskrodieva, T., Fedorov, E., Rudakov, K., Neskrodieva, A. «Method Detection Audit Data Anomalies on Basis Restricted Cauchy Machine» *CEUR Workshop Proceedings*, Volume 3187, 2022, pp. 1-12.
22. Smirnov O., Smirnova T., Anas M. Al-Oraiqat, Drieiev O., Polishchuk L., Sheroz Khan, Yassin M. Y. Hasan, Aladdein M. Amro, Hazim S. AlRawashdeh

«Method for Determining Treated Metal Surface Quality Using Computer Vision Technology». Sensors (Basel, Switzerland) Volume 22, Issue 16, 6223, 2022.

23. Smirnov, O., Lakhno, V., Akhmetov, B., Chubaievskyi, V., Khorolska, K., Bebesko, B. «Selection of a Rational Composition of Information Protection Means Using a Genetic Algorithm». In: Rajakumar, G., Du, KL., Vuppapalapati, C., Beligiannis, G.N. (eds) Intelligent Communication Technologies and Virtual Mobile Networks. Lecture Notes on Data Engineering and Communications Technologies, vol 131. 2023. Springer, Singapore. pp. 21-34.

24. Kuznetsov, A., Oleshko, I., Chernov, K., Bagmut, M., Smirnova, T. «Biometric authentication using convolutional neural networks». Lecture Notes in Networks and Systems. Volume 152, 2021, Pages 85-98.

25. Smirnov O., Kuznetsov A., Kryvinska N., Kiian A., Kuznetsova K. «Full Non-Binary Constant-Weight Codes». SN Computer Science, Vol 2, 337, 2021. <https://doi.org/10.1007/s42979-021-00739-w>.

26. Smirnov O., Neskorodieva T., Fedorov E., Rymar P. «Neural Network Modeling Method of Transformations Data of Audit Production with Returnable Waste». CEUR Workshop Proceedings Volume 3101, 2021, Pages 192-207.

27. Smirnov, O., Kuznetsov, A., Potii, O., Poluyanenko, N., Stelnyk, I., Mialkovsky, D. «Combining and filtering functions in the framework of nonlinear-feedback shift register». International Journal of Computing; 2020, Volume 19, Issue 2 – Research Institute for Intelligent Computer Systems – 2020. – P. 247-256.

28. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova T. «Non-binary constant weight coding technique». CEUR Workshop Proceedings. Volume 2740, 2020, Pages 102-114.

29. Smirnov O., Kuznetsov A., Kiian A., Cherep A., Kanabekova M., Chepurko I. «Testing of code-based pseudorandom number generators for post-quantum application». 2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT), Ukraine, Kyiv, May 14-18. 2020. P. 172-177.

30. Smirnov O., Kuznetsov A., Pushkar'ov A., Serhienko R., Babenko V., Kuznetsova T., «Representation of Cascade Codes in the Frequency Domain». In: Radivilova T., Ageyev D., Kryvinska N. (eds) Data-Centric Business and Applications. Lecture Notes on Data Engineering and Communications Technologies, vol 48. Springer, Cham. 2021. pp 557-587.

31. Smirnov, O., Drieieva, H., Drieiev, O., Polishchuk, Y., Brzhanov, R., Aleksander, M. «Method of fractal traffic generation by a model of generator on the graph». CEUR Workshop Proceedings Volume 2616, 2020, Pages 366-379.

32. Smirnov, O., Drieieva, H., Drieiev, O., Simakhin, V., Bondar, S., Odarchenko, R. «Managing multifractal properties of the binary sequence generated with the Markov chains», CEUR Workshop Proceedings Volume 2608, 2020, Pages 633-645.

33. Smirnov O. Kuznetsov A., Zaichenko Yu., Pastukhov M., Oleshko O., Kuznetsova K., «Formation of Discrete Signals with Special Correlation Properties». International Conference on Information and Telecommunication Technologies and Radio Electronics, UkrMiCo 2019; Odessa; Ukraine; 9-13 September 2019. P.22-28.

34. Smirnov, O., Kuznetsov, A., Kolovanova, I., Kuznetsova, T., «Noise immunity of the algebraic geometric codes». International Journal of Computing; 2019, Volume 18, Issue 4 – Research Institute for Intelligent Computer Systems – 2019. – P. 393-407.

35. Smirnov, O., Kuznetsov, A., Reshetniak, O., Ivko, N., Katkova, T., Kuznetsova, T., «Generators of Pseudorandom Sequence with Multilevel Function of Correlation». 2019 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T), Kyiv, Ukraine, 8 – 11 October 2019 . P.517-522.

36. Smirnov, O., Ulichev, O., Meleshko, Y., Khokh, V., Goncharenko, I. «Method of Choosing Objects for Informational Influence in Social Networks during Information Campaign Based on the Analytic Hierarchy Process». CEUR Workshop Proceedings, Vol 2588, P. 215-227, 2019.

					БКРМ-122.24.0019.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		72

37. Smirnov, O., Krasnobayev, V., Yanko, A., Kuznetsova, T. «Methods of nulling numbers in the system of residual classes». CEUR Workshop Proceedings, Vol 2588, P. 90-106, 2019.

38. Smirnov, O., Kuznetsov, A., Kovalchuk, D., Averchev, A., Pastukhov, M., Kuznetsova, K., «Formation of Pseudorandom Sequences with Special Correlation Properties», 2019 3rd International Conference on Advanced Information and Communications Technologies, AICT -2019/ Lviv, Ukraine, 2-6 July, 2019, P. 395-399.

39. Smirnov, O., Kuznetsov, A., Kiian, A., Zamula, A., Rudenko, S., Hryhorenko, V., «Variance Analysis of Networks Traffic for Intrusion Detection in Smart Grids», 2019 IEEE 6th International Conference On Energy Smart Systems (2019 IEEE ESS), Kyiv, Ukraine April 17-19, 2019 P. 353-358.

40. Smirnov, O., Kuznetsov, A., Kavun, S., Babenko, B., Nakisko, O., Kuznetsova, K., «Malware Correlation Monitoring in Computer Networks of Promising Smart Grids», 2019 IEEE 6th International Conference On Energy Smart Systems (2019 IEEE ESS), Kyiv, Ukraine April 17-19, 2019 P. 347-352.

41. Smirnov, O., Kuznetsov, A., Kovalchuk, D., Pastukhov, M., Kuznetsova, K., Prokopovych-Tkachenko, D., «Discrete Signals with Special Correlation Properties», CEUR Workshop Proceedings Volume 2353, CEUR Workshop Proceedings 2019, Pages 618-629.

42. Smirnov, S., Bulekbaeva, G., Kikvidze, O.G., Lakhno, V., Brzhanov, R., Tabylov, A. «Computer simulation in the MathCAD package of plastic deformation of the deposited layer on the flat surface of the part». Journal of Theoretical and Applied Information Technology Volume 97, Issue 20, 2019, Pages 2467-2484. (Scopus).

43. Smirnov A.A., Kuznetsov A.A., Danilenko D.A., Berezovsky A., «The statistical analysis of a network traffic for the intrusion detection and prevention systems», Telecommunications and Radio Engineering. – Volume 74, Issue 1. – Begel House Inc. – 2015. – P. 61-78.

44. Аль-Мудхафар Акіл Абдулхуссейн М., Смірнова Т.В., Буравченко К.О., Смірнов О.А. «Метод оцінки та підвищення користувальницького досвіду абонентів в програмно-конфігурованих мережах на основі використання машинного навчання». Сучасні інформаційні системи, 2023, том 7, № 2, С. 49-56.

45. Смірнов О.А., Смірнова Т.В., Буравченко К.О., Кравченко С.С., Горбов В.О., «Хмарна система підтримки прийняття рішень технологічного процесу відновлення поверхонь конструкцій і деталей машин». Сучасні інформаційні системи. 2021. Т. 5, № 4. С. 79-95

46. Смірнов, О.А., Усік П.С., Полігенько О.О., Одарченко Р.С., Терещенко Л.Ю. «Інформаційна технологія та програмне забезпечення для підвищення ефективності планування підсистеми базових станцій стільникового зв'язку». Проблеми телекомунікацій. № 1(26). С. 83-96. 2020.

47. Смірнов О.А., Усік П.С., Миронець І.В., Буравченко К.О., Якименко Н.М. «Метод підвищення ефективності розподіленої обробки даних у комп'ютерних системах операторів стільникового зв'язку» Вісник Черкаського державного технологічного університету. Технічні науки. №4. С. 103-110. 2020.

48. О.А.Смірнов, Т.В.Смірнова, Л.І. Поліщук, К.О. Буравченко, А.О.Макевнін, «Дослідження хмарних технологій як сервісів», Кібербезпека: освіта, наука, техніка. № 3(7). С. 43-62. 2020.

49. Смірнов, С.А., Смірнова, Т.В., Минайленко, Р.М., Доренський, О.П., Сисоєнко С.В. «Хмарна автоматизована система інтелектуальної підтримки прийняття рішень для технологічних процесів». Вісник Черкаського державного технологічного університету. Технічні науки. №4, 2020, С. 84-92.

50. Смірнов О.А., Дреєва Г.М., Дреєв О.М., Смірнова Т.В. «Фрактальний аналіз генератора самоподібного трафіку на основі ланцюга Маркова». Центральнотраїнський науковий вісник. Технічні науки. № 2(33). с. 161-172, 2019.

51. Smirnov, O., Kuznetsov, A., Kuznetsova., K. Synthesis of Discrete Signals with Improved Correlation Properties. Монографія: In.: ISCI'2019:

					ВКРМ-122.24.0019.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		74

Information Security in Critical Infrastructures. Collective monograph. Edited by Ivan D. Gorbenko and Alexandr A. Kuznetsov, ASC Academic Publishing, USA, 2019, pp. 281-299. – ISBN: 978-0-9989826-8-7 (Hardback), ISBN: 978-0-9989826-9-4 (Ebook).

52. О.А. Смірнов, Т.В. Смірнова, О.М. Дреєв, Є.К. Солових, «Методи оптимізації технологічних процесів відновлення сталевих покриттів», Shipbuilding & marine infrastructure / Суднобудування і морська інфраструктура № 1 (11). с. 48-57, 2019.

53. Смірнова Т.В., Солових Є.К., Смірнов О.А., Дреєв О.М. Побудова хмарних інформаційних технологій оптимізації технологічного процесу відновлення та зміцнення поверхонь деталей. Центральноукраїнський науковий вісник. Технічні науки. № 1(32). с. 184-194, 2019.

54. Смірнова Т.В., Дреєв О.М., Смірнов О.А. Експертна система оптимізації процесу відновлення та зміцнення поверхонь деталей типу «вал» електродуговим напиленням. Системи управління, навігації та зв'язку, № 2 (54). С. 149-154, 2019.

55. Смірнов О.А., Котелянець В.В. Стійкі до колізій стохастичні моделі функціонування безпроводових сенсорних мереж. Вісник інженерної академії України, №3, с. 145-152, 2018

					ВКРМ-122.24.0019.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		75

Додаток А
(обов'язковий)

Технічне завдання

Зміст

1 Найменування та область застосування.....	2
2 Підстава для розробки.....	2
3 Мета та призначення розробки.....	2
4 Джерела розробки.....	2
5 Технічні вимоги.....	2
5.1 Вміст проекту.....	2
5.2 Показники призначення.....	3
5.3 Вимоги до функціональних характеристик.....	3
5.4 Вимоги до архітектури.....	3
5.5 Вимоги до надійності.....	3
5.6 Умови експлуатації.....	4
5.7 Вимоги до складу та параметрів технічних засобів.....	4
5.8 Вимоги до інформаційної і програмної сумісності.....	4
5.8.1 Обладнання.....	4
5.8.2 Мова програмування.....	4
5.8.3 Вхідні дані.....	5
5.8.4 Вихідні дані.....	5
6 Вимоги до програмної документації.....	5
7 Економічні вимоги.....	5
8 Вимоги щодо охорони праці.....	5
9 Перелік документів, що розробляються.....	6
10 Етапи розробки.....	6
11 Порядок контролю та приймання.....	6

					ВКРМ-122.24.0019.00.00.ТЗ			
Вим.	Арк.	№ документа	Підпис	Дата				
Розробив	Чирва М.В.				<i>Дослідження та програмна реалізація системи генерування складних динамічних структур</i>	Літ.	Аркуш	Аркушів
Перевірів	Коваленко А.С.					М	1	6
Н. Контр.	Коваленко А.С.				ЦНТУ КН-23М			
Затв.	Смірнов О.А.							

1 Найменування та область застосування

Це технічне завдання розповсюджується на дослідження та програмну реалізацію системи генерування складних динамічних структур.

2 Підстава для розробки

Підставою для розробки служить завдання на випускню кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, видане на кафедрі кібербезпеки та програмного забезпечення (нак. № 18-13 від 07.08.2024 року).

3 Мета та призначення розробки

Метою випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти є дослідження та програмна реалізація системи генерування складних динамічних структур.

4 Джерела розробки

Джерелом цієї випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти є стосовна до теми література і існуючі аналоги.

5 Технічні вимоги

5.1 Склад продукції

Складниками розробки є:

- вибір і обґрунтування методів реалізації проекту;
- розробка програмної частин системи, а також розробка взаємодії системи з ОС та з користувачем;

					ВКРМ-122.24.0019.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

- техніко-економічне обґрунтування доцільності прийнятого до розробки програмного забезпечення;
- аналіз умов праці;
- розробка програми, що реалізує спроектовані алгоритми роботи системи.

5.2 Показники призначення

Система повинна забезпечувати:

- програмну реалізацію системи генерування складних динамічних структур;
- цілісність даних у процесі роботи та при зберіганні;
- простий, інтуїтивно зрозумілий інтерфейс.

5.3 Вимоги до функціональних характеристик

Розроблене програмне забезпечення не повинно мати обмежень на версію драйверів та операційної системи.

5.4 Вимоги до архітектури

Компонент, що розробляється повинен використовувати системні засоби та апаратні засоби, що на даному етапі розвитку обчислювальної техніки найбільше поширені.

5.5 Вимоги до надійності

Програмні модулі написані по всім правилам, які стосуються стандартних викликів процедур, функцій, методів і форм, визначених технічною документацією на середовище розробки.

					ВКРМ-122.24.0019.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		3

5.6 Умови експлуатації

Робочі місця користувачів ПЗ повинні задовольняти наступним умовам експлуатації:

- температура повітря: 19-20 град. по Цельсію;
- відносна вологість повітря до 80%;
- атмосферний тиск 107 кПа.

5.7 Вимоги до складу та параметрів технічних засобів

Програмне забезпечення повинно бути реалізоване на ПЕОМ, працювати в ОС Windows 10/11 і з сумісними з цією платформою пристроями і прикладним програмним забезпеченням.

5.8 Вимоги до інформаційної і програмної сумісності

Переносність програмного забезпечення повинна бути забезпечена за рахунок його реалізації стандартного інтерфейсу взаємодії з ОС, що працюють під управлінням ОС Windows 10/11.

5.8.1 Обладнання

Комп'ютер Intel® Celeron/8 Mb/1.2 Gb/SVGA 14" 1Mb або сумісні з ним.

5.8.2 Мова програмування

Середовище PHP фреймворку Yii2.

					ВКРМ-122.24.0019.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

5.8.3 Вхідні дані

Опис алгоритму роботи запропонованої системи.

5.8.4 Вихідні дані

Робоча програма.

6 Вимоги до програмної документації

Програмна продукція повинна бути представлена у виді опису структури даних, схем та опису алгоритму, а також текстів вихідних модулів програмного забезпечення згідно ЄСПД .

7 Економічні вимоги

7.1 Для ПЗ необхідно виробити функціонально-вартісний аналіз варіантів розробки.

7.2 Виконати маркетингове та економічне обґрунтування ІТ-проєкту з урахуванням цін на 3 вересня 2024 року.

8 Вимоги щодо охорони праці

В частині охорони праці випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти повинна бути розглянута характеристика умов праці програміста.

					ВКРМ-122.24.0019.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		5

9 Перелік документів, що розробляються

- Наукова новизна – 1 аркуш.
- Структурна схема системи – 1 аркуш.
- Функціональна схема системи – 1 аркуш.
- Діаграма процесів – 1 аркуш.
- Блок-схема алгоритму роботи програми – 2 аркуша.
- Показники економічної ефективності – 1 аркуш.
- Пояснювальна записка – 75 аркушів.

10 Етапи розробки

10.1 Збір і обробка інформації по темі випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти. Постановка задачі на виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти (складання ТЗ).

10.2 Проведення досліджень або експериментальних робіт для уточнення основних положень випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.

10.3 Розробка функціональних схем, блок схем алгоритмів роботи програмного забезпечення.

10.4 Побудова схем взаємодії даних.

10.5 Створення прототипу ПЗ.

10.6 Віднаходження ПЗ, аналіз отриманих результатів.

10.7 Робота над питанням охорони праці і техніки безпеки.

10.8 Маркетингове та економічне обґрунтування ІТ-проєкту.

10.9 Оформлення пояснювальної записки і виконання робіт по графічній частині.

11 Порядок контролю та приймання

11.1 Подання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти на попередній захист 02.12.2024 р.

11.2 Подання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти на захист 16.12.2024 р.

					ВКРМ-122.24.0019.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		6

Додаток Б
(обов'язковий)

Міністерство освіти і науки України
Центральноукраїнський національний технічний університет

ЗАТВЕРДЖУЮ

Керівник випускної кваліфікаційної роботи за
другим (магістерським) рівнем вищої освіти

_____ Коваленко А.С.

*Дослідження та програмна реалізація
системи генерування складних динамічних структур*

Лістинг програми

Код документу 12

Носій: CD/DVD-диск / USB-флеш-накопичувач

Загальна кількість аркушів: 18

Літера: РП

Кропивницький – 2024 року

Файл PostController - файл автентифікації

```
<?php

namespace app\controllers;

use Yii;
use app\models\Post;
use app\models\PostsSearch;
use yii\web\Controller;
use yii\web\NotFoundHttpException;
use yii\filters\VerbFilter;

/**
 * PostController імplementує CRUD дії для моделі Post.
 */
class PostController extends Controller
{

    public function behaviors()
    {
        return [
            'verbs' => [
                'class' => VerbFilter::className(),
                'actions' => [
                    'delete' => ['POST'],
                ],
            ],
        ];
    }

    /**
     * Список всіх моделей Post.
     */
    public function actionIndex()
    {
        $searchModel = new PostsSearch();
        $dataProvider = $searchModel->search(Yii::$app->request->queryParams);

        return $this->render('index', [
            'searchModel' => $searchModel,
            'dataProvider' => $dataProvider,
        ]);
    }

    /**
     * Відображення моделі Post.
     */
    public function actionView($id)
    {
        return $this->render('view', [
            'model' => $this->findModel($id),
        ]);
    }

    /**
     * Створення моделі нового поста.
     * Якщо створення пройшло успішно браузер переходить на сторінку 'view'
     */
    public function actionCreate()
    {
        $model = new Post();

        if ($model->load(Yii::$app->request->post()) && $model->save()) {
            return $this->redirect(['view', 'id' => $model->id]);
        } else {
            return $this->render('create', [
                'model' => $model,
            ]);
        }
    }
}
```

```
        ]);
    }
}

/**
 * Оновлення існуючої моделі Post.
 */
public function actionUpdate($id)
{
    $model = $this->findModel($id);

    if ($model->load(Yii::$app->request->post()) && $model->save()) {
        return $this->redirect(['view', 'id' => $model->id]);
    } else {
        return $this->render('update', [
            'model' => $model,
        ]);
    }
}

/**
 * Видалення моделі Post.
 * Якщо видалення пройшло успішно браузер переходить на сторінку 'index'
 */
public function actionDelete($id)
{
    $this->findModel($id)->delete();

    return $this->redirect(['index']);
}

/**
 * Знаходить модель Post по ключу БД.
 * якщо модель не знайдена генерується виключна ситуація
 *NotFoundException
 */
protected function findModel($id)
{
    if (($model = Post::findOne($id)) !== null) {
        return $model;
    } else {
        throw new NotFoundException('The requested page does not
exist.');
```

Файл SiteController.php

```
<?php

namespace app\controllers;

use Yii;
use yii\filters\AccessControl;
use yii\web\Controller;
use yii\filters\VerbFilter;
use app\models\LoginForm;
use app\models\ContactForm;
use app\models\Signup;
use app\models>Login;

class SiteController extends Controller
{
    public function behaviors()
    {
        return [
            'access' => [
                'class' => AccessControl::className(),
                'only' => ['logout'],
                'rules' => [
                    [
                        'actions' => ['logout'],
                        'allow' => true,
                        'roles' => ['@'],
                    ],
                ],
            ],
            'verbs' => [
                'class' => VerbFilter::className(),
                'actions' => [
                    'logout' => ['post'],
                ],
            ],
        ];
    }

    public function actions()
    {
        return [
            'error' => [
                'class' => 'yii\web>ErrorAction',
            ],
            'captcha' => [
                'class' => 'yii\captcha\CaptchaAction',
                'fixedVerifyCode' => YII_ENV_TEST ? 'testme' : null,
            ],
        ];
    }

    public function actionIndex()
    {
        return $this->render('index');
    }

    //реєстрація користувача
    public function actionSignup(){
        $model = new Signup();
        if(isset($_POST['Signup'])){
            $model->attributes= Yii::$app->request->post('Signup');

            if($model->validate() && $model->signup()){
                return $this->redirect(['index']);
            }
        }
    }
}
```

```
    }
    return $this->render('signup', ['model'=>$model]);
}

//функція входу на свій акаунт
public function actionLogin(){
    $login_model = new Login();
    if(Yii::$app->request->post('Login')){
        $login_model->attributes = Yii::$app->request->post('Login');

        if($login_model->validate())
        {

            Yii::$app->user->login($login_model->getUser());
            return $this->goHome();
        }
    }
    return $this->render('login', ['login_model'=>$login_model]);
}

public function actionLogout()
{
    if(!Yii::$app->user->isGuest)
    {
        Yii::$app->user->logout();
        return $this->redirect(['login']);
    }
}

public function actionContact()
{
    $model = new ContactForm();
    if ($model->load(Yii::$app->request->post()) && $model->contact(Yii::$app->params['adminEmail'])) {
        Yii::$app->session->setFlash('contactFormSubmitted');

        return $this->refresh();
    }
    return $this->render('contact', [
        'model' => $model,
    ]);
}

public function actionAbout()
{
    return $this->render('about');
}
}
```

Файл UserController.php

```
<?php

namespace app\controllers;

use Yii;
use app\models\User;
use yii\data\ActiveDataProvider;
use yii\web\Controller;
use yii\web\NotFoundHttpException;
use yii\filters\VerbFilter;

/**
 * UserController імплементує CRUD події для User.
 */
class UserController extends Controller
{
    /**
     * @inheritdoc
     */
    public function behaviors()
    {
        return [
            'verbs' => [
                'class' => VerbFilter::className(),
                'actions' => [
                    'delete' => ['POST'],
                ],
            ],
        ];
    }

    /**
     * Список всіх моделей User.
     * @return mixed
     */
    public function actionIndex()
    {
        $dataProvider = new ActiveDataProvider([
            'query' => User::find(),
        ]);

        return $this->render('index', [
            'dataProvider' => $dataProvider,
        ]);
    }

    /**
     * Відображає модель User.
     * @param string $id
     * @return mixed
     */
    public function actionView($id)
    {
        return $this->render('view', [
            'model' => $this->findModel($id),
        ]);
    }

    /**
     * Створення нового користувача.
     * @return mixed
     */
    public function actionCreate()
    {
        $model = new User();

        if ($model->load(Yii::$app->request->post()) && $model->save()) {
```

```

        return $this->redirect(['view', 'id' => $model->id]);
    } else {
        return $this->render('create', [
            'model' => $model,
        ]);
    }
}

/**
 * Оновлення існуючої моделі User.
 * @param string $id
 * @return mixed
 */
public function actionUpdate($id)
{
    $model = $this->findModel($id);

    if ($model->load(Yii::$app->request->post()) && $model->save()) {
        return $this->redirect(['view', 'id' => $model->id]);
    } else {
        return $this->render('update', [
            'model' => $model,
        ]);
    }
}

/**
 * Видалення користувача .
 */
public function actionDelete($id)
{
    $this->findModel($id)->delete();

    return $this->redirect(['index']);
}

/**
 * Знаходить модель User по ключу БД.
 * If the model is not found, a 404 HTTP exception will be thrown.
 */
protected function findModel($id)
{
    if (($model = User::findOne($id)) !== null) {
        return $model;
    } else {
        throw new NotFoundHttpException('The requested page does not
exist.');
```

Class KMeans.php

```

<?php
use Exception;
class KMeans
{
    // ініціалізація поля даних
    protected $data;
    protected $clustered_data;
    // масив центроїдів
    protected $centroids;
    // масив відстаней центроїдів
    protected $centroid_distance;
    // доступні методи для кластеризації
    protected static $ACCEPTED_CLUSTERING_METHODS = [
        'random',
        'forgy',
    ];
    /**
     * ініціалізація
     *
     * @param $data array */
    public function __construct(array $data)
    {
        if (count($data) < 2) {
            throw new Exception('Data must have more than one row');
        }
        $this->data = $data;
    }
    /**
     * основна функція
     * @param $cluster_count integer кількість кластерів
     * @param $method string метод кластеризації
     * @return масив кластеризованих даних (getClusteredData)
     */
    public function cluster($cluster_count, $method = 'forgy')
    {
        if ($cluster_count < 2) {
            throw new Exception('Cluster count must be greater than 1');
        }
        if ($cluster_count > count($this->data)) {
            throw new Exception('Cluster count must be greater than the number
of data points');
        }
        if (!in_array($method, self::$ACCEPTED_CLUSTERING_METHODS)) {
            throw new Exception("Unrecognized method passed into cluster:
{$method}");
        }
        do {
            if (empty($centroids)) {
                $centroids = $this->getInitialCentroids($cluster_count,
$method);
            } else {
                $centroids = $this->calculateCentroids($this->clustered_data);
            }
            $new_clustered_data = array_fill(0, $cluster_count, []);
            foreach ($this->data as $observation) {
                $closest_centroid = $this->
>calculateClosestCentroid($observation, $centroids);
                array_push($new_clustered_data[$closest_centroid],
$observation);
            }
        } while ($this->assignmentConvergenceCheck((array) $this->
>clustered_data, $new_clustered_data) === false);
        $this->centroids = $centroids;

        return $this->getClusteredData();
    }
    /**

```

```

    * Отримання центроїдів
    *
    * @return array list of centroids
    */
public function getCentroids()
{
    if (empty($this->centroids)) {
        throw new Exception('Centroids have not been hydrated yet - run
cluster method first');
    }
    return $this->centroids;
}
/**
 * Отримання кластеризованих даних
 *
 * @return array multi-dimensional array of clustered data
 */
public function getClusteredData()
{
    if (empty($this->clustered_data)) {
        throw new Exception('Clustered data have not been hydrated yet - run
cluster method first');
    }
    return $this->clustered_data;
}
/**
 * Отримання відстаней центроїдів
 * will throw an exception if cluster has not been run yet
 *
 * @return array list
 */
/*
public function getCentroidDistance()
{
    if (empty($this->centroid_distance)) {
        throw new Exception('Centroid distance has not been hydrated yet -
run cluster method first');
    }
    return $this->centroid_distance;
}
*/
/**
 * Ініціалізація центроїдів
 *
 * @param $cluster_count integer кількість кластерів
 * @param $method string тип ініціалізації
 * @return array список центроїдів для ініціалізації
 */
protected function getInitialCentroids($cluster_count, $method)
{
    if ($method == 'forgy') {
        return $this->getForgyInitialization($cluster_count);
    }
    if ($method == 'random') {
        return $this->getRandomInitialization($cluster_count);
    }
}
/**
 * Ініціалізація випадковим чином
 *
 * @param $cluster_count integer кількість кластерів
 * @return array список
 */
protected function getRandomInitialization($cluster_count)
{
    $random_keys = array_rand($this->data, $cluster_count);
    $random_keys = array_flip($random_keys);
    return array_intersect_key($this->data, $random_keys);
}

```

```

/**
 *
 * @param $cluster_count integer кількість кластерів
 * @return array список
 */
protected function getForgyInitialization($cluster_count)
{
    $data_range = $this->calculateRange($this->data);
    $random_points = [];
    for ($i = 0; $i < $cluster_count; $i++) {
        $random_points[$i] = array_fill(0, count($this->data), null);
        foreach ($data_range as $key => $range) {
            $random_points[$i][$key] = ($range['min'] + lcg_value() *
($range['max'] - $range['min']));
        }
    }
    return $random_points;
}
/**
 * Обчислити центроїди
 *
 * @param $clustered_data array масив кластеризованих даних
 * @return array список центроїдів
 */
protected function calculateCentroids(array $clustered_data)
{
    $centroids = [];
    foreach ($clustered_data as $cluster) {
        $cluster_sum = array_fill(0, count(current($cluster)), 0);
        foreach ($cluster as $observation) {
            foreach ($observation as $key => $value) {
                $cluster_sum[$key] += $value;
            }
        }
        $centroid = array_fill(0, count(current($cluster)), 0);
        foreach ($cluster_sum as $key => $value) {
            $centroid[$key] = $value / count($cluster);
        }
        array_push($centroids, $centroid);
    }
    return $centroids;
}
/**
 * Обчислити найближчу відстань до центроїда
 *
 * @param $observation array
 * @param $centroids array список центроїдів
 * @return integer index that observation should be
clustered into
 */
protected function calculateClosestCentroid(array $observation, array
$centroids)
{
    $centroid_distance = [];
    foreach ($centroids as $centroid) {
        array_push($centroid_distance, $this->calculateDistance($observation, $centroid));
    }
    asort($centroid_distance);
    $centroid_distance = array_keys($centroid_distance);
    return array_shift($centroid_distance);
}
/**
 * Перевірити кластеризовані дані
 *
 * @param $clustered_data array the old holder of clustered_data

```

```

    * @param $new_clustered_data array new clustered_data to check
    against
    * @return boolean whether or not convergence has
    occurred
    */
    protected function assignmentConvergenceCheck(array $clustered_data, array
    $new_clustered_data)
    {
        if (empty($clustered_data)) {
            $this->clustered_data = $new_clustered_data;
            return false;
        }
        foreach ($clustered_data as $key => $cluster) {
            foreach ($cluster as $observation) {
                if (!in_array($observation, $new_clustered_data[$key])) {
                    $this->clustered_data = $new_clustered_data;
                    return false;
                }
            }
        }
        return true;
    }
}
/**
 * Допоміжний метод обчислення розмірності даних
 *
 * @param $data array list of points to determine range of
 * @return array formatted return of range based on the data
 */
protected function calculateRange($data)
{
    $data_range = array_fill(0, count(current($data)), ['min' => null, 'max'
=> null]);
    foreach ($data as $observation) {
        $key = 0;
        foreach ($observation as $value) {
            if ($data_range[$key]['min'] === null ||
            $data_range[$key]['min'] > $value) {
                $data_range[$key]['min'] = $value;
            }
            if ($data_range[$key]['max'] === null ||
            $data_range[$key]['max'] < $value) {
                $data_range[$key]['max'] = $value;
            }
        }
        $key++;
    }
    return $data_range;
}
/**
 * Обчислення евклідової відстані
 *
 * @param $point_a array list of numeric values that determine a point
 * @param $point_b array list of numeric values that determine a point
 * @return float distance between the points
 */
protected function calculateDistance($point_a, $point_b)
{
    $distance = 0;
    for ($i = 0, $count = count($point_a); $i < $count; $i++) {
        $difference = $point_a[$i] - $point_b[$i];
        $distance += pow($difference, 2);
    }
    return $distance;
}
}

```

Login.php

```

<?php
namespace app\models;
use yii\base\Model;
class Login extends Model
{
    public $email;
    public $password;
    public function rules()
    {
        return [
            [['email','password'],'required'],
            ['email','email'],
            ['password','string','min'=>2,'max'=>10],
            ['password','validatePassword'] //собственная функция для валидации
пароля
        ];
    }
    public function attributeLabels()
    {
        return [
            'email' => 'Электронна пошта:',
            'password' => 'Пароль:',
        ];
    }
    public function validatePassword($attribute,$params)
    {
        if(!$this->hasErrors()) // если нет ошибок в валидации
        {
            $user = $this->getUser(); // получаем пользователя для дальнейшего
сравнения пароля
            if(!$user || !$user->validatePassword($this->password))
            {
                //якщо користувач не знайдений в БД
                $this->addError($attribute,'Пароль или имейл введены неверно');
                //виведення помилки
            }
        }
    }
    public function getUser()
    {
        return User::findOne(['email'=>$this->email]); // а получаем мы его по
введенному имейлу
    }
}

```

Signup.php

```

<?php

namespace app\models;

use yii\base\Model;

/**
 *
 */
class Signup extends Model
{
    public $email;
    public $password;

    public function rules(){
        return[
            [['email','password'],'required'],
            [['email'],'email'],
            [['email'],'unique','targetClass'=>'app\models\User'],
            [['password'],'string','min'=>2,'max'=>10]

        ];
    }
    /**
     * @inheritdoc
     */
    public function attributeLabels()
    {
        return [
            'email' => 'Електронна пошта:',
            'password' => 'Пароль:',
        ];
    }
    public function signup(){
        $user = new User();
        $user->email = $this->email;
        $user->setPassword($this->password);

        return $user->save(); // true or false
    }
}
?> <?php

namespace app\models;

use yii\base\Model;

/**
 *
 */
class Signup extends Model
{
    public $email;
    public $password;

    public function rules(){
        return[
            [['email','password'],'required'],
            [['email'],'email'],
            [['email'],'unique','targetClass'=>'app\models\User'],
            [['password'],'string','min'=>2,'max'=>10]

        ];
    }
}
/**

```

```

    * @inheritdoc
    */
    public function attributeLabels()
    {
        return [
            'email' => 'Електронна пошта:',
            'password' => 'Пароль:',
        ];
    }

    public function signup() {

        $user = new User();
        $user->email = $this->email;
        $user->setPassword($this->password);

        return $user->save(); // true or false
    }
}
?> <?php

namespace app\models;

use yii\base\Model;

/**
 *
 */
class Signup extends Model
{
    public $email;
    public $password;

    public function rules() {
        return [
            [['email', 'password'], 'required'],
            [['email', 'email'],
            [['email', 'unique', 'targetClass'=>'app\models\User'],
            [['password', 'string', 'min'=>2, 'max'=>10]

        ];
    }

    public function attributeLabels()
    {
        return [
            'email' => 'Електронна пошта:',
            'password' => 'Пароль:',
        ];
    }

    public function signup() {

        $user = new User();
        $user->email = $this->email;
        $user->setPassword($this->password);

        return $user->save(); // true or false
    }
}
?>

User.php

<?php

namespace app\models;

use Yii;
use yii\db\ActiveRecord;
use yii\web\IdentityInterface;

```

```

/**
 * This is the model class for table "tbl_user".
 *
 * @property string $id
 * @property string $login
 * @property string $password
 * @property string $email
 * @property string $creation_date
 * @property string $reputation
 * @property string $location
 * @property string $about
 *
 * @property TblComments[] $tblComments
 * @property TblPost[] $tblPosts
 */
class User extends ActiveRecord implements IdentityInterface
{
    /**
     * @inheritdoc
     */
    public static function tableName()
    {
        return 'tbl_user';
    }

    /**
     * @inheritdoc
     */
    /* public function rules()
    {
        return [
            [['password', 'email', 'creation_date'], 'required'],
            [['creation_date'], 'safe'],
            [['reputation'], 'integer'],
            [['about'], 'string'],
            [['login', 'location'], 'string', 'max' => 40],
            [['password'], 'string', 'max' => 10],
            [['email'], 'string', 'max' => 20],
        ];
    }*/

    /**
     * @inheritdoc
     */
    public function attributeLabels()
    {
        return [
            'id' => 'ID',
            'login' => 'Login',
            'password' => 'Password',
            'email' => 'Email',
            'creation_date' => 'Creation Date',
            'reputation' => 'Reputation',
            'location' => 'Location',
            'about' => 'About',
        ];
    }

    public function setPassword($password) {
        return $this->password = sha1($password);
    }

    public function setEmail($email) {
        return $this->email = $email;
    }

    public function validatePassword($password)
    {
        return $this->password === sha1($password);
    }
}

```

```
//=====
public static function findIdentity($id)
{
    return self::findOne($id);
}
public function getId()
{
    return $this->id;
}
public static function findIdentityByAccessToken($token, $type = null)
{
}
public function getAuthKey()
{
}
public function validateAuthKey($authKey)
{
}

/**
 * @return \yii\db\ActiveQuery
 */
public function getTblComments()
{
    return $this->hasMany(TblComments::className(), ['user_id' => 'id']);
}

/**
 * @return \yii\db\ActiveQuery
 */
public function getTblPosts()
{
    return $this->hasMany(TblPost::className(), ['author_id' => 'id']);
}
}
```

ContactUs.php

```
<?php

namespace app\models;

use Yii;
use yii\base\Model;

/**
 * ContactForm
 */
class ContactForm extends Model
{
    public $name;
    public $email;
    public $subject;
    public $body;
    public $verifyCode;

    /**
     * @return масив правил валідації
     */
    public function rules()
    {
        return [
            // name, email, subject and body are required
            [['name', 'email', 'subject', 'body'], 'required'],
            // email has to be a valid email address
            ['email', 'email'],
            // verifyCode needs to be entered correctly
            ['verifyCode', 'captcha'],
        ];
    }

    /**
     * @return масив міток атрибутів
     */
    public function attributeLabels()
    {
        return [
            'name' => 'Імя:',
            'email' => 'Електронна пошта:',
            'subject' => 'Тема',
            'body' => 'Текст повідомлення',
            'verifyCode' => 'Код верифікації',
        ];
    }

    /**
     * Надсилає повідомлення
     * @param string $email
     * @return boolean пройшла модель валідацію чи ні
     */
    public function contact($email)
    {
        if ($this->validate()) {
            Yii::$app->mailer->compose()
                ->setTo($email)
                ->setFrom([$this->email => $this->name])
                ->setSubject($this->subject)
                ->setTextBody($this->body)
                ->send();

            return true;
        }
        return false;
    }
}
```

login.php

```
<?php

/* @var $this yii\web\View */
/* @var $form yii\bootstrap\ActiveForm */
/* @var $model app\models\LoginForm */

use yii\helpers\Html;
use yii\widgets\ActiveForm;

$this->title = 'Вхід на сайт';
$this->params['breadcrumbs'][] = $this->title;
?>
<div class="site-login">
    <h1><?= Html::encode($this->title) ?></h1>

    <p>Будь-ласка заповніть поля, щоб ввійти на сайт:</p>

    <?php $form = ActiveForm::begin([
        'id' => 'login-form',
        'options' => ['class' => 'form-horizontal'],
        'fieldConfig' => [
            'template' => "{label}\n<div class=\"col-lg-3\">{input}</div>\n<div
class=\"col-lg-8\">{error}</div>",
            'labelOptions' => ['class' => 'col-lg-1 control-label'],
        ],
    ]); ?>

    <?= $form->field($login_model, 'email')->textInput(['autofocus' =>
true]) ?>

    <?= $form->field($login_model, 'password')->passwordInput() ?>

    <div class="form-group">
        <div class="col-lg-offset-1 col-lg-11">
            <?= Html::submitButton('Ввійти', ['class' => 'btn btn-primary',
'name' => 'login-button']) ?>
        </div>
    </div>

    <?php ActiveForm::end(); ?>
</div>
```